

9.2

*IBM MQ w kontenerach*

**IBM**

**Uwaga**

Przed skorzystaniem z niniejszych informacji oraz produktu, którego one dotyczą, należy zapoznać się z informacjami zamieszczonymi w sekcji “Uwagi” na stronie 165.

To wydanie dotyczy wersji 9 wydanie 2 produktu IBM® MQ oraz wszystkich kolejnych wydań i modyfikacji, o ile nie zostanie to określone inaczej w nowych edycjach.

Wysyłając informacje do IBM, użytkownik przyznaje IBM niewyłączne prawo do używania i rozpowszechniania informacji w dowolny sposób, jaki uzna za właściwy, bez żadnych zobowiązań wobec ich autora.

© **Copyright International Business Machines Corporation 2007, 2024.**

---

# Spis treści

<b>IBM MQ w kontenerach i IBM Cloud Pak for Integration.....</b>	<b>5</b>
Planowanie wdrożenia IBM MQ w kontenerach.....	5
Wybór sposobu używania IBM MQ w kontenerach.....	5
Obsługa operatora IBM MQ Operator.....	6
Zależności operatora IBM MQ Operator.....	10
Uprawnienia o zasięgu klastra wymagane przez IBM MQ Operator.....	11
Uwagi dotyczące pamięci masowej dla IBM MQ Operator.....	11
Obsługa budowania własnych obrazów kontenera menedżera kolejek IBM MQ.....	13
Wysoka dostępność produktu IBM MQ w kontenerach.....	16
Odtwarzanie po awarii w przypadku IBM MQ w kontenerach.....	18
Uwierzytelnianie i autoryzacja użytkowników oprogramowania IBM MQ w kontenerach.....	19
Planowanie skalowalności i wydajności dla produktu IBM MQ w kontenerach.....	20
Używanie IBM MQ w IBM Cloud Pak for Integration i Red Hat OpenShift.....	20
Historia wydań operatora IBM MQ Operator.....	21
Migracja produktu IBM MQ do rozwiązania IBM Cloud Pak for Integration.....	38
Instalowanie i deinstalowanie produktu IBM MQ Operator w rozwiązaniu Red Hat OpenShift.....	61
Aktualizowanie operatora IBM MQ Operator i menedżerów kolejek.....	73
Wdrażanie i konfigurowanie menedżerów kolejek przy użyciu operatora IBM MQ Operator.....	82
Użytkowanie produktu IBM MQ przy wykorzystaniu operatora IBM MQ Operator.....	118
Rozwiązywanie problemów związanych z IBM MQ Operator.....	128
Skorowidz interfejsu API operatora IBM MQ Operator.....	130
Budowanie własnego kontenera IBM MQ i kodu wdrażania.....	151
Planowanie wdrożenia własnego obrazu menedżera kolejek IBM MQ przy użyciu kontenera.....	152
Budowanie przykładowego obrazu kontenera menedżera kolejek IBM MQ.....	152
Uruchamianie aplikacji z powiązaniem lokalnymi w oddzielnych kontenerach.....	155
Tworzenie własnej grupy wysokiej dostępności w przypadku tworzenia własnych kontenerów....	157
<b>Uwagi.....</b>	<b>165</b>
Informacje dotyczące interfejsu programistycznego.....	166
Znaki towarowe.....	167




## Multi IBM MQ w kontenerach i IBM Cloud Pak for Integration

Kontenery umożliwiają spakowanie menedżera kolejek IBM MQ lub aplikacji klienckiej IBM MQ wraz ze wszystkimi komponentami zależnymi w zestandaryzowanej jednostce na potrzeby tworzenia oprogramowania.

Produkt IBM MQ można uruchomić przy użyciu IBM MQ Operator w Red Hat® OpenShift®. Można to zrobić za pomocą IBM Cloud Pak for Integration, IBM MQ Advanced lub IBM MQ Advanced for Developers.

Produkt IBM MQ można także uruchomić w kontenerze, który został samodzielnie zbudowany przez użytkownika.

 Więcej informacji na temat IBM MQ Operator można znaleźć na następujących stronach.

## Multi Planowanie wdrożenia IBM MQ w kontenerach

Planując wdrożenie IBM MQ w kontenerach, należy wziąć pod uwagę, że IBM MQ udostępnia różne opcje wsparcia dotyczące architektury, takie jak sposób zapewnienia wysokiej dostępności oraz sposób zabezpieczania menedżerów kolejek.

### O tym zadaniu

Przed rozpoczęciem planowania IBM MQ w architekturze kontenerów należy zapoznać się z podstawowymi pojęciami IBM MQ (patrz [Przegląd techniczny produktu IBM MQ](#)), a także podstawowymi pojęciami Kubernetes/Red Hat OpenShift (patrz [architektura Red Hat OpenShift Container Platform](#)).

### Procedura

- [“Wybór sposobu używania IBM MQ w kontenerach”](#) na stronie 5.
- [“Obsługa operatora IBM MQ Operator”](#) na stronie 6.
- [“Obsługa budowania własnych obrazów kontenera menedżera kolejek IBM MQ”](#) na stronie 13.
- [“Uwagi dotyczące pamięci masowej dla IBM MQ Operator”](#) na stronie 11.
- [“Wysoka dostępność produktu IBM MQ w kontenerach”](#) na stronie 16.
- [“Odtwarzanie po awarii w przypadku IBM MQ w kontenerach”](#) na stronie 18.
- [“Uwierzytelnianie i autoryzacja użytkowników oprogramowania IBM MQ w kontenerach”](#) na stronie 19.

## Wybór sposobu używania IBM MQ w kontenerach

Istnieje wiele opcji korzystania z produktu IBM MQ w kontenerach: można użyć operatora IBM MQ Operator, który korzysta ze wstępnie spakowanych obrazów kontenerów, lub zbudować własne obrazy i kod wdrożenia.

### Korzystanie z IBM MQ Operator



Jeśli planowane jest wdrożenie na serwerze Red Hat OpenShift Container Platform, prawdopodobnie ma być używany produkt IBM MQ Operator.

IBM MQ Operator dodaje nowy dostosowany zasób QueueManager do produktu Red Hat OpenShift Container Platform. Element wykonawczy czuwa nad nowymi definicjami menedżera kolejek, a następnie przekształca je w niezbędne zasoby niskopoziomowe, takie jak zasoby StatefulSet czy Service.

W przypadku rodzimej wysokiej dostępności element wykonawczy może również wykonać złożoną sekwencyjną aktualizację instancji menedżera kolejek. Patrz [“Uwagi dotyczące wykonywania własnej sekwencyjnej aktualizacji menedżera kolejek rodzimej wysokiej dostępności”](#) na stronie 159

Niektóre funkcje produktu IBM MQ nie są obsługiwane w przypadku korzystania z operatora IBM MQ Operator. Aby wykonać jedną z następujących czynności, należy zbudować własne obrazy i wykresy:

- Korzystanie z interfejsów API REST do administrowania lub przesyłania komunikatów
- Użyj dowolnego z następujących komponentów produktu MQ:
  - Agenty Managed File Transfer i ich zasoby. Można jednak użyć konsoli IBM MQ Operator w celu udostępnienia jednej lub większej liczby menedżerów kolejek koordynacji, komend lub agentów.
  - AMQP
  - IBM MQ Bridge to Salesforce
  - IBM MQ Bridge to blockchain (produkt nieobsługiwany w kontenerach)
  - IBM MQ Telemetry Transport (MQTT).
- Opcje dostosowywania w komendach **crtmqm**, **strmqm** i **endmqm**, np. służące do konfigurowanie stron plików dziennika. Większość opcji można skonfigurować za pośrednictwem pliku INI.

Należy pamiętać, że IBM MQ Operator i kontenery szybko ewoluują, dlatego nie są obsługiwane w wydaniach Long Term Support.

IBM MQ Operator zawiera zarówno wstępnie zbudowane obrazy kontenerów, jak i kod wdrożenia przeznaczony do uruchamiania na platformie Red Hat OpenShift Container Platform. IBM MQ Operator może posłużyć do wdrożenia udostępnionego obrazu kontenera IBM MQ lub obrazu kontenera nałożonego na ten udostępniony obraz, ale nie pozwala na wdrażanie niestandardowych, utworzonych przez użytkownika obrazów kontenerów MQ.

## Budowanie własnych obrazów i kodu wdrażania



Jest to najbardziej elastyczne rozwiązanie kontenerowe, ale wymaga zaawansowanych kompetencji w dziedzinie konfigurowania kontenerów i "posiadania na własność" kontenera wynikowego. Jeśli nie planuje się używania Red Hat OpenShift Container Platform, należy zbudować własne obrazy i kod wdrożenia.

Dostępne są przykłady przydatne przy budowaniu własnych obrazów. Patrz sekcja [“Budowanie własnego kontenera IBM MQ i kodu wdrażania”](#) na stronie 151.

### Pojęcia pokrewne

[“Obsługa operatora IBM MQ Operator”](#) na stronie 6

IBM MQ Operator jest obsługiwany tylko w przypadku wdrożenia na platformie Red Hat OpenShift Container Platform.

[“Obsługa budowania własnych obrazów kontenera menedżera kolejek IBM MQ”](#) na stronie 13

Produkt IBM MQ udostępnia kod do budowania kontenera menedżera kolejek produktu IBM MQ w serwisie GitHub. To rozwiązanie jest oparte na procesie, z którego korzysta IBM w celu zbudowania własnego obsługiwanego kontenera. Można użyć tego repozytorium GitHub, aby uprościć i przyspieszyć budowanie własnych obrazów kontenerów.



IBM MQ Operator jest obsługiwany tylko w przypadku wdrożenia na platformie Red Hat OpenShift Container Platform.

Produkt IBM MQ Operator korzysta z obrazów opartych na wersjach IBM MQ Continuous Delivery (CD). Wersja Extended Update Support (EUS) jest jednak dostępna w przypadku produktu IBM Cloud Pak for Integration. Wydania CD są obsługiwane przez maksymalnie jeden rok lub przez okres dwóch wydań CD, w zależności od tego, który z tych okresów jest dłuższy. Wydania Long Term Support produktu IBM MQ nie są dostępne za pośrednictwem IBM MQ Operator. IBM Cloud Pak for Integration 2020.4.1 to

wydanie z rozszerzoną obsługą aktualizacji (EUS), które jest obsługiwane przez 18 miesięcy, jeśli używana jest wersja produktu IBM MQ oznaczona jako -eus. W przeciwnym przypadku produkt IBM MQ 9.2 jest traktowany jako wydanie Continuous Delivery z IBM MQ Operator.

IBM MQ Operator używa obrazów kontenerów zawierających instalację IBM MQ w systemie Red Hat Universal Base Image (UBI). Instalacja obejmuje biblioteki kluczy Linux® i programy narzędziowe używane przez IBM MQ. System UBI jest obsługiwany przez produkt Red Hat na platformie Red Hat OpenShift.

Produkt IBM MQ Operator jest obsługiwany w architekturach amd64i s390x (z/Linux).

### Pojęcia pokrewne

“Obsługa budowania własnych obrazów kontenera menedżera kolejek IBM MQ” na stronie 13  
 Produkt IBM MQ udostępnia kod do budowania kontenera menedżera kolejek produktu IBM MQ w serwisie GitHub. To rozwiązanie jest oparte na procesie, z którego korzysta IBM w celu zbudowania własnego obsługiwanego kontenera. Można użyć tego repozytorium GitHub, aby uprościć i przyspieszyć budowanie własnych obrazów kontenerów.

## OpenShift CP4I CD EUS Wersja zgodna z operatorem IBM MQ Operator

Odwzorowanie między obsługiwanymi wersjami produktów IBM MQ, Red Hat OpenShift Container Platform i IBM Cloud Pak for Integration.

- [“Dostępne wersje produktu IBM MQ” na stronie 7](#)
- [“Kompatybilne wersje środowiska Red Hat OpenShift Container Platform” na stronie 8](#)
- [“Wersja IBM Cloud Pak for Integration” na stronie 8](#)
- [“Dostępne wersje produktu IBM MQ w starszych operatorach” na stronie 8](#)
- [“Zgodne wersje produktu Red Hat OpenShift Container Platform dla starszych operatorów” na stronie 9](#)

### Dostępne wersje produktu IBM MQ

Kanał operatora	Wersja operatora	Wersja IBM MQ							
		9.1.5	9.2.0 CD	9.2.0 EUS	9.2.1	9.2.2	9.2.3	9.2.4	9.2.5
v1.6	1.6	⚠	⚠	→	⚠	●	●		
v1.7	1.7	⚠	⚠	→	⚠	●	●	●	
v1.8	1.8	⚠	⚠	→	⚠	⚠	●	●	●

Klucz:

- Dostępna obsługa rozwiązania Continuous Delivery
- Dostępna obsługa rozwiązania Extended Update Support
- Dostępne tylko podczas migracji z operandu Extended Update Support do operandu Continuous Delivery.
- ⚠ Nieaktualne. Ponieważ obsługa wersji produktu IBM MQ dobiega końca, mogą być one nadal konfigurowane w operatorze, ale nie kwalifikują się już do wsparcia i mogą zostać usunięte w przyszłych wersjach.

Sekcja [“Historia wydań operatora IBM MQ Operator” na stronie 21](#) zawiera szczegółowe informacje na temat każdej wersji, w tym szczegóły dotyczące składników, zmian i poprawek.

## Kompatybilne wersje środowiska Red Hat OpenShift Container Platform

Kanał operatora	Wersja operatora	Wersja Red Hat OpenShift Container Platform <sup>1</sup>				
		4.6	4.7 <sup>2</sup>	4.8	4.9	4.10
v1.6	1.6	●	●	●	●	●
v1.7	1.7	●	●	●	●	●
v1.8	1.8	●	●	●	●	●

Klucz:

- Dostępna obsługa rozwiązania Continuous Delivery
- Dostępna obsługa rozwiązania Extended Update Support

### Wersja IBM Cloud Pak for Integration

Wersja IBM MQ Operator 1.8.x jest obsługiwana jako część wersji IBM Cloud Pak for Integration 2021.4.1 lub niezależnie.

Produkt IBM MQ Operator 1.7.x jest obsługiwany jako część produktu IBM Cloud Pak for Integration w wersji 2021.4.1 lub jako autonomiczny produkt.

Produkt IBM MQ Operator 1.6.x jest obsługiwany jako część produktu IBM Cloud Pak for Integration w wersji 2021.2.1, 2021.3.1 lub jako autonomiczny produkt.

Serwer IBM MQ Operator 1.5.x nie jest już obsługiwany.

Serwer IBM MQ Operator 1.4.x nie jest już obsługiwany.

Serwer IBM MQ Operator 1.3.x nie jest już obsługiwany.

Serwer IBM MQ Operator 1.2.x nie jest już obsługiwany.

Pliki IBM MQ Operators 1.1.x i 1.0.x nie są już obsługiwane.

### Dostępne wersje produktu IBM MQ w starszych operatorach

Poniższa tabela ma zastosowanie do wersji produktu IBM MQ Operator, których cykl życia został zakończony.

Kanał operatora	Wersja operatora	Wersja IBM MQ							
		9.1.5	9.2.0 CD	9.2.0 EUS	9.2.1	9.2.2	9.2.3	9.2.4	9.2.5
v1.0	1.0	⚠							
v1.1	1.1	⚠	⚠						
v1.2	1.2	⚠	⚠						
v1.3-eus	1.3	⚠	⚠	⚠					

<sup>1</sup> Wersje produktu Red Hat OpenShift Container Platform są objęte własnymi datami wsparcia. Więcej informacji na ten temat zawiera sekcja dotycząca [strategii cyklu życia Red Hat OpenShift Container Platform](#).

<sup>2</sup> IBM MQ Operator zależy od IBM Cloud Pak foundational services. Jeśli ma być używany produkt Red Hat OpenShift Container Platform 4.7, należy najpierw zaktualizować wersję IBM Cloud Pak foundational services.



Kanał operatora	Wersja operatora	Wersja IBM MQ							
		9.1.5	9.2.0 CD	9.2.0 EUS	9.2.1	9.2.2	9.2.3	9.2.4	9.2.5
v1.4	1.4	⚠	⚠	→	⚠				
v1.5	1.5	⚠	⚠	→	⚠	⚠			

Klucz:

→

Dostępne tylko podczas migracji z operandu Extended Update Support do operandu Continuous Delivery.

⚠

Nieaktualne. Ponieważ obsługa wersji produktu IBM MQ dobiega końca, mogą być one nadal konfigurowane w produkcie IBM MQ Operator, ale nie kwalifikują się już do wsparcia.

Sekcja [“Historia wydań operatora IBM MQ Operator”](#) na stronie 21 zawiera szczegółowe informacje na temat każdej wersji, w tym szczegóły dotyczące składników, zmian i poprawek.

## Zgodne wersje produktu Red Hat OpenShift Container Platform dla starszych operatorów

Poniższa tabela ma zastosowanie do wersji produktu IBM MQ Operator, których cykl życia został zakończony.

Kanał operatora	Wersja operatora	Wersja Red Hat OpenShift Container Platform <sup>3</sup>						
		4.4 <sup>4</sup>	4.5 <sup>5</sup>	4.6	4.7 <sup>6</sup>	4.8	4.9	4.10
v1.0	1.0	⚠	⚠	⚠	⚠			
v1.1	1.1	⚠	⚠	⚠	⚠	⚠		
v1.2	1.2	⚠	⚠	⚠	⚠	⚠		
v1.3-eus	1.3			⚠	→	→	→	→
v1.4	1.4			⚠	⚠	⚠	⚠	
v1.5	1.5			⚠	⚠	⚠	⚠	⚠

Klucz:

→

Dostępne tylko podczas migracji z operandu Extended Update Support do operandu Continuous Delivery.

<sup>3</sup> Wersje produktu Red Hat OpenShift Container Platform są objęte własnymi datami wsparcia. Więcej informacji na ten temat zawiera sekcja dotycząca [strategii cyklu życia Red Hat OpenShift Container Platform](#).

<sup>4</sup> Cykl życia produktu Red Hat OpenShift Container Platform 4.4 został zakończony. Więcej informacji na ten temat zawiera sekcja dotycząca [strategii cyklu życia Red Hat OpenShift Container Platform](#).

<sup>5</sup> Cykl życia produktu Red Hat OpenShift Container Platform 4.5 został zakończony. Więcej informacji na ten temat zawiera sekcja dotycząca [strategii cyklu życia Red Hat OpenShift Container Platform](#).

<sup>6</sup> IBM MQ Operator zależy od IBM Cloud Pak foundational services. Jeśli ma być używany produkt Red Hat OpenShift Container Platform 4.7, należy najpierw zaktualizować wersję IBM Cloud Pak foundational services.



Cykl życia wersji IBM MQ Operator został zakończony, ale wcześniej był dostępny w tej wersji produktu Red Hat OpenShift Container Platform

OpenShift

CP4I

## Zależności operatora IBM MQ Operator

Produkt IBM MQ Operator zależy od operatora IBM Cloud Pak foundational services Operator, który także instaluje operatora IBM Operand Deployment Lifecycle Manager (ODLM) Operator. Te operatory zostaną zainstalowane automatycznie podczas instalowania operatora IBM MQ Operator. Te zależne operatory mają niewielki wpływ na procesor i pamięć. Są one używane do wdrażania dodatkowych zasobów w niektórych okolicznościach.

Podczas tworzenia zasobu QueueManager operator IBM MQ Operator utworzy żądanie OperandRequest dotyczące dodatkowych potrzebnych dla niego usług. Żądanie OperandRequest jest realizowane przez operator ODLM, który w razie potrzeby zainstaluje i utworzy instancję wymaganych usług. Wymagane usługi są określane na podstawie zaakceptowanej umowy licencyjnej podczas wdrażania menedżera kolejek oraz na podstawie których żądane są komponenty menedżera kolejek.

- Jeśli zostanie wybrana licencja produktu IBM MQ Advanced lub IBM MQ Advanced for Developers, nie są wymagane żadne dodatkowe usługi. Na przykład w następującym przypadku IBM Cloud Pak foundational services nie są używane:

```
spec:
  license:
    accept: true
    license: L-APIG-BZDDDY
    use: "Production"
```

- Jeśli zostanie wybrana licencja produktu IBM Cloud Pak for Integration i zostanie wybrana opcja aktywowania serwera WWW, produkt IBM MQ Operator utworzy również instancję operatora IBM Identity and Access Management (IAM) Operator w celu włączenia pojedynczego logowania. IAM Operator jest już dostępny, jeśli wcześniej zainstalowano operator IBM Cloud Pak for Integration Operator. Na przykład:

```
spec:
  license:
    accept: true
    license: L-RJON-BUVMQX
    use: "Production"
```

Jeśli jednak serwer WWW zostanie wyłączony, nie będą wymagane żadne IBM Cloud Pak foundational services. Na przykład:

```
spec:
  license:
    accept: true
    license: L-RJON-BUVMQX
    use: "Production"
  web:
    enabled: false
```

Starsze wersje produktu IBM MQ Operator zawsze żądały instalacji operatora IBM Licensing Operator (i jego zależności) w celu śledzenia użycia licencji. Począwszy od IBM MQ Operator 1.5 usługa licencjonowania nie jest wymagana, a użytkownik musi ją zażądać oddzielnie.

IBM MQ Operator wymaga 1 rdzenia CPU i 1 GB pamięci. Szczegółowe informacje na temat wymagań sprzętowych i programowych dla operatorów zależnych można znaleźć w sekcji [Wymagania sprzętowe i zalecenia dla usług podstawowych](#).

Można wybrać poziom wykorzystania CPU i pamięci przez menedżery kolejek. Więcej informacji zawiera temat [“.spec.queueManager.resources”](#) na stronie 140.

### Odsyłacze pokrewne

[“Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1”](#) na stronie 130

## Operator

IBM MQ Operator wymaga uprawnień o zasięgu klastra do zarządzania elementami webhook i przykładami oraz do odczytu informacji o klasach pamięci masowych i wersji klastra.

IBM MQ Operator wymaga następujących uprawnień o zasięgu klastra:

- Uprawnienia do zarządzania elementami web hook przyjęcia. Umożliwiają tworzenie, pobieranie i aktualizowanie konkretnych elementów webhook używanych w procesie tworzenia kontenerów i zarządzania kontenerami udostępnianymi przez operator.
  - Grupy interfejsu API: **admissionregistration.k8s.io**
  - Zasoby: **validatingwebhookconfigurations**
  - Czasowniki: **create, get, update**
- Uprawnienie do tworzenia zasobów i zarządzania zasobami, które są używane w konsoli produktu Red Hat OpenShift na potrzeby udostępniania przykładów i fragmentów kodu podczas tworzenia zasobów niestandardowych.
  - Grupy interfejsu API: **console.openshift.io**
  - Zasoby: **consoleyamlsamples**
  - Czasowniki: **create, get, update, delete**
- Uprawnienie do odczytu wersji klastra. Dzięki niemu operator może przekazać dalej wszystkie problemy ze środowiskiem klastrowym.
  - Grupy interfejsu API: **config.openshift.io**
  - Zasoby: **clusterversions**
  - Czasowniki: **get, list, watch**
- Uprawnienie do odczytu klas pamięci masowej w klastrze. Dzięki nim operator może z powrotem kierować problemy z wybranymi klasami pamięci masowej w kontenerach.
  - Grupy interfejsu API: **storage.k8s.io**
  - Zasoby: **storageclasses**
  - Czasowniki: **get, list**

## Operator

IBM MQ Operator może działać w dwóch trybach pamięci masowej:

- **Efemeryczna pamięć masowa** jest używana, gdy wszystkie informacje o stanie kontenera mogą zostać bezpiecznie usunięte po zrestartowaniu kontenera. Ten tryb jest powszechnie używany w środowiskach demonstracyjnych, a także podczas programowania z autonomicznymi menedżerami kolejek.
- **Trwała pamięć masowa** to typowa konfiguracja używana dla produktu IBM MQ, która zapewnia dostępność istniejących konfiguracji, dzienników i trwałych komunikatów po zrestartowaniu kontenera.

Operator IBM MQ Operator umożliwia dostosowanie parametrów pamięci masowej, które mogą znacznie różnić się w zależności od środowiska i trybu.



### Efemeryczna pamięć masowa

IBM MQ jest aplikacją stanową i zachowuje swój stan na wypadek restartu. Jeśli używana jest efemeryczna pamięć masowa, wszystkie informacje o stanie menedżera kolejek zostaną utracone podczas restartu. Dotyczy to:

- wszystkich komunikatów;

- całości stanu komunikacji między menedżerami kolejek (numerów kolejnych komunikatów w kanale);
- tożsamość klastra MQ menedżera kolejek;
- stanu wszystkich transakcji;
- całości konfiguracji menedżera kolejek;
- wszystkich lokalnych danych diagnostycznych.

Z tego powodu należy rozważyć, czy pamięć efemeryczna jest faktycznie odpowiednim wyborem w scenariuszu produkcyjnym, testowym lub programistycznym. Czy, na przykład, wiadomo, że wszystkie komunikaty są nietrwałe, a menedżer kolejek nie jest elementem klastra produktu MQ? Przy restarcie usuwany jest nie tylko cały stan środowiska przesyłania komunikatów, lecz także konfiguracja menedżera kolejek. Aby można było korzystać z całkowicie efemerycznego kontenera, konfiguracja IBM MQ musi być dodana do samego obrazu kontenera (więcej informacji na ten temat zawiera sekcja [“Budowanie obrazu z niestandardowymi plikami MQSC i INI przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift”](#) na stronie 115). Jeśli ten warunek nie zostanie spełniony, konieczne będzie skonfigurowanie IBM MQ przy każdym restarcie kontenera.

  Na przykład, aby skonfigurować IBM MQ z efemeryczną pamięcią masową, typ pamięci masowej w zasobie QueueManager powinien być następujący:

```
queueManager:
  storage:
    queueManager:
      type: ephemeral
```

## Trwała pamięć masowa



Produkt IBM MQ zwykle działa z pamięcią trwałą, która zachowuje komunikaty i konfigurację menedżera kolejek po każdym restarcie. Jest to konfiguracja domyślna. Ze względu na różnice między rozwiązaniami różnych dostawców pamięci masowej często wymagane jest zmodyfikowanie konfiguracji. Poniżej przedstawiono, jako przykład, często używane pola służące do modyfikowania konfiguracji pamięci masowej MQ w interfejsie API v1beta1:

- [spec.queueManager.availability](#) steruje trybem dostępności. W trybie `SingleInstance` wymagana jest tylko pamięć masowa `ReadWriteOnce`, natomiast tryb `MultiInstance` wymaga klasy pamięci, która obsługuje zapis/odczyt `ReadWriteMany` z właściwą charakterystyką blokowania plików. Produkt IBM MQ udostępnia [instrukcję obsługi](#) i [instrukcję testowania](#). Tryb dostępności wpływa również na układ trwałego woluminu. Więcej informacji na ten temat zawiera sekcja [“Wysoka dostępność produktu IBM MQ w kontenerach”](#) na stronie 16.
- [spec.queueManager.storage](#) steruje ustawieniami poszczególnych pamięci masowych. Menedżer kolejek może być skonfigurowany tak, by używał od jednego do czterech woluminów trwałych.

W poniższym przykładzie przedstawiono fragment prostej konfiguracji menedżera kolejek z jedną instancją:

```
spec:
  queueManager:
    storage:
      queueManager:
        enabled: true
```

W poniższym przykładzie przedstawiono fragment kodu konfiguracji menedżera kolejek z wieloma instancjami, który ma inną niż domyślną klasę pamięci masowej, a także plikową pamięć masową, która wymaga grup dodatkowych:

```
spec:
  queueManager:
    availability:
      type: MultiInstance
    storage:
      queueManager:
        class: ibmc-file-gold-gid
```

```
persistedData:
  enabled: true
  class: ibmc-file-gold-gid
recoveryLogs:
  enabled: true
  class: ibmc-file-gold-gid
securityContext:
  supplementalGroups: [99]
```

**Uwaga:** Można także skonfigurować grupy dodatkowe z menedżerami kolejek z jedną instancją.

**Uwaga:** Jeśli używana jest rodzima HA (patrz “Wysoka dostępność produktu IBM MQ w kontenerach” na stronie 16), systemy plików współużytkowanych nie są wymagane. W szczególności nie należy używać systemu NFSv3.

## Linux Obługa budowania własnych obrazów kontenera menedżera kolejek IBM MQ

Produkt IBM MQ udostępnia kod do budowania kontenera menedżera kolejek produktu IBM MQ w serwisie GitHub. To rozwiązanie jest oparte na procesie, z którego korzysta IBM w celu zbudowania własnego obsługiwanego kontenera. Można użyć tego repozytorium GitHub, aby uprościć i przyspieszyć budowanie własnych obrazów kontenerów.

Kod jest dostępny w repozytorium mq-container w serwisie GitHub tutaj: <https://github.com/ibm-messaging/mq-container>. Jest on udostępniany na licencji Apache 2.0. Wsparcie jest udzielane przez społeczność.

Repozytorium nie używa standardowych pakietów Linux rpm, ale skompresowanego pakietu dla wdrożeń kontenerów. Korzyścią z tego podejścia jest to, że można korzystać z bezpieczniejszych środowisk kontenerów bez konieczności eskalowania uprawnień. Jednak to ma wpływ na dostępne opcje zabezpieczeń, ponieważ IBM MQ tradycyjnie używa eskalowania uprawnień do uwierzytelniania opartego na systemie operacyjnym. W przypadku wdrożenia kontenera używanie uwierzytelniania opartego na systemie operacyjnym nie jest zwykle dobrym rozwiązaniem. Zamiast tego można korzystać z uwierzytelniania wzajemnego TLS lub LDAP. Przy użyciu produktu IBM MQ Advanced for Developers można również korzystać z uwierzytelniania opartego na plikach, co pozwala użytkownikom na szybkie rozpoczęcie pracy.

Menedżer kolejek replikowanych danych (RDQM) nie jest obsługiwany w środowisku kontenerowym. Podobne możliwości jak w przypadku RDQM można uzyskać za pomocą produktu “Rodzima wysoka dostępność (Native HA)” na stronie 95.

### Pojęcia pokrewne

“Obługa operatora IBM MQ Operator” na stronie 6

IBM MQ Operator jest obsługiwany tylko w przypadku wdrożenia na platformie Red Hat OpenShift Container Platform.

[Obrazy nieinstalacyjne produktu IBM MQ](#)

## Linux Adnotacje licencji podczas budowania własnego obrazu kontenera IBM MQ

Adnotacje licencji umożliwiają śledzenie wykorzystania na podstawie limitów zdefiniowanych w kontenerze, a nie na komputerze bazowym. Istnieje możliwość takiego skonfigurowania klientów, by kontener został wdrożony z konkretnymi adnotacjami, które będą używane przez IBM License Service do śledzenia wykorzystania.

Podczas wdrażania samodzielnie zbudowanego obrazu kontenera IBM MQ stosowana jest zwykle jedna z dwóch strategii licencjonowania:

- Licencja na cały komputer, na którym działa kontener.
- Licencja na kontener z określonymi limitami.

Obie opcje są dostępne dla klientów, a dodatkowe szczegóły można znaleźć na [stronie licencji](#) na kontenery IBM w serwisie Passport Advantage.

Jeśli kontener IBM MQ ma być licencjonowany na podstawie limitów kontenera, konieczne jest zainstalowanie produktu IBM License Service w celu śledzenia wykorzystania. Więcej informacji na temat obsługiwanych środowisk oraz instrukcje instalacji można znaleźć na stronie [ibm-licensing-operator](#) w serwisie GitHub.

Produkt IBM License Service jest zainstalowany w klastrze Kubernetes, w którym wdrożono kontener IBM MQ, a do śledzenia wykorzystania używane są adnotacje zasobnika. Z tego powodu klienci muszą wdrażać zasobnik z konkretnymi adnotacjami, które będą następnie używane przez produkt IBM License Service. W zależności od posiadanych upoważnień i możliwości wdrożonych w kontenerze należy użyć jednej lub kilku następujących adnotacji:

- [“Kontener IBM MQ Advanced” na stronie 14](#)
- [“Kontener IBM MQ Advanced High Availability Replica” na stronie 14](#)
- [“Kontener IBM MQ Base” na stronie 14](#)
- [“Kontener IBM MQ Base High Availability Replica” na stronie 15](#)
- [“Kontener IBM MQ Advanced for Developers” na stronie 15](#)
- [“Kontener IBM MQ Advanced z upoważnieniem CP4I \(produkcyjnym\)” na stronie 15](#)
- [“Kontener IBM MQ Advanced High Availability Replica z upoważnieniem CP4I \(produkcyjnym\)” na stronie 15](#)
- [“Kontener IBM MQ Advanced z upoważnieniem CP4I \(nieprodukcyjnym\)” na stronie 15](#)
- [“Kontener IBM MQ Advanced High Availability Replica z upoważnieniem CP4I \(nieprodukcyjnym\)” na stronie 15](#)
- [“IBM MQ Base z uprawnieniem CP4I \(produkcyjnym\)” na stronie 16](#)
- [“IBM MQ Base High Availability Replica z uprawnieniem CP4I \(produkcyjnym\)” na stronie 16](#)
- [“IBM MQ Base z uprawnieniem CP4I \(nieprodukcyjnym\)” na stronie 16](#)
- [“IBM MQ Base High Availability Replica z uprawnieniem CP4I \(nieprodukcyjnym\)” na stronie 16](#)

## Kontener IBM MQ Advanced

```
productName: "IBM MQ Advanced"  
productID: "208423bb063c43288328b1d788745b0c"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
```

## Kontener IBM MQ Advanced High Availability Replica

```
productName: "IBM MQ Advanced High Availability Replica"  
productID: "546cb719714942c18748137ddd8d5659"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
```

## Kontener IBM MQ Base

```
productName: "IBM MQ"  
productID: "c661609261d5471fb4ff8970a36bccea"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
```

## Kontener IBM MQ Base High Availability Replica

```
productName: "IBM MQ High Availability Replica"
productID: "2a2a8e0511c849969d2f286670ea125e"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
```

## Kontener IBM MQ Advanced for Developers

```
productName: "IBM MQ Advanced for Developers"
productID: "2f886a3eefbe4ccb89b2adb97c78b9cb"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "FREE"
```

## Kontener IBM MQ Advanced z upoważnieniem CP4I (produkcyjnym)

```
productName: "IBM MQ Advanced with CP4I License"
productID: "208423bb063c43288328b1d788745b0c"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productCloudpakRatio: "2:1"
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

## Kontener IBM MQ Advanced High Availability Replica z upoważnieniem CP4I (produkcyjnym)

```
productName: "IBM MQ Advanced High Availability Replica with CP4I License"
productID: "546cb719714942c18748137ddd8d5659"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productCloudpakRatio: "10:1"
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

## Kontener IBM MQ Advanced z upoważnieniem CP4I (nieprodukcyjnym)

```
productName: "IBM MQ Advanced for Non-Production with CP4I License"
productID: "21dfe9a0f00f444f888756d835334909"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productCloudpakRatio: "4:1"
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

## Kontener IBM MQ Advanced High Availability Replica z upoważnieniem CP4I (nieprodukcyjnym)

```
productName: "IBM MQ Advanced High Availability Replica for Non-Production with CP4I License"
productID: "b3f8f984007d47fb981221589cc50081"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productCloudpakRatio: "20:1"
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

## IBM MQ Base z uprawnieniem CP4I (produkcyjnym)

```
productName: "IBM MQ with CP4I License"
productID: "c661609261d5471fb4ff8970a36bccea"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productCloudpakRatio: "4:1"
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

## IBM MQ Base High Availability Replica z uprawnieniem CP4I (produkcyjnym)

```
productName: "IBM MQ High Availability Replica with CP4I License"
productID: "2a2a8e0511c849969d2f286670ea125e"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productCloudpakRatio: "20:1"
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

## IBM MQ Base z uprawnieniem CP4I (nieprodukcyjnym)

```
productName: "IBM MQ with CP4I License Non-Production"
productID: "151bec68564a4a47a14e6fa99266def"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productCloudpakRatio: "8:1"
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

## IBM MQ Base High Availability Replica z uprawnieniem CP4I (nieprodukcyjnym)

```
productName: "IBM MQ High Availability Replica with CP4I License Non-Production"
productID: "f5d0e21c013c4d4b8b9b2ce701f31928"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productCloudpakRatio: "40:1"
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

## **Wysoka dostępność produktu IBM MQ w kontenerach**

Dostępne są trzy konfiguracje z operatorem IBM MQ Operator zapewniające wysoką dostępność: **menedżer kolejek o rodzimej wysokiej dostępności** (replika aktywna i dwie repliki rezerwowe), **wieloinstancyjny menedżer kolejek** (para aktywny-rezerwowy ze wspólnym sieciowym systemem plików) i **pojedynczy menedżer kolejek elastycznych** (proste rozwiązanie zapewniające wysoką dostępność z wykorzystaniem sieciowej pamięci masowej). W tych dwóch ostatnich konfiguracjach dostępność odtwarzalnych danych zależy od systemu plików. W związku z tym, gdy nie jest stosowana rodzima wysoka dostępność, dostępność systemu plików jest krytyczna dla dostępności menedżera kolejek. Jeśli odtwarzanie danych jest istotne, system plików powinien zapewnić nadmiarowość przez replikację.

Dostępność **komunikatów** i **usługi** należy rozpatrywać oddzielnie. Gdy używany jest IBM MQ for Multiplatforms, każdy komunikat jest przechowywany dokładnie w jednym menedżerze kolejek. Jeśli więc menedżer kolejek stanie się niedostępny, użytkownik tymczasowo utraci dostęp do przechowywanych w nim komunikatów. Aby uzyskać wysoką dostępność komunikatów, należy jak najszybciej przywrócić sprawność menedżera kolejek. Wysoką dostępność usługi można osiągnąć, korzystając z wielu instancji kolejek dla aplikacji klienckich, na przykład przy użyciu jednolitego klastra IBM MQ.



Menedżer kolejek może być rozpatrywany w dwóch aspektach: danych przechowywanych na dysku i działających procesów, które umożliwiają dostęp do danych. Każdy menedżer kolejek może być przeniesiony do innego węzła Kubernetes, o ile zachowa te same dane (w woluminach trwałych Kubernetes Persistent Volume) i będzie nadal adresowalny w sieci w aplikacji klienckich. W środowisku Kubernetes usługa (Service) jest używana do zapewnienia spójności tożsamości sieciowej.

Produkt IBM MQ polega na dostępności danych w woluminach trwałych. Dlatego dostępność pamięci masowej, w której ulokowane są woluminy trwałe, ma zasadnicze znaczenie dla dostępności menedżera kolejek, ponieważ IBM MQ nie może osiągnąć większej dostępności niż pamięć masowa, z której korzysta. Aby zapewnić odporność na wyłączenie całej strefy dostępności, należy użyć dostawcy woluminu, który replikuje zapisy na dysku do innej strefy.

## Menedżer kolejek o rodzimej wysokiej dostępności



Menedżery kolejek o rodzimej wysokiej dostępności są dostępne począwszy od wersji IBM Cloud Pak for Integration 2021.2.1 z operatorem IBM MQ Operator w wersji 1.6 lub wyższej oraz produktem IBM MQ w wersji 9.2.3 lub wyższej.

Menedżery kolejek z rodzimą wysoką dostępnością mają zasobnik **aktywny** i dwa zasobniki **replik** Kubernetes, które działają jako część zestawu stanowego Kubernetes z dokładnie trzema replikami, z których każda ma własny zestaw woluminów trwałych Kubernetes. Wymagania IBM MQ dotyczące współużytkowanych systemów plików mają także zastosowanie w przypadku używania pojedynczego menedżera kolejek o rodzimej wysokiej dostępności (z wyjątkiem blokowania opartego na dzierżawie), ale nie jest konieczne korzystanie ze współużytkowanego systemu plików. Można użyć blokowej pamięci masowej z odpowiednim systemem plików. Na przykład *xfs* lub *ext4*. Na czas przywracania sprawności pojedynczego menedżera kolejek o rodzimej wysokiej dostępności wpływają następujące czynniki:

1. Czas, po jakim instancje replik wykrywają, że instancja aktywna uległa awarii. Ten czynnik jest konfigurowalny.
2. Czas potrzebny sondzie gotowości zasobnika Kubernetes na wykrycie, że gotowy jest inny kontener, i przekierowanie ruchu sieciowego. Ten czynnik jest konfigurowalny.
3. Czas potrzebny na ponowne nawiązanie połączenia przez klienty IBM MQ.

Więcej informacji na ten temat zawiera sekcja [“Rodzima wysoka dostępność \(Native HA\)”](#) na stronie 95

## Wieloinstancyjny menedżer kolejek



Wieloinstancyjne menedżery kolejek korzystają z **aktywnego i rezerwowego** Kubernetes zasobnika, które działają jako część zestawu stanowego Kubernetes z dokładnie dwiema replikami i jednym zestawem woluminów trwałych Kubernetes. Dzienniki transakcji i dane menedżera kolejek są przechowywane na dwóch woluminach trwałych, we współużytkowanym systemie plików.

Wieloinstancyjny menedżer kolejek wymaga, aby zarówno zasobnik **aktywny**, jak i **rezerwowo** miały współbieżny dostęp do woluminu trwałego. Aby uzyskać taką konfigurację, należy użyć trwałych woluminów Kubernetes z parametrem **access mode** ustawionym na `ReadWriteMany`. Woluminy muszą również spełniać wymagania produktu IBM MQ dotyczące współużytkowanych systemów plików, ponieważ produkt IBM MQ zakłada automatyczne zwalnianie blokad plików, gdy inicjuje przetęczenie awaryjne menedżera kolejek. IBM MQ generuje listę przetestowanych systemów plików.

Na czas przywracania sprawności wieloinstancyjnego menedżera kolejek menedżera kolejek wpływają następujące czynniki:

1. Czas, jaki upływa od awarii do zwolnienia we współużytkowanym systemie plików blokad, które zostały pierwotnie uzyskane przez instancję aktywną.
2. Czas potrzebny instancji rezerwowej na uzyskanie blokad, a następnie rozpoczęcie działania.
3. Czas potrzebny sondzie gotowości zasobnika Kubernetes na wykrycie, że gotowy jest inny kontener, i przekierowanie ruchu sieciowego. Ten czynnik jest konfigurowalny.

4. Czas potrzebny na ponowne nawiązanie połączenia przez klienty IBM MQ.

## Pojedynczy menedżer kolejek elastycznych

Multi

Pojedynczy menedżer kolejek elastycznych jest pojedynczą instancją menedżera kolejek uruchomioną w jednym zasobniku Kubernetes, gdzie Kubernetes monitoruje menedżera kolejek i w razie potrzeby zastępuje zasobnik nowym.

Wymagania IBM MQ dotyczące współużytkowanych systemów plików mają także zastosowanie w przypadku używania pojedynczego menedżera kolejek elastycznych (z wyjątkiem blokowania opartego na dzierżawie), ale nie jest konieczne korzystanie ze współużytkowanego systemu plików. Można użyć blokowej pamięci masowej z odpowiednim systemem plików. Na przykład *xfs* lub *ext4*.

Na czas przywracania sprawności pojedynczego menedżera kolejek elastycznych wpływają następujące czynniki:

1. Czas pracy sondy działania i liczba tolerowanych przez nią niepowodzeń. Ten czynnik jest konfigurowalny.
2. Czas potrzebny programowi planującemu Kubernetes na przeniesienie harmonogramu niesprawnego zasobnika do nowego węzła.
3. Czas potrzebny na pobranie obrazu kontenera do nowego węzła. Jeśli parametr strategii **imagePullPolicy** jest ustawiony na `IfNotPresent`, obraz może już być dostępny w tym węźle.
4. Czas potrzebny na uruchomienie nowej instancji menedżera kolejek.
5. Czas potrzebny na wykrycie gotowości kontenera przez sondę gotowości zasobnika Kubernetes. Ten czynnik jest konfigurowalny.
6. Czas potrzebny na ponowne nawiązanie połączenia przez klienty IBM MQ.

### Ważne:

Mimo że pojedynczy menedżer kolejek elastycznych oferuje pewne korzyści, należy wziąć pod uwagę faktyczną możliwość lub brak możliwości osiągnięcia wymaganej dostępności w razie awarii węzłów.

W środowisku Kubernetes niesprawny zasobnik zwykle szybko odzyskuje sprawność, ale awaria całego węzła jest obsługiwana w inny sposób. W przypadku korzystania z obciążenia stanowego, takiego jak IBM MQ z atrybutem Kubernetes `StatefulSet`, jeśli węzeł główny Kubernetes utraci kontakt z węzłem roboczym, nie może określić, czy węzeł uległ awarii lub czy po prostu utracił połączenie sieciowe. Oznacza to, że w takim przypadku Kubernetes **nie podejmuje żadnego działania** do momentu wystąpienia jednego z następujących zdarzeń:

1. Węzeł wróci do stanu, w którym węzeł główny Kubernetes będzie mógł się z nim komunikować.
2. W węźle głównym Kubernetes zostanie podjęte działanie administracyjne w celu jawnego usunięcia zasobnika. Nie zawsze oznacza to zatrzymanie zasobnika — powoduje po prostu usunięcie go ze składnicy Kubernetes. Dlatego wspomniane działanie administracyjne należy wykonać z zachowaniem ostrożności.

### Zadania pokrewne

[“Konfigurowanie wysokiej dostępności na potrzeby menedżerów kolejek przy użyciu rozwiązania IBM MQ Operator” na stronie 95](#)

### Odsyłacze pokrewne

[Konfiguracje o wysokiej dostępności](#)

OpenShift

CP4I

Kubernetes

## Odtwarzanie po awarii w przypadku IBM MQ

### w kontenerach

Należy wziąć pod uwagę rodzaj awarii, na którą chcemy się przygotować. W środowiskach chmurowych korzystanie ze stref dostępności zapewnia pewien poziom tolerancji dla awarii i jest znacznie łatwiejsze w użyciu. Jeśli istnieje nieparzysta liczba centrów przetwarzania danych (dla kworum) i łącze sieciowe

o niskim opóźnieniu, można potencjalnie uruchomić jeden klaster Red Hat OpenShift Container Platform lub Kubernetes z wieloma strefami dostępności, każdy w oddzielnym położeniu fizycznym. W tym temacie opisano zagadnienia dotyczące odtwarzania po awarii w przypadku, gdy nie można spełnić tych kryteriów: oznacza to, że istnieje parzysta liczba centrów przetwarzania danych lub łącze sieciowe o dużym opóźnieniu.

W kontekście odtwarzania po awarii należy wziąć pod uwagę następujące kwestie:

- Replikacja danych produktu IBM MQ (znajdujących się w co najmniej jednym zasobie produktu PersistentVolume) do miejsca usuwania skutków awarii
- Ponowne utworzenie menedżera kolejek przy użyciu replikowanych danych
- Identyfikator sieciowy menedżera kolejek widoczny dla aplikacji klienckich IBM MQ oraz innych menedżerów kolejek. Tym identyfikatorem może być, na przykład, wpis DNS.

Dane trwałe muszą być replikowane, synchronicznie albo asynchronicznie, do ośrodka odtwarzania po awarii. Jest to zazwyczaj kwestia specyficzna dla dostawcy pamięci masowej, ale można tego również dokonać za pomocą VolumeSnapshot. Więcej informacji na temat obrazów stanu woluminów zawiera sekcja [Obrazy stanu woluminów CSI](#).

Podczas odtwarzania po awarii konieczne będzie ponowne utworzenie instancji menedżera kolejek w nowym klastrze Kubernetes przy użyciu replikowanych danych. Jeśli używasz IBM MQ Operator, będziesz potrzebować kodu YAML QueueManager, a także kodu YAML dla innych zasobów pomocniczych takich jak ConfigMap czy Secret.

### Informacje pokrewne



[ha\\_for\\_ctr.dita](#)

## Uwierzytelnianie i autoryzacja użytkowników oprogramowania IBM MQ w kontenerach

Produkt IBM MQ może być skonfigurowany tak, by używał kont użytkowników i grup LDAP. Można też używać w obrazie kontenera kont użytkowników i grup użytkowników lokalnego systemu operacyjnego. Ze względów bezpieczeństwa IBM MQ Operator nie zezwala na korzystanie z kont i grup użytkowników systemu operacyjnego.

W środowisku z wieloma podmiotami użytkującymi zwykle stosowane są restrykcyjne zabezpieczenia w celu zapobiegania potencjalnym problemom, na przykład:

- **Zapobieganie użyciu konta użytkownika "root" wewnątrz kontenera**
- **Wymuszenie użycia losowego identyfikatora UID.** Na przykład w produkcie Red Hat OpenShift Container Platform domyślny zasób SecurityContextConstraints (o nazwie restricted) używa losowego identyfikatora użytkownika dla każdego kontenera.
- **Blokowanie wykorzystania eskalacji uprawnień.** Produkt IBM MQ w systemie Linux używa eskalacji uprawnień do sprawdzania haseł użytkowników — korzysta w tym celu z programu "setuid", aby stać się użytkownikiem "root".

  Aby zapewnić zgodność z tymi środkami bezpieczeństwa, IBM MQ Operator nie obsługuje użycia w kontenerze identyfikatorów zdefiniowanych w bibliotekach systemu operacyjnego. W kontenerze nie ma zdefiniowanego identyfikatora użytkownika lub grupy mqm. W przypadku używania IBM MQ w IBM Cloud Pak for Integration i Red Hat OpenShift należy skonfigurować menedżera kolejek do używania LDAP na potrzeby uwierzytelniania i autoryzacji użytkowników. Informacje na temat odpowiedniej konfiguracji produktu IBM MQ zawiera sekcja [Uwierzytelnianie połączenia: repozytoria użytkowników i Autoryzacja LDAP](#)

## Planowanie skalowalności i wydajności dla produktu IBM MQ w kontenerach

W większości przypadków skalowanie i wydajność produktu IBM MQ w kontenerach jest taka sama, jak produkt IBM MQ for Multiplatforms. Istnieje jednak kilka dodatkowych limitów, które mogą zostać narzucone przez platformę kontenerową.

### O tym zadaniu

Podczas planowania skalowalności i wydajności dla produktu IBM MQ w kontenerach należy rozważyć następujące opcje:

### Procedura

- **Ogranicz liczbę wątków i procesów.**

Produkt IBM MQ używa wątków do zarządzania współbieżnością. W programie Linux wątki są implementowane jako procesy, dzięki czemu można napotkać ograniczenia narzucone przez platformę kontenerową lub system operacyjny w odniesieniu do maksymalnej liczby procesów. W składniku Red Hat OpenShift Container Platform istnieje domyślny limit 4096 procesów na kontener (1024 procesy do OpenShift 4.11). Chociaż jest to odpowiednie dla większości scenariuszy, mogą wystąpić przypadki, w których może to mieć wpływ na liczbę połączeń klienckich dla menedżera kolejek.

Limit procesu w programie Kubernetes może zostać skonfigurowany przez administratora klastra przy użyciu ustawienia konfiguracji kubelet **podPidsLimit**. Więcej informacji na ten temat zawiera sekcja [Limity identyfikatorów procesów i rezerwacje](#) w dokumentacji Kubernetes. W programie Red Hat OpenShift Container Platform można także [utworzyć niestandardowy zasób ContainerRuntimeConfig](#), aby edytować parametry CRI-O.

W konfiguracji produktu IBM MQ można również ustawić maksymalną liczbę połączeń klienckich dla menedżera kolejek. Sekcja [Limity kanałów połączenia z serwerem](#) umożliwia zastosowanie limitów dla pojedynczego kanału połączenia z serwerem oraz atrybut [MAXCHANNELS INI](#) w celu zastosowania limitów dla całego menedżera kolejek.

- **Ogranicz liczbę woluminów.**

W systemach chmurowych i kontenerowych powszechnie używane są woluminy pamięci masowej podłączone do sieci. Istnieją ograniczenia dotyczące liczby woluminów, które mogą być przyłączone do węzłów produktu Linux. Na przykład wartość AWS EC2 ogranicza się do nie więcej niż 30 woluminów na maszynę wirtualną. Red Hat OpenShift Container Platform [ma podobny limit](#), podobnie jak Microsoft Azure i Google Cloud Platform.

Rodzimy menedżer kolejek wysokiej dostępności wymaga jednego woluminu dla każdej z trzech instancji i wymusza rozmieszczenie instancji w węzłach. Można jednak skonfigurować menedżer kolejek w taki sposób, aby używała trzech woluminów na instancję (dane menedżera kolejek, dzienniki odtwarzania i utrwalone dane).

- **Użyj technik skalowania IBM MQ.**

Zamiast niewielkiej liczby dużych menedżerów kolejek, korzystne może być użycie technik skalowania IBM MQ, takich jak IBM MQ jednolite klastry w celu uruchomienia wielu menedżerów kolejek z tą samą konfiguracją. Ma to dodatkową korzyść, gdy zmniejszy się wpływ restartu pojedynczego kontenera (na przykład w ramach obsługi platformy kontenerowej).

IBM MQ Operator wdraża produkt IBM MQ i zarządza nim w ramach produktu IBM Cloud Pak for Integration lub jako autonomiczny produkt na platformie Red Hat OpenShift Container Platform

## Procedura

- [“Historia wydań operatora IBM MQ Operator” na stronie 21.](#)
- [“Migracja produktu IBM MQ do rozwiązania IBM Cloud Pak for Integration” na stronie 38.](#)
- [“Instalowanie i deinstalowanie produktu IBM MQ Operator w rozwiązaniu Red Hat OpenShift” na stronie 61.](#)
- [“Aktualizowanie operatora IBM MQ Operator i menedżerów kolejek” na stronie 73.](#)
- [“Wdrażanie i konfigurowanie menedżerów kolejek przy użyciu operatora IBM MQ Operator” na stronie 82.](#)
- [“Użytkowanie produktu IBM MQ przy wykorzystaniu operatora IBM MQ Operator” na stronie 118.](#)
- [“Skorowidz interfejsu API operatora IBM MQ Operator” na stronie 130.](#)

## **Historia wydań operatora IBM MQ Operator**

### *IBM MQ Operator*

#### **IBM MQ Operator 1.8.2**



##### **Wersja IBM Cloud Pak for Integration**

IBM Cloud Pak for Integration 2021.4.1

##### **Kanał operatora**

v1.8

##### **Dozwolone wartości dla .spec.version**

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3

##### **Wersja Red Hat OpenShift Container Platform**

Red Hat OpenShift Container Platform 4.6 i nowsze

##### **Wersja IBM Cloud Pak foundational services**

IBM Cloud Pak foundational services 3.8 i nowsze (kanał v3)

##### **Co nowego**

- Aktualizacja tylko dla zabezpieczeń wbudowana w IBM MQ Operator 1.8.0.
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

#### **IBM MQ Operator 1.8.1**



##### **Wersja IBM Cloud Pak for Integration**

IBM Cloud Pak for Integration 2021.4.1

##### **Kanał operatora**

v1.8

##### **Dozwolone wartości dla .spec.version**

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2

##### **Wersja Red Hat OpenShift Container Platform**

Red Hat OpenShift Container Platform 4.6 i nowsze

## Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.8 i nowsze (kanał v3)

### Co nowego

- Aktualizacja tylko dla zabezpieczeń wbudowana w [IBM MQ Operator 1.8.0](#).
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

## IBM MQ Operator 1.8.0



### Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2021.4.1

### Kanał operatora

v1.8

### Dozwolone wartości dla .spec.version

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1, 9.2.3.0-r1, 9.2.4.0-r1, [9.2.5.0-r1](#)

### Wersja Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 i nowsze

### Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.8 i nowsze (kanał v3)

### Co nowego

- Dodano warunki statusu dla nieaktualnych wersji produktu IBM MQ.

### Co się zmieniło

- Obrazy przeniesione z koncentratora programu Docker do IBM Container Registry.
  - Klienci z regułami firewalla mogą być zmuszeni do dopasowania ich, aby uzyskać dostęp do obrazów na IBM Container Registry.
  - Klienci w środowisku z separacją doświadczą restartu węzła sieci podczas aktualizacji produktu IBM MQ Operator do wersji 1.8.0.
- Nieaktualne wersje: IBM MQ 9.1.5, 9.2.0 CD, 9.2.1, 9.2.2. Te wersje mogą nie być uzgadniane z kolejnymi wersjami produktu IBM MQ Operator.
- Zmiany w logice licencji: klienci dokonujący aktualizacji do produktu IBM MQ 9.2.5 mogą używać wyłącznie tych licencji, które zostały określone do pracy z produktem IBM MQ 9.2.5. Więcej informacji zawiera sekcja [“Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1”](#) na stronie 130.
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

## IBM MQ Operator 1.7.0



### Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2021.4.1

### Kanał operatora

v1.7

### Dozwolone wartości dla .spec.version

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1, 9.2.3.0-r1, [9.2.4.0-r1](#)

### Wersja Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 i nowsze

## Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.8 i nowsze (kanał v3)

### Co nowego

- Dodaje produkt IBM MQ 9.2.4 jako wersję dostarczania ciągłego

## IBM MQ Operator 1.6.0



### Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2021.2.1

### Kanał operatora

v1.6

### Dozwolone wartości dla .spec.version

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, [9.2.0.2-r1-eus](#), [9.2.0.2-r2-eus](#), 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1, [9.2.3.0-r1](#)

### Wersja Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 i nowsze

### Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.7 i nowsze (kanał v3)

### Co nowego

- Dodano IBM MQ 9.2.3 jako wydanie dostarczania ciągłego (amd64 wyłącznie dla produktu IBM Cloud Pak for Integration 2021.2.1; amd64 lub s390x w przypadku korzystania z licencji IBM MQ)
- Nowy typ dostępności dla menedżerów kolejek: [rodzima wysoka dostępność](#). Funkcja dostępna produkcyjnie jako część wersji IBM Cloud Pak for Integration 2021.2.1.

### Co się zmieniło

- IBM MQ Operator 1.6 i wyższe wersje używają rejestru IBM Container Registry zamiast Docker Hub. Oznacza to, że konieczne jest użycie CatalogSource produktu [icr.io](#). Patrz [“Instalowanie i deinstalowanie produktu IBM MQ Operator w rozwiązaniu Red Hat OpenShift” na stronie 61](#).
- Sekwencyjna aktualizacja rodzimej wysokiej dostępności nie czeka już na zsynchronizowanie repliki przed przejściem do następnej repliki.
- Wyeliminowano problemy z powinowactwem rodzimej wysokiej dostępności w wersji OCP 4.7 i wyższych.
- Wyeliminowano problemy występujące, gdy z rodzimą wysoką dostępnością były używane certyfikaty podpisane przez ośrodek certyfikacji.

## IBM MQ Operator 1.5.0



### Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2021.1.1

### Kanał operatora

v1.5

### Dozwolone wartości dla .spec.version

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, [9.2.0.0-r3](#), 9.2.0.1-r1-eus, 9.2.1.0-r1, [9.2.1.0-r2](#), [9.2.2.0-r1](#)

### Wersja Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 i nowsze

### Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.7 i nowsze (kanał v3)

## Co nowego

- Dodano IBM MQ 9.2.2 jako wydanie dostarczania ciągłego (amd64 wyłącznie dla produktu IBM Cloud Pak for Integration 2021.1.1; amd64 lub s390x w przypadku korzystania z licencji IBM MQ)
- Nowy typ dostępności dla menedżerów kolejek: rodzima wysoka dostępność. Funkcja dostępna tylko do oceny, jako część wersji IBM Cloud Pak for Integration 2021.1.1.
- Integracja z metrykami Red Hat OpenShift Container Platform Cluster Monitoring dla programu Prometheus poprzez udostępnianie zasobu ServiceMonitor

## Co się zmieniło

- Operator IBM Licensing Operator nie jest już tworzony domyślnie podczas tworzenia menedżera kolejek.
- Zmiany w wieloinstancyjnych menedżerach kolejek są teraz obsługiwane krocząco. W ramach tej zmiany wprowadzono sondę uruchamiania Kubernetes, co ma wpływ na wartości używane podczas konfigurowania sondy działania. Sonda uruchamiania zostanie uruchomiona natychmiast, a następnie będzie oczekiwać na pomyślne uruchomienie menedżera kolejek. Jeśli sonda uruchamiania da wynik pozytywny w dowolnym momencie tego okresu oczekiwania, uruchamiane są sondy działania i gotowości. Poprzednio, w przypadku menedżera kolejek, który uruchamiał się powoli można było zwiększyć ustawienie `initialDelaySeconds` na sondzie działania. W takim przypadku należy teraz przywrócić wcześniejsze ustawienie `initialDelaySeconds`.
- Produkt `CustomResourceDefinition` jest aktualizowany z `apiextensions.k8s.io/v1beta1` do wersji `apiextensions.k8s.io/v1`

## Znane problemy i ograniczenia

- Wymaga wersji IBM Cloud Pak foundational services 3.7, który zawiera niezgodną zmianę w komponencie Identity and Access Management (IAM). Jeśli istnieją menedżery kolejek, które używają licencji IBM Cloud Pak for Integration, po wykonaniu tej aktualizacji wymagane jest zrestartowanie menedżera kolejek w celu uzyskania dostępu do konsoli WWW. Ponadto zostaną zgłoszone inne błędy na konsoli WWW. Te błędy można naprawić, aktualizując do najnowszej wartości `.spec.version` dla wybranej wersji produktu IBM MQ po zakończeniu aktualizacji elementu wykonawczego.
- Aktualizacja krocząca sekwencyjna nie rozpoczyna się automatycznie w przypadku aktualizacji produktu MQ do nowej wersji. Konieczne jest ręczne usunięcie zasobników.

## IBM MQ Operator 1.4.0



### Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1 (IBM MQ Operator 1.4.0 to wersja CD i nie kwalifikuje się do wydłużonego czasu obsługi (EUS))

### Kanał operatora

v1.4

### Dozwolone wartości dla `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.1.0-r1

### Wersja Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 i nowsze

## Co nowego

- Dodano IBM MQ 9.2.1 jako wydanie Continuous Delivery (CD)
- Teraz można zapobiec tworzeniu domyślnej ścieżki menedżera kolejek, ustawiając wartość `.spec.queueManager.route.enabled` na `false`



## Znane problemy i ograniczenia

- W przypadku aktualizowania QueueManager o typie dostępności MultiInstance oba zasobniki zostaną natychmiast usunięte. Oba powinny zostać szybko zrestartowane przez środowisko Red Hat OpenShift Container Platform.

## IBM MQ Operator 1.3.8 (EUS)



### Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

### Kanał operatora

v1.3-eus

### Dozwolone wartości dla .spec.version

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus

### Wersja Red Hat OpenShift Container Platform

Tylko Red Hat OpenShift Container Platform 4.6

### Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (kanał stable-v1)

### Co nowego

- Dodaje nowy operand w wersji 9.2.0.6-r3-eus.
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

## IBM MQ Operator 1.3.7 (EUS)



### Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

### Kanał operatora

v1.3-eus

### Dozwolone wartości dla .spec.version

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus

### Wersja Red Hat OpenShift Container Platform

Tylko Red Hat OpenShift Container Platform 4.6

### Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (kanał stable-v1)

### Co nowego

- Dodaje nowy operand w wersji 9.2.0.6-r2-eus.
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

## IBM MQ Operator 1.3.6 (EUS)



### Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

### Kanał operatora

v1.3-eus

### Dozwolone wartości dla `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, [9.2.0.6-r1-eus](#)

### Wersja Red Hat OpenShift Container Platform

Tylko Red Hat OpenShift Container Platform 4.6

### Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (kanał `stable-v1`)

### Co nowego

- Dodaje nowy operand w wersji [9.2.0.6-r1-eus](#).
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

## IBM MQ Operator 1.3.5 (EUS)



### Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

### Kanał operatora

v1.3-eus

### Dozwolone wartości dla `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, [9.2.0.5-r3-eus](#)

### Wersja Red Hat OpenShift Container Platform

Tylko Red Hat OpenShift Container Platform 4.6

### Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (kanał `stable-v1`)

### Co nowego

- Dodaje nowy operand w wersji [9.2.0.5-r3-eus](#).
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

## IBM MQ Operator 1.3.4 (EUS)



### Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

### Kanał operatora

v1.3-eus

### Dozwolone wartości dla `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, [9.2.0.5-r2-eus](#)

### Wersja Red Hat OpenShift Container Platform

Tylko Red Hat OpenShift Container Platform 4.6

### Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (kanał `stable-v1`)

### Co nowego

- Dodaje nowy operand w wersji [9.2.0.5-r2-eus](#)
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

## IBM MQ Operator 1.3.3 (EUS)

EUS

### Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

### Kanał operatora

v1.3-eus

### Dozwolone wartości dla `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, [9.2.0.5-r1-eus](#)

### Wersja Red Hat OpenShift Container Platform

Tylko Red Hat OpenShift Container Platform 4.6

### Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (kanał `stable-v1`)

### Co nowego

- Dodaje nowy operand w wersji [9.2.0.5-r1-eus](#)
- Informacje o usuwanych starych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

## IBM MQ Operator 1.3.2 (EUS)

EUS

### Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

### Kanał operatora

v1.3-eus

### Dozwolone wartości dla `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, [9.2.0.4-r1-eus](#)

### Wersja Red Hat OpenShift Container Platform

Tylko Red Hat OpenShift Container Platform 4.6

### Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (kanał `stable-v1`)

### Co nowego

- Dodaje nowy operand w wersji [9.2.0.4-r1-eus](#)

## IBM MQ Operator 1.3.1 (EUS)

EUS

### Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

### Kanał operatora

v1.3-eus

### Dozwolone wartości dla `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, [9.2.0.2-r1-eus](#)

### Wersja Red Hat OpenShift Container Platform

Tylko Red Hat OpenShift Container Platform 4.6

### Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (kanał `stable-v1`)

### Co nowego

- Dodano nową wersję operandu [9.2.0.2-r1-eus](#)

## IBM MQ Operator 1.3.0 (EUS)

EUS

### Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

### Kanał operatora

v1.3-eus

### Dozwolone wartości dla `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus

### Wersja Red Hat OpenShift Container Platform

Tylko Red Hat OpenShift Container Platform 4.6

### Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (kanał `stable-v1`)

### Co nowego

- Rozszerzona obsługa aktualizacji (EUS) jest oferowana dla pól `.spec.version` kończących się na `-eus` w przypadku korzystania z licencji IBM Cloud Pak for Integration
- Dodano nowy sposób ustawiania etykiet i adnotacji na zasobie `QueueManager` przy użyciu `.spec.labels` i `.spec.annotations`

### Co się zmieniło

- Usprawniono obsługę błędów przy próbie zmiany z pojedynczej instancji na wiele instancji
- Udoskonalenia dotyczące sposobu renderowania właściwości `QueueManager` w produkcie IBM Cloud Pak for Integration Platform Navigator oraz "Widoku formularza" konsoli WWW produktu Red Hat OpenShift Container Platform
- Naprawiono domyślną metrykę licencji podczas korzystania z licencji IBM Cloud Pak for Integration, obecnie: `VirtualProcessorCore`
- Naprawiono kartę **Zasoby** dla produktu `QueueManager` w konsoli WWW Red Hat OpenShift Container Platform, która teraz poprawnie wyświetla zasoby zarządzane przez produkt IBM MQ Operator dla tego menedżera kolejek

### Znane problemy i ograniczenia

- W przypadku aktualizowania `QueueManager` o typie dostępności `MultiInstance` oba zasobniki zostaną natychmiast usunięte. Oba powinny zostać szybko zrestartowane przez środowisko Red Hat OpenShift Container Platform.

## IBM MQ Operator 1.2.0

CD

### Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.3.1

### Kanał operatora

v1.2

### Dozwolone wartości dla `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2

### Wersja Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.4 i nowsze

### Co nowego

- Dodano obsługę systemu z/Linux
- Dodano bardziej szczegółowe warunki statusu do zasobu `QueueManager`. Więcej informacji na ten temat zawiera sekcja "Warunki stanu menedżera kolejek ([mq.ibm.com/v1beta1](http://mq.ibm.com/v1beta1))" na stronie 149.

- Dodano dodatkowe kontrole w czasie wykonywania, które zapobiegają użyciu niepoprawnych klas pamięci masowej. Więcej informacji na ten temat zawiera sekcja [“Wyłączanie kontroli elementu webhook w czasie wykonywania”](#) na stronie 118
- Uproszczono obsługę menedżerów kolejek z wieloma instancjami: można teraz wybrać tylko jedną właściwość (`.spec.queueManager.availability.type`) w zasobie `QueueManager`
- Uproszczono wybór innej niż domyślna klasy pamięci masowej przez wprowadzanie właściwości `.spec.queueManager.storage.defaultClass` w zasobie `QueueManager`

#### Co się zmieniło

- Udoskonalenia dotyczące sposobu renderowania właściwości `QueueManager` w produkcie IBM Cloud Pak for Integration Platform Navigator oraz "Widoku formularza" konsoli WWW produktu Red Hat OpenShift Container Platform
- Dostępność zaktualizowanej wersji menedżera kolejek jest teraz sygnalizowana w programie IBM Cloud Pak for Integration Platform Navigator

## IBM MQ Operator 1.1.0



### Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.2.1

### Kanał operatora

v1.1

### Dozwolone wartości dla `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1

### Wersja Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.4 i nowsze

#### Co nowego

- Dodano IBM MQ Advanced 9.2.0 jako wydanie Continuous Delivery (CD)
- Dodano możliwość określania informacji INI i MQSC w zasobach `ConfigMap` i `Secret`.
- Udostępniono nawigację schematów podczas korzystania z konsoli WWW środowiska Red Hat OpenShift Container Platform

#### Co się zmieniło

- Rozwiązano problem dotyczący strategii sieciowej, która wpływa na produkt Red Hat OpenShift w rozwiązaniu IBM Cloud
- Udoskonalenia sprawdzania poprawności elementu webhook, aby zapobiec niepoprawnym kombinacjom ustawień w zasobach `QueueManager`

## IBM MQ Operator 1.0.0



### Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.2.1

### Kanał operatora

v1.0

### Dozwolone wartości dla `.spec.version`

9.1.5.0-r2

### Wersja Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.4 i nowsze

## Co nowego

- Początkowa wersja elementu wykonawczego, wprowadzenie interfejsu API produktu [mq.ibm.com/v1beta1](https://mq.ibm.com/v1beta1)

## Obrazy kontenerów menedżera kolejek do użytku z operatorem IBM MQ Operator

### 9.2.5.0-r3

CD

#### Wymagana wersja operatora

[1.8.2](#) lub nowsza

#### Obsługiwane architektury

amd64, s390x

#### Obrazy

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r3`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.2.5.0-r3`
- `icr.io/ibm-messaging/mq:9.2.5.0-r3`

#### Co nowego

- [Co nowego w produkcie IBM MQ 9.2.5](#)

#### Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.2.5](#)
- Na podstawie obrazu [Red Hat Universal Base Image 8.6-751](#)

### 9.2.5.0-r2

CD

#### Wymagana wersja operatora

[1.8.1](#) lub nowsza

#### Obsługiwane architektury

amd64, s390x

#### Obrazy

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.2.5.0-r2`
- `icr.io/ibm-messaging/mq:9.2.5.0-r2`

#### Co nowego

- [Co nowego w produkcie IBM MQ 9.2.5](#)

#### Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.2.5](#)
- Na podstawie obrazu [Red Hat Universal Base Image 8.5-240.1648458092](#)

### 9.2.5.0-r1

CD

#### Wymagana wersja operatora

[1.8.0](#) lub nowsza

#### Obsługiwane architektury

amd64, s390x

## Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r1](https://cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r1)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.5.0-r1](https://cp.icr.io/cp/ibm-mqadvanced-server:9.2.5.0-r1)
- [icr.io/ibm-messaging/mq:9.2.5.0-r1](https://icr.io/ibm-messaging/mq:9.2.5.0-r1)

## Co nowego

- [Co nowego w produkcie IBM MQ 9.2.5](#)

## Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.2.5](#)
- Niepoprawna opcja Menedżery kolejek zdalnych została usunięta z IBM MQ Console
- Na podstawie obrazu [Red Hat Universal Base Image 8.5-240](#)

## 9.2.4.0-r1



### Wymagana wersja operatora

[1.7.0](#) lub nowsza

### Obsługiwane architektury

amd64, s390x

## Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.4.0-r1](https://cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.4.0-r1)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.4.0-r1](https://cp.icr.io/cp/ibm-mqadvanced-server:9.2.4.0-r1)
- [docker.io/ibmcom/mq:9.2.4.0-r1](https://docker.io/ibmcom/mq:9.2.4.0-r1)

## Co nowego

- [Co nowego w produkcie IBM MQ 9.2.4](#)

## Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.2.4](#)
- Na podstawie obrazu [Red Hat Universal Base Image 8.5-204](#)

## 9.2.3.0-r1



### Wymagana wersja operatora

[1.6.0](#) lub nowsza

### Obsługiwane architektury

amd64, s390x

## Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.3.0-r1](https://cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.3.0-r1) (tylko amd64)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.3.0-r1](https://cp.icr.io/cp/ibm-mqadvanced-server:9.2.3.0-r1)
- [docker.io/ibmcom/mq:9.2.3.0-r1](https://docker.io/ibmcom/mq:9.2.3.0-r1)

## Co nowego

- [Co nowego w produkcie IBM MQ 9.2.3](#)
- [Obsługa rodzimej wysokiej dostępności MQ](#) w zastosowaniach produkcyjnych, gdy jest używana licencja IBM Cloud Pak for Integration. Uwaga: menedżerów kolejek korzystających z rodzimej wysokiej dostępności na licencji do oceny w wersji IBM MQ 9.2.2 nie można zaktualizować do wersji 9.2.3. Okres oceny zakończył się.

## Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.2.3](#)
- Na podstawie [Red Hat Universal Base Image 8.4-205](#)

### 9.2.2.0-r1



#### Wymagana wersja operatora

[1.5.0](#) lub nowsza

#### Obsługiwane architektury

amd64, s390x

#### Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.2.0-r1](#) (tylko amd64)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.2.0-r1](#)
- [docker.io/ibmcom/mq:9.2.2.0-r1](#)

#### Co nowego

- [Co nowego w produkcie IBM MQ 9.2.2](#)
- Obsługa [rodzimej wysokiej dostępności](#) MQ (funkcja wstępna, wyłącznie do oceny), gdy jest używana licencja IBM Cloud Pak for Integration

## Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.2.2](#)
- Wyeliminowanie problemu powodującego FDC podczas zamykania menedżera kolejek IBM MQ Advanced for Developers
- Na podstawie obrazu [Red Hat Universal Base Image 8.3-291](#)

### 9.2.1.0-r2



#### Wymagana wersja operatora

[1.5.0](#) lub nowsza

#### Obsługiwane architektury

amd64, s390x

#### Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.1.0-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.1.0-r2](#)
- [docker.io/ibmcom/mq:9.2.1.0-r2](#)

#### Co się zmieniło

- Wyeliminowano problem z pojedynczym logowaniem w wersji IBM Cloud Pak foundational services 3.7 i nowszych.
- Na podstawie obrazu [Red Hat Universal Base Image 8.3-291](#)

### 9.2.1.0-r1



#### Wymagana wersja operatora

[1.4.0](#) lub nowsza



## Obsługiwane architektury

amd64, s390x

## Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.1.0-r1](https://cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.1.0-r1)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.1.0-r1](https://cp.icr.io/cp/ibm-mqadvanced-server:9.2.1.0-r1)
- [docker.io/ibmcom/mq:9.2.1.0-r1](https://docker.io/ibmcom/mq:9.2.1.0-r1)

## Co nowego

- [Co nowego w produkcie IBM MQ 9.2.1](#)
- Informacje o połączeniu dla domyślnej trasy są dostępne w konsoli WWW produktu MQ

## Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.2.1](#)
- Na podstawie obrazu [Red Hat Universal Base Image 8.3-230](#)

## 9.2.0.6-r3-eus



### Wymagana wersja operatora

[1.3.8](#) i przyszłe pakiety poprawek

### Obsługiwane architektury

amd64, s390x

### Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.6-r3-eus](https://cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.6-r3-eus)

### Co się zmieniło

- Zawiera produkt IBM MQ 9.2.0 Fix Pack 6. Więcej informacji na ten temat zawiera sekcja [Lista poprawek dla produktu IBM MQ w wersji 9.2 LTS](#).
- Na podstawie obrazu [Red Hat Universal Base Image 8.6-941](#).

## 9.2.0.6-r2-eus



### Wymagana wersja operatora

[1.3.7](#) i przyszłe pakiety poprawek

### Obsługiwane architektury

amd64, s390x

### Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.6-r2-eus](https://cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.6-r2-eus)

### Co się zmieniło

- Zawiera produkt IBM MQ 9.2.0 Fix Pack 6. Więcej informacji na ten temat zawiera sekcja [Lista poprawek dla produktu IBM MQ w wersji 9.2 LTS](#).
- W oparciu o [Red Hat Universal Base Image 8.6-902](#).

## 9.2.0.6-r1-eus



### Wymagana wersja operatora

[1.3.6](#) i przyszłe pakiety poprawek

## Obsługiwane architektury

amd64, s390x

## Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.6-r1-eus](https://cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.6-r1-eus)

## Co się zmieniło

- Zawiera produkt IBM MQ 9.2.0 Fix Pack 6. Więcej informacji na ten temat zawiera sekcja [Lista poprawek dla produktu IBM MQ w wersji 9.2 LTS](#).
- W oparciu o [Red Hat Universal Base Image 8.6-854](#).

## 9.2.0.5-r3-eus



### Wymagana wersja operatora

[1.3.5](#) i przyszłe pakiety poprawek

### Obsługiwane architektury

amd64, s390x

## Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.5-r3-eus](https://cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.5-r3-eus)

## Co się zmieniło

- Zawiera produkt IBM MQ 9.2.0 Fix Pack 5. Więcej informacji na ten temat zawiera sekcja [Co się zmieniło w produkcie IBM MQ 9.2.0 Fix Pack 5](#) i [Lista poprawek dla produktu IBM MQ w wersji 9.2 LTS](#).
- W oparciu o [Red Hat Universal Base Image 8.6-751.1655117800](#).

## 9.2.0.5-r2-eus



### Wymagana wersja operatora

[1.3.4](#) i przyszłe pakiety poprawek

### Obsługiwane architektury

amd64, s390x

## Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.5-r2-eus](https://cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.5-r2-eus)

## Co się zmieniło

- Zawiera produkt IBM MQ 9.2.0 Fix Pack 5. Aby uzyskać więcej informacji, patrz [Co zostało zmienione w produkcie IBM MQ 9.2.0 Fix Pack 5](#) i [Lista poprawek dla produktu IBM MQ, wersja 9.2 LTS](#).
- Na podstawie obrazu [Red Hat Universal Base Image 8.6-751](#)

## 9.2.0.5-r1-eus



### Wymagana wersja operatora

[1.3.3](#) i przyszłe pakiety poprawek

### Obsługiwane architektury

amd64, s390x

## Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.5-r1-eus](https://cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.5-r1-eus)

## Co się zmieniło

- Zawiera produkt IBM MQ 9.2.0 Fix Pack 5. Aby uzyskać więcej informacji, patrz [Co zostało zmienione w produkcie IBM MQ 9.2.0 Fix Pack 5](#) i [Lista poprawek dla produktu IBM MQ, wersja 9.2 LTS](#).
- Na podstawie obrazu [Red Hat Universal Base Image 8.5-240.1648458092](#)

### 9.2.0.4-r1-eus



#### Wymagana wersja operatora

[1.3.2](#) i przyszłe pakiety poprawek

#### Obsługiwane architektury

amd64, s390x

#### Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.4-r1-eus](https://cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.4-r1-eus)

## Co się zmieniło

- Zawiera produkt IBM MQ 9.2.0 Fix Pack 4. Aby uzyskać więcej informacji, patrz [Co zostało zmienione w produkcie IBM MQ 9.2.0 Fix Pack 4](#) i [Lista poprawek dla produktu IBM MQ, wersja 9.2 LTS](#).
- Na podstawie obrazu [Red Hat Universal Base Image 8.5-204](#)

### 9.2.0.2-r2-eus



#### Wymagana wersja operatora

[1.6.0](#) lub nowsza

#### Obsługiwane architektury

amd64, s390x

#### Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.2-r2-eus](https://cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.2-r2-eus)

## Co się zmieniło

- Wyeliminowano problem z pojedynczym logowaniem w wersji IBM Cloud Pak foundational services 3.7 i nowszych, co jest istotne tylko przy migracji z wydania EUS do wydania CD.
- Na podstawie obrazu [Red Hat Universal Base Image 8.4-200.1622548483](#)

### 9.2.0.2-r1-eus



#### Wymagana wersja operatora

[1.3.1](#) i przyszłe pakiety poprawek ; wersja 1.6.0 lub nowsza

#### Obsługiwane architektury

amd64, s390x

#### Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.2-r1-eus](https://cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.2-r1-eus)

## Co się zmieniło

- Do integracji panelu Operations Dashboard używany jest agent śledzenia i kolektor w wersji 1.0.8

- Zawiera produkt IBM MQ 9.2.0 Fix Pack 2. Aby uzyskać więcej informacji, patrz [Co zostało zmienione w produkcie IBM MQ 9.2.0 Fix Pack 2 i Lista poprawek dla produktu IBM MQ, wersja 9.2 LTS.](#)
- Na podstawie obrazu [Red Hat Universal Base Image 8.4-200.1622548483](#)

### 9.2.0.1-r1-eus



#### Wymagana wersja operatora

[1.3.0](#) lub nowsza

#### Obsługiwane architektury

amd64, s390x

#### Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.1-r1-eus](#)

#### Co nowego

- Dostępne tylko w przypadku korzystania z licencji na IBM Cloud Pak for Integration
- Wydłużona obsługa aktualizacji (EUS — Extended Update Support) jest dostępna w przypadku używania produktów IBM MQ Operator 1.3.x i IBM Common Services 3.6 w środowisku Red Hat OpenShift Container Platform 4.6

#### Co się zmieniło

- Zawiera produkt IBM MQ 9.2.0 Fix Pack 1. Aby uzyskać więcej informacji, patrz [Co zostało zmienione w produkcie IBM MQ 9.2.0 Fix Pack 1 i Lista poprawek dla produktu IBM MQ, wersja 9.2 LTS.](#)
- Na podstawie obrazu [Red Hat Universal Base Image 8.3-201](#)
- Naprawiono problem z sondą działania (chkmqhealthy) i sondą gotowości (chkmqready) w przypadku działania w SecurityContextConstraints, co umożliwiła eskalację uprawnień.

### 9.2.0.0-r3



#### Wymagana wersja operatora

[1.5.0](#) lub nowsza

#### Obsługiwane architektury

amd64, s390x

#### Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.0-r3](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.0.0-r3](#)
- [docker.io/ibmcom/mq:9.2.0.0-r3](#)

#### Co się zmieniło

- Na podstawie obrazu [Red Hat Universal Base Image 8.3-291](#)

### 9.2.0.0-r2



#### Wymagana wersja operatora

[1.2.0](#) lub nowsza

#### Obsługiwane architektury

amd64, s390x

## Obrazy

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.0-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.2.0.0-r2`
- `docker.io/ibmcom/mq:9.2.0.0-r2`

## Co nowego

- Teraz produkt jest dostępny w systemie z/Linux

## Co się zmieniło

- Na podstawie obrazu [Red Hat Universal Base Image 8.2-349](#)

## 9.2.0.0-r1



### Wymagana wersja operatora

[1.1.0](#) lub nowsza

### Obsługiwane architektury

amd64

## Obrazy

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.0-r1-amd64`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.2.0.0-r1-amd64`
- `docker.io/ibmcom/mq:9.2.0.0-r1`

## Co nowego

- [Co nowego w produkcie IBM MQ 9.2.0](#)

## Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.2.0](#)
- Użycie argumentu `-ic` do `crtmqm` w celu automatycznego zastosowania plików MQSC. Zastąpienie poprzedniego użycia komend produktu `runmqsc`
- Na podstawie obrazu [Red Hat Universal Base Image 8.2-301.1593113563](#)

## 9.1.5.0-r2



### Wymagana wersja operatora

[1.0.0](#) lub nowsza

### Obsługiwane architektury

amd64

## Obrazy

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.1.5.0-r2-amd64`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.1.5.0-r2-amd64`
- `docker.io/ibmcom/mq:9.1.5.0-r2`

## Co się zmieniło

- Na podstawie obrazu [Red Hat Universal Base Image 8.2-267](#)

# Migracja produktu IBM MQ do rozwiązania IBM Cloud Pak for Integration

Ten zestaw tematów zawiera opis kluczowych etapów migracji istniejącego menedżera kolejek IBM MQ do środowiska kontenerowego z operatorem IBM MQ Operator w rozwiązaniu IBM Cloud Pak for Integration.

## O tym zadaniu

Klienci, które wdrażają produkt IBM MQ w rozwiązaniu Red Hat OpenShift, można podzielić na podstawie następujących scenariuszy:

1. Tworzenie nowego wdrożenia produktu IBM MQ w rozwiązaniu Red Hat OpenShift dla nowych aplikacji.
2. Rozszerzanie sieci produktu IBM MQ do produktu Red Hat OpenShift na potrzeby nowych aplikacji w produkcie Red Hat OpenShift.
3. Przenoszenie wdrożenia produktu IBM MQ do produktu Red Hat OpenShift w celu kontynuowania obsługi istniejących aplikacji.

W przypadku scenariusza 3 konieczne jest przeprowadzenie migracji konfiguracji IBM MQ. Pozostałe scenariusze są uznawane za nowe wdrożenia.

Ten zestaw tematów zawiera opis scenariusza 3 i kluczowych etapów migracji istniejącego menedżera kolejek IBM MQ do środowiska kontenerowego z operatorem IBM MQ Operator. Ze względu na elastyczność i szerokie zastosowanie produktu IBM MQ istnieje kilka kroków opcjonalnych. Każdy z nich zawiera sekcję "Czy muszę to zrobić?". Właściwe zweryfikowanie potrzeb pozwoli zaoszczędzić czas podczas migracji.

Należy również wziąć pod uwagę, które dane mają być migrowane:

1. Migracja IBM MQ z tą samą konfiguracją, ale bez żadnych istniejących komunikatów w kolejkach.
2. Migracja produktu IBM MQ z tą samą konfiguracją i istniejącymi komunikatami.

Typowa migracja z wersji do wersji może być realizowana według dowolnego z tych dwóch wariantów. W momencie rozpoczęcia migracji w typowym menedżerze kolejek IBM MQ nie ma wielu komunikatów w kolejkach (albo nie ma żadnych komunikatów), co powoduje, że wariant 1 jest odpowiedni w wielu przypadkach. W przypadku migracji do platformy kontenerowej bardziej powszechne jest użycie wariantu 1, który zmniejsza złożoność migracji i pozwala na wdrożenie "niebiesko-zielone". W związku z tym instrukcje koncentrują się na tym scenariuszu.

Celem tego scenariusza jest utworzenie w środowisku kontenera menedżera kolejek, który będzie zgodny z definicją istniejącego menedżera kolejek. Dzięki temu istniejące aplikacje przyłączone do sieci można po prostu przekonfigurować w taki sposób, aby wskazywały na nowy menedżer kolejek, bez konieczności zmiany innych ustawień konfiguracji lub logiki aplikacji.

W trakcie migracji generowanych jest wiele plików konfiguracyjnych, które będą stosowane w nowym menedżerze kolejek. Aby uprościć zarządzanie tymi plikami, należy utworzyć katalog i wygenerować je w tym katalogu.

## Procedura

1. ["Sprawdzenie, czy wymagane funkcje są dostępne" na stronie 39](#)
2. ["Wyodrębnianie konfiguracji menedżera kolejek" na stronie 39](#)
3. Opcjonalnie: ["Opcjonalnie: wyodrębnianie i pozyskanie kluczy i certyfikatów menedżera kolejek" na stronie 40](#)
4. Opcjonalnie: ["Opcjonalnie: konfigurowanie protokołu LDAP" na stronie 42](#)
5. Opcjonalnie: ["Opcjonalnie: zmiana adresów IP i nazw hostów w konfiguracji produktu IBM MQ" na stronie 49](#)
6. ["Aktualizowanie konfiguracji menedżera kolejek dla środowiska kontenera" na stronie 51](#)
7. ["Wybieranie docelowej architektury wysokiej dostępności dla programu IBM MQ działającego w kontenerach" na stronie 53](#)

8. [“Tworzenie zasobów dla menedżera kolejek” na stronie 54](#)
9. [“Tworzenie nowego menedżera kolejek w środowisku Red Hat OpenShift” na stronie 55](#)
10. [“Sprawdzanie poprawności nowego wdrożenia w kontenerze” na stronie 59](#)



## Sprawdzanie, czy wymagane funkcje są dostępne

IBM MQ Operator nie zawiera wszystkich funkcji dostępnych w produkcie IBM MQ Advanced, a użytkownik musi sprawdzić, czy te funkcje nie są wymagane. Inne funkcje są częściowo obsługiwane i można je przekonfigurować w taki sposób, aby były zgodne z możliwościami dostępnymi kontenerze.

### Zanim rozpoczniesz

Jest to pierwszy krok w procedurze [“Migracja produktu IBM MQ do rozwiązania IBM Cloud Pak for Integration” na stronie 38.](#)

### Procedura

1. Sprawdź, czy docelowy obraz kontenera zawiera wszystkie wymagane funkcje.  
Więcej informacji na ten temat zawiera sekcja [“Wybór sposobu używania IBM MQ w kontenerach” na stronie 5.](#)
2. IBM MQ Operator ma jeden port obsługujący ruch IBM MQ, nazywany portem nasłuchującym. Jeśli istnieje wiele portów nasłuchujących, należy uprościć konfigurację, używając jednego portu nasłuchującego w kontenerze. Ponieważ nie jest to typowy scenariusz, ta modyfikacja nie została tutaj szczegółowo udokumentowana.
3. Jeśli używane są procedury zewnętrzne IBM MQ, zmigruj je do kontenera, dodając odpowiednią warstwę w plikach binarnych IBM MQ. Jest to zaawansowany scenariusz migracji i w związku z tym nie został tutaj opisany. Ogólny schemat kroków zawiera sekcja [“Budowanie obrazu z niestandardowymi plikami MQSC i INI przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift” na stronie 115.](#)
4. Jeśli w systemie IBM MQ używany jest mechanizm wysokiej dostępności, przeanalizuj dostępne opcje.  
Patrz sekcja [“Wysoka dostępność produktu IBM MQ w kontenerach” na stronie 16.](#)

### Co dalej

Teraz można przystąpić do [wyodrębnienia konfiguracji menedżera kolejek.](#)



## Wyodrębnianie konfiguracji menedżera kolejek

Większość konfiguracji jest przenośna między menedżerami kolejek. Dotyczy to na przykład elementów, z którymi aplikacje wchodzi w interakcję, takich jak definicje kolejek, tematów i kanałów. To zadanie służy do wyodrębnienia konfiguracji z istniejącego menedżera kolejek IBM MQ.

### Zanim rozpoczniesz

W przypadku tego zadania założono, że użytkownik [sprawdził, czy wymagane funkcje są dostępne.](#)

### Procedura

1. Zaloguj się do komputera przy użyciu istniejącej instalacji produktu IBM MQ.
2. Utwórz kopię zapasową konfiguracji.

Uruchom następującą komendę:

```
dmpmqcfig -m QMGR_NAME > /tmp/backup.mqsc
```

Uwagi dotyczące użycia tej komendy:

- Ta komenda zapisuje kopię zapasową w katalogu tmp. Można wybrać inne miejsce, ale w tym scenariuszu założono, że z katalogu tmp korzystają także następne komendy.
- Zastąp wartość `QMGR_NAME` nazwą menedżera kolejek ze swojego środowiska. Jeśli nie masz pewności co do wartości, uruchom komendę `dspsmq`, aby wyświetlić dostępne na komputerze menedżery kolejek. Poniżej znajdują się przykładowe wyniki komendy `dspsmq` dla menedżera kolejek o nazwie qm1:

```
QMNAME(qm1)                STATUS(Running)
```

Komenda `dspsmq` wymaga, aby menedżer kolejek IBM MQ był uruchomiony. W przeciwnym razie zostanie wyświetlony następujący błąd:

```
AMQ8146E: IBM MQ queue manager not available.
```

Jeśli jest to wymagane, uruchom menedżer kolejek, wywołując następującą komendę:

```
strmqm QMGR_NAME
```

## Co dalej

Teraz można wyodrębnić i uzyskać klucze i certyfikaty menedżera kolejek.

## Opcjonalnie: wyodrębnianie i pozyskanie kluczy i certyfikatów menedżera kolejek

Produkt IBM MQ można skonfigurować tak, by używał protokołu TLS do szyfrowania ruchu w menedżerze kolejek. To zadanie pozwala sprawdzić, czy menedżer kolejek używa protokołu TLS, wyodrębnić klucze i certyfikaty, a także skonfigurować protokół TLS w migrowanym menedżerze kolejek.

## Zanim rozpocznie

W tym zadaniu przyjęto założenie, że została wyodrębniona konfiguracja menedżera kolejek.

## O tym zadaniu

### Czy muszę to zrobić?

Produkt IBM MQ można skonfigurować w taki sposób, aby szyfrował ruch w menedżerze kolejek. Szyfrowanie jest realizowane przy użyciu repozytorium kluczy skonfigurowanego w menedżerze kolejek. Kanaty IBM MQ umożliwiają wówczas komunikację TLS. Jeśli nie wiadomo, czy ten mechanizm skonfigurowany w danym środowisku, uruchom następującą komendę, aby to sprawdzić:

```
grep 'SECCOMM(ALL\|SECCOMM(ANON\|SSLIPH' backup.mqsc
```

Brak wyników będzie oznaczał, że protokół TLS nie jest używany. Nie oznacza to jednak, że protokół TLS nie powinien być skonfigurowany w migrowanym menedżerze kolejek. Istnieje kilka powodów, dla których celowa jest zmiana takiej konfiguracji:

- Podejście do zabezpieczeń w środowisku Red Hat OpenShift powinno zostać rozszerzone w porównaniu z poprzednim środowiskiem.
- Jeśli konieczne jest uzyskanie dostępu do migrowanego menedżera kolejek spoza środowiska Red Hat OpenShift, wymagane jest użycie protokołu TLS do przejścia przez trasę Red Hat OpenShift.

## Procedura

1. Wyodrębnij wszystkie zaufane certyfikaty z istniejącej składnicy.



Jeśli protokół TLS jest obecnie używany w menedżerze kolejek, menedżer kolejek może przechowywać jakieś zaufane certyfikaty. Te certyfikaty muszą zostać wyodrębnione i skopiowane do nowego menedżera kolejek. Wykonaj jeden z następujących opcjonalnych kroków:

- Aby usprawnić wyodrębnianie certyfikatów, uruchom następujący skrypt w systemie lokalnym:

```
#!/bin/bash

keyr=$(grep SSLKEYR $1)
if [ -n "${keyr}" ]; then
    keyrlocation=$(sed -n "s/^.*\(.*\).*$/\1/ p" <<< ${keyr})
    mapfile -t runmqckmResult < <(runmqckm -cert -list -db ${keyrlocation}.kdb -stashed)
    cert=1
    for i in "${runmqckmResult[@]:1}"
    do
        certlabel=$(echo ${i} | xargs)
        echo Extracting certificate $certlabel to $cert.cert
        runmqckm -cert -extract -db ${keyrlocation}.kdb -label "$certlabel" -target $
    {cert}.cert -stashed
        cert=$((cert+1))
    done
fi
```

Uruchamiając skrypt, należy podać położenie kopii zapasowej IBM MQ jako argument, a certyfikaty zostaną wyodrębnione. Jeśli na przykład skrypt nosi nazwę `extractCert.sh`, a kopia zapasowa IBM MQ znajduje się w pliku `/tmp/backup.mqsc`, należy uruchomić następującą komendę:

```
extractCert.sh /tmp/backup.mqsc
```

- Zamiast tego można również uruchomić następujące komendy w podanej kolejności:

- a. Zidentyfikuj położenie bazy TLS:

```
grep SSLKEYR /tmp/backup.mqsc
```

Przykładowy wynik:

```
SSLKEYR('/run/runmqserver/tls/key') +
```

gdzie składnica kluczy znajduje się w katalogu `/run/runmqserver/tls/key.kdb`

- b. W oparciu o te informacje o położeniu sprawdź w pliku kluczy, które certyfikaty są w nim zapisane:

```
runmqckm -cert -list -db /run/runmqserver/tls/key.kdb -stashed
```

Przykładowy wynik:

```
Certificates in database /run/runmqserver/tls/key.kdb:
  default
  CN=cs-ca-certificate,0=cert-manager
```

- c. Wyodrębnij każdy z wymienionych certyfikatów. W tym celu należy uruchomić następującą komendę:

```
runmqckm -cert -extract -db KEYSTORE_LOCATION -label "LABEL_NAME" -target OUTPUT_FILE
-stashed
```

W przykładach, które były wcześniej przedstawione, oznacza to:

```
runmqckm -cert -extract -db /run/runmqserver/tls/key.kdb -label "CN=cs-ca-
certificate,0=cert-manager" -target /tmp/cert-manager.crt -stashed
```

```
runmqckm -cert -extract -db /run/runmqserver/tls/key.kdb -label "default" -target /tmp/default.crt -stashed
```

## 2. Uzyskiwanie nowego klucza i certyfikatu dla menedżera kolejek

Aby skonfigurować protokół TLS w migrowanym menedżerze kolejek, należy wygenerować nowy klucz i certyfikat. Jest on następnie używany podczas wdrażania. W wielu organizacjach oznacza to konieczność zwrócenia się do zespołu ds. bezpieczeństwa o klucz i certyfikat. W niektórych organizacjach ta możliwość nie jest dostępna i używane są certyfikaty samopodpisane.

W poniższym przykładzie generowany jest certyfikat samopodpisany z terminem ważności ustawionym na 10 lat:

```
openssl req \
  -newkey rsa:2048 -nodes -keyout qmgr.key \
  -subj "/CN=mq queuemanager/OU=ibm mq" \
  -x509 -days 3650 -out qmgr.crt
```

Tworzone są dwa nowe pliki:

- `qmgr.key` jest kluczem prywatnym dla menedżera kolejek
- `qmgr.crt` jest certyfikatem publicznym

## Co dalej

Teraz można przystąpić do [konfigurowania protokołu LDAP](#).

## OpenShift V 9.2.1 CD EUS Opcjonalnie: konfigurowanie protokołu LDAP

Produkt IBM MQ Operator można skonfigurować w taki sposób, aby używał kilku różnych strategii zabezpieczeń. Zwykle protokół LDAP jest najbardziej efektywny w przypadku wdrożenia korporacyjnego. Dlatego to właśnie protokół LDAP jest używany w opisywanym tutaj scenariuszu migracji.

## Zanim rozpocznie

W tym zadaniu przyjęto założenie, że użytkownik [wyodrębnił i uzyskał klucze i certyfikaty menedżera kolejek](#).

## O tym zadaniu

### Czy muszę to zrobić?

Jeśli do uwierzytelniania i autoryzacji jest już używany protokół LDAP, zmiany nie są wymagane.

Jeśli nie masz pewności, czy protokół LDAP jest używany, uruchom następującą komendę:

```
connauthname="$(grep CONNAUTH backup.mqsc | cut -d "(" -f2 | cut -d ")" -f1"; grep -A 20 AUTHINFO\($connauthname\) backup.mqsc
```

Przykładowy wynik:

```
DEFINE AUTHINFO('USE.LDAP') +
  AUTHTYPE(IDPWLDAP) +
  ADOPTCTX(YES) +
  CONNAME('ldap-service.ldap(389)') +
  CHCKCLNT(REQUIRED) +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  * LDAPPWD('*****') +
  SHORTUSR('uid') +
  GRPFIELD('cn') +
```

```
USRFIELD('uid') +
AUTHORMD(SEARCHGRP) +
* ALTDATA(2020-11-26) +
* ALTTIME(15.44.38) +
REPLACE
```

W wynikach znajdują się dwa atrybuty, które są szczególnie interesujące:

### **AUTHTYPE**

Jeśli atrybut ten ma wartość IDPWLDAP, oznacza to, że protokół LDAP jest używany do uwierzytelniania.

Jeśli wartość jest pusta lub inna, protokół LDAP nie jest skonfigurowany. W takim przypadku należy odczytać atrybut AUTHORMD, aby sprawdzić, czy konta użytkowników LDAP są używane do autoryzacji.

### **AUTHORMD**

Jeśli atrybut ta ma wartość OS, protokół LDAP nie jest używany do autoryzacji.

Aby zmodyfikować ustawienia autoryzacji i uwierzytelniania, tak by używany był protokół LDAP, wykonaj następujące czynności:

## **Procedura**

1. Zaktualizuj kopię zapasową IBM MQ dla serwera LDAP.
2. Zaktualizuj kopię zapasową IBM MQ z uwzględnieniem informacji o autoryzacji LDAP.

### **LDAP, część 1: aktualizowanie kopii zapasowej IBM MQ dla serwera LDAP**

Obszerny opis sposobu konfigurowania protokołu LDAP nie należy do tego scenariusza. W tym temacie znajduje się podsumowanie procesu, przykład i odsyłacze do dalszych informacji.

## **Zanim rozpocznie**

W tym zadaniu przyjęto założenie, że użytkownik wyodrębnił i uzyskał klucze i certyfikaty menedżera kolejek.

## **O tym zadaniu**

### **Czy muszę to zrobić?**

Jeśli do uwierzytelniania i autoryzacji jest już używany protokół LDAP, zmiany nie są wymagane. Jeśli nie masz pewności, czy używany jest protokół LDAP, patrz sekcja “Opcjonalnie: konfigurowanie protokołu LDAP” na stronie 42.

Konfigurowanie serwera LDAP przebiega w dwóch etapach:

1. Zdefiniowanie konfiguracji LDAP.
2. Powiązanie konfiguracji LDAP z definicją menedżera kolejek.

Więcej informacji na temat tej konfiguracji można znaleźć w następujących informacjach:

- Przegląd repozytorium użytkowników
- Skorowidz komendy AUTHINFO

## **Procedura**

1. Zdefiniuj konfigurację LDAP.

Zmodyfikuj plik backup .mqsc, aby zdefiniować nowy obiekt **AUTHINFO** dla systemu LDAP. Na przykład:

```
DEFINE AUTHINFO(USE_LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
```

```
LDAPUSER('cn=admin,dc=ibm,dc=com') +
LDAPPWD('admin') +
SECCOMM(NO) +
USRFIELD('uid') +
SHORTUSR('uid') +
BASEDNU('ou=people,dc=ibm,dc=com') +
AUTHORMD(SEARCHGRP) +
BASEDNG('ou=groups,dc=ibm,dc=com') +
GRPFIELD('cn') +
CLASSGRP('groupOfUniqueNames') +
FINDGRP('uniqueMember')
REPLACE
```

where

- **CONNNAME** to nazwa hosta i port odpowiadające serwerowi LDAP. Jeśli dla zapewnienia odporności stosowanych jest wiele adresów, można je skonfigurować w formie listy rozdzielanej przecinkami.
- **LDAPUSER** jest nazwą wyróżniającą odpowiadającą użytkownikowi, który jest używany przez produkt IBM MQ podczas nawiązywania połączenia z serwerem LDAP w celu wykonywania zapytań o rekordy użytkowników.
- **LDAPPWD** jest hasłem użytkownika **LDAPUSER**.
- **SECCOM** określa, czy w komunikacji z serwerem LDAP powinna być używany protokół TLS. Możliwe wartości:
  - YES: używany jest protokół TLS, a serwer IBM MQ legitymuje się certyfikatem.
  - ANON: protokół TLS jest używany bez legitymowania się przez serwer IBM MQ certyfikatem.
  - NO: protokół TLS nie jest używany podczas nawiązywania połączenia.
- **USRFIELD** określa pole w rekordzie LDAP, do którego będzie dopasowywana nazwa użytkownika.
- **SHORTUSR** to pole w obrębie rekordu LDAP, którego długość nie może przekraczać 12 znaków. Wartość w tym polu jest zakładaną tożsamością, jeśli uwierzytelnianie powiodło się.
- **BASEDNU** to podstawowa nazwa wyróżniająca, która powinna być używana do przeszukiwania katalogu LDAP.
- **BASEDNG** jest podstawową nazwą wyróżniającą dla grup w katalogu LDAP.
- **AUTHORMD** definiuje mechanizm używany do rozstrzygnięcia przypisania użytkowników do grup. Dostępne są cztery opcje:
  - OS: kieruje do systemu operacyjnego zapytania o grupy powiązane z krótką nazwą.
  - SEARCHGRP: służy do wyszukiwania w katalogu LDAP grup uwierzytelnionego użytkownika.
  - SEARCHUSR: umożliwia wyszukiwanie informacji o uwierzytelnionym użytkowniku w celu uzyskania informacji o przypisaniach do grup.
  - SRCHGRPSN: służy do wyszukiwania pozycji grupy w katalogu LDAP na podstawie krótkiej nazwy użytkownika uwierzytelnionego (zdefiniowanej przez pole SHORTUSR).
- **GRPFIELD** jest atrybutem w rekordzie grupy LDAP, który odpowiada prostej nazwie. Jeśli ten parametr jest określony, można go używać do definiowania rekordów autoryzacji.
- **CLASSUSR** to klasa obiektów LDAP, która odpowiada użytkownikowi.
- **CLASSGRP** to klasa obiektu LDAP, która odpowiada grupie.
- **FINDGRP** jest atrybutem w rekordzie LDAP, który odpowiada przynależności do grupy.

Nowy wpis może być umieszczony w dowolnym miejscu w pliku, jednak praktyczne jest dodawanie nowych wpisów na początku:

```
Open [icon]
backup.mqsc
*****
* Script generated on 2020-10-21 at 11.48.32
* Script generated by user ' CallumJackso' on host 'LAPTOP-VLQ
* Queue manager name: qm1
* Queue manager platform: Windows
* Queue manager command level: (920/920)
* Command issued: dmpmqcfg -m qm1
*****
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
  SECCOMM(NO) +
  USRFIELD('uid') +
  SHORTUSR('uid') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  AUTHORMD(SEARCHGRP) +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  GRPFIELD('cn') +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  REPLACE
ALTER QMGR +
* ALTDATE(2020-10-26) +
* ALTTIME(11.43.11) +
```

2. Powiąż konfigurację LDAP z definicją menedżera kolejek.

Konieczne jest powiązanie konfiguracji LDAP z definicją menedżera kolejek. Bezpośrednio pod pozycją DEFINE AUTHINFO znajduje się pozycja ALTER QMGR. Zmodyfikuj pozycję CONNAUTH w taki sposób, aby odpowiadała nowo utworzonej nazwie AUTHINFO. Na przykład w poprzednim przykładzie użyto zapisu AUTHINFO(USE.LDAP), co oznacza, że nazwa to USE.LDAP. Należy zatem zmienić CONNAUTH('SYSTEM.DEFAULT.AUTHINFO.IDPWOS') na CONNAUTH('USE.LDAP'):

```
Open [icon]
backup.mqsc
*****
* Script generated on 2020-10-21 at 11.48.32
* Script generated by user ' CallumJackso' on host 'l
* Queue manager name: qm1
* Queue manager platform: Windows
* Queue manager command level: (920/920)
* Command issued: dmpmqcfg -m qm1
*****
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
  SECCOMM(NO) +
  USRFIELD('uid') +
  SHORTUSR('uid') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  AUTHORMD(SEARCHGRP) +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  GRPFIELD('cn') +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  REPLACE
ALTER QMGR +
* ALTDATE(2020-10-26) +
* ALTTIME(11.43.11) +
  CCSID(850) +
  CERTLABL('default') +
  CLWLUSEQ(LOCAL) +
* COMMANDO(SYSTEM_ADMIN_COMMAND.QUEUE) +
  CONNAUTH('USE.LDAP') +
```

Aby przełączyć się na protokół LDAP natychmiast, wywołaj komendę REFRESH SECURITY, dodając wiersz bezpośrednio po komendzie ALTER QMGR:

```

*backup.mqsc
*****
* Script generated on 2020-10-21 at 11.48.32
* Script generated by user ' CallumJackso' on host 'LAPTOP-VLQKJ5UH'
* Queue manager name: qm1
* Queue manager platform: Windows
* Queue manager command level: (920/920)
* Command issued: dmpmqcfg -m qm1
*****
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
  SECCOMM(NO) +
  USRFIELD('uid') +
  SHORTUSR('uid') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  AUTHORMD(SEARCHGRP) +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  GRPFIELD('cn') +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  REPLACE
ALTER QMGR +
* ALTDATE(2020-10-26) +
* ALTTIME(11.43.11) +
  CCSID(850) +
  CERTLABL('default') +
  CLWLUSEQ(LOCAL) +
* COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE) +
  CONNAUTH('USE.LDAP') +
* CRDATE(2020-10-26) +
* CRTIME(11.43.11) +
* QMID(qm1_2020-10-26_11.43.11) +
  SSLCRYP(' ') +
  SSLKEYR('/run/runmqserver/tls/key') +
  SUITEB(NONE) +
* VERSION(09020000) +
  FORCE
REFRESH SECURITY

```

## Co dalej

Teraz można zaktualizować kopię zapasową IBM MQ z uwzględnieniem informacji o autoryzacji LDAP.

### OpenShift V 9.2.1 CD EUS **LDAP, część 2: aktualizowanie kopii zapasowej IBM MQ z uwzględnieniem informacji o autoryzacji LDAP**

IBM MQ udostępnia szczegółowe reguły autoryzacji, które sterują dostępem do obiektów IBM MQ. Jeśli mechanizm uwierzytelniania i autoryzacji został zmieniony na LDAP, reguły autoryzacji mogą być niepoprawne i wymagać aktualizacji.

## Zanim rozpoczniesz

W tym zadaniu przyjęto założenie, że użytkownik zaktualizował kopię zapasową dla serwera LDAP.

## O tym zadaniu

### Czy muszę to zrobić?

Jeśli do uwierzytelniania i autoryzacji jest już używany protokół LDAP, zmiany nie są wymagane. Jeśli nie masz pewności, czy używany jest protokół LDAP, patrz sekcja [“Opcjonalnie: konfigurowanie protokołu LDAP” na stronie 42](#).

Aby zaktualizować informacje o autoryzacji LDAP, należy wykonać dwa etapy:

1. Usunięcie wszystkich istniejących uprawnień z pliku.
2. Zdefiniowanie nowych informacji o autoryzacji dla protokołu LDAP.

## Procedura

1. Usunięcie wszystkich istniejących uprawnień z pliku

W pliku kopii zapasowej, w pobliżu końca pliku, odszukaj kilka wpisów, które rozpoczynają się od SET AUTHREC:



```
Open [icon] *backup.mqsc /tmp
OBJTYPE(PROCESS) +
AUTHADD(CRT)
SET AUTHREC +
PROFILE('@CLASS') +
PRINCIPAL('CallumJackson@AzureAD') +
OBJTYPE(QMGR) +
AUTHADD(CRT)
SET AUTHREC +
PROFILE('@CLASS') +
GROUP('mqm@LAPTOP-VLQKJ5UH') +
OBJTYPE(QMGR) +
AUTHADD(CRT)
SET AUTHREC +
PROFILE('SYSTEM.ADMIN.CHANNEL.EVENT') +
PRINCIPAL('CallumJackson@AzureAD') +
OBJTYPE(Queue) +
AUTHADD(BROWSE,CHG,CLR,DLT,DSP,GET,INQ,PUT,PASSALL,PASSID,SET,SETALL,SETID)
SET AUTHREC +
PROFILE('SYSTEM.ADMIN.CHANNEL.EVENT') +
GROUP('mqm@LAPTOP-VLQKJ5UH') +
OBJTYPE(Queue) +
AUTHADD(BROWSE,CHG,CLR,DLT,DSP,GET,INQ,PUT,PASSALL,PASSID,SET,SETALL,SETID)

* Script ended on 2020-10-26 at 11.48.32
* Number of Inquiry commands issued: 14
* Number of Inquiry commands completed: 14
* Number of Inquiry responses processed: 295
* QueueManager count: 1
* Queue count: 57
* NameList count: 3
* Process count: 1
* Channel count: 11
* AuthInfo count: 4
* Listener count: 4
* Service count: 2
* CommInfo count: 1
* Topic count: 6
* Subscription count: 1
* ChlAuthRec count: 3
* AuthRec count: 199
* Number of objects/records: 293
*****
```

Znajdź istniejące wpisy i usuń je. Najprostszym podejściem jest usunięcie wszystkich istniejących reguł SET AUTHREC, a następnie utworzenie nowych pozycji na podstawie pozycji LDAP.



## 2. Zdefiniowanie nowych informacji o autoryzacji dla protokołu LDAP

W zależności od konfiguracji menedżera kolejek oraz liczby zasobów i grup może to być działanie czasochłonne lub przeciwnie, proste. W poniższym przykładzie założono, że menedżer kolejek ma tylko jedną kolejkę o nazwie Q1, a użytkownik chce zezwolić grupie LDAP apps na dostęp do niej.

```
SET AUTHREC GROUP('apps') OBJTYPE(QMGR) AUTHADD(ALL)
SET AUTHREC PROFILE('Q1') GROUP('apps') OBJTYPE(Queue) AUTHADD(ALL)
```

Pierwsza komenda AUTHREC dodaje uprawnienie dostępu do menedżera kolejek, a drugi zapewnia dostęp do kolejki. Jeśli wymagane jest uzyskanie dostępu do drugiej kolejki, niezbędna jest trzecia komenda AUTHREC, chyba że zastosuje się symbole wieloznaczne w celu zapewnienia bardziej ogólnego dostępu.

Poniżej przedstawiono inny przykład. Jeśli grupa administratorów (o nazwie admins) wymaga pełnego dostępu do menedżera kolejek, dodaj następujące komendy:

```
SET AUTHREC PROFILE('*') OBJTYPE(Queue) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Topic) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Channel) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(CLNTCONN) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(AUTHINFO) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Listener) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(NAMELIST) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Process) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Service) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(QMGR) GROUP('admins') AUTHADD(ALL)
```

### Co dalej

Teraz można zmienić adresy IP i nazwy hostów w konfiguracji produktu IBM MQ.

## Opcjonalnie: zmiana adresów IP i nazw hostów w konfiguracji produktu IBM MQ

Konfiguracja IBM MQ może mieć określone adresy IP i określone nazwy hostów. W niektórych sytuacjach mogą one pozostać bez zmian, podczas gdy w innych sytuacjach muszą zostać zmienione.

### Zanim rozpoczniesz

W tym zadaniu przyjęto założenie, że użytkownik ma skonfigurowany protokół LDAP.

### O tym zadaniu

#### Czy muszę to zrobić?

Najpierw należy określić, czy są określone jakiejkolwiek adresy IP lub nazwy hostów, poza konfiguracją LDAP zdefiniowaną w poprzedniej sekcji. W tym celu należy uruchomić następującą komendę:

```
grep 'CONNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc
```

Przykładowy wynik:

```
*****
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
--
DEFINE AUTHINFO('SYSTEM.DEFAULT.AUTHINFO.IDPWLDAP') +
  AUTHTYPE(IDPWLDAP) +
  ADOPTCTX(YES) +
  CONNAME(' ') +
--
REPLACE
```

```
DEFINE AUTHINFO('SYSTEM.DEFAULT.AUTHINFO.CRLLDAP') +  
  AUTHTYPE(CRLLDAP) +  
  CONNAME(' ') +
```

W tym przykładzie wyszukiwanie zwraca trzy wyniki. Jeden wynik odpowiada konfiguracji LDAP zdefiniowanej poprzednio. Można go zignorować, ponieważ nazwa hosta serwera LDAP pozostaje bez zmian. Pozostałe dwa wyniki to puste pozycje połączeń, więc można je również zignorować. Jeśli nie ma żadnych dodatkowych pozycji, można pominąć pozostałą część tego tematu.

## Procedura

### 1. Należy właściwie zinterpretować zwrócone wyniki.

Konfiguracja IBM MQ może w wielu aspektach zawierać adresy IP, nazwy hostów i porty. Można je sklasyfikować w dwóch kategoriach:

- a. **Położenie tego menedżera kolejek:** informacje o położeniu używane lub publikowane przez menedżer kolejek, które mogą być używane przez inne menedżery kolejek lub aplikacje w ramach sieci IBM MQ na potrzeby połączeń.
- b. **Położenie zależności menedżera kolejek:** położenia innych menedżerów kolejek lub systemów, które musi znać ten menedżer kolejek.

Ponieważ omawiany scenariusz dotyczy tylko zmian w konfiguracji tego menedżera kolejek, obsługiwane są tylko zmiany konfiguracji z kategorii (a). Jeśli jednak położenie tego menedżera kolejek jest przywoływane przez inne menedżery kolejek lub aplikacje, ich konfiguracje mogą wymagać zmiany w celu dopasowania do nowego położenia tego menedżera kolejek.

Istnieją dwa kluczowe obiekty, które mogą zawierać informacje wymagające zmiany:

- Procesy nasłuchujące: reprezentują adres sieciowy, na którym nasłuchuje produkt IBM MQ.
- Kanał odbiorczy klastra: istnieje, jeśli menedżer kolejek jest częścią klastra IBM MQ. Określa adres sieciowy, z którym mogą się łączyć inne menedżery kolejek.

### 2. W oryginalnych danych wyjściowych komendy `grep 'CONNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc` zidentyfikuj, czy są zdefiniowane kanały ODBIORNIKA KLASTRA. Jeśli tak, zaktualizuj adresy IP.

Aby sprawdzić, czy są zdefiniowane jakiegokolwiek kanały CLUSTER RECEIVER, znajdź wszystkie pozycje z parametrem CHLTYPE (CLUSRCVR) w pierwotnych wynikach:

```
DEFINE CHANNEL(ANY_NAME) +  
  CHLTYPE(CLUSRCVR) +
```

Jeśli istnieją pozycje, zaktualizuj element CONNAME przy użyciu IBM MQ Red Hat OpenShift Route. Ta wartość jest oparta na środowisku Red Hat OpenShift i używa przewidywalnej składni:

```
queue_manager_resource_name-ibm-mq-qm-openshift_project_name.openshift_app_route_hostname
```

Jeśli na przykład wdrożenie menedżera kolejek ma nazwę `qm1` w przestrzeni nazw `cp4i`, a `openshift_app_route_hostname` to `apps.callumj.icp4i.com`, adres URL trasy jest następujący:

```
qm1-ibm-mq-qm-cp4i.apps.callumj.icp4i.com
```

Numerem portu dla trasy jest zwykle 443. O ile administrator Red Hat OpenShift nie uzna inaczej, jest to zwykle poprawna wartość. Korzystając z tych informacji, należy zaktualizować pola CONNAME. Na przykład:

```
CONNAME('qm1-ibm-mq-qm-cp4i.apps.callumj.icp4i.com(443)')
```

W oryginalnych danych wyjściowych komendy `grep 'CONNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc` sprawdź, czy istnieją jakiegokolwiek pozycje dla produktu LOCLADDR lub IPADDRV. Jeśli tak, usuń je. Nie są one istotne w środowisku kontenera.

## Co dalej

Teraz można przystąpić do aktualizacji konfiguracji menedżera kolejek dla środowiska kontenera.

## Aktualizowanie konfiguracji menedżera kolejek dla środowiska kontenera

W przypadku pracy w kontenerze pewne aspekty konfiguracyjne są definiowane przez kontener i mogą być sprzeczne z wyeksportowaną konfiguracją.

## Zanim rozpoczniesz

W tym zadaniu założono, że użytkownik zmienił konfigurację IBM MQ w zakresie adresów IP i nazw hostów.

## O tym zadaniu

W kontenerze zdefiniowane są następujące aspekty konfiguracji:

- Definicje nasłuchiwania (które odpowiadają eksponowanym portom).
- Położenie ewentualnej składnicy TLS.

Dlatego konieczne jest zaktualizowanie wyeksportowanej konfiguracji:

1. Usuń wszystkie definicje nasłuchiwania.
2. Zdefiniuj położenie repozytorium kluczy TLS.

## Procedura

1. Usuń wszystkie definicje nasłuchiwania.

W konfiguracji zapasowej wyszukaj łańcuch `DEFINE LISTENER`. Powinna znajdować się między definicjami `AUTHINFO` i `SERVICE`. Zaznacz fragment i usuń go.

```
*backup.mqsc
** ALTDATA(2020-11-20) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE AUTHINFO('SYSTEM.DEFAULT.AUTHINFO.CRLLDAP') +
  AUTHTYPE(CRLLDAP) +
  CONNAME(' ') +
* ALTDATA(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE LISTENER('SYSTEM.DEFAULT.LISTENER.LU62') +
  TRPTYPE(LU62) +
  CONTROL(MANUAL) +
* ALTDATA(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE LISTENER('SYSTEM.DEFAULT.LISTENER.NETBIOS') +
  TRPTYPE(NETBIOS) +
  CONTROL(MANUAL) +
  LOCLNAME(' ') +
* ALTDATA(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE LISTENER('SYSTEM.DEFAULT.LISTENER.SPX') +
  TRPTYPE(SPX) +
  CONTROL(MANUAL) +
* ALTDATA(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE LISTENER('SYSTEM.DEFAULT.LISTENER.TCP') +
  TRPTYPE(TCP) +
  CONTROL(MANUAL) +
* ALTDATA(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE SERVICE('SYSTEM.AMQP.SERVICE') +
  CONTROL(QMGR) +
  SERVTYPE(SERVER) +
  STARTCMD('+MQ_INSTALL_PATH+\bin\amqp.bat') +
  STARTARG('start -m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+\.'
  STOPCMD('+MQ_INSTALL_PATH+\bin64\endmqsd.exe') +
```

2. Zdefiniuj położenie repozytorium kluczy TLS.

Kopia zapasowa menedżera kolejek zawiera konfigurację TLS oryginalnego środowiska. Różni się ona od stosowanej w środowisku kontenerów, dlatego wymaga kilku zmian:

- Zmień pozycję **CERTLABL** na default.
- Zmień położenie repozytorium kluczy TLS (**SSLKEYR**) na: /run/runmqserver/tls/key

Aby znaleźć atrybut **SSLKEYR** w pliku, należy wyszukać hasło **SSLKEYR**. Zwykle znajduje się tylko jedną pozycję. Jeśli zostanie znalezionych wiele pozycji, należy sprawdzić, czy edytowany jest obiekt **QMGR** przedstawiony na poniższej ilustracji:

```

*backup.mqsc
*****
* Script generated on 2020-10-21   at 11.48.32
* Script generated by user ' CallumJackso' on host 'LAPTOP-VLQKJ5UH'
* Queue manager name: qm1
* Queue manager platform: Windows
* Queue manager command level: (920/920)
* Command issued: dmpmqcfg -m qm1
*****
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
  SECCOMM(NO) +
  USRFIELD('uid') +
  SHORTUSR('uid') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  AUTHORMD(SEARCHGRP) +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  GRPFIELD('cn') +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  REPLACE
ALTER QMGR +
* ALTDATA(2020-10-26) +
* ALTTIME(11.43.11) +
  CCSTD(850) +
  CERTLABL('default') +
  CLWLUSEQ(LOCAL) +
* COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE) +
  CONNAUTH('USE.LDAP') +
* CRDATE(2020-10-26) +
* CRTIME(11.43.11) +
* QMID(qm1_2020-10-26_11.43.11) +
  SSLCRYP(' ') +
  SSLKEYR('/run/runmqserver/tls/key') +
  SUITEB(NONE) +
* VERSION(09020000) +
  FORCE
REFRESH SECURITY

```

## Co dalej

Teraz można wybrać docelową architekturę produktu IBM MQ działającego w kontenerach.

## OpenShift V 9.2.1 CD EUS Wybieranie docelowej architektury wysokiej dostępności dla programu IBM MQ działającego w kontenerach

Można wdrożyć instancję (jeden zasobnik Kubernetes) albo wiele instancji (dwa zasobniki), aby spełnić wymagania w zakresie wysokiej dostępności.

## Zanim rozpoczniesz

W tym zadaniu założono, że konfiguracja menedżera kolejek została zaktualizowana dla środowiska kontenera.

### O tym zadaniu

W produkcie IBM MQ Operator dostępne są dwa sposoby realizacji wysokiej dostępności:

- **Pojedyncza instancja:** działa pojedynczy kontener (zasobnik), który odpowiada za zrestartowanie produktu Red Hat OpenShift w przypadku wystąpienia awarii. Ze względu na charakterystykę zestawu stanowego w produkcie Kubernetes istnieje kilka sytuacji, w których to przełączenie awaryjne może zająć dłuższy czas lub wymagać wykonania działania administracyjnego.
- **Wiele instancji:** działają dwa kontenery (każdy w osobnym zasobniku), jeden w trybie aktywnym, a drugi w trybie rezerwowym. Ta topologia umożliwia znacznie szybsze przełączenie awaryjne. Wymaga ona systemu plików z dostępem Read Write Many, który spełnia wymagania produktu IBM MQ.

W tym zadaniu wybierana jest tylko docelowa architektura wysokiej dostępności. Czynności związane z konfigurowaniem wybranej architektury są opisane w kolejnym zadaniu w tym scenariuszu ("Tworzenie nowego menedżera kolejek w środowisku Red Hat OpenShift" na stronie 55).

### Procedura

1. Porównaj obie dostępne opcje.

Ich obszerny opis znajduje się w sekcji "Wysoka dostępność produktu IBM MQ w kontenerach" na stronie 16.

2. Wybierz docelową architekturę wysokiej dostępności.

Jeśli nie masz pewności, którą opcję wybrać, zacznij od **jednej instancji** i sprawdź, czy taka konfiguracja spełni wymagania w zakresie dostępności.

### Co dalej

Teraz można przystąpić do tworzenia zasobów menedżera kolejek.

## Tworzenie zasobów dla menedżera kolejek

Zaimportuj konfigurację produktu IBM MQ oraz certyfikaty TLS i klucze TLS do środowiska Red Hat OpenShift.

## Zanim rozpoczniesz

W tym zadaniu założono, że użytkownik wybrał architekturę docelową dla produktu IBM MQ działającego w kontenerach.

### O tym zadaniu

W poprzednich sekcjach wyodrębniono, zaktualizowano i zdefiniowano dwa zasoby:

- IBM MQ Konfiguracja
- Certyfikaty TLS i klucze

Należy zaimportować te zasoby do środowiska Red Hat OpenShift przed wdrożeniem menedżera kolejek.

### Procedura

1. Zaimportuj konfigurację IBM MQ do Red Hat OpenShift.

W poniższych instrukcjach przyjęto założenie, że konfiguracja produktu IBM MQ znajduje się w bieżącym katalogu, w pliku o nazwie `backup.mqsc`. W przeciwnym razie należy dostosować nazwę pliku do specyfiki środowiska.

- a) Zaloguj się do klastra przy użyciu komendy `oc login`.
- b) Załaduj konfigurację IBM MQ do programu `configmap`.

Uruchom następującą komendę:

```
oc create configmap my-mqsc-migrated --from-file=backup.mqsc
```

- c) Sprawdź, czy plik został załadowany pomyślnie.

Uruchom następującą komendę:

```
oc describe configmap my-mqsc-migrated
```

## 2. Importowanie zasobów TLS produktu IBM MQ

Tak, jak opisano to w sekcji [“Opcjonalnie: wyodrębnianie i pozyskanie kluczy i certyfikatów menedżera kolejek”](#) na stronie 40, dla wdrożenia menedżera kolejek może być wymagany protokół TLS. W takim przypadku należy mieć już kilka plików kończących się na `.crt` i `.key`. Należy dodać je do kluczy tajnych programu Kubernetes, aby menedżer kolejek mógł się do nich odwoływać podczas wdrażania.

Założmy, że masz klucz i certyfikat dla menedżera kolejek w plikach o nazwach:

- `qmgr.crt`
- `qmgr.key`

Aby zaimportować te pliki, uruchom następującą komendę:

```
oc create secret tls my-tls-migration --cert=qmgr.crt --key=qmgr.key
```

Program Kubernetes udostępnia ten pomocny program narzędziowy, gdy importowany jest zgodny klucz publiczny i prywatny. Jeśli istnieją dodatkowe certyfikaty, które można dodać, na przykład do magazynu zaufanych certyfikatów menedżera kolejek, uruchom następującą komendę:

```
oc create secret generic my-extra-tls-migration --from-file=comma_separated_list_of_files
```

Jeśli na przykład importowane pliki to `trust1.crt`, `trust2.crt` i `trust3.crt`, komenda jest następująca:

```
oc create secret generic my-extra-tls-migration --from-file=trust1.crt,trust2.crt,trust3.crt
```

## Co dalej

Teraz można przystąpić do [tworzenia nowego menedżera kolejek w środowisku Red Hat OpenShift](#).

## Tworzenie nowego menedżera kolejek w środowisku Red Hat OpenShift

W środowisku Red Hat OpenShift można przeprowadzić wdrożenie menedżera kolejek z jedną instancją lub wieloinstancyjnego.

## Zanim rozpocznesz

W tym zadaniu przyjęto założenie, że użytkownik [utworzył zasoby menedżera kolejek i zainstalował produkt IBM MQ Operator w rozwiązaniu Red Hat OpenShift](#).

## O tym zadaniu

Tak, jak wyjaśniono w sekcji [“Wybieranie docelowej architektury wysokiej dostępności dla programu IBM MQ działającego w kontenerach”](#) na stronie 53, istnieją dwie możliwe topologie wdrażania. W związku z tym w niniejszym temacie przedstawiono dwa różne szablony postępowania:

- Wdrażanie menedżera kolejek z jedną instancją.
- Wdrażanie menedżera kolejek z wieloma instancjami.

**Ważne:** Należy wykonać tylko jeden z dwóch szablonów, w zależności od preferowanej topologii.

## Procedura

- Wdrażanie menedżera kolejek z jedną instancją.

Migrowany menedżer kolejek jest wdrażany w rozwiązaniu Red Hat OpenShift przy użyciu pliku YAML. Poniżej znajduje się przykład przygotowany w oparciu o nazwy używane w poprzednich tematach:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1
spec:
  version: 9.2.5.0-r3
  license:
    accept: true
    license: L-RJON-C7QG3S
    use: "Production"
  pki:
    keys:
      - name: default
        secret:
          secretName: my-tls-migration
          items:
            - tls.key
            - tls.crt
  web:
    enabled: true
  queueManager:
    name: QM1
  mqsc:
    - configMap:
        name: my-mqsc-migrated
        items:
          - backup.mqsc
```

W zależności od wykonanych kroków może być konieczne dostosowanie wcześniejszego kodu YAML. Dla ułatwienia zamieszczono tutaj objaśnienie tego kodu YAML:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1
```

Definiuje obiekt Kubernetes, jego typ i nazwę. Jedyнным polem wymagającym dostosowania jest pole `name`.

```
spec:
  version: 9.2.5.0-r3
  license:
    accept: true
    license: L-RJON-C7QG3S
    use: "Production"
```



Te dane odpowiadają informacjom o wersji i licencji właściwym dla wdrożenia. Jeśli chcesz je zmienić, skorzystaj z informacji zamieszczonych w sekcji [“Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1”](https://mq.ibm.com/v1beta1) na stronie 130.

```
pki:
  keys:
  - name: default
    secret:
      secretName: my-tls-migration
      items:
      - tls.key
      - tls.crt
```

Menedżer kolejek, który ma być skonfigurowany do używania protokołu TLS, musi odwoływać się do odpowiednich certyfikatów i kluczy. Pole `secretName` odwołuje się do klucza tajnego Kubernetes utworzonego w sekcji Importowanie zasobów protokołu TLS IBM MQ, a lista elementów (`tls.key` i `tls.crt`) przedstawia standardowe nazwy, jakie Kubernetes przypisuje, korzystając ze składni `oc create secret tls`. Jeśli istnieją dodatkowe certyfikaty, które mają zostać dodane do magazynu zaufanych certyfikatów, można je dodać w podobny sposób, ale pozycje są odpowiednimi nazwami plików używanymi podczas importowania. Na przykład w celu utworzenia certyfikatów magazynu zaufanych certyfikatów można użyć następującego kodu:

```
oc create secret generic my-extra-tls-migration --from-file=trust1.crt,trust2.crt,trust3.crt
```

```
pki:
  trust:
  - name: default
    secret:
      secretName: my-extra-tls-migration
      items:
      - trust1.crt
      - trust2.crt
      - trust3.crt
```

**Ważne:** Jeśli protokół TLS nie jest wymagany, usuń sekcję TLS z kodu YAML.

```
web:
  enabled: true
```

Umożliwia włączenie konsoli WWW dla wdrożenia.

```
queueManager:
  name: QM1
```

Definiuje nazwę menedżera kolejek jako QM1. Menedżer kolejek jest dostosowany na podstawie wymagań użytkownika, na przykład zgodnie z nazwą pierwotnego menedżera kolejek.

```
mjsc:
  - configMap:
      name: my-mjsc-migrated
      items:
      - backup.mjsc
```

Poprzedni kod pobiera konfigurację menedżera kolejek, która została zaimportowana w sekcji [Importowanie konfiguracji produktu IBM MQ](#). Jeśli używane są inne nazwy, należy zmodyfikować `my-mjsc-migrated` i `backup.mjsc`.

Należy zauważyć, że przykład YAML zakłada, że domyślna klasa pamięci masowej dla środowiska Red Hat OpenShift jest zdefiniowana jako klasa pamięci RWX lub RWO. Jeśli wartość domyślna nie jest zdefiniowana w danym środowisku, należy określić klasę pamięci masowej, która ma być używana. Można to zrobić, uzupełniając kod YAML w następujący sposób:

```
queueManager:
  name: QM1
  storage:
    defaultClass: my_storage_class
```

```
queueManager:
  type: persistent-claim
```

Dodaj podświetlony tekst wraz z atrybutem klasy dostosowanym do specyfiki danego środowiska. Aby ustalić nazwy klas pamięci masowej istniejących w środowisku, uruchom następującą komendę:

```
oc get storageclass
```

Poniżej znajdują się przykładowe dane zwracane przez tę komendę:

NAME	PROVISIONER	RECLAIMPOLICY
aws-efs	openshift.org/aws-efs	Delete
gp2 (default)	kubernetes.io/aws-efs	Delete

Poniższy kod przedstawia sposób odwoływania się do konfiguracji IBM MQ, która została zaimportowana w sekcji [Importowanie konfiguracji produktu IBM MQ](#). Jeśli używane są inne nazwy, należy zmodyfikować `my-mqsc-migrated` i `backup.mqsc`.

```
mqsc:
  - configMap:
      name: my-mqsc-migrated
    items:
      - backup.mqsc
```

Wdrożono menedżer kolejek z pojedynczą instancją. Na tym szablon kończy się. Teraz można przystąpić do [zweryfikowania nowego wdrożenia w kontenerze](#).

- Wdrażanie menedżera kolejek z wieloma instancjami.

Migrowany menedżer kolejek jest wdrażany w rozwiązaniu Red Hat OpenShift przy użyciu pliku YAML. Poniższy przykład bazuje na nazwach używanych w poprzednich sekcjach.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1mi
spec:
  version: 9.2.5.0-r3
  license:
    accept: true
    license: L-RJON-C7QG3S
    use: "Production"
  pki:
    keys:
      - name: default
        secret:
          secretName: my-tls-migration
        items:
          - tls.key
          - tls.crt
  web:
    enabled: true
  queueManager:
    name: QM1
    availability: MultiInstance
  storage:
    defaultClass: aws-efs
    persistedData:
      enabled: true
    queueManager:
      enabled: true
    recoveryLogs:
      enabled: true
  mqsc:
    - configMap:
        name: my-mqsc-migrated
      items:
        - backup.mqsc
```

Zamieszczono tutaj objaśnienie tego kodu YAML. Większość konfiguracji będzie przebiegać zgodnie z tym samym podejściem, co wdrażanie menedżera kolejek z pojedynczą instancją, dlatego w tym miejscu opisano tylko zagadnienia dotyczące dostępności i pamięci masowej menedżera kolejek.

```
queueManager:  
  name: QM1  
  availability: MultiInstance
```

Nazwa menedżera kolejek jest określana jako QM1, a następnie ustawiana dla wdrożenia na wartość MultiInstance zamiast domyślnej instancji pojedynczej.

```
storage:  
  defaultClass: aws-efs  
  persistedData:  
    enabled: true  
  queueManager:  
    enabled: true  
  recoveryLogs:  
    enabled: true
```

Wieloinstancyjny menedżer kolejek IBM MQ jest zależny od pamięci masowej RWX. Domyślnie menedżer kolejek jest wdrażany w trybie pojedynczej instancji, dlatego w przypadku zmiany na tryb wieloinstancyjny wymagane są dodatkowe opcje pamięci masowej. W poprzednim przykładzie kodu YAML zdefiniowane są trzy trwałe woluminy pamięci masowej i klasa woluminu utrwalonego. Ta klasa woluminu utrwalonego musi być klasą pamięci RWX. W razie wątpliwości co do nazw klas pamięci masowej w danym środowisku, można uruchomić następującą komendę, aby je wykryć:

```
oc get storageclass
```

Poniżej znajdują się przykładowe dane zwracane przez tę komendę:

NAME	PROVISIONER	RECLAIMPOLICY
aws-efs	openshift.org/aws-efs	Delete
gp2 (default)	kubernetes.io/aws-efs	Delete

Poniższy kod przedstawia sposób odwoływania się do konfiguracji IBM MQ, która została zaimportowana w sekcji [Importowanie konfiguracji produktu IBM MQ](#). Jeśli używane są inne nazwy, należy zmodyfikować my-mqsc-migrated i backup.mqsc.

```
mqsc:  
  - configMap:  
      name: my-mqsc-migrated  
      items:  
        - backup.mqsc
```

Wieloinstancyjny menedżer kolejek został wdrożony. Na tym szablon kończy się. Teraz można przystąpić do [zweryfikowania nowego wdrożenia w kontenerze](#).

## **Sprawdzanie poprawności nowego wdrożenia w kontenerze**

Po wdrożeniu produktu IBM MQ w środowisku Red Hat OpenShift można zweryfikować środowisko przy użyciu przykładów produktu IBM MQ.

### Zanim rozpocznieš

W tym zadaniu przyjęto założenie, że [utworzony został nowy menedżer kolejek w środowisku Red Hat OpenShift](#).

**Ważne:** W tym zadaniu przyjęto założenie, że w menedżerze kolejek nie jest włączony protokół TLS.

## O tym zadaniu

W tym zadaniu można uruchomić przykłady produktu IBM MQ z wnętrza kontenera zmigrowanego menedżera kolejek. Użytkownik może jednak preferować korzystanie z własnych aplikacji działających w innym środowisku.

Wymagane są następujące informacje:

- Nazwa użytkownika LDAP
- Hasło LDAP
- IBM MQ Nazwa kanału
- Nazwa kolejki

W tym przykładowym kodzie używane są następujące ustawienia. Należy pamiętać, że ustawienia w środowisku użytkownika będą inne.

- Nazwa użytkownika LDAP: mqapp
- Hasło LDAP: mqapp
- IBM MQ Nazwa kanału: DEV.APP.SVRCONN
- Nazwa kolejki: Q1

## Procedura

1. Wykonaj komendę Exec względem działającego kontenera IBM MQ.

Użyj następującej komendy:

```
oc exec -it qm1-ibm-mq-0 /bin/bash
```

gdzie `qm1-ibm-mq-0` to produkt, który wdrożyliśmy w systemie [“Tworzenie nowego menedżera kolejek w środowisku Red Hat OpenShift” na stronie 55](#). Jeśli wdrożenie ma inną nazwę, należy zmienić tę wartość.

2. Wyślij komunikat.

Wykonaj następujące komendy:

```
cd /opt/mqm/samp/bin
export IBM MQSAMP_USER_ID=mqapp
export IBM MQSERVÉR=DEV.APP.SVRCONN/TCP/'localhost(1414) '
./amqsputc Q1 QM1
```

Użytkownik zostanie poproszony o podanie hasła, a następnie może wysłać komunikat.

3. Sprawdź, czy komunikat został odebrany pomyślnie.

Uruchom przykład GET:

```
./amqsgetc Q1 QM1
```

## Wyniki

Procedura [“Migracja produktu IBM MQ do rozwiązania IBM Cloud Pak for Integration” na stronie 38](#) została zakończona.

## Co dalej

Użyj poniższych informacji, aby uzyskać pomoc w bardziej złożonych scenariuszach migracji:

## Migracja komunikatów z kolejek

Aby przeprowadzić migrację komunikatów istniejących w kolejkach, należy postępować zgodnie ze wskazówkami zamieszczonymi w następującym temacie w celu eksportowania i importowania komunikatów po utworzeniu nowego menedżera kolejek: [Używanie programu narzędziowego dmpmqmsg między dwoma systemami](#).

## Nawiązywanie połączenia z produktem IBM MQ spoza środowiska Red Hat OpenShift

Wdrożony menedżer kolejek może być narażony na działanie klientów IBM MQ i menedżerów kolejek spoza środowiska Red Hat OpenShift. Proces jest zależny od wersji produktu IBM MQ łączącego się ze środowiskiem Red Hat OpenShift. Patrz sekcja ["Konfigurowanie trasy w celu nawiązania połączenia z menedżerem kolejek spoza klastra produktu Red Hat OpenShift"](#) na stronie 112.

## Instalowanie i deinstalowanie produktu IBM MQ Operator w rozwiązaniu Red Hat OpenShift

Produkt IBM MQ Operator można zainstalować w rozwiązaniu Red Hat OpenShift przy użyciu rozwiązania Operator Hub.

### Procedura

- ["Zależności operatora IBM MQ Operator"](#) na stronie 10.
- ["Uprawnienia o zasięgu klastra wymagane przez IBM MQ Operator"](#) na stronie 11.
- ["Instalowanie produktu IBM MQ Operator przy użyciu konsoli WWW Red Hat OpenShift"](#) na stronie 61.
- ["Instalowanie produktu IBM MQ Operator przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift"](#) na stronie 63.
- ["Instalowanie operatora IBM MQ Operator w środowisku z separacją"](#) na stronie 67.

### Zadania pokrewne

["Deinstalowanie produktu IBM MQ Operator przy użyciu konsoli WWW Red Hat OpenShift"](#) na stronie 63  
Przy użyciu konsoli WWW Red Hat OpenShift można zdeinstalować IBM MQ Operator z Red Hat OpenShift.

["Deinstalowanie produktu IBM MQ Operator przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift"](#) na stronie 66

Przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift można zdeinstalować produkt IBM MQ Operator z Red Hat OpenShift. Proces deinstalacji różni się w zależności od tego, czy IBM MQ Operator jest zainstalowany w jednej przestrzeni nazw, czy też został zainstalowany i jest dostępny we wszystkich przestrzeniach nazw w klastrze.


## Instalowanie produktu IBM MQ Operator przy użyciu konsoli WWW Red Hat OpenShift

Produkt IBM MQ Operator można zainstalować w rozwiązaniu Red Hat OpenShift przy użyciu rozwiązania Operator Hub.

### Zanim rozpoczniesz

Zaloguj się do konsoli WWW klastra produktu Red Hat OpenShift.

### Procedura

1.  Opcjonalne: Dodawanie operatorów IBM Common Services do listy instalowalnych operatorów.

#### Uwaga:

Ten krok dotyczy wydań IBM MQ Operator 1.5 i wcześniejszych. Ten krok dodaje osobny katalog Common Services. W przypadku nowszych wydań operatora usługi Common Services są zawarte w katalogu IBM.

- a) Kliknij ikonę plusa w prawym górnym rogu ekranu. Zostanie wyświetlone okno dialogowe **Import YAML**.
- b) Wklej następującą definicję zasobu w oknie dialogowym.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: opencloud-operators
  namespace: openshift-marketplace
spec:
  displayName: IBMCS Operators
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-common-service-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```

- c) Kliknij opcję **Create**.
2. Dodawanie operatorów IBM do listy instalowalnych operatorów
- a) Kliknij ikonę plusa w prawym górnym rogu ekranu. Zostanie wyświetlone okno dialogowe **Import YAML**.
  - b) Wklej następującą definicję zasobu w oknie dialogowym.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: IBM Operator Catalog
  image: icr.io/cpopen/ibm-operator-catalog:latest
  publisher: IBM
  sourceType: grpc
  updateStrategy:
    registryPoll:
      interval: 45m
```

- c) Kliknij opcję **Create**.
3. Tworzenie przestrzeni nazw, która ma być używana dla operatora IBM MQ Operator
- Operator IBM MQ Operator może być zainstalowany w zasięgu jednej przestrzeni nazw lub wszystkich przestrzeni nazw. Ten krok jest wymagany tylko wtedy, gdy chcesz zainstalować operator w określonej przestrzeni nazw, która jeszcze nie istnieje.
- a) Na panelu nawigacyjnym kliknij opcję **Home > Projects**.  
Zostanie wyświetlona strona Projects.
  - b) Kliknij opcję **Create Project**. Zostanie wyświetlony obszar Create Project.
  - c) Wprowadź szczegóły tworzonej przestrzeni nazw. Na przykład jako nazwę można podać „ibm-mq”.
  - d) Kliknij opcję **Create**. Zostanie utworzona przestrzeń nazw dla operatora IBM MQ Operator.
4. Zainstaluj operator IBM MQ Operator.
- a) Na panelu nawigacyjnym kliknij opcję **Operators > OperatorHub**.  
Zostanie wyświetlona strona OperatorHub.
  - b) W polu **All Items** wprowadź „IBM MQ”.  
Zostanie wyświetlony wpis IBM MQ w katalogu.
  - c) Wybierz wpis **IBM MQ**.  
Zostanie wyświetlone okno IBM MQ.
  - d) Kliknij opcję **Install**.

Zostanie wyświetlona strona tworzenia subskrypcji operatora.

e) Zapoznaj się z sekcją [“Wersja zgodna z operatorem IBM MQ Operator” na stronie 7](#), aby określić, który kanał operatora ma zostać wybrany.

f) Ustaw tryb instalacji na określoną przez użytkownika przestrzeń nazw lub zasięg klastra.

Wybór zasięgu całego klastra jest zalecany, ponieważ instalowanie różnych wersji operatora w różnych przestrzeniach nazw może prowadzić do problemów. Operatorzy zaprojektowano z myślą o rozszerzeniach płaszczyzny sterowania.

g) Kliknij opcję **Subscribe**.

IBM MQ pojawi się na stronie zainstalowanych operatorów.

h) Sprawdź status operatora na stronie zainstalowanych operatorów. Po zakończeniu instalacji status zmieni się na Succeeded.

## Co dalej

[“Przygotowywanie projektu Red Hat OpenShift dla produktu IBM MQ przy użyciu konsoli WWW Red Hat OpenShift” na stronie 82](#)

## **Deinstalowanie produktu IBM MQ Operator przy użyciu konsoli WWW Red Hat OpenShift**

Przy użyciu konsoli WWW Red Hat OpenShift można zdeinstalować IBM MQ Operator z Red Hat OpenShift.

## Zanim rozpocznie

Zaloguj się do konsoli WWW klastra produktu Red Hat OpenShift.

Jeśli operator IBM MQ Operator jest zainstalowany we wszystkich projektach/przestrzeniach nazw w klastrze, powtórz kroki od 1 do 5 poniższej procedury dla każdego projektu, z którego mają zostać usunięte menedżery kolejek.

## Procedura

1. Wybierz opcję **Operators > Installed Operators**.
2. Z listy rozwijanej **Project** wybierz projekt.
3. Kliknij operator **IBM MQ**.
4. Kliknij kartę **Queue Managers**, aby wyświetlić menedżery kolejek zarządzane przez ten operator IBM MQ Operator.
5. Usuń jeden lub więcej menedżerów kolejek.

Należy zauważyć, że pomimo tego, że menedżery kolejek nadal działają, mogą nie działać zgodnie z oczekiwaniami bez operatora IBM MQ Operator.

6. Opcjonalne: W razie potrzeby powtórz kroki od 1 do 5 dla każdego projektu, z którego mają zostać usunięte menedżery kolejek.
7. Wróć do opcji **Operators > Installed Operators**.
8. Obok operatora **IBM MQ** kliknij wielokropek i wybierz opcję **Uninstall Operator**.
9. Jeśli używany jest produkt Red Hat OpenShift Container Platform 4.7, może być konieczne ręczne usunięcie haka WWW sprawdzania poprawności przy użyciu wiersza komend:

```
oc delete validatingwebhookconfiguration namespace.validator.queuemanagers.mq.ibm.com
```

## **Instalowanie produktu IBM MQ Operator przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift**

Produkt IBM MQ Operator można zainstalować w rozwiązaniu Red Hat OpenShift przy użyciu rozwiązania Operator Hub.

## Zanim rozpocznieš

Zaloguj się do interfejsu wiersza komend produktu Red Hat OpenShift przy użyciu programu **oc login**. Aby wykonać te kroki, należy być administratorem klastra.

## Procedura

### 1. EUS

Opcjonalne: Utwórz **CatalogSource** dla elementów wykonawczych Common Services IBM.

#### Uwaga:

Ten krok dotyczy wydań IBM MQ Operator 1.5 i wcześniejszych. Ten krok dodaje osobny katalog Common Services. W przypadku nowszych wydań operatora usługi Common Services są zawarte w katalogu IBM.

#### a) Utwórz plik YAML definiujący zasób **CatalogSource**.

Utwórz plik o nazwie „operator-source-cs.yaml” o następującej zawartości:

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: opencloud-operators
  namespace: openshift-marketplace
spec:
  displayName: IBMCS Operators
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-common-service-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```

#### b) Zastosuj **CatalogSource** do serwera.

```
oc apply -f operator-source-cs.yaml -n openshift-marketplace
```

### 2. Utwórz **CatalogSource** dla elementów wykonawczych IBM

#### a) Utwórz plik YAML definiujący zasób **CatalogSource**

Utwórz plik o nazwie „operator-source-ibm.yaml” o następującej zawartości:

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: IBM Operator Catalog
  image: icr.io/cpopen/ibm-operator-catalog:latest
  publisher: IBM
  sourceType: grpc
  updateStrategy:
    registryPoll:
      interval: 45m
```

#### b) Zastosuj **CatalogSource** do serwera.

```
oc apply -f operator-source-ibm.yaml -n openshift-marketplace
```

### 3. Tworzenie przestrzeni nazw, która ma być używana dla operatora IBM MQ Operator

Operator IBM MQ Operator może być zainstalowany w zasięgu jednej przestrzeni nazw lub wszystkich przestrzeni nazw. Ten krok jest wymagany tylko wtedy, gdy chcesz zainstalować operator w określonej przestrzeni nazw, która jeszcze nie istnieje.

```
oc new-project ibm-mq
```

### 4. Wyświetlenie listy operatorów dostępnych dla klastra z serwisu OperatorHub



```
oc get packagemanifests -n openshift-marketplace
```

5. Sprawdzanie obsługiwanych trybów instalacji (InstallMode) i dostępnych kanałów operatora IBM MQ Operator

```
oc describe packagemanifests ibm-mq -n openshift-marketplace
```

6. Utwórz plik YAML obiektu **OperatorGroup**

**OperatorGroup** to zasób OLM, który wybiera docelowe przestrzenie nazw, w których ma zostać wygenerowany wymagany dostęp RBAC dla wszystkich elementów wykonawczych w tej samej przestrzeni nazw co **OperatorGroup**.

Przestrzeń nazw, do której subskrybujesz element wykonawczy, musi mieć składnik **OperatorGroup** odpowiadający składnikowi **InstallMode** elementu wykonawczego, albo AllNamespaces, albo SingleNamespace. Jeśli element wykonawczy, który ma zostać zainstalowany, korzysta z przestrzeni nazw AllNamespaces, to przestrzeń nazw openshift-operators ma już odpowiedni składnik **OperatorGroup**.

Jeśli jednak elementy wykonawczy korzysta z przestrzeni nazw SingleNamespace, a użytkownik nie ma odpowiedniego składnika **OperatorGroup**, należy go utworzyć.

- a) Utwórz plik o nazwie „mq-operator-group.yaml” o następującej zawartości:

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: <operatorgroup_name>
  namespace: <namespace_name>
spec:
  targetNamespaces:
  - <namespace_name>
```

- b) Utwórz obiekt **OperatorGroup**

```
oc apply -f mq-operator-group.yaml
```

7. Utwórz plik YAML obiektu **Subscription**, aby zasubskrybować przestrzeń nazw dla IBM MQ Operator

- a) Zapoznaj się z sekcją [“Wersja zgodna z operatorem IBM MQ Operator”](#) na stronie 7, aby określić, który kanał operatora ma zostać wybrany.
- b) Utwórz plik o nazwie "mq-sub.yaml" z następującą zawartością, ale zmieniając wartość **channel**, aby była zgodna z kanałem dla wersji produktu IBM MQ Operator, który ma zostać zainstalowany.

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-mq
  namespace: openshift-operators
spec:
  channel: <ibm-mq-operator-channel>
  name: ibm-mq
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
```

W przypadku użycia przestrzeni nazw AllNamespaces **InstallMode** określ **openshift-operators** w przestrzeni nazw. W przeciwnym przypadku określ odpowiednią pojedynczą przestrzeń nazw dla użycia przestrzeni SingleNamespace **InstallMode**. Uwaga: należy zmienić wyłącznie pole **namespace**, pozostawiając pole **sourceNamespace** bez zmian.

- c) Utwórz obiekt **Subscription**

```
oc apply -f mq-sub.yaml
```

8. Sprawdź status elementu wykonawczego

Po pomyślnym zakończeniu instalacji elementu wykonawczego status zasobnika jest wyświetlany jako *Działający*. W przypadku użycia przestrzeni nazw AllNamespaces **InstallMode** określ **openshift-**

**operators** jako przestrzeń nazw. W przeciwnym przypadku określ odpowiednią pojedynczą przestrzeń nazw dla użycia przestrzeni `SingleNamespace` **InstallMode**.

```
oc get pods -n <namespace_name>
```

## Co dalej

[“Przygotowywanie projektu produktu Red Hat OpenShift dla produktu IBM MQ przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift” na stronie 83](#)

## **Deinstalowanie produktu IBM MQ Operator przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift**

Przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift można zdeinstalować produkt IBM MQ Operator z Red Hat OpenShift. Proces deinstalacji różni się w zależności od tego, czy IBM MQ Operator jest zainstalowany w jednej przestrzeni nazw, czy też został zainstalowany i jest dostępny we wszystkich przestrzeniach nazw w klastrze.

## Zanim rozpocznie

Zaloguj się do klastra produktu Red Hat OpenShift przy użyciu programu `oc login`.

## Procedura

- Jeśli IBM MQ Operator jest zainstalowany w jednej przestrzeni nazw, wykonaj następujące kroki podrzędne:

- a) Upewnij się, że jesteś w poprawnym projekcie:

```
oc project <project_name>
```

- b) Wyświetl menedżery kolejek zainstalowane w projekcie:

```
oc get qmgr
```

- c) Usuń jeden lub więcej menedżerów kolejek:

```
oc delete qmgr <qmgr_name>
```

Należy zauważyć, że pomimo tego, że menedżery kolejek nadal działają, mogą nie działać zgodnie z oczekiwaniami bez operatora IBM MQ Operator.

- d) Wyświetl instancje **ClusterServiceVersion**:

```
oc get csv
```

- e) Usuń IBM MQ **ClusterServiceVersion**:

```
oc delete csv <ibm_mq_csv_name>
```

- f) Wyświetl subskrypcje:

```
oc get subscription
```

- g) Usuń wszystkie subskrypcje:

```
oc delete subscription <ibm_mq_subscription_name>
```

- h) Opcjonalne: Jeśli żadna inna usługa nie używa wspólnych usług, celowe może być zdeinstalowanie operatora usług wspólnych i usunięcie grupy operatorów:

- a. Aby zdeinstalować operatora usług wspólnych, postępuj zgodnie z instrukcjami w sekcji [Deinstalowanie usług wspólnych](#) w dokumentacji produktu IBM Cloud Pak foundational services.

b. Wyświetl grupę operatorów:

```
oc get operatorgroup
```

c. Usuń grupę operatorów:

```
oc delete OperatorGroup <operator_group_name>
```

- Jeśli IBM MQ Operator jest zainstalowany i dostępny dla wszystkich przestrzeni nazw w klastrze, wykonaj następujące kroki podrzędne:

a) Wyświetl wszystkie zainstalowane menedżery kolejek:

```
oc get qmgr -A
```

b) Usuń jeden lub więcej menedżerów kolejek:

```
oc delete qmgr <qmgr_name> -n <namespace_name>
```

Należy zauważyć, że pomimo tego, że menedżery kolejek nadal działają, mogą nie działać zgodnie z oczekiwaniami bez operatora IBM MQ Operator.

c) Wyświetl instancje **ClusterServiceVersion**:

```
oc get csv -A
```

d) Usuń IBM MQ **ClusterServiceVersion** z klastra:

```
oc delete csv <ibm_mq_csv_name> -n openshift-operators
```

e) Wyświetl subskrypcje:

```
oc get subscription -n openshift-operators
```

f) Usuń subskrypcje:

```
oc delete subscription <ibm_mq_subscription_name> -n openshift-operators
```

g) Jeśli używany jest produkt Red Hat OpenShift Container Platform 4.7, może być konieczne ręczne usunięcie haka WWW sprawdzania poprawności:

```
oc delete validatingwebhookconfiguration namespace.validator.queuemanagers.mq.ibm.com
```

h) Opcjonalne: Jeśli żadna inna usługa nie używa wspólnych usług, celowe może być zdeinstalowanie operatora usług wspólnych:

Należy postępować zgodnie z instrukcjami w sekcji [Deinstalowanie usług wspólnych](#) w dokumentacji produktu IBM Cloud Pak foundational services.

## Instalowanie operatora IBM MQ Operator w środowisku z separacją

Ten kurs dotyczy instalowania IBM MQ Operator w klastrze Red Hat OpenShift, który nie ma połączenia z Internetem. Operator IBM MQ Operator można zainstalować w środowisku z separacją przy użyciu przenośnego urządzenia pamięci masowej lub przy użyciu komputera bastionowego.

### Instalowanie operatora IBM MQ Operator w środowisku z separacją przy użyciu przenośnego urządzenia pamięci masowej

Kroki instalacji opisano w sekcji [Tworzenie kopii lustrzanych obrazów z użyciem przenośnego urządzenia pamięci masowej](#) w dokumentacji produktu IBM Cloud Pak for Integration. Jeśli

instalowany jest tylko produkt IBM MQ, należy zastąpić wszystkie wystąpienia następujących zmiennych środowiskowych wartościami podanymi w tym miejscu:

```
export CASE_NAME=ibm-mq
export CASE_ARCHIVE_VERSION=version_number
export CASE_INVENTORY_SETUP=ibmMQOperator
```

gdzie *numer\_wersji* jest wersją przypadku, która ma być używana do wykonania instalacji z separacją. Listę dostępnych wersji przypadków można znaleźć pod adresem <https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq>. Zapoznaj się z sekcją [“Wersja zgodna z operatorem IBM MQ Operator”](#) na stronie 7, aby określić, który kanał operatora ma zostać wybrany.

## Instalowanie operatora IBM MQ Operator w środowisku z separacją przy użyciu komputera-bastionu

1. [“Wymagania wstępne”](#) na stronie 68
2. [“Przygotowanie rejestru Docker”](#) na stronie 68
3. [“Przygotowanie hosta bastionowego”](#) na stronie 69
4. [“Tworzenie zmiennych środowiskowych dla instalatora i spisu obrazów”](#) na stronie 70
5. [“Pobieranie instalatora IBM MQ i spisu obrazów”](#) na stronie 70
6. [“Logowanie się do klastra Red Hat OpenShift Container Platform w roli administratora klastra”](#) na stronie 70
7. [“Utworzenie przestrzeni nazw Kubernetes dla operatora IBM MQ Operator”](#) na stronie 71
8. [“Tworzenie kopii lustrzanych obrazów i konfigurowanie klastra”](#) na stronie 71
9. [“Zainstaluj operator IBM MQ Operator.”](#) na stronie 73
10. [“Wdrażanie menedżera kolejek IBM MQ”](#) na stronie 73

## Wymagania wstępne

1. Musi być zainstalowany klaster Red Hat OpenShift Container Platform . Aby uzyskać informacje na temat obsługiwanych wersji Red Hat OpenShift Container Platform, patrz [“Wersja zgodna z operatorem IBM MQ Operator”](#) na stronie 7.
2. Musi być dostępny rejestr Docker. Więcej informacji na ten temat zawiera sekcja [“Przygotowanie rejestru Docker”](#) na stronie 68.
3. Musi być skonfigurowany serwer bastionowy. Więcej informacji na ten temat zawiera [“Przygotowanie hosta bastionowego”](#) na stronie 69.

## Przygotowanie rejestru Docker

Lokalny rejestr Docker jest używany do przechowywania wszystkich obrazów w środowisku lokalnym. Należy utworzyć taki rejestr spełniający następujące wymagania:

- Musi obsługiwać [Docker Manifest V2, Schema 2](#).
- Musi obsługiwać obrazy z więcej niż jedną architekturą.
- Musi być dostępny zarówno z serwera bastionowego, jak i z węzłów klastra Red Hat OpenShift Container Platform.
- Musi mieć nazwę użytkownika i hasło użytkownika z uprawnieniami do zapisu w rejestrze docelowym z hosta bastionowego.
- Musi mieć nazwę użytkownika i hasło użytkownika z uprawnieniami do zapisu w rejestrze docelowym w węzłach klastra Red Hat OpenShift.
- Musi dopuszczać separatory ścieżek w nazwach obrazów.

Po utworzeniu rejestru Docker należy go skonfigurować:

1. Tworzenie przestrzeni nazw rejestru

- `ibmcom` – przestrzeń nazw do przechowywania wszystkich obrazów z przestrzeni nazw `dockerhub.io/ibmcom`.

Przeźnię nazw `ibmcom` jest przeznaczona dla wszystkich obrazów IBM, które są publicznie dostępne i do pobrania nie wymagają podawania referencji.

- `cp` – przestrzeń nazw do przechowywania obrazów IBM z repozytorium `cp.icr.io/cp`.

Przeźnię nazw `cp` jest przeznaczona dla obrazów z całego rejestru IBM Entitled Registry, które do pobrania wymagają klucza uprawnień do produktu i referencji. Aby uzyskać klucz, zaloguj się do biblioteki [MyIBM Container Software Library](#) przy użyciu identyfikatora IBM ID i hasła, które są powiązane z uprawnieniem do oprogramowania. W sekcji **Entitlement keys** wybierz opcję **Copy key**, aby skopiować klucz uprawnień do schowka, a następnie zapisz go do wykorzystania w kolejnych krokach.

- `opencloudio` – przestrzeń nazw do przechowywania obrazów z repozytorium `quay.io/opencloudio`.

Przeźnię nazw `opencloudio` jest przeznaczona dla wybranych obrazów komponentów Open Source oprogramowania IBM, które są dostępne w serwisie `quay.io`. Obrazy produktu IBM Cloud Pak foundational services są utrzymywane w repozytorium `opencloudio`.

## 2. Upewnij się, że każda przestrzeń nazw spełnia następujące wymagania:

- Obsługuje automatyczne tworzenie repozytorium.
- Ma referencje użytkownika z uprawnieniami do zapisu w repozytoriach i tworzenia repozytoriów. Host bastionowy używa tych referencji.
- Ma referencje użytkownika z uprawnieniami do odczytu wszystkich repozytoriów. Klaster Red Hat OpenShift Container Platform używa tych referencji.

## Przygotowanie hosta bastionowego

Przygotuj host bastionowy, który może uzyskać dostęp do klastra Red Hat OpenShift Container Platform, do lokalnego rejestru Docker i do Internetu. Host bastionowy musi znajdować się na platformie Linux for x86-64 z dowolnym systemem operacyjnym obsługiwany przez interfejs wiersza komend IBM Cloud Pak i interfejs wiersza komend Red Hat OpenShift Container Platform.

Wykonaj następujące kroki w węźle bastionowym:

1. Zainstaluj program OpenSSL w wersji 1.11.1 lub wyższej.
2. Zainstaluj program Docker lub Podman w węźle bastionowym.

- Aby zainstalować program Docker, uruchom następujące komendy:

```
yum check-update
yum install docker
```

- Aby zainstalować program Podman, patrz [Instrukcja instalacji programu Podman](#)

3. Zainstaluj program skopeo 1.x.x w węźle bastionowym. Aby zainstalować program skopeo, uruchom następujące komendy:

```
yum check-update
yum install skopeo
```

4. Zainstaluj interfejs wiersza komend IBM Cloud Pak. Zainstaluj najnowszą wersję pliku binarnego dla używanej platformy. Aby uzyskać więcej informacji, patrz [cloud-pak-cli](#).

- a. Pobierz plik binarny.

```
wget https://github.com/IBM/cloud-pak-cli/releases/download/vversion-number/binary-file-name
```

Na przykład:

```
wget https://github.com/IBM/cloud-pak-cli/releases/latest/download/cloudctl-linux-amd64.tar.gz
```

b. Wyodrębnij plik binarny.

```
tar -xf binary-file-name
```

c. Uruchom następujące komendy, aby zmodyfikować i przenieść plik

```
chmod 755 file-name  
mv file-name /usr/local/bin/cloudctl
```

d. Upewnij się, że program `cloudctl` jest zainstalowany:

```
cloudctl --help
```

5. Zainstaluj narzędzie interfejsu wiersza komend oc Red Hat OpenShift Container Platform.

Więcej informacji na ten temat zawiera sekcja [Narzędzia interfejsu wiersza komend Red Hat OpenShift Container Platform](#).

6. Utwórz katalog, który będzie służył jako składnica w trybie bez połączenia.

Poniżej znajduje się przykładowy katalog. Ten przykład jest używany w kolejnych krokach.

```
mkdir $HOME/offline
```

**Uwaga:** Ta składnica w trybie bez połączenia musi być trwała, aby dane nie były przesyłane więcej niż jeden raz. Trwałość ułatwia również jednokrotne lub regularne uruchamianie procesu zapisu lustrzanego.

## Tworzenie zmiennych środowiskowych dla instalatora i spisu obrazów

Utwórz następujące zmienne środowiskowe z nazwą obrazu instalatora i spisem zasobów obrazu:

```
export CASE_ARCHIVE_VERSION=version_number  
export CASE_ARCHIVE=ibm-mq-$CASE_ARCHIVE_VERSION.tgz  
export CASE_INVENTORY=ibmMQoperator
```

gdzie *numer\_wersji* jest wersją przypadku, która ma być używana do wykonania instalacji z separacją. Listę dostępnych wersji przypadków można znaleźć pod adresem <https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq>. Zapoznaj się z sekcją [Obsługa wersji dla produktu IBM MQ Operator](#), aby określić, który kanał operatora wybrać.

## Pobieranie instalatora IBM MQ i spisu obrazów

Pobierz instalator `ibm-mq` i spis obrazów do hosta bastionowego:

```
cloudctl case save \  
  --case https://github.com/IBM/cloud-pak/raw/master/repo/case/ibm-mq/$CASE_ARCHIVE_VERSION/  
  $CASE_ARCHIVE \  
  --outputdir $HOME/offline/
```

## Logowanie się do klastra Red Hat OpenShift Container Platform w roli administratora klastra

Poniżej znajduje się przykładowa komenda zalogowania do klastra Red Hat OpenShift Container Platform:

```
oc login cluster_host:port --username=cluster_admin_user --password=cluster_admin_password
```

## Utworzenie przestrzeni nazw Kubernetes dla operatora IBM MQ Operator

Utwórz zmienną środowiskową z przestrzenią nazw, aby zainstalować produkt IBM MQ Operator, a następnie utwórz przestrzeń nazw:

```
export NAMESPACE=ibm-mq-test
oc create namespace ${NAMESPACE}
```

## Tworzenie kopii lustrzanych obrazów i konfigurowanie klastra

Aby utworzyć kopię lustrzaną obrazów i skonfigurować klaster, wykonaj następujące kroki:

**Uwaga:** W żadnej komendzie nie należy używać znaku tyldy w cudzysłowie. Na przykład nie należy używać wartości args "--registry registry --user registry\_userid --pass registry\_password --inputDir ~/offline". Tylda nie jest rozwijana, a wykonanie komendy może zakończyć się niepowodzeniem.

1. Zapisz referencje uwierzytelniania dla wszystkich źródłowych rejestrów Docker.

Wszystkie obrazy IBM Cloud Platform Common Services, operatora IBM MQ Operator i IBM MQ Advanced Developer są przechowywane w rejestrach publicznych, które nie wymagają uwierzytelniania. Jednak IBM MQ Advanced Server (nie Developer), inne produkty i komponenty innych firm wymagają jednego lub większej liczby rejestrów objętych uwierzytelnianiem. Następujące rejestry wymagają uwierzytelniania:

- cp.icr.io
- registry.redhat.io
- registry.access.redhat.com

Więcej informacji na temat tych rejestrów można znaleźć w sekcji [Tworzenie przestrzeni nazw rejestrów](#).

Aby skonfigurować referencje dla wszystkich rejestrów, które wymagają uwierzytelniania, uruchom następującą komendę. Uruchom komendę osobno dla każdego takiego rejestru:

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action configure-creds-airgap \
--namespace ${NAMESPACE} \
--args "--registry registry --user registry_userid --pass registry_password --inputDir $HOME/offline"
```

Komenda zachowuje i buforuje referencje rejestru w pliku, w katalogu \$HOME/.airgap/secrets w systemie plików.

2. Utwórz zmienne środowiskowe na podstawie lokalnych informacji o połączeniu z rejestrem Docker.

```
export LOCAL_DOCKER_REGISTRY=IP_or_FQDN_of_local_docker_registry
export LOCAL_DOCKER_USER=username
export LOCAL_DOCKER_PASSWORD=password
```

**Uwaga:** Rejestr Docker używa standardowych portów, takich jak 80 i 443. Jeśli w rejestrze Docker używany jest port niestandardowy, należy podać port, używając składni `host:port`. Na przykład:

```
export LOCAL_DOCKER_REGISTRY=myregistry.local:5000
```

3. Skonfiguruj sekret uwierzytelniania dla lokalnego rejestru Docker.

**Uwaga:** Ten krok trzeba wykonać tylko jeden raz.

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action configure-creds-airgap \
--namespace ${NAMESPACE} \
```

```
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass $
{LOCAL_DOCKER_PASSWORD}"
```

Komenda zachowuje i buforuje referencje rejestru w pliku, w katalogu \$HOME/.airgap/secrets w systemie plików.

#### 4. Skonfiguruj globalny sekret do ciągnięcia obrazu i zasób **ImageContentSourcePolicy**.

##### a. Sprawdź, czy wymagany jest restart węzła.

- W produkcie Red Hat OpenShift Container Platform w wersji 4.4 i nowszych oraz w nowej instalacji produktu IBM MQ Operator z użyciem luki powietrznej ten krok powoduje zrestartowanie wszystkich węzłów klastra. Zasoby klastra mogą być niedostępne do momentu zastosowania nowego sekretu pobierania.
- W wersji IBM MQ Operator 1.8 instrukcja CASE jest aktualizowana w celu uwzględnienia dodatkowego źródła zapisu lustrzanego dla obrazów. Dlatego w przypadku aktualizacji z wcześniejszych wersji produktu IBM MQ Operator do wersji 1.8 lub nowszej wyzwany jest restart węzła.
- Aby sprawdzić, czy ten krok wymaga restartu węzła, należy dodać opcję `--dry-run` do kodu tego kroku. Spowoduje to wygenerowanie najnowszego pliku **ImageContentSourcePolicy** i wyświetlenie go w oknie konsoli (**stdout**). Jeśli ta **ImageContentSourcePolicy** różni się od skonfigurowanego klastra **ImageContentSourcePolicy**, następuje restart.

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action configure-cluster-airgap \
--namespace ${NAMESPACE} \
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass $
{LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline --dryRun"
```

##### b. Aby skonfigurować dane niejawne pobierania obrazu globalnego i plik **ImageContentSourcePolicy**, należy uruchomić kod dla tego kroku bez opcji `--dry-run` :

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action configure-cluster-airgap \
--namespace ${NAMESPACE} \
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass $
{LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline"
```

#### 5. Upewnij się, że zasób **ImageContentSourcePolicy** został utworzony.

```
oc get imageContentSourcePolicy
```

#### 6. Opcjonalnie: Jeśli używany jest rejestr niezabezpieczony, należy dodać rejestr lokalny do listy **insecureRegistries** klastra.

```
oc patch image.config.openshift.io/cluster --type=merge -p '{"spec":{"registrySources":
{"insecureRegistries":["${LOCAL_DOCKER_REGISTRY}"]}}'
```

#### 7. Sprawdź status węzła klastra.

```
oc get nodes
```

Po zastosowaniu zasobu **imageContentsourcePolicy** i globalnego sekretu pobierania węzeł może mieć status **Ready**, **Scheduling** lub **Disabled**. Poczekaj, aż wszystkie węzły będą mieć status **Ready**.

#### 8. Utwórz kopię lustrzaną obrazów w rejestrze lokalnym.

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action mirror-images \
--namespace ${NAMESPACE} \
```



```
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass $  
{LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline"
```

## Zainstaluj operator IBM MQ Operator.

1. Zaloguj się do konsoli WWW klastra produktu Red Hat OpenShift.
2. Utwórz źródło katalogu. Użyj tego samego terminala, którego użyto do wykonania poprzednich kroków.

```
cloudctl case launch \  
--case $HOME/offline/${CASE_ARCHIVE} \  
--inventory ${CASE_INVENTORY} \  
--action install-catalog \  
--namespace ${NAMESPACE} \  
--args "--registry ${LOCAL_DOCKER_REGISTRY} --recursive"
```

3. Sprawdź, czy dla operatora Common Services Installer został utworzony zasób **CatalogSource**.

```
oc get pods -n openshift-marketplace  
oc get catalogsource -n openshift-marketplace
```

4. Zainstaluj operator IBM MQ Operator przy użyciu OLM.

- a. Na panelu nawigacyjnym kliknij opcję **Operators > OperatorHub**.

Zostanie wyświetlona strona **OperatorHub**.

- b. W polu **All Items** wprowadź wartość IBM MQ.

Zostanie wyświetlony wpis w katalogu IBM MQ.

- c. Wybierz wpis **IBM MQ**.

Zostanie wyświetlone okno **IBM MQ**.

- d. Kliknij opcję **Install**.

Zostanie wyświetlona strona **Create Operator Subscription**.

- e. Zapoznaj się z sekcją [“Wersja zgodna z operatorem IBM MQ Operator”](#) na stronie 7, aby określić, który kanał operatora ma zostać wybrany.

- f. Ustaw opcję **Installation Mode** na konkretną utworzoną przestrzeń nazw lub zasięg całego klastra.

- g. Kliknij opcję **Subscribe**.

Operator **IBM MQ** zostanie dodany do strony **Installed Operators**.

- h. Sprawdź status operatora na stronie **Installed Operators**. Po zakończeniu instalacji status zmienia się na **Succeeded**.

## Wdrażanie menedżera kolejek IBM MQ

Aby uzyskać instrukcję tworzenia nowego menedżera kolejek w ramach zainstalowanego operatora, patrz [“Wdrażanie i konfigurowanie menedżerów kolejek przy użyciu operatora IBM MQ Operator”](#) na stronie 82.

### Zadania pokrewne

[“Przygotowanie do aktualizacji menedżera kolejek produktu IBM MQ Operator lub menedżera kolejek w środowisku airgap”](#) na stronie 74

W klastrze produktu Red Hat OpenShift, który nie ma połączenia z Internetem, należy wykonać kroki przygotowawcze przed zaktualizowaniem produktu IBM MQ Operator.

## Aktualizowanie operatora IBM MQ Operator i menedżerów kolejek

Aktualizowanie operatora IBM MQ Operator pozwoli na zaktualizowanie menedżerów kolejek.

## Procedura

- [“Aktualizowanie produktu IBM MQ Operator przy użyciu konsoli WWW Red Hat OpenShift” na stronie 77.](#)
- [“Aktualizowanie produktu IBM MQ Operator przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift” na stronie 78.](#)
- [“Aktualizowanie menedżera kolejek produktu IBM MQ przy użyciu konsoli WWW Red Hat OpenShift” na stronie 80.](#)
- [“Wdrażanie menedżera kolejek produktu IBM MQ przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift” na stronie 81.](#)

## Przygotowanie do aktualizacji menedżera kolejek produktu IBM MQ Operator lub menedżera kolejek w środowisku airgap

W klastrze produktu Red Hat OpenShift, który nie ma połączenia z Internetem, należy wykonać kroki przygotowawcze przed zaktualizowaniem produktu IBM MQ Operator.

### Zanim rozpocznie

W tym temacie założono, że skonfigurowano już lokalny rejestr obrazów, w którym są tworzone kopie lustrzane poprzednich wersji obrazów produktu IBM Cloud Pak for Integration.

### O tym zadaniu

Przed zaktualizowaniem produktu IBM MQ Operator lub menedżera kolejek w środowisku z luką powietrzną należy utworzyć kopię lustrzaną najnowszych obrazów produktu IBM Cloud Pak for Integration.

Należy pamiętać, że pierwsze cztery kroki w tym zadaniu są takie same, jak kroki, które należy wykonać, gdy [“Instalowanie operatora IBM MQ Operator w środowisku z separacją” na stronie 67.](#)

## Procedura

1. Utwórz zmienne środowiskowe dla instalatora i magazynu obrazów.

Utwórz następujące zmienne środowiskowe z nazwą obrazu instalatora i spisem zasobów obrazu:

```
export CASE_ARCHIVE_VERSION=version_number
export CASE_ARCHIVE=ibm-mq-$CASE_ARCHIVE_VERSION.tgz
export CASE_INVENTORY=ibmMQoperator
```

gdzie *numer\_wersji* jest wersją przypadku, która ma być używana do wykonania instalacji z separacją. Listę dostępnych wersji przypadków można znaleźć pod adresem <https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq>. Zapoznaj się z sekcją [Obsługa wersji dla produktu IBM MQ Operator](#), aby określić, który kanał operatora wybrać.

2. Pobierz instalator produktu IBM MQ i spis zasobów obrazów.

Pobierz instalator `ibm-mq` i spis obrazów do hosta bastionowego:

```
cloudctl case save \
  --case https://github.com/IBM/cloud-pak/raw/master/repo/case/ibm-mq/
  $CASE_ARCHIVE_VERSION/$CASE_ARCHIVE \
  --outputdir $HOME/offline/
```

3. Zaloguj się do klastra Red Hat OpenShift Container Platform jako administrator klastra.

Poniżej znajduje się przykładowa komenda zalogowania do klastra Red Hat OpenShift Container Platform:

```
oc login cluster_host:port --username=cluster_admin_user --password=cluster_admin_password
```

#### 4. Utwórz kopię lustrzaną obrazów i skonfiguruj klaster.

Aby utworzyć kopię lustrzaną obrazów i skonfigurować klaster, wykonaj następujące kroki:

**Uwaga:** W żadnej komendzie nie należy używać znaku tyldy w cudzysłowie. Na przykład nie należy używać wartości args "--registry registry --user registry\_userid --pass registry\_password --inputDir ~/offline". Tylda nie jest rozwijana, a wykonanie komendy może zakończyć się niepowodzeniem.

##### a. Zapisz referencje uwierzytelniania dla wszystkich źródłowych rejestrów Docker.

Wszystkie obrazy IBM Cloud Platform Common Services, operatora IBM MQ Operator i IBM MQ Advanced Developer są przechowywane w rejestrach publicznych, które nie wymagają uwierzytelniania. Jednak IBM MQ Advanced Server (nie Developer), inne produkty i komponenty innych firm wymagają jednego lub większej liczby rejestrów objętych uwierzytelnianiem. Następujące rejestry wymagają uwierzytelniania:

- cp.icr.io
- registry.redhat.io
- registry.access.redhat.com

Więcej informacji na temat tych rejestrów można znaleźć w sekcji [Tworzenie przestrzeni nazw rejestrów](#).

Aby skonfigurować referencje dla wszystkich rejestrów, które wymagają uwierzytelniania, uruchom następującą komendę. Uruchom komendę osobno dla każdego takiego rejestru:

```
cloudctl case launch \  
--case $HOME/offline/${CASE_ARCHIVE} \  
--inventory ${CASE_INVENTORY} \  
--action configure-creds-airgap \  
--namespace ${NAMESPACE} \  
--args "--registry registry --user registry_userid --pass registry_password --inputDir $HOME/offline"
```

Komenda zachowuje i buforuje referencje rejestru w pliku, w katalogu \$HOME/.airgap/secrets w systemie plików.

##### b. Utwórz zmienne środowiskowe na podstawie lokalnych informacji o połączeniu z rejestrem Docker.

```
export LOCAL_DOCKER_REGISTRY=IP_or_FQDN_of_local_docker_registry  
export LOCAL_DOCKER_USER=username  
export LOCAL_DOCKER_PASSWORD=password
```

**Uwaga:** Rejestr Docker używa standardowych portów, takich jak 80 i 443. Jeśli w rejestrze Docker używany jest port niestandardowy, należy podać port, używając składni `host:port`. Na przykład:

```
export LOCAL_DOCKER_REGISTRY=myregistry.local:5000
```

##### c. Skonfiguruj sekret uwierzytelniania dla lokalnego rejestru Docker.

**Uwaga:** Ten krok trzeba wykonać tylko jeden raz.

```
cloudctl case launch \  
--case $HOME/offline/${CASE_ARCHIVE} \  
--inventory ${CASE_INVENTORY} \  
--action configure-creds-airgap \  
--namespace ${NAMESPACE} \  
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass ${LOCAL_DOCKER_PASSWORD}"
```

Komenda zachowuje i buforuje referencje rejestru w pliku, w katalogu \$HOME/.airgap/secrets w systemie plików.

##### d. Skonfiguruj globalny sekret do ciągnięcia obrazu i zasób **ImageContentSourcePolicy**.

- i) Sprawdź, czy wymagany jest restart węzła.

- W produkcji Red Hat OpenShift Container Platform w wersji 4.4 i nowszych oraz w nowej instalacji produktu IBM MQ Operator z użyciem luki powietrznej ten krok powoduje zrestartowanie wszystkich węzłów klastra. Zasoby klastra mogą być niedostępne do momentu zastosowania nowego sekretu pobierania.
- W wersji IBM MQ Operator 1.8 instrukcja CASE jest aktualizowana w celu uwzględnienia dodatkowego źródła zapisu lustrzanego dla obrazów. Dlatego w przypadku aktualizacji z wcześniejszych wersji produktu IBM MQ Operator do wersji 1.8 lub nowszej wyzwalany jest restart węzła.
- Aby sprawdzić, czy ten krok wymaga restartu węzła, należy dodać opcję `--dry-run` do kodu tego kroku. Spowoduje to wygenerowanie najnowszego pliku **ImageContentSourcePolicy** i wyświetlenie go w oknie konsoli (**stdout**). Jeśli ta **ImageContentSourcePolicy** różni się od skonfigurowanego klastra **ImageContentSourcePolicy**, następuje restart.

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action configure-cluster-airgap \
--namespace ${NAMESPACE} \
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass $
{LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline --dryRun"
```

- ii) Aby skonfigurować dane niejawne pobierania obrazu globalnego i plik **ImageContentSourcePolicy**, należy uruchomić kod dla tego kroku bez opcji `--dry-run` :

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action configure-cluster-airgap \
--namespace ${NAMESPACE} \
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass $
{LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline"
```

- e. Upewnij się, że zasób **ImageContentSourcePolicy** został utworzony.

```
oc get imageContentSourcePolicy
```

- f. Opcjonalnie: Jeśli używany jest rejestr niezabezpieczony, należy dodać rejestr lokalny do listy **insecureRegistries** klastra.

```
oc patch image.config.openshift.io/cluster --type=merge -p '{"spec":{"registrySources":
{"insecureRegistries":["${LOCAL_DOCKER_REGISTRY}"]}}'
```

- g. Sprawdź status węzła klastra.

```
oc get nodes
```

Po zastosowaniu zasobu **imageContentsourcePolicy** i globalnego sekretu pobierania węzeł może mieć status **Ready**, **Scheduling** lub **Disabled**. Poczekaj, aż wszystkie węzły będą mieć status **Ready**.

- h. Utwórz kopię lustrzaną obrazów w rejestrze lokalnym.

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action mirror-images \
--namespace ${NAMESPACE} \
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass $
{LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline"
```

5. Zaktualizuj źródło katalogu.

Użyj tego samego terminala, którego użyto do wykonania poprzednich kroków.

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action install-catalog \
```

```
--namespace ${NAMESPACE} \  
--args "--registry ${LOCAL_DOCKER_REGISTRY} --recursive"
```

## Co dalej

Teraz można przeprowadzić aktualizację produktu IBM MQ Operator i menedżera kolejek, wykonując jedną z następujących czynności:

- [“Aktualizowanie produktu IBM MQ Operator przy użyciu konsoli WWW Red Hat OpenShift” na stronie 77](#)
- [“Aktualizowanie produktu IBM MQ Operator przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift” na stronie 78](#)
- [“Aktualizowanie menedżera kolejek produktu IBM MQ przy użyciu konsoli WWW Red Hat OpenShift” na stronie 80](#)
- [“Wdrażanie menedżera kolejek produktu IBM MQ przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift” na stronie 81](#)
- [“Aktualizowanie menedżera kolejek produktu IBM MQ w rozwiązaniu Red Hat OpenShift przy użyciu Platform Navigator” na stronie 81](#)

## Aktualizowanie produktu IBM MQ Operator przy użyciu konsoli WWW Red Hat OpenShift

Operator IBM MQ Operator można zaktualizować przy użyciu strony Operator Hub.

## Zanim rozpoczniesz

Zaloguj się do konsoli WWW klastra produktu Red Hat OpenShift.

Przed zaktualizowaniem produktu IBM MQ Operator w środowisku z luką powietrzną należy utworzyć kopię lustrzaną najnowszych obrazów IBM Cloud Pak for Integration. Więcej informacji na ten temat zawiera sekcja [Przygotowanie do aktualizacji produktu IBM MQ Operator lub menedżera kolejek w środowisku z luką powietrzną](#).

## Procedura

1. Zapoznaj się z sekcją [“Wersja zgodna z operatorem IBM MQ Operator” na stronie 7](#), aby określić, który kanał operatora wybrać do aktualizacji.
2. Opcjonalne: Jeśli wykonywana jest aktualizacja z wersji IBM MQ Operator starszej niż 1.5 do wersji IBM MQ Operator 1.5 lub nowszej, należy najpierw zaktualizować wersję produktu IBM Cloud Pak foundational services.  
  
Więcej informacji na ten temat zawiera [“Aktualizowanie produktu IBM Cloud Pak foundational services przy użyciu konsoli WWW produktu Red Hat OpenShift” na stronie 78](#).
3. Zaktualizuj IBM MQ Operator. Nowe wersje główne lub podrzędne produktu IBM MQ Operator są dostarczane za pośrednictwem nowych kanałów subskrypcji. Aby zaktualizować element wykonawczy do nowej wersji głównej lub podrzędnej, należy zaktualizować wybrany kanał w subskrypcji produktu IBM MQ Operator.
  - a) Na panelu nawigacyjnym kliknij opcję **Operators > Installed Operators**.  
Zostaną wyświetlone wszystkie operatory zainstalowane w określonym projekcie.
  - b) Wybierz pozycję **IBM MQ Operator**
  - c) Przejdź do karty **Subskrypcja**
  - d) Kliknij opcję **Channel**  
Zostanie wyświetlone okno **Change Subscription Update Channel**.
  - e) Wybierz żądany kanał i kliknij opcję **Save**.

Operator zostanie zaktualizowany do najnowszej wersji dostępnej dla nowego kanału. Patrz sekcja [“Wersja zgodna z operatorem IBM MQ Operator”](#) na stronie 7.

## Co dalej


W przypadku aktualizacji do wersji IBM Cloud Pak foundational services 3.7 wszystkie menedżery kolejek, które korzystają z licencji IBM Cloud Pak for Integration, muszą zostać zaktualizowane lub zrestartowane. Więcej informacji na temat sposobu wykonania tej czynności zawiera sekcja [“Aktualizowanie menedżera kolejek produktu IBM MQ przy użyciu konsoli WWW Red Hat OpenShift”](#) na stronie 80.


### **Aktualizowanie produktu IBM Cloud Pak foundational services przy użyciu konsoli WWW produktu Red Hat OpenShift**

Jeśli wykonywana jest aktualizacja z wersji IBM MQ Operator starszej niż 1.5 do wersji IBM MQ Operator 1.5 lub nowszej, należy najpierw zaktualizować wersję produktu IBM Cloud Pak foundational services.

## Zanim rozpocznie

**Uwaga:** Tę czynność należy wykonać jedynie wtedy, gdy wykonywana jest aktualizacja z wersji IBM MQ Operator starszej niż 1.5 do wersji IBM MQ Operator 1.5 lub nowszej.

 Jeśli istnieją menedżery kolejek, które używają licencji IBM Cloud Pak for Integration, po wykonaniu tej aktualizacji wymagane jest zrestartowanie menedżera kolejek w celu uzyskania dostępu do konsoli WWW. Ponadto zostaną zgłoszone [inne błędy](#) na konsoli WWW. Te błędy można naprawić, aktualizując `.spec.version` do najnowszej wartości dla wybranej wersji IBM MQ, po zakończeniu aktualizacji elementu wykonawczego.

 Jeśli już istnieją menedżery kolejek, a użytkownik korzysta z rozwiązania IBM Cloud Pak for Integration Operations Dashboard, przed aktualizacją należy zapoznać się z [“Wdrażanie lub aktualizowanie IBM MQ 9.2.2 lub 9.2.3 z integracją panelu kontrolnego Operations Dashboard w IBM Cloud Pak for Integration 2021.4”](#) na stronie 115.

## Procedura

1. Zaloguj się do konsoli WWW klastra produktu Red Hat OpenShift.
2. Na panelu nawigacyjnym kliknij opcję **Operators > Installed Operators**.  
Zostaną wyświetlone wszystkie operatory zainstalowane w określonym projekcie.
3. Wybierz pozycję **IBM Cloud Pak foundational services Operator**. Uwaga: w wersjach wcześniejszych niż 3.7 ten operator nosił nazwę **IBM Common Services Operator**
4. Przejdź do karty **Subskrypcja**.
5. Kliknij **Kanał**.  
Zostanie wyświetlone okno **Change Subscription Update Channel**.
6. Wybierz kanał **v3**, a następnie kliknij **Zapisz**.  
Element wykonawczy IBM Cloud Pak foundational services zostanie zaktualizowany do najnowszej wersji dostępnej dla nowego kanału. Patrz [“Wersja zgodna z operatorem IBM MQ Operator”](#) na stronie 7.

## Co dalej

Teraz można przystąpić do [Aktualizowania produktu IBM MQ Operator](#).

### **Aktualizowanie produktu IBM MQ Operator przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift**

Produkt IBM MQ Operator można zaktualizować z poziomu wiersza komend.

## Zanim rozpocznie

Zaloguj się do klastra, używając komendy **cloudctl login** (w przypadku IBM Cloud Pak for Integration) lub **oc login**.

Przed zaktualizowaniem produktu IBM MQ Operator w środowisku z luką powietrzną należy utworzyć kopię lustrzaną najnowszych obrazów IBM Cloud Pak for Integration . Więcej informacji na ten temat zawiera sekcja [Przygotowanie do aktualizacji produktu IBM MQ Operator lub menedżera kolejek w środowisku z luką powietrzną](#).

## Procedura

1. Zapoznaj się z sekcją [“Wersja zgodna z operatorem IBM MQ Operator”](#) na stronie 7, aby określić, który kanał operatora wybrać do aktualizacji.
2. Opcjonalne: Jeśli wykonywana jest aktualizacja z wersji IBM MQ Operator starszej niż 1.5 do wersji IBM MQ Operator 1.5 lub nowszej, należy najpierw zaktualizować wersję produktu IBM Cloud Pak foundational services.

Więcej informacji na ten temat zawiera [“Aktualizowanie produktu IBM Cloud Pak foundational services przy użyciu interfejsu CLI produktu Red Hat OpenShift”](#) na stronie 79.

3. Zaktualizuj IBM MQ Operator. Nowe główne/podrzędne wersje operatora IBM MQ Operator są dostarczane za pośrednictwem nowych kanałów subskrypcji. Aby zaktualizować operator do nowej wersji głównej/podrzędnej, należy zaktualizować wybrany kanał w subskrypcji operatora IBM MQ Operator.

- a) Upewnij się, że dostępny jest wymagany kanał aktualizacji produktu IBM MQ Operator.

```
oc get packagemanifest ibm-mq -o=jsonpath='{.status.channels[*].name}'
```

- b) Zastosuj poprawkę Subscription, aby przejść do żądanego kanału aktualizacji (gdzie vX.Y jest żądanym kanałem aktualizacji określonym w poprzednim kroku.

```
oc patch subscription ibm-mq --patch '{"spec":{"channel":"vX.Y"}}' --type=merge
```

## Co dalej


W przypadku aktualizacji do wersji IBM Cloud Pak foundational services 3.7 wszystkie menedżery kolejek, które korzystają z licencji IBM Cloud Pak for Integration, muszą zostać zaktualizowane lub zrestartowane. Więcej informacji na temat sposobu wykonania tej czynności zawiera sekcja [“Wdrażanie menedżera kolejek produktu IBM MQ przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift”](#) na stronie 81.


## **Aktualizowanie produktu IBM Cloud Pak foundational services przy użyciu interfejsu CLI produktu Red Hat OpenShift**

Jeśli wykonywana jest aktualizacja z wersji IBM MQ Operator starszej niż 1.5 do wersji IBM MQ Operator 1.5 lub nowszej, należy najpierw zaktualizować wersję produktu IBM Cloud Pak foundational services.

## Zanim rozpocznie

**Uwaga:** Tę czynność należy wykonać jedynie wtedy, gdy wykonywana jest aktualizacja z wersji IBM MQ Operator starszej niż 1.5 do wersji IBM MQ Operator 1.5 lub nowszej.

 Jeśli istnieją menedżery kolejek, które używają licencji IBM Cloud Pak for Integration, po wykonaniu tej aktualizacji wymagane jest zrestartowanie menedżera kolejek w celu uzyskania dostępu do konsoli WWW. Ponadto zostaną zgłoszone [inne błędy](#) na konsoli WWW. Te błędy można naprawić, aktualizując `.spec.version` do najnowszej wartości dla wybranej wersji IBM MQ, po zakończeniu aktualizacji elementu wykonawczego.

 Jeśli już istnieją menedżery kolejek, a użytkownik korzysta z rozwiązania IBM Cloud Pak for Integration Operations Dashboard, przed aktualizacją należy zapoznać się z [“Wdrażanie lub](#)

aktualizowanie IBM MQ 9.2.2 lub 9.2.3 z integracją panelu kontrolnego Operations Dashboard w IBM Cloud Pak for Integration 2021.4” na stronie 115.

## Procedura

1. Zaloguj się do klastra, używając komendy **cloudctl login** (w przypadku IBM Cloud Pak for Integration) lub **oc login**.
2. Upewnij się, że kanał aktualizacji v3 IBM Cloud Pak foundational services jest dostępny.

```
oc get packagemanifest -n ibm-common-services ibm-common-service-operator  
-o=jsonpath='{.status.channels[*].name}'
```

3. Zastosuj poprawkę do Subscription, aby przejść do żądanego kanału aktualizacji: v3

```
oc patch subscription ibm-common-service-operator --patch '{"spec":{"channel":"v3"}}' --  
type=merge
```

## Co dalej

Teraz można przystąpić do [Aktualizowania produktu IBM MQ Operator](#).

## Aktualizowanie menedżera kolejek produktu IBM MQ przy użyciu konsoli WWW Red Hat OpenShift

Menedżer kolejek produktu IBM MQ wdrożony za pomocą rozwiązania IBM MQ Operator może zostać zaktualizowany w produkcie Red Hat OpenShift przy użyciu rozwiązania Operator Hub.

## Zanim rozpocznie

- Zaloguj się do konsoli WWW klastra produktu Red Hat OpenShift.
- Upewnij się, że IBM MQ Operator używa żądanego kanału aktualizacji. Patrz [“Aktualizowanie operatora IBM MQ Operator i menedżerów kolejek”](#) na stronie 73.

Przed zaktualizowaniem menedżera kolejek w środowisku z luką powietrzną należy utworzyć kopię lustrzaną najnowszych obrazów IBM Cloud Pak for Integration . Więcej informacji na ten temat zawiera sekcja [Przygotowanie do aktualizacji produktu IBM MQ Operator lub menedżera kolejek w środowisku z luką powietrzną](#).

## Procedura

1. Na panelu nawigacyjnym kliknij opcję **Operators > Installed Operators**.  
Zostaną wyświetlone wszystkie operatory zainstalowane w określonym projekcie.
2. Wybierz pozycję **IBM MQ Operator**.  
Zostanie wyświetlone okno **IBM MQ Operator**.
3. Przejdź do karty **Menedżer kolejek**.  
Zostanie wyświetlone okno **Szczegóły menedżera kolejek**.
4. Wybierz menedżera kolejek, który ma zostać zaktualizowany.
5. Przejdź do karty YAML.
6. W razie potrzeby zmień następujące pola, aby były zgodne z żądaną aktualizacją wersji menedżera kolejek IBM MQ.
  - spec.version
  - spec.license.licence

Informacje na temat odwzorowania kanałów na wersje [“Wersja zgodna z operatorem IBM MQ Operator”](#) na stronie 7 i wersje menedżera kolejek IBM MQ Operator można znaleźć w sekcji IBM MQ.



7. Zapisz zmieniony kod YAML menedżera kolejek.

## **Wdrażanie menedżera kolejek produktu IBM MQ przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift**

Menedżer kolejek produktu IBM MQ wdrożony za pomocą rozwiązania IBM MQ Operator może zostać zaktualizowany w produkcie Red Hat OpenShift przy użyciu wiersza komend.

### Zanim rozpoczniesz

Aby wykonać te kroki, należy być administratorem klastra.

- Zaloguj się do interfejsu wiersza komend produktu Red Hat OpenShift przy użyciu programu `oc login`.
- Upewnij się, że IBM MQ Operator używa żądanego kanału aktualizacji. Patrz [“Aktualizowanie operatora IBM MQ Operator i menedżerów kolejek” na stronie 73](#).

Przed zaktualizowaniem menedżera kolejek w środowisku z luką powietrzną należy utworzyć kopię lustrzaną najnowszych obrazów IBM Cloud Pak for Integration . Więcej informacji na ten temat zawiera sekcja [Przygotowanie do aktualizacji produktu IBM MQ Operator lub menedżera kolejek w środowisku z luką powietrzną](#).

### Procedura

Zmodyfikuj zasób **QueueManager**, w razie potrzeby zmieniając następujące pola tak, aby były zgodne z żądaną aktualizacją wersji menedżera kolejek IBM MQ.

- `spec.version`
- `spec.license.licence`

Informacje na temat odwzorowania kanałów na wersje [“Wersja zgodna z operatorem IBM MQ Operator” na stronie 7](#) i wersje menedżera kolejek IBM MQ Operator można znaleźć w sekcji IBM MQ.

Użyj następującej komendy:

```
oc edit queuemanager my_qmgr
```

gdzie `mój_qmgr` to nazwa zasobu QueueManager, który ma zostać zaktualizowany.

## **Aktualizowanie menedżera kolejek produktu IBM MQ w rozwiązaniu Red Hat OpenShift przy użyciu Platform Navigator**

Menedżer kolejek produktu IBM MQ wdrożony za pomocą rozwiązania IBM MQ Operator może zostać zaktualizowany w produkcie Red Hat OpenShift przy użyciu rozwiązania IBM Cloud Pak for Integration Platform Navigator.

### Zanim rozpoczniesz

- Zaloguj się do programu IBM Cloud Pak for Integration Platform Navigator w przestrzeni nazw zawierającej menedżer kolejek, który ma zostać zaktualizowany.
- Upewnij się, że IBM MQ Operator używa żądanego kanału aktualizacji. Patrz [“Aktualizowanie operatora IBM MQ Operator i menedżerów kolejek” na stronie 73](#).

Przed zaktualizowaniem menedżera kolejek w środowisku z luką powietrzną należy utworzyć kopię lustrzaną najnowszych obrazów IBM Cloud Pak for Integration . Więcej informacji na ten temat zawiera sekcja [Przygotowanie do aktualizacji produktu IBM MQ Operator lub menedżera kolejek w środowisku z luką powietrzną](#).

### Procedura

1. Na stronie głównej programu IBM Cloud Pak for Integration Platform Navigator kliknij kartę **Runtimes**.

2. Menedżery kolejek z dostępnymi aktualizacjami są oznaczone niebieskimi ikonami **i** obok pozycji **Version**. Kliknij ikonę **i**, aby wyświetlić opcję **New version available**.
3. Kliknij trzy kropki po prawej stronie menedżera kolejek, który ma zostać zaktualizowany, a następnie kliknij opcję **Change version**.
4. W sekcji **Select a new channel or version** wybierz wymaganą wersję aktualizacji.
5. Kliknij opcję **Change version**.

## Wyniki

Menedżer kolejek zostanie zaktualizowany do nowej wersji.

## **Wdrażanie i konfigurowanie menedżerów kolejek przy użyciu operatora IBM MQ Operator**

IBM MQ 9.1.5 i nowsze wersje są wdrażane w produkcie Red Hat OpenShift przy użyciu IBM MQ Operator.

### O tym zadaniu

#### Procedura

- [“Przygotowywanie projektu Red Hat OpenShift dla produktu IBM MQ” na stronie 82.](#)
- [“Wdrażanie menedżera kolejek w klastrze platformy kontenerowej Red Hat OpenShift” na stronie 84.](#)

## **Przygotowywanie projektu Red Hat OpenShift dla produktu IBM MQ**

Przygotuj klaster platformy Red Hat OpenShift Container Platform, aby był gotowy do wdrożenia menedżera kolejek.

#### Procedura

- [“Przygotowywanie projektu Red Hat OpenShift dla produktu IBM MQ przy użyciu konsoli WWW Red Hat OpenShift” na stronie 82.](#)
- [“Przygotowywanie projektu produktu Red Hat OpenShift dla produktu IBM MQ przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift” na stronie 83.](#)

#### Zadania pokrewne

[“Wdrażanie menedżera kolejek w klastrze platformy kontenerowej Red Hat OpenShift” na stronie 84](#)  
Użyj zasobu niestandardowego QueueManager, aby wdrożyć menedżer kolejek na klastrze platformy Red Hat OpenShift Container Platform.

## **Przygotowywanie projektu Red Hat OpenShift dla produktu IBM MQ przy użyciu konsoli WWW Red Hat OpenShift**

Przygotuj klaster platformy Red Hat OpenShift Container Platform, aby był gotowy do wdrożenia menedżera kolejek za pomocą operatora IBM MQ Operator. To zadanie powinien wykonać administrator projektu.

### Zanim rozpoczniesz

**Uwaga:** Jeśli produkt IBM MQ ma być używany w projekcie z innymi komponentami IBM Cloud Pak for Integration, które są już zainstalowane, wykonanie tych instrukcji może okazać się zbędne.

Zaloguj się do konsoli WWW klastra produktu Red Hat OpenShift.

## O tym zadaniu

Obrazy IBM MQ Operator są pobierane z rejestru kontenerów, który sprawdza uprawnienia licencyjne. To sprawdzenie wymaga klucza uprawnień, który jest przechowywany w secrecie pobierania docker-registry. Jeśli klucz uprawnień nie jest jeszcze dostępny, należy wykonać poniższe instrukcje, aby uzyskać klucz i utworzyć sekret pobierania.

## Procedura

1. Uzyskaj klucz uprawnień przypisany do swojego identyfikatora.
  - a) Zaloguj się do biblioteki [MyIBM Container Software Library](#) przy użyciu identyfikatora IBM ID i hasła, które są powiązane z uprawnieniem do oprogramowania.
  - b) W sekcji **Entitlement keys** wybierz opcję **Copy key**, aby skopiować klucz uprawnień do schowka.
2. Utwórz sekret zawierający klucz uprawnień w projekcie, w którym ma zostać wdrożony menedżer kolejek.
  - a) Na panelu nawigacyjnym kliknij opcję **Workloads > Secret**.  
Zostanie wyświetlona strona Secrets.
  - b) W menu rozwijanym **Project** wybierz projekt, w którym ma zostać zainstalowany produkt IBM MQ
  - c) Kliknij przycisk **Create** i wybierz opcję **Image Pull Secret**
  - d) W polu **Name** wprowadź `ibm-entitlement-key`.
  - e) W polu **Registry Server Address** wprowadź `cp.icr.io`
  - f) W polu **Username** wprowadź `cp`
  - g) W polu **Password** wprowadź klucz uprawnień skopiowany w poprzednim kroku
  - h) W polu **Email** wprowadź IBM ID skojarzony z uprawnieniem do oprogramowania

## Co dalej

[“Wdrażanie menedżera kolejek przy użyciu konsoli WWW Red Hat OpenShift” na stronie 85](#)

## **Przygotowywanie projektu produktu Red Hat OpenShift dla produktu IBM MQ przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift**

Przygotuj klaster platformy Red Hat OpenShift Container Platform, aby był gotowy do wdrożenia menedżera kolejek za pomocą operatora IBM MQ Operator. To zadanie powinien wykonać administrator projektu.

## Zanim rozpoczniesz

**Uwaga:** Jeśli produkt IBM MQ ma być używany w projekcie z innymi komponentami IBM Cloud Pak for Integration, które są już zainstalowane, wykonanie tych instrukcji może okazać się zbędne.

Zaloguj się do klastra, używając komendy **cloudctl login** (w przypadku IBM Cloud Pak for Integration) lub **oc login**.

## O tym zadaniu

Obrazy IBM MQ Operator są pobierane z rejestru kontenerów, który sprawdza uprawnienia licencyjne. To sprawdzenie wymaga klucza uprawnień, który jest przechowywany w secrecie pobierania docker-registry. Jeśli klucz uprawnień nie jest jeszcze dostępny, należy wykonać poniższe instrukcje, aby uzyskać klucz i utworzyć sekret pobierania.

## Procedura

1. Uzyskaj klucz uprawnień przypisany do swojego identyfikatora.
  - a) Zaloguj się do biblioteki [MyIBM Container Software Library](#) przy użyciu identyfikatora IBM ID i hasła, które są powiązane z uprawnieniem do oprogramowania.

- b) W sekcji **Entitlement keys** wybierz opcję **Copy key**, aby skopiować klucz uprawnień do schowka.
2. Utwórz sekret zawierający klucz uprawnień w projekcie, w którym ma zostać wdrożony menedżer kolejek.

Uruchom następującą komendę, gdzie `<entitlement-key>` jest kluczem pobranym w kroku 1, a `<user-email>` jest identyfikatorem IBM powiązany z przystępującym oprogramowaniem.

```
oc create secret docker-registry ibm-entitlement-key \
--docker-server=cp.icr.io \
--docker-username=cp \
--docker-password=<entitlement-key> \
--docker-email=<user-email>
```


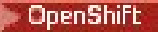

## Co dalej

[“Wdrażanie menedżera kolejek przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift” na stronie 86](#)

## **Wdrażanie menedżera kolejek w klastrze platformy kontenerowej Red Hat OpenShift**

Użyj zasobu niestandardowego QueueManager, aby wdrożyć menedżer kolejek na klastrze platformy Red Hat OpenShift Container Platform.

### Procedura

-  [“Wdrażanie menedżera kolejek przy użyciu produktu IBM Cloud Pak for Integration Platform Navigator” na stronie 84.](#)
-  [“Wdrażanie menedżera kolejek przy użyciu konsoli WWW Red Hat OpenShift” na stronie 85.](#)
-  [“Wdrażanie menedżera kolejek przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift” na stronie 86.](#)

### Zadania pokrewne

[“Przykłady konfigurowania menedżera kolejek” na stronie 88](#)

Menedżer kolejek można skonfigurować, modyfikując zawartość niestandardowego zasobu QueueManager.

## **Wdrażanie menedżera kolejek przy użyciu produktu IBM Cloud Pak for Integration Platform Navigator**

Użyj zasobu niestandardowego QueueManager, aby wdrożyć menedżer kolejek na klastrze platformy Red Hat OpenShift Container Platform za pomocą produktu IBM Cloud Pak for Integration Platform Navigator. To zadanie powinien wykonać administrator projektu.

### Zanim rozpocznie

W przeglądarce uruchom IBM Cloud Pak for Integration Platform Navigator.

Jeśli jest to pierwsze wdrożenie menedżera kolejek w tym projekcie Red Hat OpenShift, należy wykonać kroki opisane w sekcji [“Przygotowywanie projektu Red Hat OpenShift dla produktu IBM MQ” na stronie 82.](#)

### Procedura

1. Wdróż menedżera kolejek.

W poniższym przykładzie wdrażany jest menedżer kolejek „quick start”, który używa pamięci masowej efemerycznej (nietrwalej) i wyłącza zabezpieczenia produktu MQ. Komunikaty nie będą utrwalane między restartami menedżera kolejek. Można zmienić konfigurację, aby zmodyfikować różne ustawienia menedżera kolejek.

- a) W rozwiązywaniu IBM Cloud Pak for Integration Platform Navigator kliknij opcję **Administration**, a następnie **Integration Runtimes**. W starszych wersjach IBM Cloud Pak for Integration Platform Navigator kliknij opcję **Runtime and instances**.
- b) Kliknij opcję **Create instance**.
- c) Wybierz opcję **Messaging**, a następnie kliknij przycisk **Next**. W starszych wersjach IBM Cloud Pak for Integration Platform Navigator kliknij opcję **Queue Manager**, a następnie kliknij przycisk **Next**. Zostanie wyświetlony formularz do tworzenia instancji obiektu QueueManager.

**Uwaga:** Można również kliknąć opcję **Code**, aby wyświetlić lub zmienić kod YAML konfiguracji zasobu QueueManager.

- d) W sekcji **Details** sprawdź lub zaktualizuj pole **Name**, a następnie w polu **Namespace** określ przestrzeń nazw, w której ma zostać utworzona instancja menedżera kolejek.
- e) Jeśli akceptujesz umowę licencyjną na produkt IBM Cloud Pak for Integration, zmień wartość opcji **License acceptance** na **On**.  
Aby wdrożyć menedżer kolejek, należy zaakceptować licencję.
- f) W sekcji **Queue Manager** sprawdź lub zaktualizuj w polu **Name** nazwę bazowego menedżera kolejek. W starszych wersjach IBM Cloud Pak for Integration Platform Navigator użyj sekcji **Queue Manager Config**.  
Domyślnie nazwa menedżera kolejek używanego przez aplikacje klienckie IBM MQ będzie taka sama, jak nazwa zasobu QueueManager, ale z usuniętymi znakami niedozwolonymi (takimi jak łączniki).
- g) Kliknij opcję **Create**.  
Zostanie teraz wyświetlona lista menedżerów kolejek w bieżącym projekcie (przestrzeni nazw). Nowy zasób QueueManager powinien mieć status Pending.

## 2. Sprawdzanie, czy menedżer kolejek jest uruchomiony

Status Running zasobu QueueManager oznacza, że tworzenie zostało zakończone.

### Zadania pokrewne

[“Konfigurowanie trasy w celu nawiązania połączenia z menedżerem kolejek spoza klastra produktu Red Hat OpenShift” na stronie 112](#)

Do połączenia aplikacji z menedżerem kolejek IBM MQ spoza klastra Red Hat OpenShift potrzebna jest trasa Red Hat OpenShift. Należy włączyć protokół TLS w menedżerze kolejek i aplikacji klienckiej IBM MQ, ponieważ protokół SNI jest dostępny w protokole TLS tylko wtedy, gdy używany jest protokół TLS w wersji 1.2 lub nowszej. Red Hat OpenShift Container Platform Router używa wskaźnika SNI do kierowania żądań do menedżera kolejek IBM MQ.

[“Nawiązywanie połączenia z rozwiązaniem IBM MQ Console wdrożonym w klastrze Red Hat OpenShift” na stronie 118](#)

W jaki sposób nawiązać połączenie z konsolą IBM MQ Console menedżera kolejek, która została wdrożona w klastrze Red Hat OpenShift Container Platform.

### **Wdrażanie menedżera kolejek przy użyciu konsoli WWW Red Hat OpenShift**

Użyj zasobu niestandardowego QueueManager, aby wdrożyć menedżer kolejek na platformie kontenerowej Red Hat OpenShift Container Platform, używając konsoli WWW Red Hat OpenShift. To zadanie powinien wykonać administrator projektu.

### Zanim rozpocznieš

Zaloguj się do konsoli WWW klastra produktu Red Hat OpenShift. Trzeba będzie wybrać istniejący projekt (przestrzeń nazw), który ma być używany, lub utworzyć nowy projekt.

Jeśli jest to pierwsze wdrożenie menedżera kolejek w tym projekcie Red Hat OpenShift, należy wykonać kroki opisane w sekcji [“Przygotowywanie projektu Red Hat OpenShift dla produktu IBM MQ”](#) na stronie 82.

## Procedura

### 1. Wdróż menedżera kolejek.

W poniższym przykładzie wdrażany jest menedżer kolejek „quick start”, który używa pamięci masowej efemerycznej (nietrwałej) i wyłącza zabezpieczenia produktu MQ. Komunikaty nie będą utrwalane między restartami menedżera kolejek. Można zmienić konfigurację, aby zmodyfikować różne ustawienia menedżera kolejek.

a) W konsoli WWW Red Hat OpenShift na panelu nawigacyjnym kliknij opcję **Operatory** > **Zainstalowane operatory**

b) Kliknij pozycję **IBM MQ**.

c) Kliknij kartę **Queue Manager**.

d) Kliknij przycisk **Create QueueManager**.

Zostanie wyświetlony edytor YAML zawierający przykładowy kod YAML zasobu QueueManager.

**Uwaga:** Można również kliknąć opcję **Edit form**, aby wyświetlić lub zmienić konfigurację zasobu QueueManager.

e) Jeśli akceptujesz umowę licencyjną, zmień wartość opcji **License acceptance** na **On**.

Produkt IBM MQ jest dostępny z kilkoma różnymi licencjami. Aby uzyskać więcej informacji na temat poprawnych licencji, patrz [“Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1”](#) na stronie 130. Aby wdrożyć menedżer kolejek, należy zaakceptować licencję.

f) Kliknij opcję **Create**.

Zostanie teraz wyświetlona lista menedżerów kolejek w bieżącym projekcie (przestrzeni nazw). Nowy zasób QueueManager powinien być w stanie Pending.

### 2. Sprawdzanie, czy menedżer kolejek jest uruchomiony

Status Running zasobu QueueManager oznacza, że tworzenie zostało zakończone.

## Zadania pokrewne

[“Konfigurowanie trasy w celu nawiązania połączenia z menedżerem kolejek spoza klastra produktu Red Hat OpenShift”](#) na stronie 112

Do połączenia aplikacji z menedżerem kolejek IBM MQ spoza klastra Red Hat OpenShift potrzebna jest trasa Red Hat OpenShift. Należy włączyć protokół TLS w menedżerze kolejek i aplikacji klienckiej IBM MQ, ponieważ protokół SNI jest dostępny w protokole TLS tylko wtedy, gdy używany jest protokół TLS w wersji 1.2 lub nowszej. Red Hat OpenShift Container Platform Router używa wskaźnika SNI do kierowania żądań do menedżera kolejek IBM MQ.

[“Nawiązywanie połączenia z rozwiązaniem IBM MQ Console wdrożonym w klastrze Red Hat OpenShift”](#) na stronie 118

W jaki sposób nawiązać połączenie z konsolą IBM MQ Console menedżera kolejek, która została wdrożona w klastrze Red Hat OpenShift Container Platform.

## **Wdrażanie menedżera kolejek przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift**

Użyj zasobu niestandardowego QueueManager, aby wdrożyć menedżer kolejek na klastrze platformy Red Hat OpenShift Container Platform za pomocą interfejsu wiersza komend. To zadanie powinien wykonać administrator projektu.

## Zanim rozpocznie

Konieczne będzie zainstalowanie [interfejsu wiersza komend Red Hat OpenShift Container Platform](#).

Zaloguj się do klastra, używając komendy **cloudctl login** (w przypadku IBM Cloud Pak for Integration) lub **oc login**.

Jeśli jest to pierwsze wdrożenie menedżera kolejek w tym projekcie Red Hat OpenShift, należy wykonać kroki opisane w sekcji [“Przygotowywanie projektu Red Hat OpenShift dla produktu IBM MQ”](#) na stronie 82.

## Procedura

### 1. Wdróż menedżera kolejek.

W poniższym przykładzie wdrażany jest menedżer kolejek „quick start”, który używa pamięci masowej efemerycznej (nietrwalej) i wyłącza zabezpieczenia produktu MQ. Komunikaty nie będą utrwalane między restartami menedżera kolejek. Można zmienić treść kodu YAML, aby modyfikować różne ustawienia menedżera kolejek.

#### a) Utwórz plik YAML QueueManager

Na przykład, aby zainstalować prosty menedżer kolejek w środowisku IBM Cloud Pak for Integration, należy utworzyć plik mq-quickstart.yaml o następującej treści:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
spec:
  version: 9.2.5.0-r3
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: NonProduction
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
    storage:
      queueManager:
        type: ephemeral
  template:
    pod:
      containers:
        - name: qmgr
          env:
            - name: MQSNOAUT
              value: "yes"
```

**Ważne:** Jeśli akceptujesz umowę licencyjną produktu IBM Cloud Pak for Integration, zmień wartość `accept: false` na `accept: true`. Aby uzyskać szczegółowe informacje na temat licencji, patrz [“Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1”](#) na stronie 130.

Ten przykład obejmuje również serwer WWW wdrożony wraz z menedżerem kolejek, przy czym konsola WWW obsługuje pojedyncze logowanie przy wykorzystaniu rozwiązania IBM Cloud Pak Identity and Access Manager.

Aby zainstalować prosty menedżer kolejek niezależnie od IBM Cloud Pak for Integration, należy utworzyć plik mq-quickstart.yaml o następującej treści:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart
spec:
  version: 9.2.5.0-r3
  license:
    accept: false
    license: L-APIG-BZDDDY
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
    storage:
      queueManager:
        type: ephemeral
```

```
template:
  pod:
    containers:
      - name: qmgr
        env:
          - name: MQSNOAUT
            value: "yes"
```

**Ważne:** Jeśli akceptujesz umowę licencyjną MQ, zmień `accept: false` na wartość `accept: true`. Aby uzyskać szczegółowe informacje na temat licencji, patrz [“Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1”](#) na stronie 130.

b) Utwórz obiekt `QueueManager`

```
oc apply -f mq-quickstart.yaml
```

2. Sprawdzanie, czy menedżer kolejek jest uruchomiony

Poprawność wdrożenia można sprawdzić, uruchamiając komendę

```
oc describe queuemanager <QueueManagerResourceName>
```

i sprawdzając status.

Na przykład uruchom komendę

```
oc describe queuemanager quickstart
```

, a następnie sprawdź, czy pole `status.Phase` wskazuje `Running`

## Zadania pokrewne

[“Konfigurowanie trasy w celu nawiązania połączenia z menedżerem kolejek spoza klastra produktu Red Hat OpenShift”](#) na stronie 112

Do połączenia aplikacji z menedżerem kolejek IBM MQ spoza klastra Red Hat OpenShift potrzebna jest trasa Red Hat OpenShift. Należy włączyć protokół TLS w menedżerze kolejek i aplikacji klienckiej IBM MQ, ponieważ protokół SNI jest dostępny w protokole TLS tylko wtedy, gdy używany jest protokół TLS w wersji 1.2 lub nowszej. Red Hat OpenShift Container Platform Router używa wskaźnika SNI do kierowania żądań do menedżera kolejek IBM MQ.

[“Nawiązywanie połączenia z rozwiązaniem IBM MQ Console wdrożonym w klastrze Red Hat OpenShift”](#) na stronie 118

W jaki sposób nawiązać połączenie z konsolą IBM MQ Console menedżera kolejek, która została wdrożona w klastrze Red Hat OpenShift Container Platform.

## Przykłady konfigurowania menedżera kolejek

Menedżer kolejek można skonfigurować, modyfikując zawartość niestandardowego zasobu `QueueManager`.

## O tym zadaniu

Poniższe przykłady ilustrują konfigurowanie menedżera kolejek przy użyciu pliku YAML zasobu `QueueManager`.

## Procedura

- [“Przykład: Udostępnianie plików MQSC i INI”](#) na stronie 88
- [“Przykład: Konfigurowanie protokołu TLS”](#) na stronie 90

## Przykład: Udostępnianie plików MQSC i INI

Ten przykład tworzy mapę Kubernetes ConfigMap, która zawiera dwa pliki MQSC i jeden plik INI. Następnie wdrażany jest menedżer kolejek, który przetwarza te pliki MQSC i INI.



## O tym zadaniu

Pliki `MQSC` i `INI` mogą być udostępniane na etapie wdrażania menedżera kolejek. Dane `MQSC` i `INI` muszą być zdefiniowane w jednym lub wielu zasobach Kubernetes `ConfigMap` i `Secret`. Te zasoby muszą zostać utworzone w przestrzeni nazw (projekcie), w której zostanie wdrożony menedżer kolejek.

**Uwaga:** Gdy plik `MQSC` lub `INI` zawiera dane poufne, należy stosować klucz tajny Kubernetes.

Do udostępnienia plików `MQSC` i `INI` w opisywany sposób wymagana jest wersja IBM MQ Operator 1.1 lub nowsza.

## Przykład

Poniższy przykład tworzy mapę Kubernetes `ConfigMap`, która zawiera dwa pliki `MQSC` i jeden plik `INI`. Następnie wdrażany jest menedżer kolejek, który przetwarza te pliki `MQSC` i `INI`.

Przykładowa mapa `ConfigMap` — zastosuj następujący kod `YAML` w klastrze:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: mqsc-ini-example
data:
  example1.mqsc: |
    DEFINE QLOCAL('DEV.QUEUE.1') REPLACE
    DEFINE QLOCAL('DEV.QUEUE.2') REPLACE
  example2.mqsc: |
    DEFINE QLOCAL('DEV.DEAD.LETTER.QUEUE') REPLACE
  example.ini: |
    Channels:
      MQIBindType=FASTPATH
```

Przykładowy menedżer kolejek — wdróż menedżer kolejek za pomocą następującej konfiguracji, korzystając z wiersza komend lub przy użyciu programu IBM Cloud Pak for Integration Platform Navigator:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mqsc-ini-cp4i
spec:
  version: 9.2.5.0-r3
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: NonProduction
  web:
    enabled: true
  queueManager:
    name: "MQSCINI"
    mqsc:
      - configMap:
          name: mqsc-ini-example
          items:
            - example1.mqsc
            - example2.mqsc
    ini:
      - configMap:
          name: mqsc-ini-example
          items:
            - example.ini
  storage:
    queueManager:
      type: ephemeral
```

**Ważne:** Jeśli akceptujesz umowę licencyjną produktu IBM Cloud Pak for Integration, zmień wartość `accept: false` na `accept: true`. Szczegółowe informacje na temat licencji można znaleźć w sekcji [Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1](#).

Informacje dodatkowe:

- Menedżer kolejek może być skonfigurowany tak, aby używał jednego zasobu Kubernetes `ConfigMap` lub `Secret` w środowisku (tak jak pokazano to w tym przykładzie) lub wielu zasobów Kubernetes `ConfigMap` i `Secret`.




- Można wybrać użycie wszystkich danych MQSC i INI z zasobu Kubernetes ConfigMap lub Secret (tak jak w tym przykładzie) lub skonfigurować każdego menedżera kolejek tak, aby używał tylko podzbioru dostępnych plików.
- Pliki MQSC i INI są przetwarzane w porządku alfabetycznym na podstawie ich klucza. Dlatego `example1.mqsc` będzie zawsze przetwarzany przed `example2.mqsc` niezależnie od kolejności ich występowania w konfiguracji menedżera kolejek.
- Jeśli więcej niż jeden plik MQSC lub INI ma ten sam klucz w wielu zasobach Kubernetes ConfigMap lub Secret, ten zestaw plików jest przetwarzany na podstawie kolejności, w jakiej pliki są zdefiniowane w konfiguracji menedżera kolejek.

### **Przykład: Konfigurowanie protokołu TLS**

W tym przykładzie menedżer kolejek jest wdrażany w Red Hat OpenShift Container Platform przy użyciu IBM MQ Operator. Konfigurowana jest jednokierunkowa komunikacja TLS między przykładowym klientem a menedżerem kolejek. Przykład potwierdza poprawność konfiguracji poprzez umieszczanie i pobieranie komunikatów.

## Zanim rozpoczniesz

Aby zrealizować ten przykład, należy najpierw spełnić następujące wymagania wstępne:

- Zainstaluj produkt IBM MQ client, a następnie dodaj `samp/bin` i `bin` do elementu `PATH`. Potrzebne będą aplikacje **runmqakm**, **amqsputc** i **amqsgetc**, które można zainstalować razem z komponentem IBM MQ client w następujący sposób:
  -   W przypadku Windows i Linux: zainstaluj klienta IBM MQ (podlegającego redystrybucji) dla używanego systemu operacyjnego z serwisu <https://ibm.biz/mq92redistclients>
  -  W systemie Mac: Pobierz i skonfiguruj IBM MQ MacOS Toolkit: <https://developer.ibm.com/tutorials/mq-macos-dev/>
- Zainstaluj narzędzie OpenSSL dla używanego systemu operacyjnego.
- Utwórz projekt/przestrzeń nazw Red Hat OpenShift Container Platform (OCP) dla tego przykładu.
- W wierszu komend zaloguj się do klastra OCP i przełącz się na powyższą przestrzeń nazw.
- Upewnij się, że operator IBM MQ Operator jest zainstalowany i dostępny w powyższej przestrzeni nazw.

## O tym zadaniu

W tym przykładzie udostępniono niestandardowy zasób YAML definiujący menedżer kolejek, który ma zostać wdrożony w środowisku Red Hat OpenShift Container Platform. Opisano tu również szczegółowe kroki wymagane do wdrożenia menedżera kolejek z włączonym protokołem TLS. Po zakończeniu możliwość umieszczania i pobierania komunikatów potwierdza poprawność menedżera kolejek, który jest skonfigurowany przy użyciu protokołu TLS.

### Tworzenie klucza prywatnego TLS i certyfikatów dla serwera IBM MQ

Poniższe przykłady kodu przedstawiają sposób tworzenia samopodpisanego certyfikatu dla menedżera kolejek oraz sposób dodawania certyfikatu do bazy danych kluczy, która będzie pełnić rolę magazynu zaufanych certyfikatów dla klienta. Jeśli masz już klucz prywatny i certyfikat, możesz ich użyć zamiast tworzyć nowe.

Należy pamiętać, że samopodpisane certyfikaty powinny być używane tylko w celach programistycznych.

### Utwórz samopodpisany klucz prywatny i certyfikat publiczny w bieżącym katalogu

Uruchom następującą komendę:

```
openssl req -newkey rsa:2048 -nodes -keyout tls.key -subj "/CN=localhost" -x509 -days 3650 -out tls.crt
```

## Dodanie klucza publicznego serwera do bazy danych kluczy klienta

Baza danych kluczy jest używana jako magazyn zaufanych certyfikatów dla aplikacji klienckiej.

Utwórz bazę danych kluczy klienta:

```
runmqakm -keydb -create -db clientkey.kdb -pw password -type cms -stash
```

Dodaj wcześniej wygenerowany klucz publiczny do bazy danych kluczy klienta:

```
runmqakm -cert -add -db clientkey.kdb -label mqservercert -file tls.crt -format ascii -stashed
```

## Konfigurowanie certyfikatów TLS na potrzeby wdrażania menedżera kolejek

Aby menedżer kolejek mógł odwoływać się do klucza i certyfikatu, należy utworzyć klucz tajny TLS Kubernetes, który odwołuje się do plików utworzonych powyżej. W tym celu należy upewnić się, że znajdują się one w przestrzeni nazw, która została utworzona przed rozpoczęciem tego zadania.

```
oc create secret tls example-tls-secret --key="tls.key" --cert="tls.crt"
```

## Utworzenie mapy konfiguracji zawierającej komendy MQSC

Utwórz mapę konfiguracji Kubernetes zawierającą komendy MQSC, aby utworzyć nową kolejkę i kanał SVRCONN, a także dodaj rekord uwierzytelniania kanału, który umożliwi dostęp do kanału, blokując tylko użytkowników o nazwie *nobody*.

Należy zauważyć, że to podejście powinno być używane tylko w celach programistycznych.

Upewnij się, że jesteś w przestrzeni nazw, która została wcześniej utworzona (patrz sekcja [Zanim rozpoczniesz](#)), a następnie wprowadź następującą komendę YAML w interfejsie użytkownika OCP lub za pomocą wiersza komend.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-tls-configmap
data:
  tls.mqsc: |
    DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
    DEFINE CHANNEL(SECUREQMCHL) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCAUTH(OPTIONAL)
    SSLCIPH('ANY_TLS12_OR_HIGHER')
    SET CHLAUTH(SECUREQMCHL) TYPE(BLOCKUSER) USERLIST('nobody') ACTION(ADD)
```

## Utworzenie wymaganej trasy OCP

Upewnij się, że jesteś w przestrzeni nazw, która została utworzona przed rozpoczęciem tego zadania, a następnie wprowadź następującą komendę YAML w interfejsie OCP lub za pomocą wiersza komend.

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: example-tls-route
spec:
  host: secureqmchl.chl.mq.ibm.com
  to:
    kind: Service
    name: secureqm-ibm-mq
  port:
    targetPort: 1414
  tls:
    termination: passthrough
```

Należy zauważyć, że Red Hat OpenShift Container Platform Router używa SNI do kierowania żądań do menedżera kolejek IBM MQ. Jeśli zostanie zmieniona nazwa kanału określona w danych MQSC w utworzonej wcześniej mapie konfiguracji, należy zmienić pole hosta w tym miejscu i w tworzonym później pliku CCDT. Więcej informacji na ten temat zawiera sekcja [“Konfigurowanie trasy w celu nawiązania połączenia z menedżerem kolejek spoza klastra produktu Red Hat OpenShift”](#) na stronie [112](#).

## Wdrażanie menedżera kolejek

**Ważne:** W tym przykładzie używana jest zmienna `MQSNOAUT` w celu wyłączenia autoryzacji w menedżerze kolejek, co pozwala skupić się na krokach wymaganych do połączenia klienta przy użyciu protokołu TLS. Nie jest to zalecane w przypadku produkcyjnego wdrożenia IBM MQ, ponieważ powoduje, że wszystkie aplikacje nawiązują połączenie z pełnymi uprawnieniami administracyjnymi, bez mechanizmu obniżania uprawnień poszczególnych aplikacji.

Utwórz nowy menedżer kolejek przy użyciu następującego zasobu niestandardowego YAML. Należy zwrócić uwagę, że odwołuje się on do mapy konfiguracji i klucza tajnego utworzonych wcześniej, a także do zmiennej `MQSNOAUT`.

Upewnij się, że jesteś w przestrzeni nazw, która została utworzona przed rozpoczęciem tego zadania, a następnie wprowadź następujący kod YAML w interfejsie użytkownika OCP lub za pomocą wiersza komend IBM Cloud Pak for Integration Platform Navigator. Sprawdź, czy określono poprawną licencję, a następnie zaakceptuj licencję, zmieniając `false` na wartość `true`.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: secureqm
spec:
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: Production
  queueManager:
    name: SECUREQM
  mqsc:
    - configMap:
        name: example-tls-configmap
        items:
          - tls.mqsc
    storage:
      queueManager:
        type: ephemeral
  template:
    pod:
      containers:
        - env:
            - name: MQSNOAUT
              value: 'yes'
          name: qmgr
  version: 9.2.5.0-r3
  web:
    enabled: true
  pki:
    keys:
      - name: example
        secret:
          secretName: example-tls-secret
          items:
            - tls.key
            - tls.crt
```

## Upewnij się, że menedżer kolejek jest uruchomiony

Menedżer kolejek jest obecnie wdrażany. Przed kontynuowaniem upewnij się, że znajduje się on w stanie `Running`. Na przykład:

```
oc get qmgr secureqm
```

## Testowanie połączenia z menedżerem kolejek

Aby potwierdzić, że menedżer kolejek został skonfigurowany do jednokierunkowej komunikacji TLS, należy użyć przykładowych aplikacji `amqsputc` i `amqsgetc`:

## Znajdowanie nazwy hosta menedżera kolejek

Użyj poniższej komendy, aby znaleźć w pełni kwalifikowaną nazwę hosta menedżera kolejek dla ścieżki `secureqm-ibm-mq-qm`:

```
oc get routes secureqm-ibm-mq-qm
```

## Określenie danych menedżera kolejek

Utwórz plik CCDT .JSON, który określa szczegóły menedżera kolejek. Zastąp wartość host nazwą hosta z poprzedniego kroku.

```
{
  "channel":
  [
    {
      "name": "SECUREQMCHL",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "<hostname from previous step>",
            "port": 443
          }
        ],
        "queueManager": "SECUREQM"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "ECDHE_RSA_AES_128_CBC_SHA256"
      },
      "type": "clientConnection"
    }
  ]
}
```

## Eksportowanie zmiennych środowiskowych

Wyeksportuj następujące zmienne środowiskowe w sposób odpowiedni dla używanego systemu operacyjnego. Te zmienne będą odczytywane przez aplikacje **amqsputc** i **amqsgetc**.

Zaktualizuj ścieżkę do plików w systemie:

```
export MQCCDTURL='<full path to file>/CCDT.JSON'
export MQSSLKEYR='<full path to file>/clientkey'
```

## Wstawianie komunikatów do kolejki

Uruchom następującą komendę:

```
amqsputc EXAMPLE.QUEUE SECUREQM
```

Jeśli nawiązanie połączenia z menedżerem kolejek powiedzie się, zostanie wygenerowana następująca odpowiedź:

```
target queue is EXAMPLE.QUEUE
```

Umieść kilka komunikatów w kolejce, wprowadzając tekst i naciskając za każdym razem klawisz **Enter**.

Aby zakończyć, naciśnij klawisz **Enter** dwa razy.

## Pobieranie komunikatów z kolejki

Uruchom następującą komendę:

```
amqsgetc EXAMPLE.QUEUE SECUREQM
```

Komunikaty dodane w poprzednim kroku zostaną skonsumowane i wyświetlone.

Po kilku sekundach komenda kończy działanie..

Na tym etapie można pogratulować sobie pomyślnego wdrożenia menedżera kolejek z włączoną obsługą protokołu TLS. Potwierdziliśmy, że klient może bezpiecznie umieścić i pobrać komunikaty do/z menedżera kolejek.

OpenShift

CP4I

## Przykład: dostosowywanie adnotacji usługi licencji

Produkt IBM MQ Operator automatycznie dodaje adnotacje produktu IBM License Service do wdrożonych zasobów. Są one monitorowane przez produkt IBM License Service. Generowane są raporty, które odpowiadają wymaganemu uprawnieniu.

## O tym zadaniu

Adnotacje dodane przez produkt IBM MQ Operator są tymi, które są oczekiwane w sytuacjach standardowych. Są one oparte na wartościach licencji wybranych podczas wdrażania menedżera kolejek.

### Przykład

Jeśli produkt **License** jest ustawiony na wartość L-RJON-BZFQU2 (IBM Cloud Pak for Integration 2021.2.1), a produkt **Use** jest ustawiony na wartość NonProduction, stosowane są następujące adnotacje:

- cloudpakId: c8b82d189e7545f0892db9ef2731b90d
- cloudpakName: IBM Cloud Pak for Integration
- productChargedContainers: qmgr
- productCloudpakRatio: '4:1'
- productID: 21dfe9a0f00f444f888756d835334909
- productName: IBM MQ Advanced for Non-Production
- productMetric: VIRTUAL\_PROCESSOR\_CORE
- productVersion: 9.2.3.0

W ramach produktu IBM Cloud Pak for Integration wdrożenia produktu IBM App Connect Enterprise obejmują ograniczone uprawnienia dla produktu IBM MQ. W takich sytuacjach adnotacje te muszą zostać przesłonięte, aby zapewnić przechwytywanie przez IBM License Service poprawnego użycia. Aby to zrobić, należy użyć podejścia opisanego w sekcji [“Dodawanie niestandardowych adnotacji i etykiet do zasobów menedżera kolejek”](#) na stronie 117.

Jeśli na przykład produkt IBM MQ zostanie wdrożony w ramach uprawnienia IBM App Connect Enterprise, należy użyć metody przedstawionej w następującym fragmencie kodu:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mq4ace
  namespace: cp4i
spec:
  annotations:
    productMetric: FREE
```

Istnieją dwa inne typowe powody, dla których adnotacje licencji mogą wymagać modyfikacji:

1. IBM MQ Advanced jest uwzględniany w uprawnieniu innego produktu IBM.
  - W takiej sytuacji należy użyć metody opisanej wcześniej dla produktu IBM App Connect Enterprise.
2. Produkt IBM MQ jest wdrażany na licencji IBM Cloud Pak for Integration.
  - Jeśli użytkownik posiada licencję na produkt IBM Cloud Pak for Integration, może podjąć decyzję o wdrożeniu menedżera kolejek w ramach współczynnika IBM MQ lub IBM MQ Advanced. W przypadku wdrożenia w oparciu o współczynnik IBM MQ należy się upewnić, że nie są używane zaawansowane możliwości, takie jak rodzima wysoka dostępność lub Advanced Message Security.
  - W tej sytuacji należy użyć następujących adnotacji do użycia w środowisku produkcyjnym:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mq4ace
  namespace: cp4i
spec:
  annotations:
    productID: c661609261d5471fb4ff8970a36bccea
    productCloudpakRatio: '4:1'
    productName: IBM MQ for Production
    productMetric: VIRTUAL_PROCESSOR_CORE
```

- W przypadku środowiska innego niż produkcyjne należy używać następujących adnotacji:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mq4ace
  namespace: cp4i
spec:
  annotations:
    productID: 151bec68564a4a47a14e6fa99266deff
    productCloudpakRatio: '8:1'
    productName: IBM MQ for Non-Production
    productMetric: VIRTUAL_PROCESSOR_CORE
```

## OpenShift > CP4I Konfigurowanie wysokiej dostępności na potrzeby menedżerów kolejek przy użyciu rozwiązania IBM MQ Operator

### O tym zadaniu

#### Procedura

- **V 9.2.3**  
“Rodzima wysoka dostępność (Native HA)” na stronie 95.
- **V 9.2.3**  
“Przykład: Konfigurowanie menedżera kolejek o rodzimej wysokiej dostępności” na stronie 97.
- “Przykład: konfigurowanie wieloinstancyjnego menedżera kolejek” na stronie 106.

#### **V 9.2.3 > CP4I > CD Rodzima wysoka dostępność (Native HA)**

Rodzima wysoka dostępność (Native HA) jest rodzimym (wbudowanym) rozwiązaniem zapewniającym wysoką dostępność środowisk IBM MQ, odpowiednim do użycia z blokową pamięcią masową w chmurze.

Konfiguracja rodzimej wysokiej dostępności obejmuje wysoce dostępny menedżer kolejek, w którym odtwarzalne dane produktu MQ (na przykład komunikaty) są replikowane w wielu zestawach pamięci masowej, co zapobiega utracie danych w wyniku awarii pamięci masowej. Menedżer kolejek składa się z wielu działających instancji — jedna jest liderem, a pozostałe są gotowe do szybkiego przejęcia obciążenia w przypadku awarii. Takie rozwiązanie maksymalizuje dostępność menedżera kolejek i jego komunikatów.

Konfiguracja rodzimej wysokiej dostępności składa się z trzech zasobników Kubernetes, z których każdy jest instancją menedżera kolejek. Jedna instancja jest aktywnym menedżerem kolejek, przetwarza komunikaty i tworzy wpisy w dzienniku odtwarzania. Za każdym razem, gdy tworzony jest wpis w dzienniku odtwarzania, aktywny menedżer kolejek wysyła dane do pozostałych dwóch instancji, zwanych replikami. Każda replika tworzy wpisy we własnym dzienniku odtwarzania, potwierdza dane, a następnie aktualizuje własne dane kolejki ze zreplikowanego dziennika odtwarzania. Jeśli w zasobniku aktywnego menedżera kolejek dojdzie do awarii, jedna z replik menedżera kolejek przejmuje rolę aktywną i od tej chwili zawiera bieżące dane do pracy.

Typ dziennika jest znany jako "replikowany dziennik". Replikowany dziennik jest w zasadzie dziennikiem liniowym, z włączonym automatycznym zarządzaniem dziennikami i automatycznymi obrazami nośników. Patrz [Typy rejestrowania](#). Te same techniki zarządzania replikowanym dziennikiem są używane do zarządzania dziennikiem liniowym.

Usługa Kubernetes jest używana do kierowania połączeń klientów TCP/IP do bieżącej aktywnej instancji, która jest identyfikowana jako jedyny zasobnik gotowy do obsługi ruchu sieciowego. Aplikacja kliencka nie musi wiedzieć o istnieniu różnych instancji.

Użycie trzech zasobników znacznie zmniejsza ryzyko rozsynchronizowania instancji. W systemie wysokiej dostępności z dwoma zasobnikami rozsynchronizowanie może wystąpić w razie przerwania połączenia między tymi zasobnikami. W przypadku braku połączenia w obu zasobnikach mogą jednocześnie działać menedżery kolejek zbierające różne dane. Po przywróceniu połączenia dostępne są dwie różne wersje

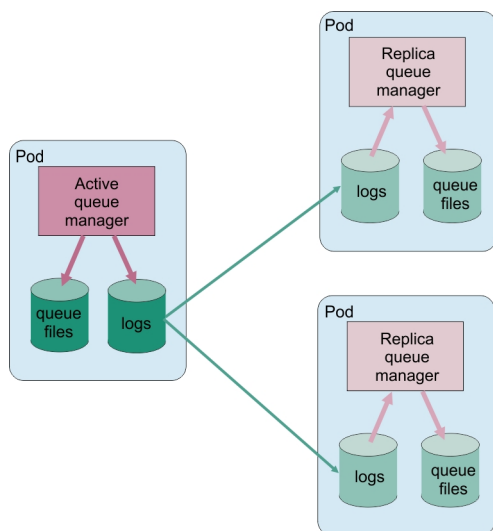
danych i wymagana jest ręczna interwencja w celu podjęcia decyzji, który zestaw danych ma być zachowany, a który ma zostać odrzucony.

Mechanizm rodzimej wysokiej dostępności trzech zasobników i wymaga istnienia kworum, aby uniknąć rozsynchronizowania. Zasobniki, które mogą komunikować się z co najmniej jednym z pozostałych zasobników, tworzą kworum. Menedżer kolejek może stać się aktywną instancją tylko w zasobniku, który ma kworum. Menedżer kolejek nie może być aktywny w zasobniku, który nie ma połączenia z co najmniej jednym innym zasobnikiem, więc nigdy nie mogą istnieć jednocześnie dwie aktywne instancje:

- Jeśli jeden zasobnik ulegnie awarii, menedżer kolejek na jednym z pozostałych dwóch zasobników może przejąć jego funkcję. Jeśli dwa zasobniki ulegną awarii, menedżer kolejek nie może stać się aktywną instancją w pozostałym zasobniku, ponieważ nie ma kworum (ten zasobnik nie może stwierdzić, czy pozostałe dwa zasobniki są w stanie awarii, czy są nadal uruchomione, ale utraciły połączenie).
- Jeśli jeden zasobnik utraci połączenie, menedżer kolejek nie może stać się aktywny w tym zasobniku, ponieważ nie ma kworum. Menedżer kolejek na jednym z pozostałych dwóch zasobników może przejąć aktywność, ponieważ te zasobniki mają kworum. Jeśli wszystkie zasobniki utracą połączenie, menedżer kolejek nie może być aktywny na żadnym z zasobników, ponieważ żaden z zasobników nie ma kworum.

Jeśli aktywny zasobnik ulegnie awarii, a następnie odzyska sprawność, może ponownie dołączyć do grupy w roli repliki.

Na poniższym rysunku przedstawiono typowe wdrożenie z trzema instancjami menedżera kolejek wdrożonymi w trzech kontenerach.



Rysunek 1. Przykład konfiguracji rodzimej wysokiej dostępności

**V9.2.3** **CP4I** **CD** Konfigurowanie rodzimej wysokiej dostępności przy użyciu operatora IBM MQ Operator

Rodzima wysoka dostępność jest konfigurowana przy użyciu interfejsu API QueueManager, a opcje zaawansowane są dostępne za pośrednictwem pliku INI.

Rodzima wysoka dostępność jest konfigurowana przy użyciu `.spec.queueManager.availability` interfejsu API QueueManager, na przykład:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: nativeha-example
spec:
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: Production
  queueManager:
    availability:
```



```
type: NativeHA
version: 9.2.5.0-r3
```

Pole `.spec.queueManager.availability.type` musi być ustawione na wartość `NativeHA`.

Rodzima wysoka dostępność jest dostępna w wersji IBM MQ 9.2.3 i nowszych.

W `.spec.queueManager.availability` można również skonfigurować klucz sekretny i szyfry TLS do wykorzystania między instancjami menedżera kolejek podczas replikacji. Jest to zalecane, a instrukcja krok po kroku jest dostępna w sekcji [“Przykład: Konfigurowanie menedżera kolejek o rodzimej wysokiej dostępności”](#) na stronie 97.

### Odsyłacze pokrewne

[“Przykład: Konfigurowanie menedżera kolejek o rodzimej wysokiej dostępności”](#) na stronie 97

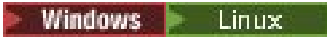
W tym przykładzie przedstawiono sposób wdrażania menedżera kolejek przy użyciu rodzimej funkcji wysokiej dostępności w produkcie Red Hat OpenShift Container Platform (OCP) przy użyciu operatora IBM MQ Operator.

 [Przykład: Konfigurowanie menedżera kolejek o rodzimej wysokiej dostępności](#)

W tym przykładzie przedstawiono sposób wdrażania menedżera kolejek przy użyciu rodzimej funkcji wysokiej dostępności w produkcie Red Hat OpenShift Container Platform (OCP) przy użyciu operatora IBM MQ Operator.

### Zanim zaczniesz

Aby zrealizować ten przykład, należy najpierw spełnić następujące wymagania wstępne:

- Zainstaluj produkt IBM MQ client i dodaj zainstalowane katalogi `samp/bin` i `bin` do zmiennej środowiskowej `PATH`. Klient udostępnia aplikacje `runmqakm`, `amqsputc` i `amqsgetc` wymagane w tym przykładzie. Zainstaluj IBM MQ client w następujący sposób:
  -  W przypadku Windows i Linux: zainstaluj klienta IBM MQ (podlegającego redystrybucji) dla używanego systemu operacyjnego z serwisu <https://ibm.biz/mq92redistclients>
  -  W systemie Mac: pobierz i skonfiguruj IBM MQ MacOS Toolkit. Patrz sekcja <https://ibm.biz/mqdevmacclient>.
- Zainstaluj narzędzie OpenSSL dla używanego systemu operacyjnego. W tym celu należy wygenerować samopodpisany certyfikat dla menedżera kolejek, jeśli nie masz jeszcze klucza prywatnego i certyfikatu.
- Utwórz projekt/przestrzeń nazw Red Hat OpenShift Container Platform (OCP) dla tego przykładu i wykonaj kroki opisane w zadaniu [“Przygotowywanie projektu Red Hat OpenShift dla produktu IBM MQ”](#) na stronie 82
- W wierszu komend zaloguj się w klastrze OCP i przetłącz się na przestrzeń nazw, która została właśnie utworzona.
- Upewnij się, że operator IBM MQ Operator jest zainstalowany i dostępny w przestrzeni nazw.
- W OCP skonfiguruj domyślną klasę pamięci masowej, która ma być używana przez menedżer kolejek. Aby ukończyć ten kurs bez ustawiania domyślnej klasy pamięci masowej, patrz [Uwaga 2: Używanie klasy pamięci masowej innej niż domyślna](#).

### Informacje o zadaniu

Rodzime menedżery kolejek o wysokiej dostępności działają na jednym aktywnym zasobniku Kubernetes i dwóch zasobnikach będących replikami. Działają one jako część zestawu stanowego Kubernetes z dokładnie trzema replikami i zestawem trwałych woluminów Kubernetes. Więcej informacji na temat menedżerów kolejek o rodzimej wysokiej dostępności zawiera sekcja [“Wysoka dostępność produktu IBM MQ w kontenerach”](#) na stronie 16.

Przykład obejmuje kod YAML zasobu niestandardowego definiujący menedżer kolejek o rodzimej wysokiej dostępności korzystający z trwałej pamięci masowej i protokołu TLS. Po wdrożeniu

menedżera kolejek w OCP zasymulujemy awarię aktywnego menedżera kolejek. Po awarii będzie można zaobserwować automatyczne odtwarzanie i potwierdzić jego powodzenie poprzez umieszczanie i pobieranie komunikatów.

## Przykład

### Tworzenie klucza prywatnego TLS i certyfikatów dla serwera MQ

Można utworzyć samopodpisany certyfikat dla menedżera kolejek oraz dodać certyfikat do bazy danych kluczy, która będzie pełnić rolę magazynu zaufanych certyfikatów dla klienta. Jeśli masz już klucz prywatny i certyfikat, możesz ich użyć zamiast tworzyć nowe. Należy pamiętać, że certyfikatów samopodpisanych powinno się używać tylko na etapie prac programistycznych.

Aby utworzyć samopodpisany klucz prywatny i certyfikat publiczny w bieżącym katalogu, uruchom następującą komendę:

```
openssl req -newkey rsa:2048 -nodes -keyout tls.key -subj "/CN=localhost" -x509 -days 3650 -out tls.crt
```

### Tworzenie klucza prywatnego TLS i certyfikatów do użytku wewnętrznego przez rodzimą wysoką dostępność

Trzy zasobniki w menedżerze kolejek o rodzimej wysokiej dostępności replikują dane w sieci. Można utworzyć certyfikat samopodpisany, który ma być używany podczas wewnętrznego replikowania. Należy pamiętać, że certyfikatów samopodpisanych powinno się używać tylko na etapie prac programistycznych.

Aby utworzyć samopodpisany klucz prywatny i certyfikat publiczny w bieżącym katalogu, uruchom następującą komendę:

```
openssl req -newkey rsa:2048 -nodes -keyout nativeha.key -subj "/CN=localhost" -x509 -days 3650 -out nativeha.crt
```

### Dodanie klucza publicznego menedżera kolejek do bazy danych kluczy klienta

Baza danych kluczy klienta jest używana jako magazyn zaufanych certyfikatów dla aplikacji klienckiej.

Utwórz bazę danych kluczy klienta:

```
runmqakm -keydb -create -db clientkey.kdb -pw password -type cms -stash
```

Dodaj wcześniej wygenerowany klucz publiczny do bazy danych kluczy klienta:

```
runmqakm -cert -add -db clientkey.kdb -label mqservercert -file tls.crt -format ascii -stashed
```

### Tworzenie klucza tajnego zawierającego certyfikaty TLS dla wdrożenia menedżera kolejek

Aby menedżer kolejek mógł odwoływać się do klucza i certyfikatu, należy utworzyć klucz tajny Kubernetes TLS, który odwołuje się do plików utworzonych powyżej. W tym celu należy upewnić się, że znajdują się one w przestrzeni nazw, która została utworzona przed rozpoczęciem tego zadania.

```
oc create secret tls example-ha-secret --key="tls.key" --cert="tls.crt"
```

### Tworzenie klucza tajnego zawierającego wewnętrzny certyfikat TLS i klucz dla rodzimej wysokiej dostępności

Aby menedżer kolejek mógł odwoływać się do klucza i certyfikatu, należy utworzyć klucz tajny Kubernetes TLS, który odwołuje się do plików utworzonych powyżej. W tym celu należy upewnić się, że znajdują się one w przestrzeni nazw, która została utworzona przed rozpoczęciem tego zadania.

```
oc create secret tls example-ha-secret-internal --key="nativeha.key" --cert="nativeha.crt"
```

### Utworzenie mapy konfiguracji zawierającej komendy MQSC

Utwórz mapę konfiguracji Kubernetes zawierającą komendy MQSC, aby utworzyć nową kolejkę i kanał SVRCONN, a także dodaj rekord uwierzytelniania kanału, który umożliwia dostęp do kanału, blokując tylko użytkowników o nazwie *nobody*.

Należy zauważyć, że to podejście powinno być używane tylko w celach programistycznych.

Upewnij się, że jesteś w przestrzeni nazw, która została wcześniej utworzona (patrz sekcja [“Zanim zaczniesz”](#) na stronie 97), a następnie wprowadź następującą komendę YAML w interfejsie użytkownika OCP lub za pomocą wiersza komend:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-mi-configmap
data:
  tls.mqsc: |
    DEFINE QLOCAL('EXAMPLE.QUEUE') DEFPSIST(YES) REPLACE
    DEFINE CHANNEL(HAQMCHL) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCAUTH(OPTIONAL)
    SSLCIPH('ANY_TLS12_OR_HIGHER')
    SET CHLAUTH(HAQMCHL) TYPE(BLOCKUSER) USERLIST('nobody') ACTION(ADD)
```

### Konfigurowanie kierowania

Jeśli używany jest IBM MQ client lub pakiet narzędzi w wersji IBM MQ 9.2.1 lub nowszej, można skonfigurować kierowanie do menedżera kolejek przy użyciu pliku konfiguracyjnego menedżera kolejek (pliku INI). W pliku należy ustawić zmienną *OutboundSNI* na trasę opartą na nazwie hosta, a nie na nazwie kanału.

Utwórz plik w katalogu, w którym są uruchomione komendy o nazwie `mqclient.ini`, zawierający dokładnie następujący tekst:

```
SSL:
  OutboundSNI=HOSTNAME
```

Nie należy zmieniać żadnych wartości w tym pliku INI. Na przykład łańcuch `HOSTNAME` nie może być ulec zmianie.

Więcej informacji na ten temat zawiera temat [Sekcja SSL pliku konfiguracyjnego klienta](#).

W przypadku użycia IBM MQ client lub pakietu narzędzi wcześniejszego niż IBM MQ 9.2.1 należy utworzyć trasę OCP zamiast poprzedniego pliku konfiguracyjnego. Wykonaj kroki opisane w sekcji [Uwaga 1: Tworzenie trasy](#).

### Wdrażanie menedżera kolejek

**Ważne:** W tym przykładzie używana jest zmienna *MQSNOAUT* w celu wyłączenia autoryzacji w menedżerze kolejek, co pozwala skupić się na krokach wymaganych do połączenia klienta przy użyciu protokołu TLS. Nie jest to zalecane w przypadku produkcyjnego wdrożenia IBM MQ, ponieważ powoduje, że wszystkie aplikacje nawiązują połączenie z pełnymi uprawnieniami administracyjnymi, bez mechanizmu obniżania uprawnień poszczególnych aplikacji.

Skopiuj i zaktualizuj następujący kod YAML.

- Upewnij się, że określono poprawną licencję. Patrz [Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1](#). W produkcji IBM Cloud Pak for Integration 2021.1.1 licencja musi być licencją na próbne użytkowanie L-RJON-BYRMYW
- Zaakceptuj licencję, zmieniając wartość `false` na `true`.

Kod YAML zasobu niestandardowego menedżera kolejek:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: nativeha-example
spec:
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: Production
  queueManager:
    name: HAEXAMPLE
    availability:
      type: NativeHA
    tls:
      secretName: example-ha-secret-internal
  mqsc:
  - configMap:
```

```

      name: example-mi-configmap
      items:
      - tls.mqsc
template:
  pod:
    containers:
    - env:
      - name: MQSNOAUT
        value: 'yes'
      name: qmgr
version: 9.2.5.0-r3
pki:
  keys:
  - name: example
  secret:
    secretName: example-ha-secret
    items:
    - tls.key
    - tls.crt

```

Będąc w przestrzeni nazw utworzonej wcześniej, należy wdrożyć zaktualizowany kod YAML przy użyciu konsoli WWW Red Hat OpenShift Container Platform, wiersza komend lub przy użyciu IBM Cloud Pak for Integration Platform Navigator.

Po krótkim opóźnieniu, w trakcie którego system konfiguruje menedżer kolejek o rodzimej wysokiej dostępności, menedżer powinien być dostępny do użycia.

## Sprawdzanie poprawności

W tej sekcji sprawdzamy, czy menedżer kolejek zachowuje się zgodnie z oczekiwaniami.

### Upewnij się, że menedżer kolejek jest uruchomiony

Menedżer kolejek jest obecnie wdrażany. Przed kontynuowaniem upewnij się, że znajduje się on w stanie Running. Na przykład:

```
oc get qmgr nativeha-example
```

### Testowanie połączenia z menedżerem kolejek

Aby potwierdzić, że menedżer kolejek został skonfigurowany do jednokierunkowej komunikacji TLS, należy użyć przykładowych aplikacji **amqspu** i **amqsgetc**:

### Znajdowanie nazwy hosta menedżera kolejek

Aby znaleźć nazwę hosta menedżera kolejek dla ścieżki `nativeha-example-ibm-mq-qm`, należy uruchomić następującą komendę. Nazwa hosta jest zwracana w polu `HOST`.

```
oc get routes nativeha-example-ibm-mq-qm
```

### Określenie danych menedżera kolejek

Utwórz plik CCDT .JSON, który określa szczegóły menedżera kolejek. Zastąp wartość `host` nazwą hosta z poprzedniego kroku.

```

{
  "channel":
  [
    {
      "name": "HAQMCHL",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "<host from previous step>",
            "port": 443
          }
        ],
        "queueManager": "HAEXAMPLE"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "ECDHE_RSA_AES_128_CBC_SHA256"
      },
      "type": "clientConnection"
    }
  ]
}

```

```
}  
  ]  
}
```

### Eksportowanie zmiennych środowiskowych

Wyeksportuj następujące zmienne środowiskowe w sposób odpowiedni dla używanego systemu operacyjnego. Te zmienne będą odczytywane przez aplikacje **amqspu** i **amqsgetc**.

Zaktualizuj ścieżkę do plików w systemie:

```
export MQCCDTURL='<full_path_to_file>/CCDT.JSON'  
export MQSSLKEYR='<full_path_to_file>/clientkey'
```

### Wstawianie komunikatów do kolejki

Uruchom następującą komendę:

```
amqspu EXAMPLE.QUEUE HAEXAMPLE
```

Jeśli nawiązanie połączenia z menedżerem kolejek powiedzie się, zostanie wygenerowana następująca odpowiedź:

```
target queue is EXAMPLE.QUEUE
```

Umieść kilka komunikatów w kolejce, wprowadzając tekst i naciskając za każdym razem klawisz **Enter**.

Aby zakończyć, naciśnij klawisz **Enter** dwa razy.

### Pobieranie komunikatów z kolejki

Uruchom następującą komendę:

```
amqsgetc EXAMPLE.QUEUE HAEXAMPLE
```

Komunikaty dodane w poprzednim kroku zostaną skonsumowane i wyświetlone.

Po kilku sekundach komenda kończy działanie..

### Wymuszanie awarii aktywnego zasobnika

Aby sprawdzić poprawność automatycznego odtwarzania menedżera kolejek, zasymuluj awarię zasobnika:

#### Wyświetlanie aktywnych i rezerwowych zasobników

Uruchom następującą komendę:

```
oc get pods --selector app.kubernetes.io/instance=nativeha-example
```

Uwaga: w polu **READY** aktywny zasobnik zwraca wartość 1/1, natomiast zasobniki repliki zwracają wartość 0/1.

#### Usuwanie aktywnego zasobnika

Uruchom następującą komendę, podając pełną nazwę aktywnego zasobnika:

```
oc delete pod nativeha-example-ibm-mq-<value>
```

#### Wyświetl ponownie status zasobnika

Uruchom następującą komendę:

```
oc get pods --selector app.kubernetes.io/instance=nativeha-example
```

#### Wyświetlanie statusu menedżera kolejek

Uruchom następującą komendę, określając pełną nazwę jednego z pozostałych zasobników:

```
oc exec -t Pod -- dspmq -o nativeha -x -m HAEXAMPLE
```

Powinien zostać wyświetlony status, który wskazuje, że aktywna instancja została zmieniona, na przykład:

```
QMNAME(HAEXAMPLE) ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)  
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
```

```
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

### Ponowne umieszczanie i pobieranie komunikatów

Gdy zasobnik rezerwowy stanie się aktywny (czyli po tym, jak wartość pola READY zmieni się na 1/1), należy ponownie użyć następujących komend, jak opisano powyżej, w celu umieszczenia komunikatów w menedżerze kolejek, a następnie pobrać komunikaty z menedżera kolejek:

```
amqsputc EXAMPLE.QUEUE HAEXAMPLE
```

```
amqsgetc EXAMPLE.QUEUE HAEXAMPLE
```

W ten sposób pomyślnie wdrożyliśmy menedżer kolejek z rodzimą wysoką dostępnością i wykazaliśmy, że może on automatycznie odzyskać sprawność w przypadku awarii zasobnika.

## Dodatkowe informacje

### Uwaga 1: Tworzenie trasy

Jeśli używany jest pakiet IBM MQ client lub pakiet narzędzi wcześniejszy niż IBM MQ 9.2.1, należy utworzyć trasę.

Aby utworzyć trasę, upewnij się, że jesteś w przestrzeni nazw, która została wcześniej utworzona (patrz sekcja “Zanim zaczniesz” na stronie 97), a następnie wprowadź następujący kod YAML w konsoli WWW Red Hat OpenShift Container Platform lub za pomocą wiersza komend:

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: example-mi-route
spec:
  host: hamqchl.chl.mq.ibm.com
  to:
    kind: Service
    name: nativeha-example-ibm-mq
  port:
    targetPort: 1414
  tls:
    termination: passthrough
```

Należy zauważyć, że Red Hat OpenShift Container Platform Router używa SNI do kierowania żądań do menedżera kolejek IBM MQ. Jeśli zostanie zmieniona nazwa kanału określona w [odwzorowaniu konfiguracji zawierającym komendy MQSC](#), należy również zmienić pole hosta w tym miejscu i w pliku CCDT .JSON, który określa szczegóły menedżera kolejek. Więcej informacji na ten temat zawiera sekcja “Konfigurowanie trasy w celu nawiązania połączenia z menedżerem kolejek spoza klastra produktu Red Hat OpenShift” na stronie 112.

### Uwaga 2: Używanie klasy pamięci masowej innej niż domyślna


W tym przykładzie oczekuje się, że domyślna klasa pamięci masowej została skonfigurowana w Red Hat OpenShift Container Platform, dlatego w kodzie YAML zasobu niestandardowego menedżera kolejek nie jest wymagana żadna informacja o pamięci masowej. Jeśli klasa pamięci masowej nie jest skonfigurowana jako domyślna lub ma być używana inna klasa pamięci masowej, należy dodać `defaultClass: <storage_class_name>` w `spec.queueManager.storage`.

Nazwa klasy pamięci masowej musi być dokładnie zgodna z nazwą klasy pamięci masowej, która już istnieje. Oznacza to, że musi być ona zgodna z nazwą zwróconej przez komendę `oc get storageclass`. Musi również obsługiwać produkt `ReadWriteMany`. Więcej informacji na ten temat zawiera sekcja “[Uwagi dotyczące pamięci masowej dla IBM MQ Operator](#)” na stronie 11.

### Zadania pokrewne

[“Wyświetlanie statusu rodzimych menedżerów kolejek HA dla kontenerów certyfikowanych IBM MQ” na stronie 103](#)

W przypadku kontenerów certyfikowanych przez produkt IBM MQ można wyświetlić status rodzimych instancji HA, uruchamiając komendę **dspmq** w jednym z działających Pods.

 Wyświetlanie statusu rodzimych menedżerów kolejek HA dla kontenerów certyfikowanych IBM MQ

W przypadku kontenerów certyfikowanych przez produkt IBM MQ można wyświetlić status rodzimych instancji HA, uruchamiając komendę **dspmq** w jednym z działających Pods.

## O tym zadaniu

### Ważne:

Aby wyświetlić status działania instancji menedżera kolejek, można użyć komendy **dspmq** w jednym z uruchomionych zasobników. Zwrócone informacje zależą od tego, czy instancja jest aktywna, czy jest repliką. Informacje dostarczone przez aktywną instancję są definitywne, a informacje z węzłów replik mogą być nieaktualne.

Można wykonać następujące czynności:

- Sprawdź, czy instancja menedżera kolejek w bieżącym węźle jest aktywna, czy jest repliką.
- Wyświetl status działania rodzimej wysokiej dostępności dla instancji w bieżącym węźle.
- Wyświetl status działania wszystkich trzech instancji w konfiguracji rodzimej wysokiej dostępności.

Następujące pola statusu są używane do raportowania statusu konfiguracji rodzimej wysokiej dostępności:

### ROLE

Określa bieżącą rolę instancji i jest jedną z następujących wartości: Active, Replica lub Unknown.

### INSTANCE

Nazwa podana dla tej instancji menedżera kolejek, gdy była ona tworzona przy użyciu opcji **-lr** komendy **crtmqm**.

### INSYNC

Wskazuje, czy w razie potrzeby instancja jest w stanie przejąć rolę aktywnej instancji.

### QUORUM

Raportuje status kworum w formacie *liczba zsynchronizowanych instancji/liczba skonfigurowanych instancji*.

### REPLADDR

Adres replikacji instancji menedżera kolejek.

### CONNECTV

Wskazuje, czy węzeł jest połączony z aktywną instancją.

### BACKLOG

Określa zaległość instancji w kB.

### CONNINST

Informuje, czy instancja o podanej nazwie jest połączona z tą instancją.

### ALTDAT

Wskazuje datę ostatniej aktualizacji tych informacji (wartość pusta, jeśli nigdy nie zostały zaktualizowane).

### ALTTIME

Wskazuje godzinę ostatniej aktualizacji tych informacji (wartość pusta, jeśli nigdy nie zostały zaktualizowane).

## Procedura

- Znajdź pods, które są częścią menedżera kolejek.

```
oc get pod --selector app.kubernetes.io/instance=nativeha-qm
```

- Uruchom dspmą w jednym z zasobników

```
oc exec -t Pod dspmą
```

```
oc rsh Pod
```

dla interaktywnej powłoki, w której można bezpośrednio uruchomić produkt dspmą.

- Aby określić, czy instancja menedżera kolejek działa jako aktywna instancja, czy jako replika:

```
oc exec -t Pod dspmą -o status -m QMgrName
```

Aktywna instancja menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          STATUS(Running)
```

Instancja-replika menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          STATUS(Replica)
```

Nieaktywna instancja zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          STATUS(Ended Immediately)
```

- Aby sprawdzić status działania rodzimej wysokiej dostępności instancji w podanym zasobniku:

```
oc exec -t Pod dspmą -o nativeha -m QMgrName
```

Aktywna instancja menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

Instancja-replika menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```

Nieaktywna menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- Aby określić status działania rodzimej wysokiej dostępności dla wszystkich instancji w konfiguracji o rodzimej wysokiej dostępności:

```
oc exec -t Pod dspmą -o nativeha -x -m QMgrName
```

Jeśli ta komenda zostanie wydana w węźle, w którym działa aktywna instancja menedżera kolejek BOB, może zostać wyświetlony następujący status:

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

Jeśli ta komenda zostanie wydana w węźle, w którym uruchomiona jest instancja-replika menedżera kolejek BOB, może zostać wyświetlony następujący status, który wskazuje, że jedna z replik ma zaległości:

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No) BACKLOG(435)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```



Jeśli ta komenda zostanie wydana w węzle, w którym uruchomiona jest nieaktywna instancja menedżera kolejek BOB, może zostać wyświetlony następujący status:

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
INSTANCE(inst1)     ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown)   CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst2)     ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown)   CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst3)     ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown)   CONNINST(No) ALTDATA() ALTTIME()
```

W przypadku wydania komendy w momencie, gdy instancje nadal negocjują, która będzie aktywna, a które będą replikami, zwrócony zostanie następujący status:

```
QMNAME(BOB)          STATUS(Negotiating)
```

## Odsyłacze pokrewne

Komenda `dspmq` (wyświetl menedżery kolejek)

“Przykład: Konfigurowanie menedżera kolejek o rodzimej wysokiej dostępności” na stronie 97

W tym przykładzie przedstawiono sposób wdrażania menedżera kolejek przy użyciu rodzimej funkcji wysokiej dostępności w produkcie Red Hat OpenShift Container Platform (OCP) przy użyciu operatora IBM MQ Operator.

**V9.2.3** **CP4I** **CD** *Strojenie zaawansowane rodzimej wysokiej dostępności*

Ustawienia zaawansowane służące do strojenia liczników czasu i interwałów. Nie powinno być konieczne używanie tych ustawień, chyba że wiadomo, iż wartości domyślne nie są zgodne z wymaganiami systemu.

Podstawowe opcje konfigurowania rodzimej wysokiej dostępności są obsługiwane za pomocą interfejsu API QueueManager, którego IBM MQ Operator używa do konfigurowania bazowych plików INI menedżera kolejek dla użytkownika. Istnieją bardziej zaawansowane opcje, które można konfigurować tylko za pomocą pliku INI, w sekcji [NativeHALocalinstancji](#). Więcej informacji na temat sposobu konfigurowania pliku INI zawiera sekcja [“Przykład: Udostępnianie plików MQSC i INI”](#) na stronie 88 .

## HeartbeatInterval

Interwał pulsu definiuje, co ile milisekund aktywna instancja menedżera kolejek o rodzimej wysokiej dostępności wysyła puls sieciowy. Poprawny zakres wartości interwału pulsu wynosi od 500 (0,5 sekundy) do 60000 (1 minuta), a wartość spoza tego zakresu powoduje, że uruchomienie menedżera kolejek nie powiedzie się. Jeśli ten atrybut zostanie pominięty, zostanie użyta wartość domyślna 5000 (5 sekund). Każda instancja musi używać tego samego interwału pulsu.

## HeartbeatTimeout

Limit czasu pulsu definiuje, jak długo instancja-replika menedżera kolejek o rodzimej wysokiej dostępności oczekuje przed podjęciem decyzji, że aktywna instancja nie odpowiada. Poprawny zakres limitu czasu interwału pulsu wynosi od 500 (0,5 sekundy) do 120000 (2 minuty). Wartość limitu czasu pulsu musi być większa niż lub równa wartości interwału pulsu.

Niepoprawna wartość powoduje, że uruchomienie menedżera kolejek nie powiedzie się. Jeśli ten atrybut zostanie pominięty, replika oczekuje na 2 x HeartbeatInterval przed rozpoczęciem procesu w celu wybrania nowej aktywnej instancji. Każda instancja musi używać tego samego limitu czasu pulsu.

## RetryInterval

Interwał ponowienia definiuje, co ile milisekund menedżer kolejek o rodzimej wysokiej dostępności powinien ponawiać próbę nawiązania połączenia przez niesprawne łącze replikacji. Poprawny zakres interwału ponowienia wynosi od 500 (0,5 sekundy) do 120000 (2 minuty). Jeśli ten atrybut zostanie pominięty, replika oczekuje na 2 x HeartbeatInterval przed ponowną próbą utworzenia łącza replikacji, które zakończyło się niepowodzeniem.

**CP4I** *Kończenie rodzimych menedżerów kolejek wysokiej dostępności*

Za pomocą komendy `endmqm` można zakończyć działanie aktywnego lub replika menedżera kolejek, który jest częścią rodzimej grupy wysokiej dostępności.

## Procedura

- Aby zakończyć aktywną instancję menedżera kolejek, należy zapoznać się z sekcją Końcowanie rodzimych menedżerów kolejek HA w sekcji Konfigurowanie tej dokumentacji.

**CP4I** **V 9.2.2** **CD** Ocena rodzimej wysokiej dostępności w wersji IBM Cloud Pak for Integration 2021.1.1

Okres oceny rodzimego rozwiązania do zapewnienia wysokiej dostępności w produkcie IBM Cloud Pak for Integration 2021.1.1 zakończył się. Należy użyć zaktualizowanego rodzimego rozwiązania do zapewnienia wysokiej dostępności dostępnego w produkcie IBM Cloud Pak for Integration 2021.2.1 z wykorzystaniem produktu IBM MQ Operator w wersji 1.6 lub nowszej z produktem IBM MQ w wersji 9.2.3 lub nowszej.

### Zadania pokrewne

“Wyświetlanie statusu rodzimych menedżerów kolejek HA dla kontenerów certyfikowanych IBM MQ” na stronie 103

W przypadku kontenerów certyfikowanych przez produkt IBM MQ można wyświetlić status rodzimych instancji HA, uruchamiając komendę **dspmq** w jednym z działających Pods.

### Odsyłacze pokrewne

“Przykład: Konfigurowanie menedżera kolejek o rodzimej wysokiej dostępności” na stronie 97

W tym przykładzie przedstawiono sposób wdrażania menedżera kolejek przy użyciu rodzimej funkcji wysokiej dostępności w produkcie Red Hat OpenShift Container Platform (OCP) przy użyciu operatora IBM MQ Operator.

## **OpenShift** **CP4I** **Przykład: konfigurowanie wieloinstancyjnego menedżera kolejek**

W tym przykładzie przedstawiono sposób wdrażania wieloinstancyjnego menedżera kolejek w programie Red Hat OpenShift Container Platform (OCP) przy użyciu programu IBM MQ Operator. W tym przykładzie będziemy również skonfigurować komunikację TLS między przykładowym klientem i menedżerem kolejek. Przykład weryfikuje konfigurację, umieszczając i pobierając komunikaty przed zasymulowaną awarią zasobnika i po tej zasymulowanej awarii.

## Zanim zaczniesz

Aby zrealizować ten przykład, należy najpierw spełnić następujące wymagania wstępne:

- Zainstaluj produkt IBM MQ client i dodaj zainstalowane katalogi `samp/bin` i `bin` do zmiennej środowiskowej `PATH`. Klient udostępnia aplikacje **runmqakm**, **amqsputc** i **amqsgetc** wymagane w tym przykładzie. Zainstaluj IBM MQ client w następujący sposób:
  - **Windows** **Linux** W przypadku Windows i Linux: zainstaluj klienta IBM MQ (podlegającego redystrybucji) dla używanego systemu operacyjnego z serwisu <https://ibm.biz/mq92redistclients>
  - **mac OS** W systemie Mac: pobierz i skonfiguruj IBM MQ MacOS Toolkit. Patrz <https://developer.ibm.com/tutorials/mq-macos-dev/>.
- Zainstaluj narzędzie OpenSSL dla używanego systemu operacyjnego. W tym celu należy wygenerować samopodpisany certyfikat dla menedżera kolejek, jeśli nie masz jeszcze klucza prywatnego i certyfikatu.
- Utwórz projekt/przestrzeń nazw Red Hat OpenShift Container Platform (OCP) dla tego przykładu.
- W wierszu komend zaloguj się do klastra OCP i przetłącz się na powyższą przestrzeń nazw.
- Upewnij się, że operator IBM MQ Operator jest zainstalowany i dostępny w powyższej przestrzeni nazw.
- W OCP skonfiguruj domyślną klasę pamięci masowej, która ma być używana przez menedżer kolejek. Aby ukończyć ten kurs bez ustawiania domyślnej klasy pamięci masowej, patrz Uwaga 2: Używanie klasy pamięci masowej innej niż domyślna.

## Informacje o zadaniu

Wieloinstancyjne menedżery kolejek korzystają z aktywnego i rezerwowego zasobnika Kubernetes. Są one uruchamiane jako część zestawu stanowego Kubernetes z dokładnie dwiema replikami i zestawem trwałych woluminów Kubernetes. Więcej informacji na temat menedżerów kolejek o wielu instancjach zawiera sekcja [“Wysoka dostępność produktu IBM MQ w kontenerach”](#) na stronie 16.

Przykład obejmuje niestandardowy kod YAML zasobu definiujący wieloinstancyjny menedżer kolejek z trwałą pamięcią masową i protokołem TLS. Po wdrożeniu menedżera kolejek w OCP zasymulujemy awarię aktywnego menedżera kolejek. Po awarii będzie można zaobserwować automatyczne odtwarzanie i potwierdzić jego powodzenie poprzez umieszczanie i pobieranie komunikatów.

## Przykład

### Tworzenie klucza prywatnego TLS i certyfikatów dla serwera MQ

Dokumenty w tej sekcji przedstawiają sposób tworzenia samopodpisanego certyfikatu dla menedżera kolejek oraz sposób dodawania certyfikatu do bazy danych kluczy, która będzie pełnił rolę magazynu zaufanych certyfikatów dla klienta. Jeśli masz już klucz prywatny i certyfikat, możesz ich użyć zamiast tworzyć nowe. Należy pamiętać, że certyfikatów samopodpisanych powinno się używać tylko na etapie prac programistycznych.

Aby utworzyć samopodpisany klucz prywatny i certyfikat publiczny w bieżącym katalogu, uruchom następującą komendę:

```
openssl req -newkey rsa:2048 -nodes -keyout tls.key -subj "/CN=localhost" -x509 -days 3650 -out tls.crt
```

### Dodanie klucza publicznego menedżera kolejek do bazy danych kluczy klienta

Baza danych kluczy klienta jest używana jako magazyn zaufanych certyfikatów dla aplikacji klienckiej.

Utwórz bazę danych kluczy klienta:

```
runmqakm -keydb -create -db clientkey.kdb -pw password -type cms -stash
```

Dodaj wcześniej wygenerowany klucz publiczny do bazy danych kluczy klienta:

```
runmqakm -cert -add -db clientkey.kdb -label mqservercert -file tls.crt -format ascii -stashed
```

### Tworzenie klucza tajnego zawierającego certyfikaty TLS dla wdrożenia menedżera kolejek

Aby menedżer kolejek mógł odwoływać się do klucza i certyfikatu, należy utworzyć klucz tajny Kubernetes TLS, który odwołuje się do plików utworzonych powyżej. W tym celu należy upewnić się, że znajdują się one w przestrzeni nazw, która została utworzona przed rozpoczęciem tego zadania.

```
oc create secret tls example-mi-secret --key="tls.key" --cert="tls.crt"
```

### Utworzenie mapy konfiguracji zawierającej komendy MQSC

Utwórz mapę konfiguracji Kubernetes zawierającą komendy MQSC, aby utworzyć nową kolejkę i kanał SVRCONN, a także dodaj rekord uwierzytelniania kanału, który umożliwia dostęp do kanału, blokując tylko użytkowników o nazwie *nobody*.

Należy zauważyć, że to podejście powinno być używane tylko w celach programistycznych.

Upewnij się, że jesteś w przestrzeni nazw, która została wcześniej utworzona (patrz sekcja [“Zanim zaczniesz”](#) na stronie 106), a następnie wprowadź następującą komendę YAML w interfejsie użytkownika OCP lub za pomocą wiersza komend:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-mi-configmap
data:
  tls.mqsc: |
    DEFINE QLOCAL('EXAMPLE.QUEUE') DEFPSIST(YES) REPLACE
    DEFINE CHANNEL(MIQMCHL) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCAUTH(OPTIONAL)
```

```
SSLCIPH('ANY_TLS12_OR_HIGHER')
SET CHLAUTH(MIQMCHL) TYPE(BLOCKUSER) USERLIST('nobody') ACTION(ADD)
```

## Konfigurowanie kierowania

Jeśli używany jest IBM MQ client lub pakiet narzędzi w wersji IBM MQ 9.2.1 lub nowszej, można skonfigurować kierowanie do menedżera kolejek przy użyciu pliku konfiguracyjnego menedżera kolejek (pliku INI). W pliku należy ustawić zmienną *OutboundSNI* na trasę opartą na nazwie hosta, a nie na nazwie kanału.

W katalogu, w którym są wykonywane komendy, utwórz plik o nazwie "mqclient.ini", w którym znajduje się następujący tekst:

```
## Module Name: mqclient.ini ##
## Type      : IBM MQ MQI client configuration file ##
## Function  : Define the configuration of a client ##
## ##
##*****##
## Notes    : ##
## 1) This file defines the configuration of a client ##
## ##
##*****##
SSL:
  OutboundSNI=HOSTNAME
```

Uwaga: na tej stronie nie należy zmieniać żadnych wartości. Na przykład tańcuch HOSTNAME powinien być pozostawiony bez zmian.

Więcej informacji na ten temat zawiera temat [Sekcja SSL pliku konfiguracyjnego klienta](#).

W przypadku użycia IBM MQ client lub pakietu narzędzi wcześniejszego niż IBM MQ 9.2.1 należy utworzyć trasę OCP zamiast poprzedniego pliku konfiguracyjnego. Wykonaj kroki opisane w sekcji [Uwaga 1: Tworzenie trasy](#).

## Wdrażanie menedżera kolejek

**Ważne:** W tym przykładzie używana jest zmienna *MQSNOAUT* w celu wyłączenia autoryzacji w menedżerze kolejek, co pozwala skupić się na krokach wymaganych do połączenia klienta przy użyciu protokołu TLS. Nie jest to zalecane w przypadku produkcyjnego wdrożenia IBM MQ, ponieważ powoduje, że wszystkie aplikacje nawiązują połączenie z pełnymi uprawnieniami administracyjnymi, bez mechanizmu obniżania uprawnień poszczególnych aplikacji.

Skopiuj i zaktualizuj następujący kod YAML.

- Upewnij się, że określono poprawną licencję. Patrz [Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1](#).
- Zaakceptuj licencję, zmieniając wartość `false` na `true`.
- Jeśli używany jest produkt IBM Cloud File Storage, patrz [Uwaga 3: Korzystanie z pamięci masowej IBM Cloud File Storage](#)

Kod YAML zasobu niestandardowego menedżera kolejek:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: miexample
spec:
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: NonProduction
  queueManager:
    name: MIEXAMPLE
    availability:
      type: MultiInstance
  mqsc:
    - configMap:
        name: example-mi-configmap
      items:
```

```

- tls.mqsc
template:
  pod:
    containers:
      - env:
          - name: MQSNOAUT
            value: 'yes'
          name: qmgr
        version: 9.2.5.0-r3
    web:
      enabled: true
    pki:
      keys:
        - name: example
          secret:
            secretName: example-mi-secret
            items:
              - tls.key
              - tls.crt

```

Będąc w przestrzeni nazw utworzonej wcześniej, należy wdrożyć zaktualizowany kod YAML w interfejsie użytkownika OCP, przy użyciu wiersza komend lub przy użyciu programu IBM Cloud Pak for Integration Platform Navigator.

## Sprawdzanie poprawności

Po krótkim opóźnieniu wieloinstancyjny menedżer kolejek powinien być skonfigurowany i dostępny do użycia. W tej sekcji sprawdzamy, czy menedżer kolejek zachowuje się zgodnie z oczekiwaniami.

### Upewnij się, że menedżer kolejek jest uruchomiony

Menedżer kolejek jest obecnie wdrażany. Przed kontynuowaniem upewnij się, że znajduje się on w stanie Running. Na przykład:

```
oc get qmgr miexample
```

### Testowanie połączenia z menedżerem kolejek

Aby potwierdzić, że menedżer kolejek został skonfigurowany do jednokierunkowej komunikacji TLS, należy użyć przykładowych aplikacji **amqspu** i **amqsgc**:

### Znajdowanie nazwy hosta menedżera kolejek

Aby znaleźć nazwę hosta menedżera kolejek dla ścieżki `miexample-ibm-mq-qm`, należy uruchomić następującą komendę. Nazwa hosta jest zwracana w polu `HOST`.

```
oc get routes miexample-ibm-mq-qm
```

### Określenie danych menedżera kolejek

Utwórz plik `CCDT`. JSON, który określa szczegóły menedżera kolejek. Zastąp wartość `host` nazwą hosta z poprzedniego kroku.

```

{
  "channel":
  [
    {
      "name": "MIQMCHL",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "<host from previous step>",
            "port": 443
          }
        ],
        "queueManager": "MIEXAMPLE"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "ECDHE_RSA_AES_128_CBC_SHA256"
      },
      "type": "clientConnection"
    }
  ]
}

```

```
} ]
```

### Eksportowanie zmiennych środowiskowych

Wyeksportuj następujące zmienne środowiskowe w sposób odpowiedni dla używanego systemu operacyjnego. Te zmienne będą odczytywane przez aplikacje **amqspu**tc i **amqsget**c.

Zaktualizuj ścieżkę do plików w systemie:

```
export MQCCDTURL='<full_path_to_file>/CCDT.JSON'  
export MQSSLKEYR='<full_path_to_file>/clientkey'
```

### Wstawianie komunikatów do kolejki

Uruchom następującą komendę:

```
amqspu
```

tc EXAMPLE.QUEUE MIEXAMPLE

Jeśli nawiązanie połączenia z menedżerem kolejek powiedzie się, zostanie wygenerowana następująca odpowiedź:

```
target queue is EXAMPLE.QUEUE
```

Umieść kilka komunikatów w kolejce, wprowadzając tekst i naciskając za każdym razem klawisz **Enter**.

Aby zakończyć, naciśnij klawisz **Enter** dwa razy.

### Pobieranie komunikatów z kolejki

Uruchom następującą komendę:

```
amqsget
```

c EXAMPLE.QUEUE MIEXAMPLE

Komunikaty dodane w poprzednim kroku zostaną skonsumowane i wyświetlone.

Po kilku sekundach komenda kończy działanie..

### Wymuszanie awarii aktywnego zasobnika

Aby sprawdzić poprawność automatycznego odtwarzania wieloinstancyjnego menedżera kolejek, zasymuluj awarię zasobnika:

### Wyświetlanie aktywnych i rezerwowych zasobników

Uruchom następującą komendę:

```
oc get pods
```

Uwaga: w polu **READY** aktywny zasobnik zwraca wartość 1/1, natomiast zasobnik rezerwowy zwraca wartość 0/1.

### Usuwanie aktywnego zasobnika

Uruchom następującą komendę, podając pełną nazwę aktywnego zasobnika:

```
oc delete pod miexample-ibm-mq-<value>
```

### Wyświetl ponownie status zasobnika

Uruchom następującą komendę:

```
oc get pods
```

### Wyświetlanie dziennika zasobnika rezerwowego

Uruchom następującą komendę, podając pełną nazwę zasobnika rezerwowego:

```
oc logs miexample-ibm-mq-<value>
```

Powinien zostać wyświetlony następujący komunikat:

```
IBM MQ queue manager 'MIEXAMPLE' becoming the active instance.
```

## Ponowne umieszczanie i pobieranie komunikatów

Gdy zasobnik rezerwowany stanie się aktywny (czyli po tym, jak wartość pola `READY` zmieni się na `1/1`), należy ponownie użyć następujących komend, jak opisano powyżej, w celu umieszczenia komunikatów w menedżerze kolejek, a następnie pobrać komunikaty z menedżera kolejek:

```
amqsputc EXAMPLE.QUEUE MIEXAMPLE
```

```
amqsgetc EXAMPLE.QUEUE MIEXAMPLE
```

W ten sposób pomyślnie wdrożyliśmy wieloinstancyjny menedżer kolejek i wykazaliśmy, że może on automatycznie odzyskać sprawność w przypadku awarii zasobnika.

## Dodatkowe informacje

### Uwaga 1: Tworzenie trasy

Jeśli używany jest pakiet IBM MQ client lub pakiet narzędzi wcześniejszy niż IBM MQ 9.2.1, należy utworzyć trasę OCP.

Aby utworzyć trasę, upewnij się, że jesteś w przestrzeni nazw, która została wcześniej utworzona (patrz sekcja [“Zanim zaczniesz”](#) na stronie 106), a następnie wprowadź następujący kod YAML w interfejsie użytkownika OCP lub za pomocą wiersza komend:

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: example-mi-route
spec:
  host: miqmchl.ch1.mq.ibm.com
  to:
    kind: Service
    name: miexample-ibm-mq
  port:
    targetPort: 1414
  tls:
    termination: passthrough
```

Należy zauważyć, że Red Hat OpenShift Container Platform Router używa SNI do kierowania żądań do menedżera kolejek IBM MQ. Jeśli zostanie zmieniona nazwa kanału określona w [odwzorowaniu konfiguracji zawierającym komendy MQSC](#), należy również zmienić pole `hosta` w tym miejscu i w pliku `CCDT.JSON`, który określa szczegóły menedżera kolejek. Więcej informacji na ten temat zawiera sekcja [“Konfigurowanie trasy w celu nawiązania połączenia z menedżerem kolejek spoza klastra produktu Red Hat OpenShift”](#) na stronie 112.

### Uwaga 2: Używanie klasy pamięci masowej innej niż domyślna

W tym przykładzie oczekuje się, że domyślna klasa pamięci masowej została skonfigurowana w OCP, dlatego w kodzie YAML zasobu niestandardowego menedżera kolejek nie jest wymagana żadna informacja o pamięci masowej. Jeśli klasa pamięci masowej nie jest skonfigurowana jako domyślna lub ma być używana inna klasa pamięci masowej, należy dodać `defaultClass: <storage_class_name>` w `spec.queueManager.storage`.

Nazwa klasy pamięci masowej musi być dokładnie zgodna z nazwą klasy pamięci masowej, która istnieje w systemie OCP. Oznacza to, że musi być ona zgodna z nazwą zwróconej przez komendę `oc get storageclass`. Musi również obsługiwać produkt `ReadWriteMany`. Więcej informacji na ten temat zawiera sekcja [“Uwagi dotyczące pamięci masowej dla IBM MQ Operator”](#) na stronie 11.

### Uwaga 3: Używanie IBM Cloud File Storage

W niektórych sytuacjach, na przykład w przypadku korzystania z produktu IBM Cloud File Storage, należy również określić pole `securityGroups` w kodzie YAML dostosowanego zasobu menedżera kolejek. Na przykład poprzez dodanie następującego pola podrzędnego bezpośrednio w `spec`:

```
securityContext:
  supplementalGroups: [99]
```

Więcej informacji na ten temat zawiera [“Uwagi dotyczące pamięci masowej dla IBM MQ Operator”](#) na stronie 11.

## OpenShift CP4I CD Konfigurowanie trasy w celu nawiązania połączenia z menedżerem kolejek spoza klastra produktu Red Hat OpenShift

Do połączenia aplikacji z menedżerem kolejek IBM MQ spoza klastra Red Hat OpenShift potrzebna jest trasa Red Hat OpenShift. Należy włączyć protokół TLS w menedżerze kolejek i aplikacji klienckiej IBM MQ, ponieważ protokół SNI jest dostępny w protokole TLS tylko wtedy, gdy używany jest protokół TLS w wersji 1.2 lub nowszej. Red Hat OpenShift Container Platform Router używa wskaźnika SNI do kierowania żądań do menedżera kolejek IBM MQ.

### O tym zadaniu



**Ostrzeżenie:** Ten dokument dotyczy wersji 9.2.1 Continuous Delivery i nowszych klientów IBM MQ. Jeśli klient korzysta z wersji 9.2.0 Long Term Support lub wcześniejszej, należy zapoznać się z informacjami na stronie dokumentacji produktu IBM MQ 9.1 dotyczącymi [nawiązywania połączenia z menedżerem kolejek wdrożonym w klastrze produktu Red Hat OpenShift](#).

**V 9.2.1** Wymagana konfiguracja Red Hat OpenShift Route zależy od zachowania aplikacji klienckiej Server Name Indication (SNI). IBM MQ obsługuje dwa różne ustawienia nagłówka SNI w zależności od konfiguracji i typu klienta. Nagłówek SNI jest ustawiony na nazwę hosta miejsca docelowego klienta lub na nazwę kanału IBM MQ. Informacje na temat odwzorowania nazwy kanału na nazwę hosta przez program IBM MQ można znaleźć w sekcji [Jak produkt IBM MQ obsługuje więcej niż jeden certyfikat](#).

**V 9.2.1** Określa, czy nagłówek SNI jest ustawiony na nazwę kanału IBM MQ, czy na nazwę hosta, za pomocą atrybutu **OutboundSNI**. Możliwe wartości to `OutboundSNI=CHANNEL` (wartość domyślna) lub `OutboundSNI=HOSTNAME`. Więcej informacji na ten temat zawiera sekcja SSL pliku konfiguracyjnego klienta. Należy zauważyć, że CHANNEL i HOSTNAME są dokładnymi wartościami, które są używane; nie są to nazwy zmiennych, które są zastępowane rzeczywistą nazwą kanału lub nazwą hosta.

**V 9.2.1**

### Zachowania klienta z różnymi ustawieniami OutboundSNI

Jeśli parametr **OutboundSNI** jest ustawiony na HOSTNAME, to następujące klienty ustawiają nazwę hosta SNI pod warunkiem, że w nazwie połączenia jest podana nazwa hosta:

- Klienci C
- Klienci .NET w trybie niezarządzanym
- Java/klienci JMS

Jeśli parametr **OutboundSNI** jest ustawiony na HOSTNAME, a w nazwie połączenia używany jest adres IP, to następujące klienty wysyłają pusty nagłówek SNI:

- Klienci C
- Klienci .NET w trybie niezarządzanym
- Klienci Java/JMS (które nie mogą wykonać odwrotnego wyszukiwania DNS nazwy hosta)

Jeśli parametr **OutboundSNI** jest ustawiony na CHANNEL lub nie jest ustawiony, zamiast tego jest używana nazwa kanału IBM MQ, która jest zawsze wysyłana, niezależnie od tego, czy używana jest nazwa hosta, czy nazwa połączenia z adresem IP.

Następujące typy klientów nie obsługują ustawiania nagłówka SNI na nazwę kanału IBM MQ, więc zawsze należy ustawić nagłówek SNI na nazwę hosta niezależnie od ustawienia **OutboundSNI**:

- Klienci AMQP
- Klienci XR
- Klienci .NET w trybie zarządzanym (Przed IBM MQ 9.2.0 Fix Pack 4 for Long Term Support i przed IBM MQ 9.2.3 for Continuous Delivery.)



W produkcie IBM MQ 9.2.0 Fix Pack 4 for Long Term Support oraz w produkcie IBM MQ 9.2.3 for Continuous Delivery klient IBM MQ zarządzany .NET został zaktualizowany i ustawił wartość SERVERNAME na odpowiednią nazwę hosta, jeśli właściwość **OutboundSNI** ma wartość HOSTNAME, co umożliwi IBM MQ zarządzanemu klientowi .NET nawiązanie połączenia z menedżerem kolejek za pomocą tras Red Hat OpenShift . Należy zauważyć, że w pliku IBM MQ 9.2.0 Fix Pack 4 właściwość **OutboundSNI** jest dodawana i obsługiwana tylko z pliku `mqclient.ini` . Nie można ustawić tej właściwości z poziomu aplikacji .NET.

Jeśli aplikacja kliencka łączy się z menedżerem kolejek wdrożonym w klastrze Red Hat OpenShift za pomocą programu IBM MQ Internet Pass-Thru (MQIPT), produkt MQIPT można skonfigurować w taki sposób, aby ustawić SNI na nazwę hosta, używając właściwości [SSLClientOutboundSNI](#) w definicji trasy.

### OutboundSNI, wiele certyfikatów i trasy Red Hat OpenShift

Produkt IBM MQ korzysta z nagłówka SNI w celu udostępnienia wielu funkcji certyfikatów. Jeśli aplikacja łączy się z kanałem IBM MQ skonfigurowanym do używania innego certyfikatu za pomocą pola CERTLABL, aplikacja musi połączyć się z ustawieniem **OutboundSNI** o wartości CHANNEL.

Jeśli konfiguracja trasy systemu Red Hat OpenShift wymaga nazwy hosta SNI, nie można użyć funkcji obsługi wielu certyfikatów programu IBM MQ i nie można ustawić ustawienia CERTLABL dla żadnego obiektu kanału IBM MQ .

Jeśli aplikacja z ustawieniem **OutboundSNI** innym niż CHANNEL nawiąże połączenie z kanałem ze skonfigurowaną etykietą certyfikatu, aplikacja zostanie odrzucona z błędem MQRC\_SSL\_INITIALIZATION\_ERROR i w dziennikach błędów menedżera kolejek zostanie zapisany komunikat AMQ9673 .

Więcej informacji na temat sposobu, w jaki program IBM MQ udostępnia funkcje obsługi wielu certyfikatów, zawiera sekcja [W jaki sposób program IBM MQ udostępnia możliwość obsługi wielu certyfikatów](#) .

### Przykład

Aplikacje klienckie, które ustawiają SNI na kanał MQ, wymagają utworzenia nowej trasy Red Hat OpenShift dla każdego kanału, z którym ma zostać nawiązane połączenie. Konieczne jest także używanie nazw kanałów unikalnych w klastrze Red Hat OpenShift Container Platform, aby możliwe było kierowanie do właściwego menedżera kolejek.

Ważne jest, aby nazwy kanałów produktu MQ nie kończyły się na małej literze ze względu na sposób, w jaki produkt IBM MQ odwzorowuje nazwy kanałów na nagłówki SNI.

Aby określić wymaganą nazwę hosta dla każdej z nowych tras produktu Red Hat OpenShift, należy odwzorować każdą nazwę kanału na adres SNI. Więcej informacji na ten temat zawiera sekcja [Jak produkt IBM MQ obsługuje więcej niż jeden certyfikat](#).

Następnie należy utworzyć nową trasę Red Hat OpenShift dla każdego kanału. W tym celu należy zastosować następującą komendę `yaml` w klastrze:

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: <provide a unique name for the Route>
  namespace: <the namespace of your MQ deployment>
spec:
  host: <SNI address mapping for the channel>
  to:
    kind: Service
    name: <the name of the Kubernetes Service for your MQ deployment (for example "Queue Manager Name)-ibm-mq">
  port:
    targetPort: 1414
```

```
tls:
  termination: passthrough
```

## Konfigurowanie szczegółów połączenia aplikacji klienckiej

Nazwę hosta, która ma być używana dla połączenia z klientem, można określić, uruchamiając następującą komendę:

```
oc get route <Name of hostname based Route (for example "<Queue Manager Name>-ibm-mq-qm")>
-n <namespace of your MQ deployment> -o jsonpath="{.spec.host}"
```

Port dla połączenia klienckiego powinien być ustawiony na port używany przez router Red Hat OpenShift Container Platform – zwykle 443.

### Zadania pokrewne

[“Nawiązywanie połączenia z rozwiązaniem IBM MQ Console wdrożonym w klastrze Red Hat OpenShift” na stronie 118](#)

W jaki sposób nawiązać połączenie z konsolą IBM MQ Console menedżera kolejek, która została wdrożona w klastrze Red Hat OpenShift Container Platform.

## CP4I Integracja z panelem kontrolnym IBM Cloud Pak for Integration Operations Dashboard

Panel kontrolny Operations Dashboard umożliwia śledzenie transakcji przechodzących przez rozwiązanie IBM Cloud Pak for Integration.

### O tym zadaniu

Włączenie integracji z panelem kontrolnym Operations Dashboard powoduje zainstalowanie funkcji API produktu MQ jako procedury zewnętrznej w menedżerze kolejek. Procedura zewnętrzna API wysyła dane śledzenia do składnicy danych panelu kontrolnego Operations Dashboard. Dane te dotyczą komunikatów, które przepływają przez menedżer kolejek.

Należy zauważyć, że śledzone są tylko komunikaty wysyłane za pomocą powiązań klienta produktu MQ.

Należy także pamiętać, że w przypadku wersji produktu IBM MQ Operator wcześniejszej niż 1.5, gdy śledzenie jest włączone, obrazy agenta śledzenia i kolektora wdrożone razem z menedżerem kolejek zawsze były dostępne w najnowszych wersjach, co może spowodować niekompatybilność, jeśli nie jest używana najnowsza wersja IBM Cloud Pak for Integration.

### Procedura

#### 1. Wdrażanie menedżera kolejek z włączonym śledzeniem

Domyślnie funkcja śledzenia jest wyłączona.

W przypadku wdrażania przy użyciu IBM Cloud Pak for Integration Platform Navigator można włączyć śledzenie podczas wdrażania, ustawiając opcję **Enable Tracing** na **On** i ustawiając **Tracing Namespace** na przestrzeń nazw, w której zainstalowany jest panel kontrolny Operations Dashboard. Aby uzyskać więcej informacji na temat wdrażania menedżera kolejek, patrz [“Wdrażanie menedżera kolejek przy użyciu produktu IBM Cloud Pak for Integration Platform Navigator” na stronie 84](#)

W przypadku wdrażania przy użyciu interfejsu wiersza komend Red Hat OpenShift lub konsoli WWW Red Hat OpenShift można włączyć śledzenie za pomocą następującego fragmentu kodu YAML:

```
spec:
  tracing:
    enabled: true
    namespace: <Operations_Dashboard_Namespace>
```

**Ważne:** Menedżer kolejek nie zostanie uruchomiony, dopóki produkt MQ nie zostanie zarejestrowany w panelu kontrolnym Operations Dashboard (patrz następny krok).

Należy zwrócić uwagę, że gdy ta funkcja jest włączona, oprócz kontenera z menedżerem kolejek działać będą także dwa kontenery pomocnicze („Agent” i „Collector”). Obrazy tych kontenerów pomocniczych będą dostępne w tym samym rejestrze, co główny obraz MQ, i będą używać tej samej strategii pobierania oraz sekretu pobierania. Dostępne są dodatkowe ustawienia służące do konfigurowania limitów procesora i pamięci.

2. Jeśli menedżer kolejek z integracją z panelem kontrolnym Operations Dashboard został po raz pierwszy wdrożony w tej przestrzeni nazw, wówczas należy dokonać rejestracji w panelu kontrolnym Operations Dashboard.

Rejestracja powoduje utworzenie obiektu Secret potrzebnego do uruchomienia zasobnika menedżera kolejek.

## **Wdrażanie lub aktualizowanie IBM MQ 9.2.2 lub 9.2.3 z integracją panelu kontrolnego Operations Dashboard w IBM Cloud Pak for Integration 2021.4**

Każda wersja produktu IBM MQ jest powiązana z konkretną wersją wdrażane wraz z menedżerem kolejek komponentów agenta i kolektora panelu kontrolnego operacji. Produkt IBM Cloud Pak for Integration 2021.4.1 wprowadza zmianę, która powoduje, że starsze komponenty agenta i kolektora nie będą działać z panelem kontrolnym operacji. Aby rozwiązać ten problem, należy przestonić wersję obrazów agenta Operations Dashboard i kolektora używaną podczas używania IBM MQ 9.2.2 lub 9.2.3.

### **Wdrażanie nowego menedżera kolejek IBM MQ 9.2.2 lub 9.2.3**

Jeśli używany jest produkt IBM Cloud Pak for Integration 2021.4.1 z produktem IBM MQ 9.2.2 lub 9.2.3, należy nadpisać obrazy agenta i kolektora panelu kontrolnego operacji do wersji 2.4 w pliku YAML produktu QueueManager. Na przykład:

```
spec:
  tracing:
    agent:
      image: cp.icr.io/cp/icp4i/od/icp4i-od-agent@sha256:27a211f0f78eff765d1f9520e0f9841f902600bb556827477b206e209cb44d20
    collector:
      image: cp.icr.io/cp/icp4i/od/icp4i-od-collector@sha256:dc70b1341b23dc72642ce68809811f9db0e8a0c46bda2508e8eb3d4035e04f4b
```

W przeciwnym razie zasobnik produktu QueueManager zablokuje się w stanie Pending. W przypadku aktualizacji do IBM MQ 9.2.4 można usunąć to przestonięcie.

### **Aktualizacja do wersji IBM Cloud Pak for Integration 2021.4.1**

**Uwaga:** Jeśli zachowujesz menedżer kolejek produktu IBM MQ 9.2.2 lub 9.2.3, nie wykonuj kroku 3.

1. Zaktualizuj produkt QueueManager w taki sposób, aby nadpisał obrazy agenta i kolektora, jak opisano powyżej.
2. Zaktualizuj operatory IBM Cloud Pak for Integration, w tym panel kontrolny Operations Dashboard, i operator IBM MQ zgodnie z opisem w sekcji “Aktualizowanie operatora IBM MQ Operator i menedżerów kolejek” na stronie 73.
3. (Opcjonalnie) Aby zaktualizować produkt do wersji IBM MQ 9.2.4 lub nowszej, zaktualizuj produkt QueueManager, aby używać `.spec.version` dla wersji produktu IBM MQ, a następnie usuń nadpisanie obrazów agenta i kolektora.

## **Budowanie obrazu z niestandardowymi plikami MQSC i INI przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift**

Używając potoku Red Hat OpenShift Container Platform, można utworzyć nowy obraz kontenera IBM MQ z plikami MQSC i INI, które mają być zastosowane do menedżerów kolejek korzystających z tego obrazu. To zadanie powinien wykonać administrator projektu.

## Zanim rozpoczniesz

Konieczne będzie zainstalowanie [interfejsu wiersza komend Red Hat OpenShift Container Platform](#).

Zaloguj się do klastra, używając komendy **cloudctl login** (w przypadku IBM Cloud Pak for Integration) lub **oc login**.

Jeśli w projekcie produktu Red Hat OpenShift nie ma klucza tajnego produktu Red Hat OpenShift dla IBM Entitled Registry, należy wykonać kroki opisane w sekcji [“Przygotowywanie projektu Red Hat OpenShift dla produktu IBM MQ” na stronie 82](#).

## Procedura

### 1. Tworzenie strumienia obrazów (ImageStream)

Strumień obrazów i powiązane z nim znaczniki stanowią abstrakcję umożliwiającą odwoływanie się do obrazów kontenerów z poziomu produktu Red Hat OpenShift Container Platform. Strumień obrazów i jego znaczniki umożliwiają sprawdzenie, które obrazy są dostępne, i uzyskanie pewności, że używany jest właściwy obraz, nawet jeśli obraz w repozytorium ulegnie zmianie.

```
oc create imagestream mymq
```

### 2. Utwórz BuildConfig dla nowego obrazu

Program BuildConfig zezwoli na kompilację dla nowego obrazu, który będzie oparty na oficjalnych obrazach IBM, ale doda dowolny plik MQSC lub INI, który ma zostać uruchomiony w trybie uruchamiania kontenera.

#### a) Utwórz plik YAML definiujący zasób BuildConfig

Na przykład utwórz plik o nazwie „mq-build-config.yaml” o następującej treści:

```
apiVersion: build.openshift.io/v1
kind: BuildConfig
metadata:
  name: mymq
spec:
  source:
    dockerfile: |-
      FROM cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r3
      RUN printf "DEFINE QLOCAL(foo) REPLACE\n" > /etc/mqm/my.mqsc \
        && printf "Channels:\n\tMQIBindType=FASTPATH\n" > /etc/mqm/my.ini
      LABEL summary "My custom MQ image"
  strategy:
    type: Docker
    dockerStrategy:
      from:
        kind: "DockerImage"
        name: "cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r3"
      pullSecret:
        name: ibm-entitlement-key
  output:
    to:
      kind: ImageStreamTag
      name: 'mymq:latest-amd64'
```

Aby wskazać poprawny obraz bazowy IBM MQ, który ma być używany, należy zastąpić nazwy w dwóch miejscach, w których jest wymieniony bazowy obraz (aby uzyskać szczegółowe informacje, patrz [“Historia wydań operatora IBM MQ Operator” na stronie 21](#)). Po zastosowaniu poprawek należy powtórzyć te kroki, aby ponownie zbudować obraz.

W tym przykładzie tworzony jest nowy obraz oparty na oficjalnym obrazie IBM i dodawane są pliki o nazwie "my.mqsc" i "my.ini" do katalogu /etc/mqm. Wszystkie pliki MQSC lub INI znajdujące się w tym katalogu będą stosowane przez kontener przy uruchomieniu. Pliki INI są stosowane przy użyciu opcji **crtmqm -ii** i scalane z istniejącymi plikami INI. Pliki MQSC są stosowane w porządku alfabetycznym.

Komendy MQSC powinny być powtarzalne, ponieważ będą wykonywane *przy każdym* uruchomieniu menedżera kolejek. Zazwyczaj oznacza to dodanie parametru REPLACE w dowolnych komendach DEFINE i dodanie parametru IGNSTATE (YES) do komend START lub STOP.

b) Zastosuj BuildConfig do serwera.

```
oc apply -f mq-build-config.yaml
```

3. Rozpoczęcie budowania w celu utworzenia obrazu

a) Rozpoczęcie budowania

```
oc start-build mymq
```

Powinny pojawić się wyniki podobne do poniższych:

```
build.build.openshift.io/mymq-1 started
```

b) Sprawdzanie statusu budowania

Na przykład można uruchomić następującą komendę, korzystając z identyfikatora budowania zwróconego w poprzednim kroku:

```
oc describe build mymq-1
```

4. Wdrażanie menedżera kolejek przy użyciu nowego obrazu

Postępuj zgodnie z procedurą opisaną w sekcji [“Wdrażanie menedżera kolejek w klastrze platformy kontenerowej Red Hat OpenShift”](#) na stronie 84, dodając nowy obraz niestandardowy do kodu YAML.

Można dodać następujący fragment kodu YAML do normalnego kodu YAML QueueManager, gdzie *my-namespace* to projekt / przestrzeń nazw Red Hat OpenShift użytkownika, a *image* to nazwa utworzonego wcześniej obrazu (na przykład "mymq:latest-amd64"):

```
spec:
  queueManager:
    image: image-registry.openshift-image-registry.svc:5000/my-namespace/my-image
```

### Zadania pokrewne

[“Wdrażanie menedżera kolejek w klastrze platformy kontenerowej Red Hat OpenShift”](#) na stronie 84  
Użyj zasobu niestandardowego QueueManager, aby wdrożyć menedżer kolejek na klastrze platformy Red Hat OpenShift Container Platform.

## Dodawanie niestandardowych adnotacji i etykiet do zasobów menedżera kolejek

Można dodawać niestandardowe adnotacje i etykiety do metadanych menedżera kolejek.

### O tym zadaniu

Niestandardowe adnotacje i etykiety są dodawane do wszystkich zasobów z wyjątkiem rezerwacji PVC. Jeśli adnotacja niestandardowa lub etykieta jest zgodna z istniejącym kluczem, obowiązuje wartość ustawiona przez parametr IBM MQ Operator.

### Procedura

- Dodaj niestandardowe adnotacje.

Aby dodać niestandardowe adnotacje do zasobów menedżera kolejek, w tym do zasobnika, dodaj adnotacje do składnika metadana. Na przykład:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    annotationKey: "value"
```

- Dodaj niestandardowe etykiety.

Aby dodać etykiety niestandardowe do zasobów menedżera kolejek, w tym do zasobnika, dodaj etykiety w sekcji metadata. Na przykład:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  labels:
    labelKey: "value"
```

## OpenShift > CP4I Wyłączanie kontroli elementu webhook w czasie wykonywania

Kontrole elementu webhook w czasie wykonywania zapewniają, że klasy pamięci masowej są użyteczne dla menedżera kolejek. Można wyłączyć te kontrole, aby zwiększyć wydajność, lub jeśli nie są one poprawne w danym środowisku.

### O tym zadaniu

Kontrole elementów webhook w czasie wykonywania odbywają się w ramach konfiguracji menedżera kolejek. Sprawdzają one, czy klasy pamięci masowej są odpowiednie dla wybranego typu menedżera kolejek.

Można wyłączyć te kontrole, aby zmniejszyć czas potrzebny na utworzenie menedżera kolejek lub jeśli kontrole nie są poprawne w konkretnym środowisku.

**Uwaga:** Po wyłączeniu kontroli elementu webhook w czasie wykonywania wszystkie wartości klas pamięci masowej są dozwolone. Może to spowodować uszkodzenie menedżera kolejek.

Kontrole w czasie wykonywania zostały wprowadzone w wersji IBM MQ Operator 1.2.

### Procedura

- Wyłącz kontrole elementu webhook w czasie wykonywania

Dodaj następującą adnotację w sekcji metadata. Na przykład:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    "com.ibm.cp4i/disable-webhook-runtime-checks" : "true"
```

## OpenShift > CP4I Użytkowanie produktu IBM MQ przy wykorzystaniu operatora IBM MQ Operator

### O tym zadaniu

### Procedura

- [“Przygotowywanie projektu Red Hat OpenShift dla produktu IBM MQ”](#) na stronie 82.
- [“Wdrażanie menedżera kolejek w klastrze platformy kontenerowej Red Hat OpenShift”](#) na stronie 84.

## OpenShift > CP4I Nawiązywanie połączenia z rozwiązaniem IBM MQ Console wdrożonym w klastrze Red Hat OpenShift

W jaki sposób nawiązać połączenie z konsolą IBM MQ Console menedżera kolejek, która została wdrożona w klastrze Red Hat OpenShift Container Platform.

## O tym zadaniu

Adres URL IBM MQ Console można znaleźć na stronie szczegółów QueueManager konsoli WWW produktu Red Hat OpenShift lub w IBM Cloud Pak for Integration Platform Navigator. Alternatywnie można go znaleźć w interfejsie wiersza komend produktu Red Hat OpenShift, uruchamiając następującą komendę:

```
oc get queuemanager <QueueManager Name> -n <namespace of your MQ deployment> --output jsonpath='{.status.adminUiUrl}'
```

Jeśli używana jest licencja IBM Cloud Pak for Integration, konsola WWW IBM MQ jest skonfigurowana pod kątem używania komponentu IBM Cloud Pak Identity and Access Manager (IAM) Komponent IAM mógł już zostać skonfigurowany przez administratora klastra. Jeśli jednak jest to pierwszy raz, gdy komponent IAM jest używany w klastrze Red Hat OpenShift, konieczne jest pobranie początkowego hasła administratora. Więcej informacji na ten temat zawiera sekcja [Uzyskiwanie początkowego hasła administratora](#).

Jeśli używana jest licencja IBM MQ, konsola WWW produktu MQ nie jest wstępnie skonfigurowana, a użytkownik musi ją skonfigurować samodzielnie. Więcej informacji zawiera sekcja [Konfigurowanie użytkowników i ról](#).

### Zadania pokrewne

“Konfigurowanie trasy w celu nawiązania połączenia z menedżerem kolejek spoza klastra produktu Red Hat OpenShift” na stronie 112

Do połączenia aplikacji z menedżerem kolejek IBM MQ spoza klastra Red Hat OpenShift potrzebna jest trasa Red Hat OpenShift . Należy włączyć protokół TLS w menedżerze kolejek i aplikacji klienckiej IBM MQ , ponieważ protokół SNI jest dostępny w protokole TLS tylko wtedy, gdy używany jest protokół TLS w wersji 1.2 lub nowszej. Red Hat OpenShift Container Platform Router używa wskaźnika SNI do kierowania żądań do menedżera kolejek IBM MQ.

### **Nadawanie uprawnień dla partycji IBM MQ Console za pomocą programu IBM Cloud Pak IAM**

Uprawnienia dla partycji IBM MQ Console są zarządzane za pomocą programu IBM Cloud Pak Administration Hub, a nie serwera IBM Cloud Pak for Integration Platform Navigator. Produkt IBM MQ nie korzysta z uprawnień "Automation" (Automatyzacja) udostępnianych przez produkt IBM Cloud Pak for Integration, ale korzysta z podstawowych uprawnień, które są włączone przez produkt IBM Cloud Pak Identity and Access Manager (IAM).

### Procedura

1. Otwórz konsolę administracyjną serwera IBM Cloud Pak .  
W IBM Cloud Pak for Integration Platform UI kliknij przełącznik Cloud Pak (ikona 9-dot) znajdujący się w prawym górnym rogu paska narzędzi, a następnie kliknij panel **IBM Cloud Pak Administration** (Administrowanie produktem IBM Cloud).
2. W menu nawigacyjnym w lewym górnym rogu wybierz opcję **Tożsamość i dostęp**, a następnie wybierz opcję **Zespoły i identyfikatory usług**.
3. Utwórz zespół, a następnie dodaj do niego użytkowników.
  - a) Wybierz opcję **Utwórz zespół**.
  - b) Wprowadź nazwę zespołu, a następnie wybierz domenę zabezpieczeń dla użytkowników, którym chcesz zarządzać.
  - c) Wyszukiwanie użytkowników.  
Ci użytkownicy muszą już istnieć w dostawcy tożsamości.
  - d) Po znalezieniu każdego użytkownika należy nadać im rolę. Musi to być "Administrator" lub "Administrator klastrów", aby administrować produktem IBM MQ przy użyciu IBM MQ Console.
4. Dodaj każdego użytkownika do przestrzeni nazw.
  - a) Wybierz zespół, aby go edytować.

b) Wybierz opcję **Zasoby > Zarządzaj zasobami**.

c) Wybierz przestrzenie nazw, którym ten zespół ma administrować. Mogą to być przestrzenie nazw z menedżerem kolejek.

## OpenShift CP4I Monitorowanie w przypadku korzystania z operatora IBM

### MQ Operator

Menedżery kolejek zarządzane przez IBM MQ Operator mogą generować metryki kompatybilne z rozwiązaniem Prometheus.

Charakterystyki te można wyświetlać przy użyciu stosu monitorowania Red Hat OpenShift Container Platform (OCP). Otwórz kartę **Pomiary** w produkcie OCP, a następnie kliknij opcję **Obserwuj > Metryki**. Pomiary menedżera kolejek są domyślnie włączone, ale można je wyłączyć, ustawiając parametr `.spec.metrics.enabled` na wartość `false`.

Prometheus jest bazą danych szeregów czasowych i mechanizmem oceny reguł dotyczących metryk. Kontenery IBM MQ eksponują punkt końcowy metryk, który może być odpytywany przez narzędzie Prometheus. Metryki generowane na podstawie tematów systemu MQ na potrzeby monitorowania i śledzenia aktywności.

Red Hat OpenShift Container Platform zawiera wstępnie skonfigurowany, wstępnie zainstalowany i samoaktualizujący się stos monitorowania wykorzystujący serwer Prometheus. Stos monitorowania Red Hat OpenShift Container Platform musi być skonfigurowany do monitorowania projektów zdefiniowanych przez użytkownika. Więcej informacji na ten temat zawiera sekcja [Włączanie monitorowania projektów zdefiniowanych przez użytkownika](#). IBM MQ Operator tworzy `ServiceMonitor` podczas tworzenia `QueueManager` z włączonymi metrykami, które element wykonawczy programu Prometheus może następnie wykryć.

W starszych wersjach produktu IBM Cloud Pak for Integration można także użyć usługi [IBM Cloud Platform Monitoring](#), aby udostępnić serwer Prometheus.

## OpenShift CP4I Metryki publikowane, gdy używany jest IBM MQ Operator

Kontenery menedżera kolejek mogą publikować metryki zgodne z programem Red Hat OpenShift Monitoring.

Pomiar	Typ	Opis
<code>ibmmq_qmgr_commit_total</code>	counter	Liczba zatwierdzeń
<code>ibmmq_qmgr_cpu_load_fifteen_minute_average_percentage</code>	gauge	Obciążenie procesora - średnia dla piętnastu minut
<code>ibmmq_qmgr_cpu_load_five_minute_average_percentage</code>	gauge	Obciążenie procesora - średnia dla pięciu minut
<code>ibmmq_qmgr_cpu_load_one_minute_average_percentage</code>	gauge	Obciążenie procesora - średnia dla jednej minuty
<code>ibmmq_qmgr_destructive_get_bytes_total</code>	counter	Łączna liczba bajtów niszczących operacji get w okresie
<code>ibmmq_qmgr_destructive_get_total</code>	counter	Łączna liczba niszczących operacji get w okresie



<b>Pomiar</b>	<b>Typ</b>	<b>Opis</b>
ibmmq_qmgr_durable_subscription_alter_total	counter	Liczba operacji zmiany trwałej subskrypcji
ibmmq_qmgr_durable_subscription_create_total	counter	Liczba operacji tworzenia trwałej subskrypcji
ibmmq_qmgr_durable_subscription_delete_total	counter	Liczba operacji usuwania trwałej subskrypcji
ibmmq_qmgr_durable_subscription_resume_total	counter	Liczba operacji wznowienia trwałej subskrypcji
ibmmq_qmgr_errors_file_system_free_space_percentage	gauge	System plików błędów MQ - wolne miejsce
ibmmq_qmgr_errors_file_system_in_use_bytes	gauge	System plików błędów MQ - bajty w użyciu
ibmmq_qmgr_expired_message_total	counter	Liczba komunikatów, które utraciły ważność
ibmmq_qmgr_failed_browse_total	counter	Liczba operacji przeglądania zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqcb_total	counter	Liczba operacji MQCB zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqclose_total	counter	Liczba operacji MQCLOSE zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqconn_mqconnx_total	counter	Liczba operacji MQCONN lub MQCONNX zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqget_total	counter	Liczba operacji MQGET zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqinq_total	counter	Liczba operacji MQINQ zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqopen_total	counter	Liczba operacji MQOPEN zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqput1_total	counter	Liczba operacji MQPUT1 zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqput_total	counter	Liczba operacji MQPUT zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqset_total	counter	Liczba operacji MQSET zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqsubrq_total	counter	Liczba operacji MQSUBRQ zakończonych niepowodzeniem

Pomiar	Typ	Opis
ibmmq_qmgr_failed_subscription_create_alter_resume_total	counter	Liczba operacji tworzenia/zmiany/wznowienia subskrypcji zakończonych niepowodzeniem
ibmmq_qmgr_failed_subscription_delete_total	counter	Liczba operacji usuwania subskrypcji zakończonych niepowodzeniem
ibmmq_qmgr_failed_topic_mqput_mqput1_total	counter	Liczba operacji MQPUT lub MQPUT1 dla tematów zakończonych niepowodzeniem
ibmmq_qmgr_fdc_files	gauge	Liczba plików FDC produktu MQ
ibmmq_qmgr_log_file_system_in_use_bytes	gauge	System plików dzienników - bajty w użyciu
ibmmq_qmgr_log_file_system_max_bytes	gauge	System plików dzienników - maksymalna liczba bajtów
ibmmq_qmgr_log_in_use_bytes	gauge	Dziennik - bajty w użyciu
ibmmq_qmgr_log_logical_written_bytes_total	counter	Dziennik - liczba zapisanych bajtów logicznych
ibmmq_qmgr_log_max_bytes	gauge	Dziennik - maksymalna liczba bajtów
ibmmq_qmgr_log_physical_written_bytes_total	counter	Dziennik - liczba zapisanych bajtów fizycznych
ibmmq_qmgr_log_primary_space_in_use_percentage	gauge	Dziennik – bieżące wykorzystanie przestrzeni podstawowej
ibmmq_qmgr_log workload_primary_space_utilization_percentage	gauge	Dziennik – obciążeniowe wykorzystanie przestrzeni podstawowej
ibmmq_qmgr_log_write_latency_seconds	gauge	Dziennik - opóźnienie zapisu
ibmmq_qmgr_log_write_size_bytes	gauge	Dziennik - wielkość zapisu
ibmmq_qmgr_mqcb_total	counter	Liczba operacji MQCB
ibmmq_qmgr_mqclose_total	counter	Liczba operacji MQCLOSE
ibmmq_qmgr_mqconn_mqconnx_total	counter	Liczba operacji MQCONN lub MQCONNX

<b>Pomiar</b>	<b>Typ</b>	<b>Opis</b>
ibmmq_qmgr_mqctl_total	counter	Liczba operacji MQCTL
ibmmq_qmgr_mqdisc_total	counter	Liczba operacji MQDISC
ibmmq_qmgr_mqinq_total	counter	Liczba operacji MQINQ
ibmmq_qmgr_mqopen_total	counter	Liczba operacji MQOPEN
ibmmq_qmgr_mqput_mqput1_bytes_total	counter	Łączna liczba bajtów operacji MQPUT lub MQPUT1 w okresie
ibmmq_qmgr_mqput_mqput1_total	counter	Łączna liczba operacji MQPUT lub MQPUT1 w okresie
ibmmq_qmgr_mqset_total	counter	Liczba operacji MQSET
ibmmq_qmgr_mqstat_total	counter	Liczba operacji MQSTAT
ibmmq_qmgr_mqsubrq_total	counter	Liczba operacji MQSUBRQ
ibmmq_qmgr_non_durable_subscription_create_total	counter	Liczba operacji tworzenia nietrwalej subskrypcji
ibmmq_qmgr_non_durable_subscription_delete_total	counter	Liczba operacji usuwania nietrwalej subskrypcji
ibmmq_qmgr_non_persistent_message_browse_bytes_total	counter	Przeglądanie komunikatów nietrwących - liczba bajtów
ibmmq_qmgr_non_persistent_message_browse_total	counter	Przeglądanie komunikatów nietrwących - liczba
ibmmq_qmgr_non_persistent_message_destructive_get_total	counter	Liczba niszczących operacji get dla komunikatów nietrwących
ibmmq_qmgr_non_persistent_message_get_bytes_total	counter	Otrzymane komunikaty nietrwące - liczba bajtów
ibmmq_qmgr_non_persistent_message_mqput1_total	counter	Liczba operacji MQPUT1 dla komunikatów nietrwących
ibmmq_qmgr_non_persistent_message_mqput_total	counter	Liczba operacji MQPUT dla komunikatów nietrwących

Pomiar	Typ	Opis
ibmmq_qmgr_non_persistent_message_put_bytes_total	counter	Komunikaty nietrwałe operacji put - liczba bajtów
ibmmq_qmgr_non_persistent_topic_mqput_mqput1_total	counter	Nietrwałe - liczba operacji MQPUT lub MQPUT1 dla tematów
ibmmq_qmgr_persistent_message_browse_bytes_total	counter	Przeglądanie komunikatów trwałych - liczba bajtów
ibmmq_qmgr_persistent_message_browse_total	counter	Przeglądanie komunikatów trwałych - liczba
ibmmq_qmgr_persistent_message_destructive_get_total	counter	Liczba niszczących operacji get dla komunikatów trwałych
ibmmq_qmgr_persistent_message_get_bytes_total	counter	Otrzymane komunikaty trwałe - liczba bajtów
ibmmq_qmgr_persistent_message_mqput1_total	counter	Liczba operacji MQPUT1 dla komunikatów trwałych
ibmmq_qmgr_persistent_message_mqput_total	counter	Liczba operacji MQPUT dla komunikatów trwałych
ibmmq_qmgr_persistent_message_put_bytes_total	counter	Komunikaty trwałe operacji put - liczba bajtów
ibmmq_qmgr_persistent_topic_mqput_mqput1_total	counter	Trwałe - liczba operacji MQPUT lub MQPUT1 dla tematów
ibmmq_qmgr_published_to_subscribers_bytes_total	counter	Liczba bajtów opublikowanych do subskrybentów
ibmmq_qmgr_published_to_subscribers_message_total	counter	Liczba komunikatów opublikowanych do subskrybentów
ibmmq_qmgr_purged_queue_total	counter	Liczba wyczyszczonych kolejek
ibmmq_qmgr_queue_manager_file_system_free_space_percentage	gauge	System plików menedżera kolejek - wolne miejsce
ibmmq_qmgr_queue_manager_file_system_in_use_bytes	gauge	System plików menedżera kolejek - bajty w użyciu
ibmmq_qmgr_ram_free_percentage	gauge	Procent wolnej pamięci RAM

Pomiar	Typ	Opis
ibmmq_qmgr_ram_usage_estimate_for_queue_manager_bytes	gauge	Łączna liczba bajtów pamięci RAM - szacunkowa wartość dla menedżera kolejek
ibmmq_qmgr_rollback_total	counter	Liczba wycofań zmian
ibmmq_qmgr_system_cpu_time_estimate_for_queue_manager_percentage	gauge	Czas pracy procesora dla systemu - szacunkowy procent dla menedżera kolejek
ibmmq_qmgr_system_cpu_time_percentage	gauge	Procent czasu pracy procesora dla systemu
ibmmq_qmgr_topic_mqput_mqput1_total	counter	Łączna liczba operacji MQPUT lub MQPUT1 dla tematów w okresie
ibmmq_qmgr_topic_mqput_bytes_total	counter	Łączna liczba bajtów operacji put w okresie
ibmmq_qmgr_trace_file_system_free_space_percentage	gauge	System plików śledzenia MQ - wolne miejsce
ibmmq_qmgr_trace_file_system_in_use_bytes	gauge	System plików śledzenia MQ - bajty w użyciu
ibmmq_qmgr_user_cpu_time_estimate_for_queue_manager_percentage	gauge	Czas pracy procesora dla użytkownika - szacunkowy procent dla menedżera kolejek
ibmmq_qmgr_user_cpu_time_percentage	gauge	Procent czasu pracy procesora dla użytkownika

CP4I

V 9.2.2

CD

## Wyświetlanie statusu rodzimych menedżerów kolejek HA dla kontenerów certyfikowanych IBM MQ

W przypadku kontenerów certyfikowanych przez produkt IBM MQ można wyświetlić status rodzimych instancji HA, uruchamiając komendę **dspmq** w jednym z działających Pods.

### O tym zadaniu

#### Ważne:

Aby wyświetlić status działania instancji menedżera kolejek, można użyć komendy **dspmq** w jednym z uruchomionych zasobników. Zwrócone informacje zależą od tego, czy instancja jest aktywna, czy jest repliką. Informacje dostarczone przez aktywną instancję są definitywne, a informacje z węzłów replik mogą być nieaktualne.

Można wykonać następujące czynności:

- Sprawdź, czy instancja menedżera kolejek w bieżącym węźle jest aktywna, czy jest repliką.
- Wyświetl status działania rodzimej wysokiej dostępności dla instancji w bieżącym węźle.
- Wyświetl status działania wszystkich trzech instancji w konfiguracji rodzimej wysokiej dostępności.

Następujące pola statusu są używane do raportowania statusu konfiguracji rodzimej wysokiej dostępności:

#### **ROLE**

Określa bieżącą rolę instancji i jest jedną z następujących wartości: Active, Replica lub Unknown.

#### **INSTANCE**

Nazwa podana dla tej instancji menedżera kolejek, gdy była ona tworzona przy użyciu opcji **-lr** komendy **crtmqm**.

#### **INSYNC**

Wskazuje, czy w razie potrzeby instancja jest w stanie przejąć rolę aktywnej instancji.

#### **QUORUM**

Raportuje status kworum w formacie *liczba zsynchronizowanych instancji/liczba skonfigurowanych instancji*.

#### **REPLADDR**

Adres replikacji instancji menedżera kolejek.

#### **CONNECTV**

Wskazuje, czy węzeł jest połączony z aktywną instancją.

#### **BACKLOG**

Określa zaległość instancji w kB.

#### **CONNINST**

Informuje, czy instancja o podanej nazwie jest połączona z tą instancją.

#### **ALTDATA**

Wskazuje datę ostatniej aktualizacji tych informacji (wartość pusta, jeśli nigdy nie zostały zaktualizowane).

#### **ALTTIME**

Wskazuje godzinę ostatniej aktualizacji tych informacji (wartość pusta, jeśli nigdy nie zostały zaktualizowane).

## **Procedura**

- Znajdź pods, które są częścią menedżera kolejek.

```
oc get pod --selector app.kubernetes.io/instance=nativeha-qm
```

- Uruchom dspmq w jednym z zasobników

```
oc exec -t Pod dspmq
```

```
oc rsh Pod
```

dla interaktywnej powłoki, w której można bezpośrednio uruchomić produkt dspmq.

- Aby określić, czy instancja menedżera kolejek działa jako aktywna instancja, czy jako replika:

```
oc exec -t Pod dspmq -o status -m QMgrName
```

Aktywna instancja menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          STATUS(Running)
```

Instancja-replika menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          STATUS(Replica)
```

Nieaktywna instancja zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          STATUS(Ended Immediately)
```

- Aby sprawdzić status działania rodzimej wysokiej dostępności instancji w podanym zasobniku:

```
oc exec -t Pod dspmq -o nativeha -m QMgrName
```

Aktywna instancja menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

Instancja-replika menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```

Nieaktywna menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- Aby określić status działania rodzimej wysokiej dostępności dla wszystkich instancji w konfiguracji o rodzimej wysokiej dostępności:

```
oc exec -t Pod dspmq -o nativeha -x -m QMgrName
```

Jeśli ta komenda zostanie wydana w węźle, w którym działa aktywna instancja menedżera kolejek BOB, może zostać wyświetlony następujący status:

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

Jeśli ta komenda zostanie wydana w węźle, w którym uruchomiona jest instancja-replika menedżera kolejek BOB, może zostać wyświetlony następujący status, który wskazuje, że jedna z replik ma zaległości:

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No) BACKLOG(435)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

Jeśli ta komenda zostanie wydana w węźle, w którym uruchomiona jest nieaktywna instancja menedżera kolejek BOB, może zostać wyświetlony następujący status:

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
```

W przypadku wydania komendy w momencie, gdy instancje nadal negocjują, która będzie aktywna, a które będą replikami, zwrócony zostanie następujący status:

```
QMNAME(BOB)          STATUS(Negotiating)
```

## Odsyłacze pokrewne

[Komenda dspmq \(wyświetl menedżery kolejek\)](#)

“Przykład: Konfigurowanie menedżera kolejek o rodzimej wysokiej dostępności” na stronie 97

W tym przykładzie przedstawiono sposób wdrażania menedżera kolejek przy użyciu rodzimej funkcji wysokiej dostępności w produkcie Red Hat OpenShift Container Platform (OCP) przy użyciu operatora IBM MQ Operator.

## Tworzenie i odtwarzanie kopii zapasowej konfiguracji menedżera kolejek przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift

Utworzenie kopii zapasowej konfiguracji menedżera kolejek może pomóc w odbudowaniu menedżera na podstawie jego definicji, jeśli konfiguracja zostanie utracona. Ta procedura nie powoduje utworzenia kopii zapasowej danych dziennika menedżera kolejek. Ze względu na przejściowy charakter komunikatów dane w dzienniku historycznym mogą być nieistotne w momencie odtwarzania.

### Zanim rozpoczniesz

Zaloguj się do klastra, używając komendy **cloudctl login** (w przypadku IBM Cloud Pak for Integration) lub **oc login**.

### Procedura

- Utwórz kopię zapasową konfiguracji menedżera kolejek.

Do zrzucenia konfiguracji menedżera kolejek produktu IBM MQ można użyć komendy **dmpmqcfig**.

- Uzyskaj nazwę zasobnika dla menedżera kolejek.

Na przykład można uruchomić następującą komendę, gdzie *nazwa\_menedzera\_kolejek* jest nazwą zasobu QueueManager:

```
oc get pods --selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/instance=queue_manager_name
```

- Uruchom komendę **dmpmqcfig** w zasobniku, kierując wyniki do pliku na komputerze lokalnym.

**dmpmqcfig** generuje na wyjściu konfigurację MQSC menedżera kolejek.

```
oc exec -it pod_name -- dmpmqcfig > backup.mqsc
```

- Odtwórz konfigurację menedżera kolejek.

Po wykonaniu procedury tworzenia kopii zapasowej opisanej w poprzednim kroku powinien być dostępny plik `backup.mqsc` zawierający konfigurację menedżera kolejek. Konfigurację tę można odtworzyć, stosując ten plik do nowego menedżera kolejek.

- Uzyskaj nazwę zasobnika dla menedżera kolejek.

Na przykład można uruchomić następującą komendę, gdzie *nazwa\_menedzera\_kolejek* jest nazwą zasobu QueueManager:

```
oc get pods --selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/instance=queue_manager_name
```

- Uruchom komendę **runmqsc** w zasobniku, kierując do niej zawartość pliku `backup.mqsc`.

```
oc exec -i pod_name -- runmqsc < backup.mqsc
```

## Rozwiązywanie problemów związanych z IBM MQ

### Operator

W przypadku wystąpienia problemów związanych z programem IBM MQ Operator należy użyć opisanych technik w celu znalezienia i rozwiązania tych problemów.

### Procedura

- [“Rozwiązywanie problemów: uzyskiwanie dostępu do danych menedżera kolejek”](#) na stronie 129



## Rozwiązywanie problemów: uzyskiwanie dostępu do danych menedżera kolejek

Użyj narzędzia sprawdzającego PVC, aby uzyskać dostęp do plików w PVC menedżera kolejek, w którym nie można ustanowić zdalnej powłoki dla zasobnika menedżera kolejek. Może to być spowodowane tym, że zasobnik jest w stanie **Error** lub **CrashLoopBackOff**. To narzędzie jest przeznaczone do użycia z menedżerami kolejek wdrożonymi przez program IBM MQ Operator.

### Zanim rozpoczniesz

Aby użyć narzędzia sprawdzającego PVC. Użytkownik musi mieć dostęp do przestrzeni nazw menedżera kolejek.

### O tym zadaniu

Aby ułatwić rozwiązywanie problemów, można uzyskać dostęp do danych przechowywanych w PVC (Persistent Volume Claims) powiązanych z danym menedżerem kolejek. W tym celu należy użyć narzędzia do podłączenia PVC do zestawu zasobników inspektora. Następnie można umieścić zdalną powłokę w dowolnym zasobnikach inspektora, aby odczytać pliki.

W zależności od typu wdrożenia tworzone są od jednego do trzech zasobników inspektora. Woluminy specyficzne dla danego zasobnika rodzimego menedżera kolejek wysokiej dostępności lub menedżera kolejek z wieloma instancjami są dostępne w powiązonym zasobniku kontrolera PVC. Współużytkowane woluminy są dostępne dla wszystkich inspektorów. Nazwa zasobnika inspektora zawiera nazwę powiązanego zasobnika menedżera kolejek.

### Procedura

1. Pobierz narzędzie inspektora PVC MQ .

Narzędzie jest dostępne pod adresem: <https://github.com/ibm-messaging/mq-pvc-tool>.

2. Upewnij się, że jesteś zalogowany w klastrze.
3. Znajdź nazwę menedżera kolejek i przestrzeń nazw, w której działa menedżer kolejek.
4. Uruchom narzędzie Inspector dla menedżera kolejek.
  - a) Uruchom następującą komendę, podając nazwę menedżera kolejek i jego nazwę przestrzeni nazw.

```
./pvc-tool.sh queue_manager_name queue_manager_namespace_name
```

- b) Po zakończeniu działania narzędzia uruchom następującą komendę, aby wyświetlić tworzone zasobniki inspektora.

```
oc get pods
```

5. Wyświetl pliki podłączone do zasobnika inspektora.

- a) Każdy zasobnik inspektora PVC jest powiązany z zasobnikiem menedżera kolejek, dlatego może istnieć wiele zasobników inspektora. Uzyskaj dostęp do jednego z tych zasobników, uruchamiając następującą komendę:

```
oc ish pvc-inspector-pod-name
```

Zostanie umieszczony w katalogu zawierającym podłączone katalogi PVC.

- b) Otwórz zdalną powłokę w zasobniku, uruchamiając następującą komendę:

```
ls
```

- c) Można wyświetlić katalogi o takiej samej nazwie jak podłączone kanały PVC. Aby uzyskać dostęp do plików w PVC menedżera kolejek, należy przejrzeć te katalogi. Aby wyświetlić listę kanałów PVC, uruchom następującą komendę poza sesją zdalnej powłoki:

```
oc get pvc
```

d) Wyczyść zasobniki utworzone przez narzędzie, uruchamiając następującą komendę:

```
'oc delete pods -l tool=mq-pvc-inspector
```

## OpenShift CP4I Skorowidz interfejsu API operatora IBM MQ Operator

Produkt IBM MQ udostępnia operatora Kubernetes, który zapewnia rodzimą integrację z produktem Red Hat OpenShift Container Platform.

## OpenShift CP4I Skorowidz interfejsu API [mq.ibm.com/v1beta1](https://mq.ibm.com/v1beta1)

Interfejs API v1beta1 może być używany do tworzenia zasobów menedżera kolejek i zarządzania nimi.

## OpenShift CP4I CD EUS Skorowidz informacji dotyczących licencjonowania [mq.ibm.com/v1beta1](https://mq.ibm.com/v1beta1)

### Bieżące wersje licencji

Pole `spec.license.license` musi zawierać identyfikator licencji dla akceptowanej licencji. Poprawne wartości są następujące:

Wartość <code>spec.license.license</code>	Wartość <code>spec.license.use</code>	Informacje o licencji	Odpowiednie wersje IBM MQ
L-RJON-C7QG3S	Production lub NonProduction	<a href="#">IBM Cloud Pak for Integration 2021.4.1</a>	9.2.4 lub 9.2.5
L-RJON-C7QFZX	Production lub NonProduction	<a href="#">IBM Cloud Pak for Integration Limited Edition 2021.4.1</a>	9.2.4 lub 9.2.5
L-RJON-C5CSNH	Production lub NonProduction	<a href="#">IBM Cloud Pak for Integration 2021.3.1</a>	9.2.3 lub 9.2.4
L-RJON-C5CSM2	Production lub NonProduction	<a href="#">IBM Cloud Pak for Integration Limited Edition 2021.3.1</a>	9.2.3 lub 9.2.4
L-RJON-BZFQU2	Production lub NonProduction	<a href="#">IBM Cloud Pak for Integration 2021.2.1</a>	9.2.3
L-RJON-BZFQSB	Production lub NonProduction	<a href="#">IBM Cloud Pak for Integration Limited Edition 2021.2.1</a>	9.2.3
L-RJON-BUVMQX	Production lub NonProduction	<a href="#">IBM Cloud Pak for Integration 2020.4.1</a>	9.2.0 EUS lub 9.2.1
L-RJON-BUVMYB	Production lub NonProduction	<a href="#">IBM Cloud Pak for Integration Limited Edition 2020.4.1</a>	9.2.0 EUS lub 9.2.1
L-APIG-BZDDDY	Production	<a href="#">IBM MQ Advanced i IBM MQ Advanced for Non-Production Environment 9.2 - 07/2021</a>	9.2.3, 9.2.4 lub 9.2.5
L-APIG-BYHCL7	Development	<a href="#">IBM MQ Advanced for Developers (Non-Warranted) 9.2 - 07/2021</a>	9.2.3, 9.2.4 lub 9.2.5
L-APIG-BVJJB3	Production	<a href="#">IBM MQ Advanced i IBM MQ Advanced for Non-Production Environment 9.2 - 03/2021</a>	9.2.2

Wartość <code>spec.license.license</code>	Wartość <code>spec.license.use</code>	Informacje o licencji	Odpowiednie wersje IBM MQ
L-APIG-BMJJBM	Production	<a href="#">IBM MQ Advanced V9.2</a>	9.2.0 CD lub 9.2.1
L-APIG-BMKG5H	Development	<a href="#">IBM MQ Advanced for Developers (Niegwarantowany) V9.2</a>	9.2.0 CD, 9.2.1 lub 9.2.2

Należy zauważyć, że określana jest wersja (*version*) licencji, która nie zawsze jest identyczna z wersją produktu IBM MQ.

## Starsze wersje licencji

Pole `spec.license.license` musi zawierać identyfikator licencji dla akceptowanej licencji. Poprawne wartości są następujące:

Wartość <code>spec.license.license</code>	Wartość <code>spec.license.use</code>	Informacje o licencji	Odpowiednie wersje IBM MQ
L-RJON-BXUPZ2	Production lub NonProduction	<a href="#">IBM Cloud Pak for Integration 2021.1.1</a>	9.2.2
L-RJON-BXUQ34	Production lub NonProduction	<a href="#">IBM Cloud Pak for Integration Limited Edition 2021.1.1</a>	9.2.2
L-RJON-BYRMYW	NonProduction	<a href="#">IBM Cloud Pak for Integration Eval-Demo 2021.1.1</a> . Wczesne wydanie tylko do użycia z rodzimą wysoką dostępnością z IBM MQ Operator 1.5.	9.2.2
L-RJON-BQPGWD	Production lub NonProduction	<a href="#">IBM Cloud Pak for Integration 2020.3.1</a>	9.2.0 CD
L-RJON-BN7PN3	Production lub NonProduction	<a href="#">IBM Cloud Pak for Integration 2020.2.1</a>	9.1.5 lub 9.2.0 CD
L-RJON-BPHL2Y	Production lub NonProduction	<a href="#">IBM Cloud Pak for Integration Limited Edition 2020.2.1</a>	9.1.5
L-APIG-BJAKBF	Production	<a href="#">IBM MQ Advanced V9.1 -04/2020</a>	9.1.5
L-APIG-BM7GDH	Development	<a href="#">IBM MQ Advanced for Developers (Niegwarantowany) V9.1 -04/2020</a>	9.1.5

Należy zauważyć, że określana jest wersja (*version*) licencji, która nie zawsze jest identyczna z wersją produktu IBM MQ.

  **Skorowidz interfejsu API menedżera kolejek ([mq.ibm.com/v1beta1](https://mq.ibm.com/v1beta1))**

## QueueManager

Menedżer kolejek (QueueManager) jest serwerem IBM MQ, który udostępnia aplikacjom usługi kolejkowania i publikowania/subskrypcji.

Pole	Opis
apiVersion string	APIVersion definiuje wersjonowany schemat tej reprezentacji obiektu. Serwery powinny przekształcać rozpoznane schematy w najnowsze wartości wewnętrzne i mogą odrzucać nierozpoznane wartości. Więcej informacji: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a> .
kind string	Kind to wartość łańcuchowa reprezentująca zasób REST reprezentowany przez ten obiekt. Serwery mogą wywodzić go z punktu końcowego, do którego klient wysyła żądania. Nie można go zmienić. MieszanaWielkośćLiter. Więcej informacji: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a> .
metadata	
spec QueueManagerSpec	Pożądaný stan menedżera kolejek.
status QueueManagerStatus	Obserwowany stan menedżera kolejek.

### .spec

Pożądaný stan menedżera kolejek.

Występuje w:

- [“QueueManager” na stronie 131](#)

Pole	Opis
affinity	Standardowe reguły powinowactwa Kubernetes. Aby uzyskać więcej informacji, patrz <a href="https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#affinity-v1-core">https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#affinity-v1-core</a> .
annotations <a href="#">Adnotacje</a>	Pole annotations służy do przekazywania adnotacji zasobnika. Użytkownik może dodać dowolną adnotację do tego pola, a zostanie ona zastosowana do zasobnika. Ewentualne adnotacje określone w tym miejscu zastępują domyślne adnotacje. Wymaga operatora MQ w wersji 1.3.0 lub nowszej.
imagePullSecrets <a href="#">LocalObjectReference</a> array	Opcjonalna lista odwołań do danych tajnych w tej samej przestrzeni nazw, które mają być używane do pobierania obrazów używanych przez ten menedżer kolejek. Te dane tajne, jeśli zostaną określone, będą przekazywane do indywidualnych implementacji programów pobierających. Na przykład w przypadku środowiska Docker honorowane są dane tajne typu DockerConfig. Więcej informacji: <a href="https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod">https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod</a> .
labels <a href="#">Etykiety</a>	Pole labels służy do przekazywania etykiet zasobnika. Użytkownik może dodać dowolną etykietę do tego pola, a zostanie ona zastosowana do zasobnika. Ewentualne etykiety określone w tym miejscu zastępują domyślne etykiety. Wymaga operatora MQ w wersji 1.3.0 lub nowszej.
license <a href="#">License</a>	Ustawienia sterujące akceptacją licencji i wyborem metryk licencji, które mają być używane.
pki <a href="#">PKI</a>	Ustawienia infrastruktury klucza publicznego, służące do definiowania kluczy i certyfikatów, które mają być używane z protokołem TLS (Transport Layer Security) lub produktem MQ Advanced Message Security (AMS).
queueManager <a href="#">QueueManagerConfig</a>	Ustawienia dla kontenera menedżera kolejek i bazowego menedżera kolejek.

Pole	Opis
securityContext <a href="#">SecurityContext</a>	Ustawienia zabezpieczeń, które mają zostać dodane do elementu securityContext zasobnika menedżera kolejek.
template <a href="#">Template</a>	Zaawansowane użycie szablonów zasobów Kubernetes. Szablon umożliwia użytkownikom zmianę sposobu generowania przez IBM MQ bazowych zasobów Kubernetes, takich jak zestawy stanowe (StatefulSet), zasobniki (Pod) i usługi (Service). Ta możliwość przeznaczona jest tylko dla zaawansowanych użytkowników, ponieważ jej niepoprawne wykorzystanie może zakłócić normalne działanie produktu MQ. Wszystkie wartości określone w menedżerze kolejek zostaną przestonięte przez odpowiednie ustawienia w szablonie.
terminationGracePeriod Seconds integer	Opcjonalny czas trwania (w sekundach), po którym zasobnik musi prawidłowo zakończyć działanie. Wartość musi być nieujemną liczbą całkowitą. Wartość zero oznacza natychmiastowe usunięcie. Docelowy czas, w którym podejmowane są próby zakończenia działania menedżera kolejek z kolejno eskalowanymi fazami odłączenia aplikacji. W razie potrzeby przerywane są zasadnicze zadania obsługi kontenera. Wartością domyślną jest 30 sekund.
tracing <a href="#">TracingConfig</a>	Ustawienia integracji śledzenia z panelem kontrolnym Cloud Pak for Integration Operations Dashboard.
version string	Ustawienie sterujące wersją produktu MQ, która będzie używana (wymagane). Na przykład: 9.1.5.0-r2 oznacza wersję produktu MQ 9.1.5.0, która korzysta z drugiej korekty obrazu kontenera. W korektach wprowadzane są często poprawki specyficzne dla kontenera, np. poprawki do obrazu podstawowego.
web <a href="#">WebServerConfig</a>	Ustawienia serwera WWW produktu MQ.

### **.spec.annotations**

Pole annotations służy do przekazywania adnotacji zasobnika. Użytkownik może dodać dowolną adnotację do tego pola, a zostanie ona zastosowana do zasobnika. Ewentualne adnotacje określone w tym miejscu zastępują domyślne adnotacje. Wymaga operatora MQ w wersji 1.3.0 lub nowszej.

Występuje w:

- [“.spec” na stronie 132](#)

### **.spec.imagePullSecrets**

LocalObjectReference zawiera informacje wystarczające do odszukania przywoływanego obiektu w tej samej przestrzeni nazw.

Występuje w:

- [“.spec” na stronie 132](#)

Pole	Opis
name string	Nazwa obiektu przywołującego. Więcej informacji: <a href="https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names">https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names</a> TODO: Add other useful fields. apiVersion, kind, uid? (Zadanie do wykonania: dodawanie innych przydatnych pól. apiVersion, kind, uid?)

## **.spec.labels**

Pole labels służy do przekazywania etykiet zasobnika. Użytkownik może dodać dowolną etykietę do tego pola, a zostanie ona zastosowana do zasobnika. Ewentualne etykiety określone w tym miejscu zastępują domyślne etykiety. Wymaga operatora MQ w wersji 1.3.0 lub nowszej.

Występuje w:

- [“.spec” na stronie 132](#)

## **.spec.license**

Ustawienia sterujące akceptacją licencji i wyborem metryk licencji, które mają być używane.

Występuje w:

- [“.spec” na stronie 132](#)

<b>Pole</b>	<b>Opis</b>
accept boolean	Określa, czy użytkownik akceptuje licencję powiązaną z tym oprogramowaniem (wymagane).
license string	Identyfikator akceptowanej licencji. Musi to być poprawny identyfikator licencji na używaną wersję produktu MQ. Poprawne wartości można znaleźć pod adresem <a href="http://ibm.biz/BdqvCF">http://ibm.biz/BdqvCF</a> .
metric string	Ustawienie określające, która metryka licencji ma być używana. Na przykład ProcessorValueUnit, VirtualProcessorCore lub ManagedVirtualServer. Wartością domyślną jest ProcessorValueUnit w przypadku korzystania z licencji na MQ i VirtualProcessorCore w przypadku korzystania z licencji na Cloud Pak for Integration.
use string	Ustawienie określające zastosowanie oprogramowania, gdy licencja dopuszcza więcej niż jedno zastosowanie. Poprawne wartości można znaleźć pod adresem <a href="http://ibm.biz/BdqvCF">http://ibm.biz/BdqvCF</a> .

## **.spec.pki**

Ustawienia infrastruktury klucza publicznego, służące do definiowania kluczy i certyfikatów, które mają być używane z protokołem TLS (Transport Layer Security) lub produktem MQ Advanced Message Security (AMS).

Występuje w:

- [“.spec” na stronie 132](#)

<b>Pole</b>	<b>Opis</b>
keys PKISource array	Klucze prywatne do dodania do repozytorium kluczy menedżera kolejek.
trust PKISource array	Certyfikaty do dodania do repozytorium kluczy menedżera kolejek.

## **.spec.pki.keys**

PKISource definiuje źródło informacji o infrastrukturze klucza publicznego, takich jak klucze lub certyfikaty.

Występuje w:

- [“.spec.pki” na stronie 134](#)

Pole	Opis
name string	Nazwa jest używana jako etykieta klucza lub certyfikatu. Musi być łańcuchem alfanumerycznym zapisanym małymi literami.
secret <a href="#">Secret</a>	Podaj klucz, używając sekretu Kubernetes.

### **.spec.pki.keys.secret**

Podaj klucz, używając sekretu Kubernetes.

Występuje w:

- [“.spec.pki.keys”](#) na stronie 134

Pole	Opis
items array	Klucze znajdujące się wewnątrz sekretu Kubernetes, które powinny zostać dodane do kontenera menedżera kolejek.
secretName string	Nazwa sekretu Kubernetes.

### **.spec.pki.trust**

PKISource definiuje źródło informacji o infrastrukturze klucza publicznego, takich jak klucze lub certyfikaty.

Występuje w:

- [“.spec.pki”](#) na stronie 134

Pole	Opis
name string	Nazwa jest używana jako etykieta klucza lub certyfikatu. Musi być łańcuchem alfanumerycznym zapisanym małymi literami.
secret <a href="#">Secret</a>	Podaj klucz, używając sekretu Kubernetes.

### **.spec.pki.trust.secret**

Podaj klucz, używając sekretu Kubernetes.

Występuje w:

- [“.spec.pki.trust”](#) na stronie 135

Pole	Opis
items array	Klucze znajdujące się wewnątrz sekretu Kubernetes, które powinny zostać dodane do kontenera menedżera kolejek.
secretName string	Nazwa sekretu Kubernetes.

### **.spec.queueManager**

Ustawienia dla kontenera menedżera kolejek i bazowego menedżera kolejek.

Występuje w:

- [“.spec”](#) na stronie 132

Pole	Opis
<code>availability</code> <a href="#">Availability</a>	Ustawienia dostępności menedżera kolejek, takie jak informacja o tym, czy ma być używana para aktywny-rezerwowy lub rodzimy mechanizm wysokiej dostępności.
<code>debug</code> boolean	Określa, czy komunikaty debugowania z kodu właściwego dla kontenera mają być rejestrowane w dzienniku kontenera. Wartością domyślną jest <code>false</code> .
<code>image</code> string	Obraz kontenera, który będzie używany.
<code>imagePullPolicy</code> string	Ustawienie wpływające na to, kiedy kubelet próbuje pobrać określony obraz. Wartością domyślną jest <code>IfNotPresent</code> .
<code>ini</code> <a href="#">INISource</a> array	Ustawienia udostępniania plików INI dla menedżera kolejek. Wymaga operatora produktu MQ 1.1.0 lub nowszej.
<code>livenessProbe</code> <a href="#">QueueManagerLivenessProbe</a>	Ustawienia sterujące sondą działania.
<code>logFormat</code> string	Format dziennika, który ma być używany dla tego kontenera. Użyj formatu <code>JSON</code> w przypadku dzienników z kontenera w formacie <code>JSON</code> . Użyj formatu <code>Basic</code> w przypadku komunikatów w formacie tekstowym. Wartością domyślną jest <code>Basic</code> .
<code>metrics</code> <a href="#">QueueManagerMetrics</a>	Ustawienia dotyczące metryk typu Prometheus.
<code>mqsc</code> <a href="#">MQSCSource</a> array	Ustawienia dotyczące udostępniania MQSC dla menedżera kolejek. Wymaga operatora produktu MQ 1.1.0 lub nowszej.
<code>name</code> string	Nazwa bazowego menedżera kolejek MQ, jeśli jest inna niż nazwa <code>metadata.name</code> . Tego pola należy użyć, jeśli nazwa menedżera kolejek nie jest zgodna z regułami nazewnictwa w środowisku Kubernetes (na przykład nazwa zawiera wielkie litery).
<code>readinessProbe</code> <a href="#">QueueManagerReadinessProbe</a>	Ustawienia sterujące sondą gotowości.
<code>resources</code> <a href="#">Resources</a>	Ustawienia, które sterują wymaganiami co do zasobów.
<code>route</code> <a href="#">Route</a>	Ustawienia trasy menedżera kolejek. Wymaga operatora produktu MQ w wersji 1.4.0 lub nowszej.
<code>startupProbe</code> <a href="#">StartupProbe</a>	Ustawienia sterujące sondą uruchamiania. Ma zastosowanie tylko do wdrożeń <code>MultiInstance</code> i <code>NativeHA</code> . Wymaga operatora MQ w wersji 1.5.0 lub nowszej.
<code>storage</code> <a href="#">QueueManagerStorage</a>	Ustawienia pamięci masowej służące do sterowania używaniem woluminów trwałych i klas pamięci przez menedżer kolejek.

### **.spec.queueManager.availability**

Ustawienia dostępności menedżera kolejek, takie jak informacja o tym, czy ma być używana para aktywny-rezerwowy lub rodzimy mechanizm wysokiej dostępności.

Występuje w:

- [“.spec.queueManager”](#) na stronie 135

Pole	Opis
<code>tls</code> <a href="#">Tls</a>	Opcjonalne ustawienia TLS służące do konfigurowania bezpiecznej komunikacji między replikami mechanizmu <code>NativeHA</code> . Wymaga operatora MQ w wersji 1.5.0 lub nowszej.



Pole	Opis
type string	Typ dostępności, który ma być używany. Użyj typu <code>SingleInstance</code> dla pojedynczego zasobnika, który zostanie automatycznie zrestartowany (w niektórych przypadkach) przez środowisko Kubernetes. Użyj typu <code>MultiInstance</code> dla pary zasobników, z których jeden jest aktywnym menedżerem kolejek, a drugi pozostaje w rezerwie. Użyj opcji <code>NativeHA</code> do realizacji rodzimej replikacji w celu zapewnienia wysokiej dostępności (wymaga to operatora MQ w wersji 1.5.0 lub nowszej). Wartością domyślną jest <code>SingleInstance</code> . Więcej szczegółowych informacji można znaleźć pod adresem <a href="http://ibm.biz/BdqAQa">http://ibm.biz/BdqAQa</a> .
updateStrategy string	Strategia aktualizacji menedżerów kolejek <code>MultiInstance</code> i <code>NativeHA</code> . Wartość <code>RollingUpdate</code> włącza aktualizacje sekwencyjne po każdej zmianie konfiguracji menedżera kolejek. Wartość <code>OnDelete</code> wyłącza automatyczne aktualizacje sekwencyjne — zmiany w menedżerze kolejek będą stosowane tylko przy usuwaniu zasobników (również gdy takie usunięcie będzie wywołane czynnikami zewnętrznymi). Wartością domyślną jest <code>RollingUpdate</code> . Wymaga operatora MQ Operator w wersji 1.6.0 lub nowszej.

### **.spec.queueManager.availability.tls**

Opcjonalne ustawienia TLS służące do konfigurowania bezpiecznej komunikacji między replikami mechanizmu `NativeHA`. Wymaga operatora MQ w wersji 1.5.0 lub nowszej.

Występuje w:

- [“.spec.queueManager.availability”](#) na stronie 136

Pole	Opis
łańcuch <code>cipherSpec</code>	Nazwa usługi <code>CipherSpec</code> dla połączenia TLS mechanizmu <code>NativeHA</code> .
<code>secretName</code> string	Nazwa sekretu Kubernetes.

### **.spec.queueManager.ini**

Źródło plików konfiguracyjnych INI.

Występuje w:

- [“.spec.queueManager”](#) na stronie 135

Pole	Opis
<code>configMap</code> <a href="#">ConfigMapINISource</a>	<code>ConfigMap</code> reprezentuje obiekt Kubernetes <code>ConfigMap</code> , który zawiera informacje INI.
<code>secret</code> <a href="#">SecretINISource</a>	Sekret reprezentuje sekret Kubernetes, który zawiera informacje INI.

### **.spec.queueManager.ini.configMap**

`ConfigMap` reprezentuje obiekt Kubernetes `ConfigMap`, który zawiera informacje INI.

Występuje w:

- [“.spec.queueManager.ini”](#) na stronie 137

Pole	Opis
<code>items</code> array	Klucze wewnątrz źródła Kubernetes, które powinny być stosowane.
<code>name</code> string	Nazwa źródła Kubernetes.

## **.spec.queueManager.ini.secret**

Sekret reprezentuje sekret Kubernetes, który zawiera informacje INI.

Występuje w:

- [“.spec.queueManager.ini”](#) na stronie 137

Pole	Opis
items array	Klucze wewnątrz źródła Kubernetes, które powinny być stosowane.
name string	Nazwa źródła Kubernetes.

## **.spec.queueManager.livenessProbe**

Ustawienia sterujące sondą działania.

Występuje w:

- [“.spec.queueManager”](#) na stronie 135

Pole	Opis
failureThreshold integer	Minimalna liczba następujących po sobie niepowodzeń, poprzedzonych powodzeniem, po której przyjmuje się, że doszło do niepowodzenia sondy. Wartością domyślną jest 1.
initialDelaySeconds integer	Liczba sekund od uruchomienia kontenera, po której sonda jest inicjowana. Wartość domyślna to 90 sekund dla wdrożenia SingleInstance. Wartość domyślna to 0 sekund dla wdrożeń MultiInstance i NativeHA. Więcej informacji: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .
periodSeconds integer	Co ile sekund ma być uruchamiana sonda. Wartością domyślną jest 10 sekund.
successThreshold integer	Minimalna liczba następujących po sobie powodzeń, poprzedzonych niepowodzeniem, po której przyjmuje się, że sonda zadziałała pomyślnie. Wartością domyślną jest 1.
timeoutSeconds integer	Limit czasu sondy w sekundach. Wartością domyślną jest 5 sekund. Więcej informacji: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .

## **.spec.queueManager.metrics**

Ustawienia dotyczące metryk typu Prometheus.

Występuje w:

- [“.spec.queueManager”](#) na stronie 135

Pole	Opis
enabled boolean	Określa, czy punkt końcowy dla metryk zgodnych z narzędziem Prometheus jest włączony, czy nie. Wartością domyślną jest true.

## **.spec.queueManager.mqsc**

Źródło plików konfiguracyjnych MQSC.

Występuje w:

- [“.spec.queueManager”](#) na stronie 135

Pole	Opis
<code>configMap</code> <a href="#">ConfigMapMQSCSource</a>	ConfigMap reprezentuje obiekt Kubernetes ConfigMap, który zawiera informacje MQSC.
<code>secret</code> <a href="#">SecretMQSCSource</a>	Sekret reprezentuje sekret Kubernetes, który zawiera informacje MQSC.

### **.spec.queueManager.mqsc.configMap**

ConfigMap reprezentuje obiekt Kubernetes ConfigMap, który zawiera informacje MQSC.

Występuje w:

- [“.spec.queueManager.mqsc”](#) na stronie 138

Pole	Opis
<code>items</code> array	Klucze wewnątrz źródła Kubernetes, które powinny być stosowane.
<code>name</code> string	Nazwa źródła Kubernetes.

### **.spec.queueManager.mqsc.secret**

Sekret reprezentuje sekret Kubernetes, który zawiera informacje MQSC.

Występuje w:

- [“.spec.queueManager.mqsc”](#) na stronie 138

Pole	Opis
<code>items</code> array	Klucze wewnątrz źródła Kubernetes, które powinny być stosowane.
<code>name</code> string	Nazwa źródła Kubernetes.

### **.spec.queueManager.readinessProbe**

Ustawienia sterujące sondą gotowości.

Występuje w:

- [“.spec.queueManager”](#) na stronie 135

Pole	Opis
<code>failureThreshold</code> integer	Minimalna liczba następujących po sobie niepowodzeń, poprzedzonych powodzeniem, po której przyjmuje się, że doszło do niepowodzenia sondy. Wartością domyślną jest 1.
<code>initialDelaySeconds</code> integer	Liczba sekund od uruchomienia kontenera, po której sonda jest inicjowana. Wartość domyślna to 10 sekund dla wdrożenia SingleInstance. Wartość domyślna to 0 dla wdrożeń MultiInstance i NativeHA. Więcej informacji: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .
<code>periodSeconds</code> integer	Co ile sekund ma być uruchamiana sonda. Wartością domyślną jest 5 sekund.
<code>successThreshold</code> integer	Minimalna liczba następujących po sobie powodzeń, poprzedzonych niepowodzeniem, po której przyjmuje się, że sonda zadziałała pomyślnie. Wartością domyślną jest 1.
<code>timeoutSeconds</code> integer	Limit czasu sondy w sekundach. Wartością domyślną są 3 sekundy. Więcej informacji: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .

## **.spec.queueManager.resources**

Ustawienia, które sterują wymaganiami co do zasobów.

Występuje w:

- [“.spec.queueManager”](#) na stronie 135

<b>Pole</b>	<b>Opis</b>
limits <a href="#">Limits</a>	Ustawienia procesora i pamięci.
requests <a href="#">Requests</a>	Ustawienia procesora i pamięci.

## **.spec.queueManager.resources.limits**

Ustawienia procesora i pamięci.

Występuje w:

- [“.spec.queueManager.resources”](#) na stronie 140

<b>Pole</b>	<b>Opis</b>
cpu	
memory	

## **.spec.queueManager.resources.requests**

Ustawienia procesora i pamięci.

Występuje w:

- [“.spec.queueManager.resources”](#) na stronie 140

<b>Pole</b>	<b>Opis</b>
cpu	
memory	

## **.spec.queueManager.route**

Ustawienia trasy menedżera kolejek. Wymaga operatora produktu MQ w wersji 1.4.0 lub nowszej.

Występuje w:

- [“.spec.queueManager”](#) na stronie 135

<b>Pole</b>	<b>Opis</b>
enabled boolean	Określa, czy trasa ma być włączona. Wartością domyślną jest true.

## **.spec.queueManager.startupProbe**

Ustawienia sterujące sondą uruchamiania. Ma zastosowanie tylko do wdrożeń MultiInstance i NativeHA. Wymaga operatora MQ w wersji 1.5.0 lub nowszej.

Występuje w:

- [“.spec.queueManager”](#) na stronie 135

Pole	Opis
failureThreshold integer	Minimalna liczba następujących kolejno po sobie niepowodzeń, po której przyjmuje się, że doszło do niepowodzenia sondy. Wartością domyślną jest 60.
initialDelaySeconds integer	Liczba sekund od uruchomienia kontenera, po której sonda jest inicjowana. Wartością domyślną jest 0 sekund. Więcej informacji: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .
periodSeconds integer	Co ile sekund ma być uruchamiana sonda. Wartością domyślną jest 5 sekund.
successThreshold integer	Minimalna liczba następujących kolejno po sobie powodzeń, po której przyjmuje się, że sonda zadziałała pomyślnie. Wartością domyślną jest 1.
timeoutSeconds integer	Limit czasu sondy w sekundach. Wartością domyślną jest 5 sekund. Więcej informacji: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .

### **.spec.queueManager.storage**

Ustawienia pamięci masowej służące do sterowania używaniem woluminów trwałych i klas pamięci przez menedżer kolejek.

Występuje w:

- [“.spec.queueManager” na stronie 135](#)

Pole	Opis
defaultClass string	Klasa pamięci masowej, która ma być domyślnie stosowana do wszystkich trwałych woluminów tego menedżera kolejek. Konkretnie trwałe woluminy mogą definiować swoje własne klasy pamięci masowej, które przestąpią to ustawienie domyślnej klasy pamięci masowej. Jeśli parametr type of availability ma wartość SingleInstance lub NativeHA, klasa pamięci masowej może być typu ReadWriteOnce lub ReadWriteMany. Jeśli parametr type of availability ma wartość MultiInstance, wymagana jest klasa pamięci masowej typu ReadWriteMany.
defaultDeleteClaim boolean	Określa, czy przy usuwaniu menedżera kolejek powinny zostać usunięte wszystkie woluminy. Dla konkretnych woluminów trwałych można określić odrębną wartość deleteClaim, która przestąpi ustawienie defaultDeleteClaim. Wartością domyślną jest false.
persistedData <a href="#">QueueManagerOptionalVolume</a>	Szczegóły woluminu trwałego (persistentVolume) dla utrwalonych danych produktu MQ, w tym konfiguracji, kolejek i komunikatów. Wymagane w przypadku korzystania z menedżera kolejek z wieloma instancjami.
queueManager <a href="#">QueueManagerVolume</a>	Wartość domyślna PersistentVolume dla wszelkich danych znajdujących się normalnie w woluminie /var/mqm. Jeśli nie zostaną określone inne woluminy, będzie zawierać wszystkie utrwalone dane i dzienniki odtwarzania.
recoveryLogs <a href="#">QueueManagerOptionalVolume</a>	Szczegóły woluminu trwałego dla dzienników odtwarzania produktu MQ. Wymagane w przypadku korzystania z menedżera kolejek z wieloma instancjami.

### **.spec.queueManager.storage.persistedData**

Szczegóły woluminu trwałego (persistentVolume) dla utrwalonych danych produktu MQ, w tym konfiguracji, kolejek i komunikatów. Wymagane w przypadku korzystania z menedżera kolejek z wieloma instancjami.

Występuje w:

- [“.spec.queueManager.storage” na stronie 141](#)

Pole	Opis
class string	Klasa pamięci masowej, która ma być używana dla tego woluminu. Wartość poprawna tylko wtedy, gdy type ma wartość persistent-claim. Jeśli parametr type of availability ma wartość SingleInstance lub NativeHA, klasa pamięci masowej może być typu ReadWriteOnce lub ReadWriteMany. Jeśli parametr type of availability ma wartość MultiInstance, wymagana jest klasa pamięci masowej typu ReadWriteMany.
deleteClaim boolean	Określa, czy ten wolumin ma zostać usunięty przy usuwaniu menedżera kolejek.
enabled boolean	Określa, czy ten wolumin powinien być włączony jako oddzielny wolumin, czy też umieszczony w domyślnym woluminie queueManager. Wartością domyślną jest false.
size string	Wielkość woluminu trwałego (PersistentVolume), która ma być przekazana do środowiska Kubernetes, w tym jednostek SI. Wartość poprawna tylko wtedy, gdy type ma wartość persistent-claim. Na przykład: 2Gi. Wartością domyślną jest 2Gi.
sizeLimit string	Limit wielkości podczas używania woluminu efemerycznego. Pliki są nadal zapisywane do katalogu tymczasowego, więc można użyć tej opcji, aby ograniczyć wielkość. Wartość poprawna tylko wtedy, gdy type ma wartość ephemeral.
type string	Typ woluminu, który ma być używany. Wybierz opcję ephemeral, aby użyć nietrwałej pamięci masowej, lub persistent-claim, aby użyć trwałego woluminu. Wartością domyślną jest persistent-claim.

### **.spec.queueManager.storage.queueManager**

Wartość domyślna PersistentVolume dla wszelkich danych znajdujących się normalnie w woluminie /var/mqm. Jeśli nie zostaną określone inne woluminy, będzie zawierać wszystkie utrwalone dane i dzienniki odtwarzania.

Występuje w:

- [“.spec.queueManager.storage” na stronie 141](#)

Pole	Opis
class string	Klasa pamięci masowej, która ma być używana dla tego woluminu. Wartość poprawna tylko wtedy, gdy type ma wartość persistent-claim. Jeśli parametr type of availability ma wartość SingleInstance lub NativeHA, klasa pamięci masowej może być typu ReadWriteOnce lub ReadWriteMany. Jeśli parametr type of availability ma wartość MultiInstance, wymagana jest klasa pamięci masowej typu ReadWriteMany.
deleteClaim boolean	Określa, czy ten wolumin ma zostać usunięty przy usuwaniu menedżera kolejek.
size string	Wielkość woluminu trwałego (PersistentVolume), która ma być przekazana do środowiska Kubernetes, w tym jednostek SI. Wartość poprawna tylko wtedy, gdy type ma wartość persistent-claim. Na przykład: 2Gi. Wartością domyślną jest 2Gi.

Pole	Opis
sizeLimit string	Limit wielkości podczas używania woluminu efemerycznego Pliki są nadal zapisywane do katalogu tymczasowego, więc można użyć tej opcji, aby ograniczyć wielkość. Wartość poprawna tylko wtedy, gdy type ma wartość ephemeral.
type string	Typ woluminu, który ma być używany. Wybierz opcję ephemeral, aby użyć nietrwałej pamięci masowej, lub persistent-claim, aby użyć trwałego woluminu. Wartością domyślną jest persistent-claim.

### **.spec.queueManager.storage.recoveryLogs**

Szczegóły woluminu trwałego dla dzienników odtwarzania produktu MQ. Wymagane w przypadku korzystania z menedżera kolejek z wieloma instancjami.

Występuje w:

- [“.spec.queueManager.storage” na stronie 141](#)

Pole	Opis
class string	Klasa pamięci masowej, która ma być używana dla tego woluminu. Wartość poprawna tylko wtedy, gdy type ma wartość persistent-claim. Jeśli parametr type of availability ma wartość SingleInstance lub NativeHA, klasa pamięci masowej może być typu ReadWriteOnce lub ReadWriteMany. Jeśli parametr type of availability ma wartość MultiInstance, wymagana jest klasa pamięci masowej typu ReadWriteMany.
deleteClaim boolean	Określa, czy ten wolumin ma zostać usunięty przy usuwaniu menedżera kolejek.
enabled boolean	Określa, czy ten wolumin powinien być włączony jako oddzielny wolumin, czy też umieszczony w domyślnym woluminie queueManager. Wartością domyślną jest false.
size string	Wielkość woluminu trwałego (PersistentVolume), która ma być przekazana do środowiska Kubernetes, w tym jednostek SI. Wartość poprawna tylko wtedy, gdy type ma wartość persistent-claim. Na przykład: 2Gi. Wartością domyślną jest 2Gi.
sizeLimit string	Limit wielkości podczas używania woluminu efemerycznego Pliki są nadal zapisywane do katalogu tymczasowego, więc można użyć tej opcji, aby ograniczyć wielkość. Wartość poprawna tylko wtedy, gdy type ma wartość ephemeral.
type string	Typ woluminu, który ma być używany. Wybierz opcję ephemeral, aby użyć nietrwałej pamięci masowej, lub persistent-claim, aby użyć trwałego woluminu. Wartością domyślną jest persistent-claim.

### **.spec.securityContext**

Ustawienia zabezpieczeń, które mają zostać dodane do elementu securityContext zasobnika menedżera kolejek.

Występuje w:

- [“.spec” na stronie 132](#)

Pole	Opis
fsGroup integer	Specjalna grupa uzupełniająca, która ma zastosowanie do wszystkich kontenerów w zasobniku. Niektóre typy woluminów umożliwiają agentowi Kubelet zmianę właściciela woluminu na zasobnik: 1. Identyfikator GID to FSGroup 2. Bit setgid jest ustawiony (nowe pliki utworzone w woluminie będą własnością FSGroup) 3. Bity uprawnień to OR'd z rw-rw----. Jeśli nie zostaną ustawione, agent Kubelet nie zmodyfikuje prawa własności i uprawnień dla woluminu.
initVolumeAsRoot boolean	Ma to wpływ na kontekst zabezpieczeń używany przez kontener, który inicjuje wolumin PersistentVolume. Ustaw na true, jeśli używany jest dostawca pamięci masowej, który wymaga, aby użytkownik root miał dostęp do nowo udostępnianych woluminów. Ustawienie true ma wpływ na to, które obiekty SCC (Security Context Constraints – ograniczenia kontekstu zabezpieczeń) mogą być używane, a menedżer kolejek może nie uruchomić się, jeśli użytkownik nie będzie mieć uprawnień do korzystania z obiektu SCC dopuszczającego użytkownika root. Wartością domyślną jest false. Aby uzyskać więcej informacji, patrz <a href="https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html">https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html</a> .
supplementalGroups array	Lista grup zastosowanych do pierwszego uruchomienia procesu w każdym kontenerze oprócz podstawowego GID kontenera. Jeśli wartość nie zostanie określona, żadne grupy nie zostaną dodane do żadnego kontenera.

### .spec.template

Zaawansowane użycie szablonów zasobów Kubernetes. Szablon umożliwia użytkownikom zmianę sposobu generowania przez IBM MQ bazowych zasobów Kubernetes, takich jak zestawy stanowe (StatefulSet), zasobniki (Pod) i usługi (Service). Ta możliwość przeznaczona jest tylko dla zaawansowanych użytkowników, ponieważ jej niepoprawne wykorzystanie może zakłócić normalne działanie produktu MQ. Wszystkie wartości określone w menedżerze kolejek zostaną przestonięte przez odpowiednie ustawienia w szablonie.

Występuje w:

- [“.spec” na stronie 132](#)

Pole	Opis
pod	Przestonięcia dla szablonu używanego dla zasobnika. Patrz <a href="https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#pods-spec-v1-core">https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#pods-spec-v1-core</a> .

### .spec.tracing

Ustawienia integracji śledzenia z panelem kontrolnym Cloud Pak for Integration Operations Dashboard.

Występuje w:

- [“.spec” na stronie 132](#)

Pole	Opis
agent <a href="#">TracingAgent</a>	Tylko w pakiecie Cloud Pak for Integration: można skonfigurować ustawienia opcjonalnego agenta śledzenia.
collector <a href="#">TracingCollector</a>	Tylko w pakiecie Cloud Pak for Integration: można skonfigurować ustawienia opcjonalnego kolektora śledzenia.



Pole	Opis
enabled boolean	Określa, czy włączyć integrację z panelem kontrolnym Cloud Pak for Integration Operations Dashboard za pośrednictwem śledzenia. Wartością domyślną jest false.
namespace string	Przestrzeń nazw, w której zainstalowany jest panel kontrolny Cloud Pak for Integration Operations Dashboard.

### **.spec.tracing.agent**

Tylko w pakiecie Cloud Pak for Integration: można skonfigurować ustawienia opcjonalnego agenta śledzenia.

Występuje w:

- [“.spec.tracing”](#) na stronie 144

Pole	Opis
image string	Obraz kontenera, który będzie używany.
imagePullPolicy string	Ustawienie wpływające na to, kiedy kubelet próbuje pobrać określony obraz. Wartością domyślną jest <code>IfNotPresent</code> .
<a href="#">livenessProbe</a> <a href="#">TracingProbe</a>	Ustawienia sterujące sondą działania.
<a href="#">readinessProbe</a> <a href="#">TracingProbe</a>	Ustawienia sterujące sondą gotowości.

### **.spec.tracing.agent.livenessProbe**

Ustawienia sterujące sondą działania.

Występuje w:

- [“.spec.tracing.agent”](#) na stronie 145

Pole	Opis
failureThreshold integer	Minimalna liczba następujących po sobie niepowodzeń, poprzedzonych powodzeniem, po której przyjmuje się, że doszło do niepowodzenia sondy. Wartością domyślną jest 1.
<a href="#">initialDelaySeconds</a> integer	Liczba sekund od uruchomienia kontenera, po której inicjowane są sondy działania. Wartością domyślną jest 10 sekund. Więcej informacji: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .
periodSeconds integer	Co ile sekund ma być uruchamiana sonda. Wartością domyślną jest 10 sekund.
successThreshold integer	Minimalna liczba następujących po sobie powodzeń, poprzedzonych niepowodzeniem, po której przyjmuje się, że sonda zadziałała pomyślnie. Wartością domyślną jest 1.
timeoutSeconds integer	Limit czasu sondy w sekundach. Wartością domyślną są 2 sekundy. Więcej informacji: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .

### **.spec.tracing.agent.readinessProbe**

Ustawienia sterujące sondą gotowości.

Występuje w:

- [“.spec.tracing.agent”](#) na stronie 145

Pole	Opis
failureThreshold integer	Minimalna liczba następujących po sobie niepowodzeń, poprzedzonych powodzeniem, po której przyjmuje się, że doszło do niepowodzenia sondy. Wartością domyślną jest 1.
initialDelaySeconds integer	Liczba sekund od uruchomienia kontenera, po której inicjowane są sondy działania. Wartością domyślną jest 10 sekund. Więcej informacji: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .
periodSeconds integer	Co ile sekund ma być uruchamiana sonda. Wartością domyślną jest 10 sekund.
successThreshold integer	Minimalna liczba następujących po sobie powodzeń, poprzedzonych niepowodzeniem, po której przyjmuje się, że sonda zadziałała pomyślnie. Wartością domyślną jest 1.
timeoutSeconds integer	Limit czasu sondy w sekundach. Wartością domyślną są 2 sekundy. Więcej informacji: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .

### **.spec.tracing.collector**

Tylko w pakiecie Cloud Pak for Integration: można skonfigurować ustawienia opcjonalnego kolektora śledzenia.

Występuje w:

- [“.spec.tracing”](#) na stronie 144

Pole	Opis
image string	Obraz kontenera, który będzie używany.
imagePullPolicy string	Ustawienie wpływające na to, kiedy kubelet próbuje pobrać określony obraz. Wartością domyślną jest IfNotPresent.
livenessProbe TracingProbe	Ustawienia sterujące sondą działania.
readinessProbe TracingProbe	Ustawienia sterujące sondą gotowości.

### **.spec.tracing.collector.livenessProbe**

Ustawienia sterujące sondą działania.

Występuje w:

- [“.spec.tracing.collector”](#) na stronie 146

Pole	Opis
failureThreshold integer	Minimalna liczba następujących po sobie niepowodzeń, poprzedzonych powodzeniem, po której przyjmuje się, że doszło do niepowodzenia sondy. Wartością domyślną jest 1.

Pole	Opis
initialDelaySeconds integer	Liczba sekund od uruchomienia kontenera, po której inicjowane są sondy działania. Wartością domyślną jest 10 sekund. Więcej informacji: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .
periodSeconds integer	Co ile sekund ma być uruchamiana sonda. Wartością domyślną jest 10 sekund.
successThreshold integer	Minimalna liczba następujących po sobie powodzeń, poprzedzonych niepowodzeniem, po której przyjmuje się, że sonda zadziałała pomyślnie. Wartością domyślną jest 1.
timeoutSeconds integer	Limit czasu sondy w sekundach. Wartością domyślną są 2 sekundy. Więcej informacji: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .

### **.spec.tracing.collector.readinessProbe**

Ustawienia sterujące sondą gotowości.

Występuje w:

- [“.spec.tracing.collector”](#) na stronie 146

Pole	Opis
failureThreshold integer	Minimalna liczba następujących po sobie niepowodzeń, poprzedzonych powodzeniem, po której przyjmuje się, że doszło do niepowodzenia sondy. Wartością domyślną jest 1.
initialDelaySeconds integer	Liczba sekund od uruchomienia kontenera, po której inicjowane są sondy działania. Wartością domyślną jest 10 sekund. Więcej informacji: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .
periodSeconds integer	Co ile sekund ma być uruchamiana sonda. Wartością domyślną jest 10 sekund.
successThreshold integer	Minimalna liczba następujących po sobie powodzeń, poprzedzonych niepowodzeniem, po której przyjmuje się, że sonda zadziałała pomyślnie. Wartością domyślną jest 1.
timeoutSeconds integer	Limit czasu sondy w sekundach. Wartością domyślną są 2 sekundy. Więcej informacji: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .

### **.spec.web**

Ustawienia serwera WWW produktu MQ.

Występuje w:

- [“.spec”](#) na stronie 132

Pole	Opis
enabled boolean	Określa, czy serwer WWW ma być włączony. Wartością domyślną jest false.

### **.status**

Obserwowany stan menedżera kolejek.

Występuje w:

- [“QueueManager” na stronie 131](#)

Pole	Opis
adminUiUrl string	Adres URL administracyjnego interfejsu użytkownika.
availability Availability	Status dostępności menedżera kolejek.
conditions QueueManagerStatusCondition array	Warunki reprezentują najnowsze dostępne obserwacje stanu menedżera kolejek.
endpoints QueueManagerStatusEndpoint array	Informacje dotyczące punktów końcowych, które eksponuje ten menedżer kolejek, takich jak punkty końcowe interfejsu API lub interfejsu użytkownika.
name string	Nazwa menedżera kolejek.
phase string	Faza stanu menedżera kolejek.
versions QueueManagerStatusVersion	Używana wersja produktu MQ oraz inne wersje dostępne w rejestrze IBM Entitled Registry.

### **.status.availability**

Status dostępności menedżera kolejek.

Występuje w:

- [“.status” na stronie 147](#)

Pole	Opis
initialQuorumEstablished boolean	Określa, czy dla NativeHA ustanowiono początkowe kworum.

### **.status.conditions**

QueueManagerStatusCondition definiuje stany menedżera kolejek.

Występuje w:

- [“.status” na stronie 147](#)

Pole	Opis
lastTransitionTime string	Czas ostatniego przejścia stanu z jednego statusu do innego.
message string	Czytelny dla człowieka komunikat opisujący szczegóły ostatniej zmiany statusu.
reason string	Przyczyna ostatniej zmiany tego statusu.
status string	Status stanu.
type string	Typ stanu.

### **.status.endpoints**

QueueManagerStatusEndpoint definiuje punkty końcowe menedżera kolejek.

Występuje w:

- [“.status” na stronie 147](#)

Pole	Opis
name string	Nazwa punktu końcowego.
type string	Typ punktu końcowego, na przykład 'UI' dla punktu końcowego interfejsu użytkownika, 'API' dla punktu końcowego interfejsu API, 'OpenAPI' dla dokumentacji API.
uri string	Identyfikator URI punktu końcowego.

### **.status.versions**

Używana wersja produktu MQ oraz inne wersje dostępne w rejestrze IBM Entitled Registry.

Występuje w:

- [“.status” na stronie 147](#)

Pole	Opis
available <a href="#">QueueManagerStatusVersionAvailable</a>	Inne wersje produktu MQ dostępne w rejestrze IBM Entitled Registry.
reconciled string	Konkretna wersja produktu IBM MQ, która jest używana. Jeśli zostanie określony obraz niestandardowy, może wystąpić niezgodność z używaną obecnie wersją produktu MQ.

### **.status.versions.available**

Inne wersje produktu MQ dostępne w rejestrze IBM Entitled Registry.

Występuje w:

- [“.status.versions” na stronie 149](#)

Pole	Opis
channels array	Kanały dostępne do automatycznego aktualizowania wersji produktu MQ.
versions <a href="#">Versions</a> array	Konkretne wersje produktu MQ, które są dostępne.

### **.status.versions.available.versions**

QueueManagerStatusVersion definiuje wersję produktu MQ.

Występuje w:

- [“.status.versions.available” na stronie 149](#)

Pole	Opis
name string	Nazwa (name) tej wersji menedżera kolejek. Są to poprawne wartości pola <code>spec.version</code> .

### **Warunki stanu menedżera kolejek ([mq.ibm.com/v1beta1](http://mq.ibm.com/v1beta1))**

Pola **status.conditions** są aktualizowane w celu odzwierciedlenia stanu zasobu QueueManager. Co do zasady stany opisują sytuacje nieprawidłowe. Menedżer kolejek bez nieprawidłowości i będący w stanie gotowości nie ma stanów **Error** ani **Pending**. Może mieć stany informacyjne **Warning**.

Obsługa stanów została wprowadzona w wersji IBM MQ Operator 1.2.

Dla zasobu QueueManager zdefiniowane są następujące stany:

Tabela 1. Stany menedżera kolejek

Komponent	Typ stanu	Kod przyczyny	Komunikat ostrzegawczy
QueueManager <sup>7</sup>	Oczekujący	Tworzenie	Menedżer kolejek MQ jest wdrażany
	Oczekujący	OidcPending	Menedżer kolejek MQ oczekuje na rejestrację klienta OIDC
	Błąd	Niepowodzenie	Wdrożenie menedżera kolejek MQ nie powiodło się
	Ostrzeżenie	UnsupportedVersion	<sup>8</sup> Operand został zainstalowany przez operator, który nie jest obsługiwany w produkcie OCP w wersji <ocp_version>. Ten operand nie jest obsługiwany.
	Ostrzeżenie	EUSSupport	<sup>9</sup> Operand EUS <mq_version> został zainstalowany, ale jest on zarządzany przez operator, który nie kwalifikuje się do rozszerzonego czasu trwania obsługi. Ten operand nie kwalifikuje się do wydłużonego czasu obsługi.
	Ostrzeżenie	EUSSupport	<sup>10</sup> Operand EUS <mq_version> został zainstalowany, ale wersja OCP <ocp_version> nie kwalifikuje się do rozszerzonego czasu trwania obsługi. Ten operand nie kwalifikuje się do wydłużonego czasu obsługi.
Ostrzeżenie	EUSSupport	<sup>11</sup> Operand EUS <mq_version> został zainstalowany, ale wersja OCP <ocp_version> nie kwalifikuje się do rozszerzonego czasu trwania obsługi.	

<sup>7</sup> Warunki Creating i Failed służą do monitorowania ogólnego postępu wdrażania menedżera kolejek. Jeśli używana jest licencja produktu IBM Cloud Pak for Integration, a konsola WWWSlogiwan MQ jest włączona, wówczas warunek OidcPending rejestruje status menedżera kolejek podczas jego wdrażania. <sup>8</sup> Ten komunikat jest wyrażony przez kod błędny 1001. <sup>9</sup> Ten komunikat jest wyrażony przez kod błędny 1002. <sup>10</sup> Ten komunikat jest wyrażony przez kod błędny 1003. <sup>11</sup> Ten komunikat jest wyrażony przez kod błędny 1004.

*Tabela 1. Stany menedżera kolejek (kontynuacja)*

Komponent	Typ stanu	Kod przyczyny	Komunikat ostrzegawczy
Zasobnik <sup>12</sup>	Oczekujący	PodPending	Trwa wdrażanie zasobnika menedżera kolejek MQ
	Błąd	PodFailed	Trwa wdrażanie zasobnika menedżera kolejek MQ
Pamięć masowa <sup>13</sup>	Oczekujący	StoragePending	Trwa udostępnianie pamięci masowej dla menedżera kolejek MQ
	Ostrzeżenie	StorageEphemeral	Dla produkcyjnego menedżera kolejek MQ używana jest efemeryczna pamięć masowa.
	Błąd	StorageFailed	Nie powiodła się próba udostępnienia pamięci masowej dla menedżera kolejek MQ

Multi

## Budowanie własnego kontenera IBM MQ i kodu wdrażania

Samodzielne budowanie kontenera. Jest to najbardziej elastyczne rozwiązanie kontenerowe, ale wymaga zaawansowanych kompetencji w dziedzinie konfigurowania kontenerów i "posiadania na własność" kontenera wynikowego.

### Zanim rozpocznie

Przed przystąpieniem do tworzenia własnego kontenera należy rozważyć, czy możliwe jest użycie jednego z gotowych kontenerów dostarczanych przez IBM. Patrz [IBM MQ w kontenerach](#)

### O tym zadaniu

W przypadku opakowania IBM MQ w postaci obrazu kontenera, zmiany wprowadzone w aplikacji można szybko i łatwo wdrażać w systemach testowych i pomostowych. Stwarza to korzystne warunki dla modelu dostaw ciągłych.

<sup>8</sup> Operator 1.4.0 i nowszy

<sup>9</sup> Operator 1.4.0 i nowszy

<sup>10</sup> Operator 1.4.0 i nowszy

<sup>11</sup> Tylko operator 1.3.0

<sup>12</sup> Warunki zasobnika służą do monitorowania statusu zasobników podczas wdrażania menedżera kolejek. Jeśli zostanie wyświetlony dowolny warunek PodFailed, ogólny warunek menedżera kolejek zostanie również ustawiony na wartość Failed.

<sup>13</sup> Warunki pamięci masowej służą do monitorowania postępu (warunek StoragePending) żądań utworzenia woluminów dla składnicy trwałej i raportowania błędów powiązania i innych awarii. Jeśli podczas udostępniania pamięci masowej wystąpi jakikolwiek błąd, warunek StorageFailed zostanie dodany do listy warunków, a ogólny warunek menedżera kolejek również zostanie ustawiony na wartość Failed.

## Procedura

- [“Planowanie wdrożenia własnego obrazu menedżera kolejek IBM MQ przy użyciu kontenera” na stronie 152](#)
- [“Budowanie przykładowego obrazu kontenera menedżera kolejek IBM MQ” na stronie 152](#)
- [“Uruchamianie aplikacji z powiązaniem lokalnymi w oddzielnych kontenerach” na stronie 155](#)

## Pojęcia pokrewne

[IBM MQ w kontenerach](#)

Multi

## Planowanie wdrożenia własnego obrazu menedżera kolejek IBM MQ przy użyciu kontenera

Istnieje kilka wymagań, które należy wziąć pod uwagę, uruchamiając menedżer kolejek IBM MQ w kontenerze. Przykładowy obraz kontenera umożliwia spełnienie tych wymagań, ale używając własnego obrazu, trzeba zadbać o ich spełnienie samodzielnie.

### Nadzór nad procesami

W momencie uruchamiania kontenera zasadniczo uruchamiany jest jeden proces (PID 1 wewnątrz kontenera), który może później tworzyć procesy potomne.

Gdy proces główny kończy działanie, środowisko wykonawcze kontenera zatrzymuje kontener. Menedżer kolejek IBM MQ wymaga działania wielu procesów w tle.

Z tego powodu należy dopilnować, aby proces główny pozostawał aktywny zawsze, gdy menedżer kolejek ma być uruchomiony. Dobrą praktyką jest sprawdzanie, czy menedżer kolejek jest aktywny, z tego procesu, na przykład poprzez kierowanie zapytań administracyjnych.

### Wypełnianie /var/mqm

Kontenery muszą zostać skonfigurowane z /var/mqm jako woluminem.

W takim przypadku przy pierwszym uruchomieniu kontenera katalog woluminu jest pusty. Ten katalog jest zwykle wypełniany podczas instalacji, ale w przypadku korzystania z kontenera środowisko instalacji i środowisko wykonawcze są oddzielnymi środowiskami.

Aby rozwiązać ten problem podczas uruchamiania kontenera, można użyć komendy `crtmqdir`, aby zapętnić produkt /var/mqm po raz pierwszy.

### Zabezpieczenia kontenera

W celu ograniczenia wymagań co do zabezpieczeń środowiska wykonawczego przykładowe obrazy kontenera są instalowane przy użyciu instalacji IBM MQ niepodlegającej kompresji. Zapewnia to ustawienie bitów `setuid` i powoduje, że kontener nie musi korzystać z eskalacji uprawnień. Niektóre systemy kontenerowe definiują identyfikatory użytkowników, które mogą być używane, a w instalacji bez kompresji nie czyni się żadnych założeń co do dostępnych użytkowników systemu operacyjnego.

Multi

## Budowanie przykładowego obrazu kontenera menedżera kolejek IBM MQ

Jest to instrukcja tworzenia przykładowego obrazu kontenera do uruchamiania menedżera kolejek IBM MQ w kontenerze.

### O tym zadaniu

Najpierw budujemy podstawowy obraz zawierający system plików Red Hat Universal Base Image i czystą instalację produktu IBM MQ.



Następnie tworzymy kolejną warstwę obrazu kontenera, dodając w niej konfigurację IBM MQ, aby umożliwić podstawową identyfikację użytkowników i zabezpieczenie hasłem.

Na koniec uruchamiamy kontener, używając tego obrazu jako jego systemu plików, z zawartością katalogu /var/mqm pochodzącą z woluminu właściwego dla kontenera w systemie plików hosta.

## Procedura

- Instrukcje budowania przykładowego obrazu kontenera do uruchamiania menedżera kolejek IBM MQ w kontenerze można znaleźć w następujących podtematach:
  - [“Budowanie przykładowego podstawowego obrazu menedżera kolejek IBM MQ” na stronie 153](#)
  - [“Budowanie przykładowego skonfigurowanego obrazu menedżera kolejek IBM MQ” na stronie 153](#)

## **Multi** Budowanie przykładowego podstawowego obrazu menedżera kolejek IBM MQ

Aby używać produktu IBM MQ we własnym obrazie kontenera, należy najpierw zbudować obraz podstawowy z czystą instalacją produktu IBM MQ. W poniższych krokach przedstawiono sposób budowania przykładowego obrazu podstawowego przy użyciu przykładowego kodu udostępnianego na serwerze GitHub.

## Procedura

- Użyj plików make dostarczonych w [repozytorium GitHub mq-container](#), aby zbudować obraz kontenera produkcyjnego.

Postępuj zgodnie z instrukcjami w sekcji [Budowanie obrazu kontenera](#) w serwisie GitHub. Jeśli planowane jest skonfigurowanie bezpiecznego dostępu za pomocą funkcji SCC ("restricted" Security Context Constraint-SCC) serwera Red Hat OpenShift Container Platform, należy użyć pakietu 'No-Install' IBM MQ.

## Wyniki

Mamy teraz podstawowy obraz kontenera z zainstalowanym produktem IBM MQ.

Teraz można przystąpić do [budowania przykładowego skonfigurowanego obrazu menedżera kolejek IBM MQ](#).

## **Multi** Budowanie przykładowego skonfigurowanego obrazu menedżera kolejek IBM MQ

Po utworzeniu ogólnego podstawowego obrazu kontenera IBM MQ należy zastosować w nim własną konfigurację, aby zabezpieczyć dostęp do niego. W tym celu należy utworzyć własną warstwę obrazu kontenera, korzystając z obrazu ogólnego jako obrazu nadrzędnego.

## Zanim rozpocznie

**V 9.2.0** W tym zadaniu przyjęto założenie, że do zbudowania [przykładowego podstawowego obrazu menedżera kolejek IBM MQ](#) użyto pakietu "No-Install" IBM MQ. W przeciwnym razie nie można będzie skonfigurować bezpiecznego dostępu przy użyciu ograniczenia kontekstu zabezpieczeń (SCC) typu Red Hat OpenShift Container Platform "restricted". Ograniczenie SCC typu "restricted", które jest stosowane domyślnie, korzysta z losowych identyfikatorów użytkowników i uniemożliwia eskalację uprawnień poprzez zmianę konta użytkownika na inne. Tradycyjny instalator IBM MQ oparty na RPM korzysta z użytkownika i grupy mqm, a także korzysta z bitów setuid w programach wykonywalnych. W programie IBM MQ 9.2, gdy używany jest pakiet "No-Install" IBM MQ, nie ma już użytkownika mqm ani grupy mqm.

## Procedura

1. Utwórz nowy katalog i dodaj do niego plik o nazwie `config.mqsc` o następującej zawartości:

```
DEFINE QLOCAL(EXAMPLE.QUEUE.1) REPLACE
```

Należy zauważyć, że w powyższym przykładzie zastosowano proces uwierzytelnianie na podstawie identyfikatora i hasła użytkownika. Istnieje jednak możliwość zastosowania dowolnej konfiguracji zabezpieczeń, której wymaga przedsiębiorstwo.

2. Utwórz plik o nazwie `Dockerfile` o następującej zawartości:

```
FROM mq
COPY config.mqsc /etc/mqm/
```

3. Zbuduj niestandardowy obraz kontenera, używając następującej komendy:

```
docker build -t mymq .
```

gdzie "." to katalog, w którym znajdują się właśnie utworzone dwa pliki.

Docker utworzy wówczas tymczasowy kontener, używając tego obrazu, a następnie uruchomi pozostałe komendy.

**Uwaga:** W systemie Red Hat Enterprise Linux (RHEL) stosuje się komendę **docker** (RHEL V7) lub **podman** (RHEL V7 lub RHEL V8). W systemie Linux komendy **docker** muszą być poprzedzane komendą **sudo** zapewniającą dodatkowe uprawnienia.

4. Uruchom nowy dostosowany obraz, aby utworzyć nowy kontener z obrazem dysku, który został właśnie utworzony.

Nowa warstwa obrazu nie określa żadnej konkretnej komendy do uruchomienia, dlatego komenda jest dziedziczona z obrazu nadrzędnego. Punkt wejścia obrazu nadrzędnego (kod jest dostępny w serwisie GitHub):

- Tworzy menedżer kolejek
- Uruchamia menedżer kolejek
- Tworzy domyślny proces nasłuchujący
- Następnie uruchamia wszystkie komendy MQSC z produktu `/etc/mqm/config.mqsc`.

Aby uruchomić nowy dostosowany obraz, wydaj następujące komendy:

```
docker run \
  --env LICENSE=accept \
  --env MQ_QMGR_NAME=QM1 \
  --volume /var/example:/var/mqm \
  --publish 1414:1414 \
  --detach \
  mymq
```

gdzie:

### Pierwszy parametr env

Przekazuje do kontenera zmienną środowiskową, która potwierdza akceptację licencji na produkt IBM IBM WebSphere MQ. Można także ustawić zmienną `LICENSE`, aby wyświetlić licencję.

Więcej informacji na temat licencji na produkt [IBM MQ](#) można znaleźć w sekcji [Informacje o licencji produktu IBM MQ](#).

### Drugi parametr env

Określa nazwę używanego menedżera kolejek.

### Parametr Volume

Przekazuje do kontenera informacje o tym, że wszystkie operacje zapisu MQ w produkcie `/var/mqm` powinny zostać zapisane w `/var/example` na hoście.

Ta opcja powoduje, że można później usunąć kontener, zachowując utrwalone dane. Ta opcja ułatwia również wyświetlanie plików dziennika.

### Parametr Publish

Odwzorowuje porty w systemie hosta na porty w kontenerze. Domyślnie kontener działa z własnym wewnętrznym adresem IP, co oznacza, że konieczne jest jawne odwzorowanie wszelkich portów, które mają być wyeksponowane.

W tym przykładzie oznacza to odwzorowanie portu 1414 na hoście na port 1414 w kontenerze.

### Parametr Detach

Uruchamia kontener w tle.

## Wyniki

Zbudowaliśmy skonfigurowany obraz kontenera i możemy przeglądać uruchomione kontenery za pomocą komendy **docker ps**. Procesy programu IBM MQ działające w kontenerze można wyświetlić za pomocą komendy **docker top**.



### Ostrzeżenie:

Dzienniki kontenera można wyświetlić przy użyciu komendy **docker logs \${CONTAINER\_ID}**.

## Co dalej

- Brak kontenera w wynikach komendy **docker ps** oznacza, że prawdopodobnie wystąpiło niepowodzenie w działaniu kontenera. Kontenery, które nie powiodły się, można wyświetlić za pomocą komendy **docker ps -a**.
- Użycie komendy **docker ps -a** powoduje wyświetlenie identyfikatora kontenera. Ten identyfikator jest również drukowany po wydaniu komendy **docker run**.
- Dzienniki kontenera można wyświetlić przy użyciu komendy **docker logs \${CONTAINER\_ID}**.

## Multi Uruchamianie aplikacji z powiązaniem lokalnymi w oddzielnych kontenerach

Dzięki współużytkowaniu przestrzeni nazw procesów między kontenerami w środowisku Docker można uruchamiać aplikacje, które wymagają powiązania lokalnego z produktami IBM MQ działającymi w innych kontenerach niż menedżer kolejek IBM MQ.

### O tym zadaniu

Ta funkcja jest obsługiwana w menedżerach kolejek w wersji IBM MQ 9.0.3 i nowszych.

Obowiązują następujące ograniczenia:

- Należy udostępnić przestrzeń nazw PID kontenerów, używając argumentu `--pid`.
- Należy udostępnić przestrzeń nazw IPC kontenerów, używając argumentu `--ipc`.
- Należy:
  1. Udostępnić przestrzeń nazw UTS kontenerów hostowi, używając argumentu `--uts`, albo
  2. Spowodować, by kontenery miały tę samą nazwę hosta, używając argumentu `-h` lub `--hostname`.
- Należy podłączyć katalog danych IBM MQ w woluminie, który jest dostępny dla wszystkich kontenerów w katalogu `/var/mqm`.

Opisaną możliwość można wypróbować, wykonując następujące kroki w systemie Linux, na którym jest już zainstalowane środowisko Docker.

W poniższym przykładzie wykorzystano przykładowy obraz kontenera IBM MQ. Szczegółowe informacje na temat tego obrazu można znaleźć w serwisie [Github](#).

### Procedura

1. Utwórz katalog tymczasowy, aby działał jako wolumin, wydając następującą komendę:

```
mkdir /tmp/dockerVolume
```

2. Utwórz menedżer kolejek (QM1) w kontenerze o nazwie `sharedNamespace`, wydając następującą komendę:

```
docker run -d -e LICENSE=accept -e MQ_QMGR_NAME=QM1 --volume /tmp/dockerVol:/mnt/mqm --uts host --name sharedNamespace ibmcom/mq
```

3. Uruchom drugi kontener o nazwie `secondaryContainer`, który jest oparty na kontenerze `ibmcom/mq`, ale nie twórz menedżera kolejek, wydając następującą komendę:

```
docker run --entrypoint /bin/bash --volumes-from sharedNamespace --pid container:sharedNamespace --ipc container:sharedNamespace --uts host --name secondaryContainer -it --detach ibmcom/mq
```

4. Uruchom komendę **dspmqr** w drugim kontenerze, aby wyświetlić status obu menedżerów kolejek, wydając następującą komendę:

```
docker exec secondaryContainer dspmq
```

5. Uruchom następującą komendę, aby przetworzyć komendy MQSC względem menedżera kolejek uruchomionego na drugim kontenerze:

```
docker exec -it secondaryContainer runmqsc QM1
```

## Wyniki

Aplikacje lokalne działają teraz w oddzielnych kontenerach i można pomyślnie uruchamiać takie komendy, jak **dspmqr**, **amqsput**, **amqsgeti** **runmqsc**, jako powiązania lokalne z menedżerem kolejek QM1 z dodatkowego kontenera.

Jeśli oczekiwane wyniki nie są wyświetlane, należy zapoznać się z informacjami podanymi tutaj: [“Rozwiązywanie problemów z aplikacjami i przestrzeniami nazw”](#) na stronie 156.

### Multi

## Rozwiązywanie problemów z aplikacjami i przestrzeniami nazw

Gdy używane są współużytkowane przestrzenie nazw, wszystkie przestrzenie (IPC, PID i UTS/nazwa hosta) oraz podłączone woluminy muszą być udostępnione. Bez tego aplikacje nie będą działać.

Lista obowiązujących ograniczeń i wymagań znajduje się w sekcji [“Uruchamianie aplikacji z powiązaniem lokalnymi w oddzielnych kontenerach”](#) na stronie 155.

Jeśli aplikacja nie spełnia wszystkich wymienionych wymagań, można napotkać problemy polegające na tym, że kontener uruchamia się, ale oczekiwane funkcje nie działają.

Na poniższej liście przedstawiono niektóre typowe przyczyny, a także działania, które być może pominięto, nie spełniając w ten sposób wymagań.

- Jeśli nie zostanie udostępniona przestrzeń nazw (UTS/PID/IPC) lub nazwa hosta kontenerów, a podłączony zostanie wolumin, kontener będzie widział menedżera kolejek, ale nie będzie mógł wchodzić z nim w interakcje.
  - Wyniki działania komend **dspmqr** są następujące:

```
docker exec container dspmq
```

```
QMNAME(QM1) STATUS(Status not available)
```

- W przypadku komend **runmqsc** lub innych komend, które próbują nawiązać połączenie z menedżerem kolejek, prawdopodobnie zostanie wyświetlony komunikat o błędzie AMQ8146:

```
docker exec -it container runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
```

```
Starting MQSC for queue manager QM1.  
AMQ8146: IBM MQ queue manager not available
```

- Jeśli są współużytkowane wszystkie wymagane przestrzenie nazw, ale wolumin współużytkowany nie został podłączony do katalogu `/var/mqm` i jest dostępna poprawna ścieżka danych IBM MQ, w przypadku komend również zostaną wyświetlone komunikaty o błędach AMQ8146.

Jednak **dspmq** w ogóle nie widzi menedżera kolejek i zwracana pustą odpowiedź:

```
docker exec container dspmq
```

- Jeśli są współużytkowane wszystkie wymagane przestrzenie nazw, ale wolumin współużytkowany nie został podłączony do katalogu `/var/mqm` i nie jest dostępna poprawna ścieżka danych IBM MQ (lub nie ma dostępnej ścieżki danych IBM MQ), wyświetlane są różne błędy, ponieważ ścieżka danych jest kluczowym komponentem instalacji produktu IBM MQ. Bez ścieżki do danych program IBM MQ nie może działać.

Jeśli zostanie uruchomiona dowolna z następujących komend, a odpowiedzi są podobne do pokazanych przykładach, należy sprawdzić, czy katalog został podłączony i czy utworzono katalog danych IBM MQ:

```
docker exec container dspmq  
'No such file or directory' from /var/mqm/mqs.ini  
AMQ6090: IBM MQ was unable to display an error message FFFFFFFF.  
AMQffff  
  
docker exec container dspmqver  
AMQ7047: An unexpected error was encountered by a command. Reason code is 0.  
  
docker exec container mqrc  
<file path>/mqrc.c[1152]  
lpi0btainQMDetails --> 545261715  
  
docker exec container crtmqm QM1  
AMQ8101: IBM MQ error (893) has occurred.  
  
docker exec container strmqm QM1  
AMQ6239: Permission denied attempting to access filesystem location '/var/mqm'.  
AMQ7002: An error occurred manipulating a file.  
  
docker exec container endmqm QM1  
AMQ8101: IBM MQ error (893) has occurred.  
  
docker exec container dlmqm QM1  
AMQ7002: An error occurred manipulating a file.  
  
docker exec container strmqweb  
<file path>/mqrc.c[1152]  
lpi0btainQMDetails --> 545261715
```

## CP4I Tworzenie własnej grupy wysokiej dostępności w przypadku tworzenia własnych kontenerów

Aby utworzyć grupę Native HA, należy utworzyć, skonfigurować i uruchomić trzy menedżery kolejek.

### O tym zadaniu

Zalecaną metodą tworzenia rodzimego rozwiązania HA jest użycie operatora IBM MQ (patrz [Rodzimy HA](#)). Alternatywnie, jeśli utworzone zostaną własne kontenery, można postępować zgodnie z tymi instrukcjami.

Aby utworzyć rodzimą grupę wysokiej dostępności, należy utworzyć trzy menedżery kolejek w trzech węzłach z typem dziennika ustawionym na wartość `log replication`. Następnie należy edytować plik `qm.ini` dla każdego menedżera kolejek w celu dodania szczegółów połączenia dla każdego z trzech węzłów, aby mogły one replikować dane dziennika do siebie nawzajem.

Następnie należy uruchomić wszystkie trzy menedżery kolejek, aby mogli sprawdzić, czy wszystkie trzy instancje mogą komunikować się ze sobą, oraz określić, które z nich będą instancją aktywną, a które będą replikami.

## Procedura

1. Na każdym z trzech węzłów należy utworzyć menedżer kolejek, określając typ dziennika repliki dziennika i podając unikalną nazwę dla każdej instancji dziennika. Każdy menedżer kolejek ma taką samą nazwę:

```
crtmqm -lr instance_name qmname
```

Na przykład:

```
node 1> crtmqm -lr qm1_inst1 qm1
node 2> crtmqm -lr qm1_inst2 qm1
node 3> crtmqm -lr qm1_inst3 qm1
```

2. Po pomyślnym utworzeniu każdego menedżera kolejek do pliku konfiguracyjnego menedżera kolejek zostanie dodana dodatkowa sekcja o nazwie `NativeHALocalInstance`, `qm.ini`. Do sekcji określającej podaną nazwę instancji dodawany jest atrybut `Name`.

Opcjonalnie można dodać następujące atrybuty do sekcji `NativeHALocalInstance` w pliku `qm.ini`:

### **KeyRepository**

Położenie repozytorium kluczy, w którym znajduje się certyfikat cyfrowy, który ma być używany do ochrony ruchu replikacji dziennika. Położenie jest podane w formacie macierzystym, oznacza to, że zawiera pełną ścieżkę i nazwę pliku bez rozszerzenia. Jeśli pominięto atrybut sekcji `KeyRepository`, dane replikacji dziennika są wymieniane między instancjami w postaci jawnego tekstu.

### **CertificateLabel**

Etykieta certyfikatu identyfikująca certyfikat cyfrowy, który ma być używany na potrzeby ochrony ruchu replikacji dziennika. Jeśli `KeyRepository` zostanie podana, ale pominięto `CertificateLabel`, zostanie użyta wartość domyślna `ibmwebspheremqqueue_manager`.

### **CipherSpec**

Wartość `CipherSpec` produktu MQ używana do zabezpieczania ruchu w replikacji dziennika. Jeśli zostanie podany ten atrybut sekcji, należy również podać wartość `KeyRepository`. Jeśli `KeyRepository` zostanie podana, ale pominięto `CipherSpec`, zostanie użyta wartość domyślna `ANY`.

### **LocalAddress**

Adres lokalnego interfejsu sieciowego, który akceptuje ruch w replikacji dziennika. Jeśli ten atrybut sekcji zostanie podany, identyfikuje on lokalny interfejs sieciowy i/lub port, używając formatu "[ addr] [ (port)]". Adres sieciowy może być podany jako nazwa hosta, IPv4 w postaci dziesiętnej z kropkami lub IPv6 w postaci szesnastkowej. Jeśli ten atrybut zostanie pominięty, menedżer kolejek podejmie próbę powiązania ze wszystkimi interfejsami sieciowymi, używa on portu określonego w sekcji `ReplicationAddress` w sekcji `NativeHAInstances` zgodnej z nazwą lokalnej instancji.

### **HeartbeatInterval**

Interwał pulsu definiuje, co ile milisekund aktywna instancja menedżera kolejek o rodzimej wysokiej dostępności wysyła puls sieciowy. Poprawny zakres wartości interwału pulsu wynosi od 500 (0,5 sekundy) do 60000 (1 minuta), a wartość spoza tego zakresu powoduje, że uruchomienie menedżera kolejek nie powiedzie się. Jeśli ten atrybut zostanie pominięty, zostanie użyta wartość domyślna 5000 (5 sekund). Każda instancja musi używać tego samego interwału pulsu.

### **HeartbeatTimeout**

Limit czasu pulsu definiuje, jak długo instancja-replika menedżera kolejek o rodzimej wysokiej dostępności oczekuje przed podjęciem decyzji, że aktywna instancja nie odpowiada. Poprawny zakres limitu czasu interwału pulsu wynosi od 500 (0,5 sekundy) do 120000 (2 minuty). Wartość limitu czasu pulsu musi być większa niż lub równa wartości interwału pulsu.

Niepoprawna wartość powoduje, że uruchomienie menedżera kolejek nie powiedzie się. Jeśli ten atrybut zostanie pominięty, replika oczekuje na 2 x `HeartbeatInterval` przed rozpoczęciem

procesu w celu wybrania nowej aktywnej instancji. Każda instancja musi używać tego samego limitu czasu pulsu.

### RetryInterval

Interwał ponowienia definiuje, co ile milisekund menedżer kolejek o rodzimej wysokiej dostępności powinien ponawiać próbę nawiązania połączenia przez niesprawne łącze replikacji. Poprawny zakres interwału ponowienia wynosi od 500 (0,5 sekundy) do 120000 (2 minuty). Jeśli ten atrybut zostanie pominięty, replika oczekuje na 2 x HeartbeatInterval przed ponowną próbą utworzenia łącza replikacji, które zakończyło się niepowodzeniem.

3. Zmodyfikuj plik `qm.ini` dla każdego menedżera kolejek i dodaj szczegóły połączenia. Do każdej instancji menedżera kolejek w rodzimej grupie wysokiej dostępności (łącznie z lokalną instancją) należy dodać trzy sekcje `NativeHAInstance`. Dodaj następujące atrybuty:

### Nazwa

Podaj nazwę instancji, która była używana podczas tworzenia instancji menedżera kolejek.

### ReplicationAddress

Podaj nazwę hosta, adres IPv4 w postaci dziesiętnej z kropkami lub adres IPv6 w formacie szesnastkowym instancji. Adres można określić jako nazwę hosta, adres IPv4 w postaci dziesiętnej z kropkami lub adres IPv6 w postaci szesnastkowej. Adres replikacji musi być tłumaczona i możliwa do przetłumaczenia z każdej instancji w grupie. Numer portu, który ma być używany dla replikacji dziennika, musi być podany w nawiasach, na przykład:

```
ReplicationAddress=host1.example.com(4444)
```

**Uwaga:** Sekcje `NativeHAInstance` są identyczne w każdej instancji i mogą być udostępniane za pomocą opcji automatycznej konfiguracji (`crtmqm -ii`).

4. Uruchom każdą z trzech instancji:

```
strmqm QMgrName
```

Po uruchomieniu instancji komunikują się one w celu sprawdzenia, czy wszystkie trzy instancje są uruchomione, a następnie decydują, które z tych trzech instancji są aktywne, podczas gdy pozostałe dwie instancje są nadal uruchamiane jako repliki.

### Przykład

W poniższym przykładzie przedstawiono sekcję pliku `qm.ini`, w której określono wymagane szczegóły rodzimego HA dla jednej z trzech instancji:

```
NativeHALocalInstance:
  LocalName=node-1

NativeHAInstance:
  Name=node-1
  ReplicationAddress=host1.example.com(4444)
NativeHAInstance:
  Name=node-2
  ReplicationAddress=host2.example.com(4444)
NativeHAInstance:
  Name=node-3
  ReplicationAddress=host3.example.com(4444)
```

## **Kubernetes** Uwagi dotyczące wykonywania własnej sekwencyjnej aktualizacji menedżera kolejek rodzimej wysokiej dostępności

Każda aktualizacja wersji IBM MQ lub specyfikacji zasobnika dla menedżera kolejek rodzimej wysokiej dostępności będzie wymagała przeprowadzenia sekwencyjnej aktualizacji instancji menedżera kolejek. Produkt IBM MQ Operator obsługuje tę opcję automatycznie, ale jeśli stworzysz własny kod wdrażania, istnieje kilka ważnych kwestii.

**Uwaga:** Przykładowy wykres programu Helm zawiera skrypt powłoki do wykonywania sekwencyjnej aktualizacji, jednak skrypt ten **nie** nadaje się do użytku produkcyjnego, ponieważ nie odnosi się do zagadnień w tym temacie.

W produkcji Kubernetes zasoby StatefulSet służą do zarządzania uporządkowanymi uruchomieniami i sekwencyjnymi aktualizacjami. Częścią procedury uruchomienia jest indywidualne uruchomienie każdego zasobnika, oczekiwanie na jego gotowość, a następnie przejście do następnego zasobnika. To nie zadziała w przypadku Rdzennej HA, ponieważ wszystkie Pody muszą być uruchomione, aby mogły przeprowadzić wybory lidera. Dlatego pole `.spec.podManagementPolicy` w StatefulSet musi zostać ustawione na wartość `Parallel`. Oznacza to również, że wszystkie zasobniki będą aktualizowane równolegle, co jest szczególnie niepożądane. Z tego powodu produkt StatefulSet powinien również używać strategii aktualizacji produktu `OnDelete`.

Niezdolność do wykorzystania kodu sekwencyjnej aktualizacji StatefulSet wymaga dostosowanego kodu sekwencyjnej aktualizacji, który powinien uwzględniać następujące kwestie:

- Ogólna procedura sekwencyjnej aktualizacji
- Minimalizowanie czasu przestoju poprzez aktualizację zasobników w najlepszej kolejności
- Obsługa zmian stanu klastra
- Obsługa błędów
- Rozwiązywanie problemów z synchronizacją

## Ogólna procedura sekwencyjnej aktualizacji

Kod sekwencyjnej aktualizacji powinien odczekać, aż każda instancja wyświetli status `REPLICA` produktu `dspm`. Oznacza to, że instancja wykonała pewien poziom uruchamiania (na przykład kontener jest uruchomiony, a procesy produktu `MQ` działają), ale niekoniecznie udało się jeszcze skomunikować z innymi instancjami. Na przykład: zasobnik A zostanie zrestartowany, a gdy tylko będzie w stanie `REPLICA`, zasobnik B zostanie zrestartowany. Gdy zasobnik B rozpoczyna się od nowej konfiguracji, powinien być w stanie komunikować się z zasobnikiem A i może tworzyć quorum; wówczas albo A, albo B stanie się nową aktywną instancją.

W ramach tego przydatne jest opóźnienie po osiągnięciu stanu `REPLICA` przez każdy zasobnik, aby umożliwić mu połączenie się z innymi zasobnikami i ustanowienie quorum.

## Minimalizowanie czasu przestoju poprzez aktualizację zasobników w najlepszej kolejności

Kod sekwencyjnej aktualizacji powinien usuwać zasobniki pojedynczo, zaczynając od zasobników, które są w znanym stanie błędów, a następnie wszystkich zasobników, które nie zostały pomyślnie uruchomione. Aktywny zasobnik menedżera kolejek powinien być na ogół aktualizowany jako ostatni.

Ważne jest również, aby wstrzymać usuwanie zasobników, jeśli ostatnia aktualizacja spowodowała przejście zasobnika do znanego stanu błędów. Zapobiega to wprowadzeniu przerwanej aktualizacji do wszystkich zasobników. Może się to zdarzyć na przykład wtedy, gdy zasobnik zostanie zaktualizowany, w celu użycia nowego obrazu kontenera, który nie jest dostępny (lub zawiera literówkę).

## Obsługa zmian stanu klastra

Kod sekwencyjnej aktualizacji musi odpowiednio reagować na zmiany w czasie rzeczywistym w stanie klastra. Na przykład jeden z zasobników menedżera kolejek może zostać usunięty z powodu restartu węzła lub z powodu nacisku węzła. Możliwe, że usunięty zasobnik może nie zostać natychmiast ponownie zaplanowany, jeśli klastr jest zajęty. W takim przypadku kod sekwencyjnej aktualizacji powinien odpowiednio odczekać przed zrestartowaniem innych zasobników.

## Obsługa błędów

Kod sekwencyjnej aktualizacji musi być odporny na awarie podczas wywoływania interfejsu API Kubernetes i innych nieoczekiwanych zachowań w klastrze.



Ponadto sam kod sekwencyjnej aktualizacji musi dopuszczać swoje restartowanie. Sekwencyjna aktualizacja może być długotrwała, a kod może wymagać zrestartowania.

## Rozwiązywanie problemów z synchronizacją

Kod sekwencyjnej aktualizacji musi sprawdzić wersje aktualizacji zasobnika, aby mógł zapewnić restart zasobnika. Pozwala to uniknąć problemów z synchronizacją, w których zasobnik może wskazywać status "Uruchomiono", gdy w rzeczywistości jeszcze się nie zakończył.

### Pojęcia pokrewne

“Wybór sposobu używania IBM MQ w kontenerach” na stronie 5

Istnieje wiele opcji korzystania z produktu IBM MQ w kontenerach: można użyć operatora IBM MQ Operator, który korzysta ze wstępnie spakowanych obrazów kontenerów, lub zbudować własne obrazy i kod wdrożenia.

## CP4I Wyświetlanie statusu rodzimych menedżerów kolejek HA dla kontenerów zbudowanych na zamówienie

W przypadku kontenerów zbudowanych na zamówienie można wyświetlić status instancji rodzimej HA za pomocą komendy **dspmq**.

### O tym zadaniu

Aby wyświetlić status działania instancji menedżera kolejek w węźle, można użyć komendy **dspmq**. Zwrócone informacje zależą od tego, czy instancja jest aktywna, czy jest repliką. Informacje dostarczone przez aktywną instancję są definitywne, a informacje z węzłów replik mogą być nieaktualne.

Można wykonać następujące czynności:

- Sprawdź, czy instancja menedżera kolejek w bieżącym węźle jest aktywna, czy jest repliką.
- Wyświetl status działania rodzimej wysokiej dostępności dla instancji w bieżącym węźle.
- Wyświetl status działania wszystkich trzech instancji w konfiguracji rodzimej wysokiej dostępności.

Następujące pola statusu są używane do raportowania statusu konfiguracji rodzimej wysokiej dostępności:

#### ROLE

Określa bieżącą rolę instancji i jest jedną z następujących wartości: Active, Replica lub Unknown.

#### INSTANCE

Nazwa podana dla tej instancji menedżera kolejek, gdy była ona tworzona przy użyciu opcji **-lr** komendy **crtmqm**.

#### INSYNC

Wskazuje, czy w razie potrzeby instancja jest w stanie przejąć rolę aktywnej instancji.

#### QUORUM

Raportuje status kworum w formacie *liczba zsynchronizowanych instancji/liczba skonfigurowanych instancji*.

#### REPLADDR

Adres replikacji instancji menedżera kolejek.

#### CONNECTV

Wskazuje, czy węzeł jest połączony z aktywną instancją.

#### BACKLOG

Określa zaległość instancji w kB.

#### CONNINST

Informuje, czy instancja o podanej nazwie jest połączona z tą instancją.

## ALTDATA

Wskazuje datę ostatniej aktualizacji tych informacji (wartość pusta, jeśli nigdy nie zostały zaktualizowane).

## ALTTIME

Wskazuje godzinę ostatniej aktualizacji tych informacji (wartość pusta, jeśli nigdy nie zostały zaktualizowane).

## Procedura

- Aby określić, czy instancja menedżera kolejek działa jako aktywna instancja, czy jako replika:

```
dspmqr -o status -m QMgrName
```

Aktywna instancja menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          STATUS(Running)
```

Instancja-replika menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          STATUS(Replica)
```

Nieaktywna instancja zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          STATUS(Ended Immediately)
```

- Aby określić status działania rodzimego HA instancji w bieżącym węźle:

```
dspmqr -o nativeha -m QMgrName
```

Aktywna instancja menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

Instancja-replika menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```

Nieaktywna menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- Aby określić status działania rodzimej wysokiej dostępności dla wszystkich instancji w konfiguracji o rodzimej wysokiej dostępności:

```
dspmqr -o nativeha -x -m QMgrName
```

Jeśli ta komenda zostanie wydana w węźle, w którym działa aktywna instancja menedżera kolejek BOB, może zostać wyświetlony następujący status:

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

Jeśli ta komenda zostanie wydana w węźle, w którym uruchomiona jest instancja-replika menedżera kolejek BOB, może zostać wyświetlony następujący status, który wskazuje, że jedna z replik ma zaległości:

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
```

```
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No) BACKLOG(435)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

Jeśli ta komenda zostanie wydana w węźle, w którym uruchomiona jest nieaktywna instancja menedżera kolejek BOB, może zostać wyświetlony następujący status:

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
```

W przypadku wydania komendy w momencie, gdy instancje nadal negocjują, która będzie aktywna, a które będą replikami, zwrócony zostanie następujący status:

```
QMNAME(BOB)          STATUS(Negotiating)
```

## Odsyłacze pokrewne

### **dspmqr**

## **CP4I Kończenie rodzimych menedżerów kolejek wysokiej dostępności**

Za pomocą komendy **endmqm** można zakończyć działanie aktywnego lub replika menedżera kolejek, który jest częścią rodzimej grupy wysokiej dostępności.

## **Procedura**

- Aby zakończyć aktywną instancję menedżera kolejek, należy zapoznać się z sekcją [Końcowanie rodzimych menedżerów kolejek HA](#) w sekcji Konfigurowanie tej dokumentacji.



## Uwagi

---

Niniejsza publikacja została opracowana z myślą o produktach i usługach oferowanych w Stanach Zjednoczonych.

IBM może nie oferować w innych krajach produktów, usług lub opcji omawianych w tej publikacji. Informacje o produktach i usługach dostępnych w danym kraju można uzyskać od lokalnego przedstawiciela IBM. Odwołanie do produktu, programu lub usługi IBM nie oznacza, że można użyć wyłącznie tego produktu, programu lub usługi IBM. Zamiast nich można zastosować ich odpowiednik funkcjonalny pod warunkiem, że nie narusza to praw własności intelektualnej firmy IBM. Jednakże cała odpowiedzialność za ocenę przydatności i sprawdzenie działania produktu, programu lub usługi pochodzących od producenta innego niż IBM spoczywa na użytkowniku.

IBM może posiadać patenty lub złożone wnioski patentowe na towary i usługi, o których mowa w niniejszej publikacji. Używanie tego dokumentu nie daje żadnych praw do tych patentów. Pisemne zapytania w sprawie licencji można przesyłać na adres:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Zapytania w sprawie licencji dotyczących informacji kodowanych przy użyciu dwubajtowych zestawów znaków (DBCS) należy kierować do lokalnych działów IBM Intellectual Property Department lub zgłaszać na piśmie pod adresem:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

**Poniższy akapit nie obowiązuje w Wielkiej Brytanii, a także w innych krajach, w których jego treść pozostaje w sprzeczności z przepisami prawa miejscowego:** INTERNATIONAL BUSINESS MACHINES CORPORATION DOSTARCZA TĘ PUBLIKACJĘ W STANIE, W JAKIM SIĘ ZNAJDUJE ("AS IS"), BEZ JAKICHKOLWIEK GWARANCJI (RĘKOJMIĘ RÓWNIEŻ WYŁĄCZA SIĘ), WYRAŻNYCH LUB DOMNIEMANYCH, A W SZCZEGÓLNOŚCI DOMNIEMANYCH GWARANCJI PRZYDATNOŚCI HANDLOWEJ, PRZYDATNOŚCI DO OKREŚLONEGO CELU ORAZ GWARANCJI, ŻE PUBLIKACJA TA NIE NARUSZA PRAW OSÓB TRZECICH. Ustawodawstwa niektórych krajów nie dopuszczają zastrzeżeń dotyczących gwarancji wyraźnych lub domniemanych w odniesieniu do pewnych transakcji; w takiej sytuacji powyższe zdanie nie ma zastosowania.

Informacje zawarte w niniejszej publikacji mogą zawierać nieścisłości techniczne lub błędy typograficzne. Informacje te są okresowo aktualizowane, a zmiany te zostaną uwzględnione w kolejnych wydaniach tej publikacji. IBM zastrzega sobie prawo do wprowadzania ulepszeń i/lub zmian w produktach i/lub programach opisanych w tej publikacji w dowolnym czasie, bez wcześniejszego powiadomienia.

Wszelkie wzmianki w tej publikacji na temat stron internetowych innych podmiotów zostały wprowadzone wyłącznie dla wygody użytkowników i w żadnym wypadku nie stanowią zachęty do ich odwiedzania. Materiały dostępne na tych stronach nie są częścią materiałów opracowanych dla tego produktu IBM, a użytkownik korzysta z nich na własną odpowiedzialność.

IBM ma prawo do używania i rozpowszechniania informacji przystanych przez użytkownika w dowolny sposób, jaki uzna za właściwy, bez żadnych zobowiązań wobec ich autora.

Licencjodawcy tego programu, którzy chcieliby uzyskać informacje na temat programu w celu: (i) wdrożenia wymiany informacji między niezależnie utworzonymi programami i innymi programami (łącznie

z tym opisywanym) oraz (ii) wspólnego wykorzystywania wymienianych informacji, powinni skontaktować się z:

IBM Corporation  
Koordynator współdziałania z oprogramowaniem, Dział 49XA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

Informacje takie mogą być udostępnione, o ile spełnione zostaną odpowiednie warunki, w tym, w niektórych przypadkach, zostanie uiszczona stosowna opłata.

Licencjonowany program opisany w niniejszej publikacji oraz wszystkie inne licencjonowane materiały dostępne dla tego programu są dostarczane przez IBM na warunkach określonych w Umowie IBM z Klientem, Międzynarodowej Umowie Licencyjnej IBM na Program lub w innych podobnych umowach zawartych między IBM i użytkownikami.

Wszelkie dane dotyczące wydajności zostały zebrane w kontrolowanym środowisku. W związku z tym rezultaty uzyskane w innych środowiskach operacyjnych mogą się znacząco różnić. Niektóre pomiary mogły być dokonywane na systemach będących w fazie rozwoju i nie ma gwarancji, że pomiary wykonane na ogólnie dostępnych systemach dadzą takie same wyniki. Niektóre z pomiarów mogły być estymowane przez ekstrapolację. Rzeczywiste wyniki mogą być inne. Użytkownicy powinni we własnym zakresie sprawdzić odpowiednie dane dla ich środowiska.

Informacje dotyczące produktów innych niż produkty IBM pochodzą od dostawców tych produktów, z opublikowanych przez nich zapowiedzi lub innych powszechnie dostępnych źródeł. Firma IBM nie testowała tych produktów i nie może potwierdzić dokładności pomiarów wydajności, kompatybilności ani żadnych innych danych związanych z tymi produktami. Pytania dotyczące możliwości produktów innych podmiotów należy kierować do dostawców tych produktów.

Wszelkie stwierdzenia dotyczące przyszłych kierunków rozwoju i zamierzeń IBM mogą zostać zmienione lub wycofane bez powiadomienia.

Publikacja ta zawiera przykładowe dane i raporty używane w codziennych operacjach działalności gospodarczej. W celu kompleksowego ich zilustrowania podane przykłady zawierają nazwiska osób prywatnych, nazwy przedsiębiorstw oraz nazwy produktów. Wszystkie te nazwy/nazwiska są fikcyjne i jakiegokolwiek podobieństwo do istniejących nazw/nazwisk i adresów jest całkowicie przypadkowe.

#### LICENCJA W ZAKRESIE PRAW AUTORSKICH:

Niniejsza publikacja zawiera przykładowe aplikacje w kodzie źródłowym, ilustrujące techniki programowania w różnych systemach operacyjnych. Użytkownik może kopiować, modyfikować i dystrybuować te programy przykładowe w dowolnej formie bez uiszczania opłat na rzecz IBM, w celu projektowania, używania, sprzedaży lub dystrybucji aplikacji zgodnych z aplikacyjnym interfejsem programistycznym dla tego systemu operacyjnego, dla którego napisane zostały programy przykładowe. Programy przykładowe nie zostały gruntownie przetestowane. IBM nie może zatem gwarantować ani sugerować niezawodności, użyteczności i funkcjonalności tych programów.

W przypadku przeglądania niniejszych informacji w formie elektronicznej, zdjęcia i kolorowe ilustracje mogą nie być wyświetlane.

## Informacje dotyczące interfejsu programistycznego

---

Informacje dotyczące interfejsu programistycznego, o ile są udostępniane, mają być pomocne podczas tworzenia oprogramowania aplikacji do użytku z tym programem.

Ten podręcznik zawiera informacje na temat planowanych interfejsów programistycznych, które umożliwiają klientom pisanie programów w celu uzyskania dostępu do usług produktu WebSphere MQ.

Informacje te mogą również zawierać informacje na temat diagnostyki, modyfikacji i strojenia. Tego typu informacje są udostępniane jako pomoc przy debugowaniu aplikacji.

**Ważne:** Informacji na temat diagnostyki, modyfikacji i strojenia nie należy używać jako interfejsu programistycznego, ponieważ może on ulec zmianie.

## Znaki towarowe

---

IBM, logo IBM, ibm.com, są znakami towarowymi IBM Corporation, zarejestrowanymi w wielu systemach prawnych na całym świecie. Aktualna lista znaków towarowych IBM jest dostępna w serwisie WWW, w sekcji "Copyright and trademark information" (Informacje o prawach autorskich i znakach towarowych), pod adresem [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml). Nazwy innych produktów lub usług mogą być znakami towarowymi IBM lub innych podmiotów.

Microsoft oraz Windows są znakami towarowymi Microsoft Corporation w Stanach Zjednoczonych i/lub w innych krajach.

UNIX jest zastrzeżonym znakiem towarowym The Open Group w Stanach Zjednoczonych i/lub w innych krajach.

Linux jest zastrzeżonym znakiem towarowym Linusa Torvaldsa w Stanach Zjednoczonych i/lub w innych krajach.

Ten produkt zawiera oprogramowanie opracowane przez Eclipse Project (<https://www.eclipse.org/>).

Java oraz wszystkie znaki towarowe i logo dotyczące języka Java są znakami towarowymi lub zastrzeżonymi znakami towarowymi Oracle i/lub przedsiębiorstw afiliowanych Oracle.









Numer pozycji:

(1P) P/N: