

9.1

*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 53](#).

To wydanie dotyczy wersji 9 wydania 1 produktu IBM® MQ oraz wszystkich kolejnych wydań i modyfikacji, o ile nie podano 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.....</b>  | <b>5</b>  |
| Planowanie wdrożenia IBM MQ w kontenerach.....  | 5         |
| Wybór sposobu używania IBM MQ w kontenerach.....  | 5         |
| Obsługa kontenerów certyfikowanych przez produkt IBM MQ.....  | 6         |
| Tworzenie własnych obrazów kontenerów i wykresów IBM MQ.....  | 8         |
| Uwagi dotyczące pamięci masowej dla IBM MQ Advanced certified container.....  | 9         |
| Wysoka dostępność dla produktu IBM MQ Advanced certified container.....   | 10        |
| Uwierzytelnianie użytkownika i autoryzacja dla produktu IBM MQ Advanced certified container....                           | 12        |
| Instalowanie i deinstalowanie operatora IBM MQ na platformie OpenShift.....   | 12        |
| Instalowanie operatora IBM MQ przy użyciu konsoli WWW OpenShift.....  | 12        |
| Instalowanie operatora IBM MQ przy użyciu interfejsu wiersza komend OpenShift.....  | 14        |
| Wdrażanie certyfikowanych kontenerów produktu IBM MQ.....   | 16        |
| Przygotowywanie projektu OpenShift dla produktu IBM MQ przy użyciu interfejsu wiersza komend OpenShift.....               | 16        |
| Wdrażanie menedżera kolejek przy użyciu produktu IBM Cloud Pak for Integration Platform Navigator.....                    | 17        |
| Wdrażanie menedżera kolejek przy użyciu konsoli WWW OpenShift.....  | 18        |
| Wdrażanie menedżera kolejek za pomocą interfejsu wiersza komend OpenShift.....  | 19        |
| Integracja z panelem kontrolnym IBM Cloud Pak for Integration Operations Dashboard.....                                   | 21        |
| Budowanie obrazu z niestandardowymi plikami MQSC i INI przy użyciu interfejsu wiersza komend OpenShift.....               | 21        |
| Wdrażanie certyfikowanych kontenerów IBM MQ za pomocą Helm.....   | 23        |
| Wdrażanie wcześniejszych wersji CD produktu IBM MQ w klastrze produktu IBM Cloud Private.....                             | 27        |
| Dodawanie wcześniejszych wersji dysku CD obrazu IBM MQ do klastra IBM Cloud Private.....                                  | 29        |
| Dodawanie wcześniejszych wersji dysku CD obrazu IBM MQ do klastra IBM Cloud Kubernetes Service.....                       | 30        |
| Nawiązywanie połączenia z menedżerem kolejek wdrożonym w klastrze OpenShift.....  | 30        |
| Nawiązywanie połączenia z konsolą IBM MQ Console wdrożoną w klastrze OpenShift.....                                       | 32        |
| Tworzenie i odtwarzanie kopii zapasowej konfiguracji menedżera kolejek za pomocą interfejsu wiersza komend OpenShift..... | 33        |
| Budowanie własnego kontenera IBM MQ.....  | 34        |
| Planowanie wdrożenia własnego obrazu menedżera kolejek IBM MQ przy użyciu kontenera.....                                  | 34        |
| Budowanie przykładowego obrazu menedżera kolejek produktu IBM MQ przy użyciu programu Docker.....                         | 35        |
| Uruchamianie aplikacji z powiązaniem lokalnymi w oddzielnych kontenerach.....   | 38        |
| Skorowidz interfejsu API operatora IBM MQ.....  | 40        |
| Skorowidz interfejsu API mq.ibm.com/v1beta1.....  | 40        |
| <b>Uwagi.....</b>   | <b>53</b> |
| Informacje dotyczące interfejsu programistycznego.....  | 54        |
| Znaki towarowe.....   | 55        |



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 w wstępnie spakowanym kontenerze udostępnionym w produkcie IBM MQ Advanced i IBM MQ Advanced for Developers. Ten produkt IBM MQ Advanced certified container oferuje obsługiwany obraz i wykres Helm. Można go użyć do wdrożenia gotowego obrazu produktu IBM MQ do produktu Red Hat® OpenShift®, IBM Cloud Private lub IBM Cloud Kubernetes Service.

IBM MQ można również uruchamiać w kontenerze IBM Cloud Pak for Integration lub w kontenerze zbudowanym samodzielnie.

  Więcej informacji na temat IBM MQ Advanced certified container można znaleźć na następujących stronach.

## 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 zaplanowaniem wdrożenia IBM MQ w architekturze kontenerowej należy zapoznać się zarówno z podstawowymi pojęciami dotyczącymi IBM MQ (patrz [Przegląd techniczny produktu IBM MQ](#)), jak i z podstawowymi pojęciami dotyczącymi środowiska Kubernetes/OpenShift (patrz [OpenShift Container Platform architecture](#)).

### Procedura

- [“Wybór sposobu używania IBM MQ w kontenerach”](#) na stronie 5.
- [“Wysoka dostępność dla produktu IBM MQ Advanced certified container”](#) na stronie 10.
- [“Uwierzytelnianie użytkownika i autoryzacja dla produktu IBM MQ Advanced certified container”](#) na stronie 12.

## w kontenerach

Istnieje wiele opcji korzystania z produktu IBM MQ w kontenerach: można użyć wstępnie spakowanych certyfikowanych kontenerów lub zbudować własne obrazy i kod wdrożenia.

### Korzystanie z certyfikowanych kontenerów IBM MQ Advanced

Jeśli planowane jest wdrożenie w środowisku Red Hat OpenShift Container Platform, to prawdopodobnie będą stosowane kontenery certyfikowane. Istnieją trzy odmiany certyfikowanego kontenera:

- IBM MQ Advanced certified container dla IBM Cloud Pak for Integration. Jest to osobny produkt IBM, który zawiera wersję certyfikowanego kontenera.
- IBM MQ Advanced certified container
- Certyfikowany kontener IBM MQ Advanced for Developers (bez gwarancji)

IBM MQ 9.1.4 i wcześniejsze wersje produktu CD były również obsługiwane w systemach IBM Cloud Private i IBM Cloud Kubernetes Service.

Należy pamiętać, że certyfikowane kontenery szybko ewoluują, dlatego są obsługiwane tylko w wydaniach [Continuous Delivery](#).

Certyfikowane kontenery obejmują zarówno wstępnie wybudowane obrazy kontenerów, jak i kod wdrażania na potrzeby uruchamiania w produkcie Red Hat OpenShift Container Platform. Począwszy od wersji IBM MQ 9.1.5 menedżery kolejek są zarządzane za pomocą operatora IBM MQ. Wcześniejsze wersje produktu IBM MQ, do wersji 9.1.5 włącznie, są zarządzane za pomocą wykresów Helm.

Niektóre funkcje produktu IBM MQ nie są obsługiwane w przypadku korzystania z certyfikowanych kontenerów. 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 certyfikowanych kontenerów w celu udostępnienia jednej lub większej liczby menedżerów kolejek koordynacji, komend lub menedżerów kolejek agenta.
  - AMQP
  - IBM MQ Bridge to Salesforce
  - IBM MQ Bridge to blockchain (produkt nieobsługiwany w kontenerach)
- Korzystanie z serwera WWW podczas wdrażania przy użyciu wykresów Helm (z wyjątkiem IBM Cloud Pak for Integration)
- Dostosowanie opcji komend `crtmqm`, `strmqm` i `endmqm`, np. skonfigurowanie dzienników odtwarzania

## Budowanie własnych obrazów i wykresów

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"](#) na stronie 34. Wykresy Helm udostępniane jako część certyfikowanych kontenerów są publikowane na serwerze GitHubi mogą być używane jako przykłady w przypadku budowania własnych obrazów:

- [WykresHelm dla IBM MQ Advanced certified container](#)
- [WykresHelm dla certyfikowanego kontenera IBM MQ Advanced for Developers](#)

## Pojęcia pokrewne

["Obsługa kontenerów certyfikowanych przez produkt IBM MQ"](#) na stronie 6

Certyfikowane kontenery produktu IBM MQ są obsługiwane tylko w niektórych środowiskach Kubernetes


["Tworzenie własnych obrazów kontenerów i wykresów IBM MQ"](#) na stronie 8

Informacje, które należy wziąć pod uwagę, używając kontenerów w systemie Linux.

Linux

## Obsługa kontenerów certyfikowanych przez produkt IBM MQ

Certyfikowane kontenery produktu IBM MQ są obsługiwane tylko w niektórych środowiskach Kubernetes

 W przypadku wersji CD V9.1.4 i nowszych produkt IBM MQ Advanced certified container jest obsługiwany w celu użycia z produktem Red Hat OpenShift. Patrz ["Wdrażanie menedżera kolejek przy użyciu interfejsu CLI Helm"](#) na stronie 25.

Wersje CD wcześniejsze niż V9.1.4 były obsługiwane w następujących środowiskach Kubernetes :

- IBM Cloud Kubernetes Service

- IBM Cloud Private
- IBM Cloud Private z Red Hat OpenShift

W przypadku konkretnych obsługiwanych wersji produktu Kubernetes należy zapoznać się z plikami `qualification.yaml` i `Chart.yaml` wewnątrz pobranego wykresu IBM MQ Advanced Helm . Wersje te różnią się od wersji do wydania.

IBM MQ Advanced certified container jest obsługiwany tylko w przypadku wdrażania za pomocą operatora IBM MQ lub w przypadku używania jednego z następujących wykresów Helm :

- `ibm-mqadvanced-server-prod`
- `ibm-mqadvanced-server-integration-prod` w IBM Cloud Pak for Integration

**Uwaga:** Użycie wykresów Helm jest nieaktualne, po zwolnieniu operatora IBM MQ .

Ponieważ technologia kontenerowa szybko się rozwija, produkt IBM MQ Advanced certified container jest obsługiwany tylko w najnowszej wersji platform obsługiwanych przez ten wykres w momencie wydania. Jeśli ma być używana starsza wersja platformy, może być konieczne użycie starszej wersji produktu IBM MQ Advanced certified container.

Obraz IBM MQ Advanced certified container jest oparty na wersjach produktu IBM MQ Continuous Delivery (CD). Są one obsługiwane przez maksymalnie jeden rok lub przez okres dwóch wydań CD, w zależności od tego, który z tych okresów jest dłuższy. Wersje produktu Long Term Support produktu IBM MQ nie są dostępne jako certyfikowany kontener.

Począwszy od wersji IBM MQ Advanced certified container V4.0 obraz udostępnia instalację produktu IBM MQ na serwerze Red Hat Universal Base Image (UBI), który zawiera biblioteki kluczy Linux i programy narzędziowe używane przez produkt IBM MQ. Agent UBI jest obsługiwany przez produkt Red Hat , gdy jest uruchamiany na hoście Red Hat Enterprise Linux . Wcześniejsze wersje produktu IBM MQ Advanced certified container korzystały z nieobsługiwanego podstawowego obrazu Ubuntu .

### Pojęcia pokrewne

“[Tworzenie własnych obrazów kontenerów i wykresów IBM MQ](#)” na stronie 8  
Informacje, które należy wziąć pod uwagę, używając kontenerów w systemie Linux.

## Linux → MQ Adv. → CD **Wersja zgodna z operatorem IBM MQ Advanced certified container**

Zestaw tabel z odwzorowaniem między obsługiwanymi wersjami IBM MQ Advanced certified container, IBM MQ, IBM Cloud Kubernetes Service, IBM Cloud Pak for Integration i IBM Cloud Private.

### IBM MQ Operator

► V 9.1.5

Operator IBM MQ jest obsługiwany jako część produktu IBM Cloud Pak for Integration w wersji 2020.2 lub niezależnie od produktu IBM MQ w wersji 9.1.5 i nowszej.

Operator IBM MQ jest obsługiwany w produkcie Red Hat OpenShift Container Platform w wersji 4.4 lub nowszej.

### IBM MQ Advanced certified container ► V 9.1.5 (wykres Helm )-nieaktualny

Zawiera wykres Helm `ibm-mqadvanced-server-prod`.

► V 9.1.5

Począwszy od wersji IBM MQ Advanced certified container V5.0.x , wykres Helm , obraz i poprawki są dostarczane za pośrednictwem serwisu IBM Entitled Catalog and Registry. Wcześniejsze wersje zostały wysłane za pośrednictwem produktu Passport Advantage, a wersje poprawek są dostępne w produkcie IBM Fix Central.

Tabela 1. Obsługa IBM MQ Advanced certified container

| Wersja | Wersja IBM MQ                     | zakończenie obsługi | Obsługiwane platformy                           |
|--------|-----------------------------------|---------------------|---|
| 6.0.x  | 9.1.5 Continuous Delivery Release | Marzec 2021 r.      | <a href="#">Szczegółowe wymagania systemowe</a> |
| 5.0.x  | 9.1.4 Continuous Delivery Release | Grudzień 2020       | <a href="#">Szczegółowe wymagania systemowe</a> |
| 4.1.x  | 9.1.3 Continuous Delivery Release | Lipiec 2020         | <a href="#">Szczegółowe wymagania systemowe</a> |

## Oprogramowanie IBM MQ Advanced certified container dla serwera IBM Cloud Pak for Integration **V 9.1.5** (wykresHelm) -nieaktualne

Zawiera wykres Helm `ibm-mqadvanced-server-integration-prod`.

Tabela 2. Obsługa wersji oprogramowania IBM MQ Advanced certified container w systemie IBM Cloud Pak for Integration

| Wersja | Wersja IBM MQ                     | Wersja IBM Cloud Pak for Integration               |
|--------|-----------------------------------|--|
| 6.0.x  | 9.1.4 Continuous Delivery Release | 2020.1.1 ( <a href="#">Wymagania Systemowe</a> )   |
| 5.0.x  | 9.1.3 Continuous Delivery Release | 2019.4.1 ( <a href="#">Wymagania Systemowe</a> )   |
| 4.1.x  | 9.1.3 Continuous Delivery Release | 2019.3.2.2 ( <a href="#">Wymagania Systemowe</a> ) |
| 4.0.x  | 9.1.3 Continuous Delivery Release | 2019.3.2 ( <a href="#">Wymagania Systemowe</a> )   |
| 3.0.x  | 9.1.3 Continuous Delivery Release | 2019.3.1 ( <a href="#">Wymagania Systemowe</a> )   |

Informacje na temat obsługiwanych wersji można znaleźć w [Uwagach do wydania produktu IBM Cloud Pak for Integration](#).

## Linux Tworzenie własnych obrazów kontenerów i wykresów IBM MQ

Informacje, które należy wziąć pod uwagę, używając kontenerów w systemie Linux.

- Obraz podstawowy używany przez obraz kontenera musi korzystać z obsługiwanego systemu operacyjnego Linux.
- Do zainstalowania produktu wewnątrz obrazu kontenera należy użyć instalatorów produktu IBM MQ.
- Lista obsługiwanych pakietów znajduje się w sekcji [Komponenty rpm IBM MQ dla systemów Linux](#).
- **V 9.1.0** Następujące pakiety nie są obsługiwane:
  - MQSeriesBCBridge
  - MQSeriesRDQM
- Katalog danych menedżera kolejek (domyślnie `/var/mqm`) musi być przechowywany w trwałym woluminie kontenera.

**Ważne:** Nie można używać systemu plików typu Union.

Należy albo podłączyć katalog hosta jako wolumin danych, albo użyć kontenera woluminu danych. Więcej informacji na ten temat zawiera sekcja [Zarządzanie danymi w kontenerach](#).



- Użytkownik musi mieć możliwość uruchamiania komend sterujących IBM MQ, takich jak **endmqm**, w obrębie kontenera.
- Dla celów diagnostycznych konieczna jest możliwość pobierania plików i katalogów z wnętrza kontenera.
- **V 9.1.0** Istnieje możliwość współużytkowania przestrzeni nazw kontenera dla menedżera kolejek z innymi kontenerami — w celu lokalnego wiązania aplikacji z menedżerami kolejek działającymi w oddzielnych kontenerach. Więcej informacji na ten temat zawiera [“Uruchamianie aplikacji z powiązaniem lokalnymi w oddzielnych kontenerach”](#) na stronie 38.

### Pojęcia pokrewne

[“Obsługa kontenerów certyfikowanych przez produkt IBM MQ”](#) na stronie 6

Certyfikowane kontenery produktu IBM MQ są obsługiwane tylko w niektórych środowiskach Kubernetes

## V 9.1.5 Linux MQ Adv. CD Uwagi dotyczące pamięci masowej dla IBM MQ Advanced certified container

IBM MQ Advanced certified container może działać w dwóch trybach pamięci masowej:

- **Efemeryczna pamięć masowa** jest stosowana, gdy w momencie restartu kontenera można usunąć wszystkie informacje o jego dotychczasowym stanie. 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 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. W przypadku użycia efemerycznej pamięci masowej stan menedżera kolejek zostanie utracony po restarcie. Dotyczy to:

- wszystkich komunikatów;
- całości stanu komunikacji między menedżerami kolejek (numerów kolejnych komunikatów w kanale);
- tożsamości 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 OpenShift”](#) na stronie 21). 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 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 dla produktu IBM MQ Advanced certified container”](#) na stronie 10.
- `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:
        enabled: true
        class: ibmc-file-gold-gid
        persistedData:
          enabled: true
          class: ibmc-file-gold-gid
        recoveryLogs:
          enabled: true
          class: ibmc-file-gold-gid
      securityContext:
        supplementalGroups: [99]
```

Linux

MQ Adv.

CD

## Wysoka dostępność dla produktu IBM MQ

### Advanced certified container

Dostępne są dwie główne konfiguracje zapewniające wysoką dostępność IBM MQ Advanced certified container: **wieloinstancyjny menedżer kolejek** (para aktywny-rezerwowo ze wspólnym sieciowym systemem plików) i **pojedynczy menedżer kolejek elastycznych** (proste rozwiązanie zapewniające wysoką dostępność z wykorzystaniem sieciowej pamięci masowej).

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 z 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.

## 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 przełączenie 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 na wykrycie gotowości kontenera przez sondę gotowości zasobnika Kubernetes. Można to skonfigurować na wykresie Helm .
4. Czas potrzebny na ponowne nawiązanie połączenia przez klienty IBM MQ.

## Pojedynczy menedżer kolejek elastycznych

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. Można to skonfigurować na wykresie Helm .
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. Można to skonfigurować na wykresie Helm .
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. Jeśli węzeł główny Kubernetes utraci kontakt z węzłem roboczym, nie może określić, czy węzeł uległ awarii, czy jedynie utracił połączenie z siecią. 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.

### Pojęcia pokrewne

[Konfiguracje o wysokiej dostępności](#)

## Linux > MQ Adv. > CD Uwierzytelnianie użytkownika i autoryzacja dla produktu IBM MQ Advanced certified container

Produkt IBM MQ może być skonfigurowany tak, by używał kont użytkowników i grup LDAP do autoryzacji. Jest to zalecane podejście w przypadku środowiska IBM MQ Advanced certified container.

W środowisku kontenerowym z wieloma podmiotami użytkującymi, takim jak Red Hat OpenShift Container Platform, stosowane są ograniczenia zabezpieczające, które mają zapobiec potencjalnym problemom z bezpieczeństwem. Na przykład w przypadku Red Hat OpenShift Container Platform domyślne ograniczenia `SecurityContextConstraints` (o nazwie `restricted`) zakładają użycie randomizowanego identyfikatora użytkownika, który „zniechęca” użytkowników lokalnych względem samego kontenera. IBM MQ zwykle używa eskalacji uprawnień do sprawdzania hasel użytkowników, co również nie jest zalecane w środowiskach kontenerów z wieloma podmiotami użytkującymi. Dlatego korzystanie z kont użytkowników zdefiniowanych w bibliotekach systemu operacyjnego wewnątrz działającego kontenera nie jest obsługiwane w kontenerach certyfikowanych dla oprogramowania IBM MQ.

Należy skonfigurować menedżera kolejek tak, aby używał protokołu LDAP do uwierzytelniania i autoryzacji użytkowników. Informacje na temat odpowiedniej konfiguracji produktu IBM MQ zawiera sekcja [Uwierzytelnianie połączenia: repozytoria użytkowników i Autoryzacja LDAP](#)

## V 9.1.5 > Linux > MQ Adv. > CD Instalowanie i deinstalowanie operatora IBM MQ na platformie OpenShift

Operator IBM MQ można zainstalować na platformie OpenShift przy użyciu strony Operator Hub.

### Zanim rozpocznesz

#### Procedura

- [“Instalowanie operatora IBM MQ przy użyciu interfejsu wiersza komend OpenShift” na stronie 14.](#)
- [“Instalowanie operatora IBM MQ przy użyciu konsoli WWW OpenShift” na stronie 12.](#)

## V 9.1.5 > Linux > MQ Adv. > CD Instalowanie operatora IBM MQ przy użyciu konsoli WWW OpenShift

Operator IBM MQ można zainstalować na platformie OpenShift przy użyciu strony Operator Hub.

## Zanim rozpoczniesz

Zaloguj się do konsoli WWW klastra OpenShift.

### Procedura

#### 1. Dodawanie operatorów IBM Common Services do listy instalowalnych operatorów

- a) Kliknij ikonę plusa. 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: docker.io/ibmcom/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. 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
  publisher: IBM Content
  sourceType: grpc
  image: docker.io/ibmcom/ibm-operator-catalog
  updateStrategy:
    registryPoll:
      interval: 45m
```

- c) Kliknij opcję **Create**.

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

Operator IBM MQ 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.

#### 4. Zainstaluj operator IBM MQ.

- 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) Ustaw tryb instalacji na konkretną utworzoną przestrzeń nazw lub zasięg całego klastra.

f) Kliknij opcję **Subscribe**.

IBM MQ pojawi się na stronie zainstalowanych operatorów.

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

## Co dalej

[“Wdrażanie certyfikowanych kontenerów produktu IBM MQ” na stronie 16](#)

## V 9.1.5 Linux MQ Adv CD Instalowanie operatora IBM MQ przy użyciu interfejsu wiersza komend OpenShift

Operator IBM MQ można zainstalować na platformie OpenShift przy użyciu strony Operator Hub.

### Zanim rozpocznie

Zaloguj się do interfejsu wiersza komend OpenShift (CLI), używając komendy **oc login**. Aby wykonać te kroki, należy być administratorem klastra.

### Procedura

1. Tworzenie zasobu OperatorSource dla operatorów IBM Common Services

a) Utwórz plik YAML definiujący zasób OperatorSource

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: docker.io/ibmcom/ibm-common-service-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```

b) Zastosuj OperatorSource do serwera.

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

2. Tworzenie zasobu OperatorSource dla operatorów IBM

a) Utwórz plik YAML definiujący zasób OperatorSource

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
  publisher: IBM Content
  sourceType: grpc
  image: docker.io/ibmcom/ibm-operator-catalog
  updateStrategy:
    registryPoll:
      interval: 45m
```

b) Zastosuj OperatorSource 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 IBM MQ może zostać 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

```
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 ma zostać zasubskrybowany operator, musi mieć wartość OperatorGroup zgodną z trybem InstallMode operatora: AllNamespaces albo SingleNamespace. Jeśli operator, który ma zostać zainstalowany, korzysta z trybu AllNamespaces, przestrzeń nazw openshift-operators będzie już miała odpowiedni zasób OperatorGroup.

Jeśli jednak operator korzysta z trybu SingleNamespace, a nie ma jeszcze odpowiedniego zasobu 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>
spec:
  targetNamespaces:
  - <namespace>
```

b) Utwórz obiekt OperatorGroup

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

### 7. Tworzenie pliku YAML obiektu subskrypcji przestrzeni nazw dla operatora MQ

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

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

W przypadku użycia AllNamespaces **InstallMode** określ przestrzeń nazw openshift-operators. W przeciwnym przypadku określ odpowiednią pojedynczą przestrzeń nazw dla użycia przestrzeni SingleNamespace **InstallMode**.

b) 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**.

## Co dalej

[“Wdrażanie certyfikowanych kontenerów produktu IBM MQ” na stronie 16](#)

## Linux > MQ Adv. > CD Wdrażanie certyfikowanych kontenerów produktu IBM MQ

Produkt IBM MQ w wersji 9.1.5 i nowszej może zostać wdrożony w produkcie Red Hat OpenShift przy użyciu operatora IBM MQ. Produkty IBM MQ w wersjach 9.1.5 i 9.1.4 można wdrażać w produkcie Red Hat OpenShift za pomocą komendy Helm. Wcześniejsze wersje CD można wdrażać w klastrze produktu IBM Cloud Private lub w klastrze produktu IBM Cloud Kubernetes Service za pomocą komendy Helm.

## O tym zadaniu

### Procedura

- [“Wdrażanie menedżera kolejek przy użyciu interfejsu CLI Helm” na stronie 25.](#)
- [“Wdrażanie wcześniejszych wersji CD produktu IBM MQ w klastrze produktu IBM Cloud Private” na stronie 27.](#)
- [“Dodawanie wcześniejszych wersji dysku CD obrazu IBM MQ do klastra IBM Cloud Private” na stronie 29.](#)
- [“Dodawanie wcześniejszych wersji dysku CD obrazu IBM MQ do klastra IBM Cloud Kubernetes Service” na stronie 30.](#)

## Linux > MQ Adv. > CD Przygotowywanie projektu OpenShift dla produktu IBM MQ przy użyciu interfejsu wiersza komend OpenShift

Przygotuj klaster platformy Red Hat OpenShift Container Platform, aby był gotowy do wdrożenia menedżera kolejek za pomocą operatora IBM MQ. To zadanie powinien wykonać administrator projektu.

## Zanim rozpocznie

**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 Advanced certified container 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 za pomocą interfejsu wiersza komend OpenShift” na stronie 19](#)

## V 9.1.5 Linux MQ Adv. CD 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 rozpoczniesz

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 OpenShift dla produktu IBM MQ przy użyciu interfejsu wiersza komend OpenShift” na stronie 16](#).

### 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 programie IBM Cloud Pak for Integration Platform Navigator kliknij opcję **Runtime and instances**.

- b) Kliknij opcję **Create instance**.

- c) Wybierz pozycję **Queue Manager** i 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 Config** sprawdź lub zaktualizuj w polu **Name** nazwę bazowego menedżera kolejek.

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). Aby wymusić użycie innej nazwy, można ją edytować tutaj.

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

“[Nawiązywanie połączenia z menedżerem kolejek wdrożonym w klastrze OpenShift](#)” na stronie 30  
Zestaw przykładów konfiguracji na potrzeby nawiązywania połączenia z menedżerem kolejek wdrożonym w klastrze Red Hat OpenShift.

“[Nawiązywanie połączenia z konsolą IBM MQ Console wdrożoną w klastrze OpenShift](#)” na stronie 32  
W jaki sposób nawiązać połączenie z konsolą IBM MQ Console menedżera kolejek, która została wdrożona w klastrze Red Hat OpenShift Container Platform.

## V 9.1.5 Linux MQ Adv. CD Wdrażanie menedżera kolejek przy użyciu konsoli WWW 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 rozpocznie

Zaloguj się do konsoli WWW klastra 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 OpenShift dla produktu IBM MQ przy użyciu interfejsu wiersza komend OpenShift](#)” na stronie 16.

### Procedura

1. Wdróż menedżera kolejek.

W poniższym przykładzie wdrażany jest menedżer kolejek „quick start”, który używa pamięci masowej efemerycznej (nie trwałe) 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 OpenShift, na panelu nawigacyjnym, kliknij opcję **Operators > Installed Operators**.

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 40. 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

“Nawiązywanie połączenia z menedżerem kolejek wdrożonym w klastrze OpenShift” na stronie 30  
Zestaw przykładów konfiguracji na potrzeby nawiązywania połączenia z menedżerem kolejek wdrożonym w klastrze Red Hat OpenShift.

“Nawiązywanie połączenia z konsolą IBM MQ Console wdrożoną w klastrze OpenShift” na stronie 32  
W jaki sposób nawiązać połączenie z konsolą IBM MQ Console menedżera kolejek, która została wdrożona w klastrze Red Hat OpenShift Container Platform.

## V 9.1.5 Linux MQ Adv. CD Wdrażanie menedżera kolejek za pomocą interfejsu wiersza komend 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 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 jest to pierwsze wdrożenie menedżera kolejek w tym projekcie Red Hat OpenShift, należy wykonać kroki opisane w sekcji “Przygotowywanie projektu OpenShift dla produktu IBM MQ przy użyciu interfejsu wiersza komend OpenShift” na stronie 16.

### 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: 9.1.5.0-r2
  license:
    accept: false
    license: L-RJON-BN7PN3
    use: NonProduction
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
    storage:
      queueManager:
        type: ephemeral
  template:
    pod:
      containers:
        - name: qmgr
          env:
            - name: MQSNOAUT
              value: "yes"
```

**Ważne:**Jeśli zaakceptujesz umowę licencyjną IBM Cloud Pak for Integration , zmień `accept: false` na `accept: true`. Aby uzyskać szczegółowe informacje na temat licencji, patrz [“Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1”](#) na stronie 40.

Ten przykład obejmuje również serwer WWW wdrożony wraz z menedżerem kolejek, przy czym konsola WWW obsługuje pojedyncze logowanie przy wykorzystaniu rozwiązania 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: 9.1.5.0-r2
  license:
    accept: false
    license: L-APIG-BM7GDH
    use: Development
  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 40.

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

[“Nawiązywanie połączenia z menedżerem kolejek wdrożonym w klastrze OpenShift”](#) na stronie 30

Zestaw przykładów konfiguracji na potrzeby nawiązywania połączenia z menedżerem kolejek wdrożonym w klastrze Red Hat OpenShift.

[“Nawiązywanie połączenia z konsolą IBM MQ Console wdrożoną w klastrze OpenShift”](#) na stronie 32

W jaki sposób nawiązać połączenie z konsolą IBM MQ Console menedżera kolejek, która została wdrożona w klastrze Red Hat OpenShift Container Platform.

## 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.

### 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 17

W przypadku wdrażania przy użyciu [interfejsu wiersza komend OpenShift](#) lub [konsoli WWW OpenShift](#) można włączyć śledzenie za pomocą następującego fragmentu kodu YAML:

```
spec:
  tracing:
    enabled: true
    namespace: <Operations_Dashboard_Namespace
```

W przypadku wdrażania za pomocą opcji Helm można włączyć śledzenie, ustawiając opcje `odTracingConfig.enabled=true` i `odTracingConfig.odTracingNamespace=<Operations_Dashboard_Namespace`. Aby włączyć integrację z panelem kontrolnym operacji na istniejącym menedżerze kolejek, można zastosować to ustawienie podczas aktualizowania wersji Helm.

**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 jest to pierwsze wdrożenie menedżera kolejek z integracją panelu kontrolnego operacji w tej przestrzeni nazw, konieczne jest [zarejestrowanie](#) w panelu kontrolnym operacji.

Rejestracja powoduje utworzenie obiektu Secret potrzebnego do uruchomienia zasobnika menedżera kolejek.

## Budowanie obrazu z niestandardowymi plikami MQSC i INI przy użyciu interfejsu wiersza komend 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 Red Hat OpenShift nie ma sekretu OpenShift dla rejestru IBM Entitled Registry, należy wykonać czynności opisane w sekcji [“Przygotowywanie projektu OpenShift dla produktu IBM MQ przy użyciu interfejsu wiersza komend OpenShift”](#) na stronie 16.

## 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:9.1.5.0-r2-amd64
      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:9.1.5.0-r2-amd64"
      pullSecret:
        name: ibm-entitlement-key
  output:
    to:
      kind: ImageStreamTag
      name: 'mymq:latest-amd64'
```

Aby wskazać poprawny obraz bazowy wersji i poprawki, który ma być używany, należy zastąpić nazwy w dwóch miejscach, w których jest wymieniony bazowy obraz IBM MQ. 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 za pomocą interfejsu wiersza komend OpenShift”](#) na stronie 19, dodając nowy obraz niestandardowy do kodu YAML.

Do normalnego kodu YAML zasobu `QueueManager` można dodać następujący fragment kodu, w którym *moja-przestrzeń nazw* to projekt/przestrzeń nazw OpenShift, a *obraz* 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 za pomocą interfejsu wiersza komend OpenShift”](#) na stronie 19

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.

Linux

MQ Adv.

CD

## Wdrażanie certyfikowanych kontenerów IBM MQ za pomocą Helm

Od wersji IBM MQ 9.1.5.0 zalecanym sposobem wdrożenia menedżera kolejek jest użycie operatora IBM MQ. Produkt IBM MQ 9.1.5.0 i poprzednie wydania CD można wdrażać za pomocą komendy Helm, korzystając z poniższych instrukcji.

### O tym zadaniu

#### Procedura

- [“Przygotowanie klastra OpenShift dla systemu IBM MQ w systemie OpenShift za pomocą komendy Helm”](#) na stronie 23.
- [“Wdrażanie menedżera kolejek przy użyciu interfejsu CLI Helm”](#) na stronie 25.

Linux

MQ Adv.

CD

## Przygotowanie klastra OpenShift dla systemu IBM MQ w systemie OpenShift za pomocą komendy Helm

Przygotuj klaster platformy kontenera produktu Red Hat OpenShift, aby był on gotowy do wdrożenia menedżera kolejek za pomocą komendy Helm. To zadanie powinno zostać wykonane przez administratora klastra.

## Zanim rozpocznieš

**Uwaga:** Jeŝli uŝywany jest produkt IBM Cloud Pak for Integration, instalator powinien przygotować projekt OpenShift (przestrzeŝ nazw), który będzie uŝywany razem z produktem IBM MQ, dlatego nie trzeba wykonywać tych instrukcji.

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

## Procedura

1. Upewnij się, ŝe dodałeś repozytorium IBM Helm do lokalnej kopii Helm.  
Na przykłał można uruchomić następującą komendę:

```
helm repo add ibm-entitled-charts https://raw.githubusercontent.com/IBM/charts/master/repo/entitled
```

2. Upewnij się, ŝe masz serwer Helm (o nazwie Tiller) zainstalowanych w klastrze.  
Postępuj zgodnie z instrukcjami w sekcji [Pierwsze kroki z programem Helm na OpenShift](#), aby zainstalować Helm w klastrze.
3. Upewnij się, ŝe konta usługi w projekcie OpenShift (przestrzeŝ nazw) są autoryzowane do korzystania z odpowiednich ograniczeŝ kontekstu zabezpieczeŝ (SCCs).

**V 9.1.5** IBM MQ działa w ramach domyŝlnego SCC "zastrzeŝonego", więc ten krok może być normalnie pominięty.

Zmiany w SCC muszą być wykonywane przez administratora klastra OpenShift. Kaŝda wersja wykresu Helm ma inne wymagania dla SCC, które są udokumentowane w indywidualnym pliku README dla tego wykresu Helm:

```
helm inspect readme ibm-entitled-charts/ibm-mqadvanced-server-prod
```

W kaŝdym pliku README znajdujĄ się instrukcje dotyczĄce konfigurowania autoryzacji dla certyfikatów SCC. Należy pamiętać, ŝe wykresy IBM MQ Helm tworzĄ konto usługi dla własnego uŝytku, co oznacza, ŝe uprawnienia SCC muszą być stosowane na poziomie "group" (dla wszystkich kont usług w przestrzeni nazw).

4. Aby pobrać obrazy z wybranego rejestru kontenerów, upewnij się, ŝe masz poprawny "ukryj tajemnicę obrazu"

Obrazy IBM MQ Advanced certified container są pobierane z rejestru kontenerów, który sprawdza uprawnienia licencyjne. To sprawdzenie wymaga klucza uprawnieŝ, który jest przechowywany w sekrecie pobierania `docker-registry`. Jeŝli klucz uprawnieŝ nie jest jeszcze dostępuj, należy wykonać poniŝsze instrukcje, aby uzyskać klucz i utworzyć sekret pobierania.

- a) Uzyskaj klucz uprawnieŝ przypisany do swojego identyfikatora.
  - i) Zaloguj się do biblioteki MyIBM Container Software Library przy uŝyciu identyfikatora IBM ID i hasła, które są powiązane z uprawnieniem do oprogramowania.
  - ii) W sekcji *Entitlement keys* wybierz opcję **Copy key**, aby skopiować klucz uprawnieŝ do schowka.
- b) Utwórz klucz tajny w przestrzeni nazw, w której ma zostać wdroŝony menedŝer kolejek.
  - Uruchom następującą komendę, gdzie `<entitlement-key>` jest kluczem pobranym w kroku 1, a `<user-email>` jest identyfikatorem IBM powiązanym z przysługującym 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

["Wdraŝanie menedŝera kolejek przy uŝyciu interfejsu CLI Helm" na stronie 25](#)



## użyciu interfejsu CLI Helm

Użyj opcji Helm , aby wdrożyć menedżera kolejek w klastrze platformy kontenera produktu Red Hat OpenShift . To zadanie powinien wykonać administrator projektu.

### Zanim rozpoczniesz

Należy zainstalować opcje Helm V2 i Interfejs wiersza komend Red Hat OpenShift Container Platform. Jeśli produkt IBM Cloud Pak for Integration jest używany, należy wykonać kroki opisane w sekcji [“Przygotowanie klastra OpenShift dla systemu IBM MQ w systemie OpenShift za pomocą komendy Helm”](#) na stronie 23.

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

### Procedura

1. Upewnij się, że dodałeś repozytorium IBM Helm do lokalnej kopii Helm.  
Na przykład można uruchomić następującą komendę:

```
helm repo add ibm-entitled-charts https://raw.githubusercontent.com/IBM/charts/master/repo/entitled
```

2. Przegląd opcji konfiguracyjnych dla menedżera kolejek

Krok wdrażania obejmuje kroki instalacji i konfiguracji. Niektóre ustawienia menedżera kolejek muszą być ustawione w czasie wdrażania, a zmiana ich wymaga ponownego wdrożenia.

Aby wyświetlić szczegółowe informacje na temat wszystkich dostępnych opcji wdrażania, można wyświetlić plik README wykresu Helm , uruchamiając jedną z następujących komend:

- W przypadku produktu IBM MQ Advanced certified container w produkcie IBM Cloud Pak for Integration:

```
helm inspect readme ibm-entitled-charts/ibm-mqadvanced-server-integration-prod
```

- IBM MQ Advanced certified container:

```
helm inspect readme ibm-entitled-charts/ibm-mqadvanced-server-prod
```

Zwykle potrzebne są co najmniej następujące parametry:

- a. Nazwa wydania. Na przykład: `my-release`
  - b. Zdalne repozytorium Helm . Na przykład: `ibm-entitled-charts`
  - c. Wykres Helm : na przykład `ibm-mqadvanced-server-prod` lub `ibm-mqadvanced-server-integration-prod`
  - d. Nazwa tajnego elementu przyciągania obrazu. Na przykład: `entitled-registry`. Należy zwrócić uwagę, że nie jest to potrzebne, jeśli wdrażany jest predefiniowany projekt dla produktu MQ w produkcie IBM Cloud Pak for Integration .
3. Wdróż menedżera kolejek.

Należy zwrócić uwagę, że domyślnie na wykresie Helm założono, że w klastrze platformy kontenera produktu Red Hat OpenShift ustawiono domyślną wartość Klasa pamięci masowej .

Na przykład, aby zainstalować podstawowy menedżer kolejek w produkcie IBM Cloud Pak for Integration, należy uruchomić następującą komendę:

```
helm install \
--tls \
--name my-release \
ibm-entitled-charts/ibm-mqadvanced-server-integration-prod \
--set license=accept \
```

```
--set tls.hostname=my.cluster \  
--set tls.generate=true
```

W polu `tls.hostname` można wprowadzić dowolną nazwę hosta (jest to pole wymagane, ale nie będzie używane, ponieważ w tym przykładzie wygenerujemy nowy certyfikat samopodpisany).

Aby zainstalować podstawowy menedżer kolejek niezależnie od produktu IBM Cloud Pak for Integration, można uruchomić następującą komendę:

```
helm install \  
--name my-release \  
ibm-entitled-charts/ibm-mqadvanced-server-prod \  
--set license=accept \  
--set image.pullSecret=ibm-entitlement-key
```

## Zadania pokrewne

[“Nawiązywanie połączenia z menedżerem kolejek wdrożonym w klastrze OpenShift” na stronie 30](#)

Zestaw przykładów konfiguracji na potrzeby nawiązywania połączenia z menedżerem kolejek wdrożonym w klastrze Red Hat OpenShift.

[“Nawiązywanie połączenia z konsolą IBM MQ Console wdrożoną w klastrze OpenShift” na stronie 32](#)

W jaki sposób nawiązać połączenie z konsolą IBM MQ Console menedżera kolejek, która została wdrożona w klastrze Red Hat OpenShift Container Platform.

## V 9.1.5 Linux MQ Adv. CD **Wdrażanie menedżera kolejek przy użyciu produktu IBM Cloud File Storage przy użyciu interfejsu CLI Helm**

Przykładowy scenariusz do użycia komendy Helm w celu wdrożenia menedżera kolejek na serwerze Red Hat OpenShift w klastrze IBM Cloud przy użyciu programu IBM Cloud File Storage. To zadanie powinien wykonać administrator projektu.

## Zanim rozpoczniesz

Należy zainstalować opcje [Helm V2](#) i [Interfejs wiersza komend Red Hat OpenShift Container Platform](#).

Jeśli produkt IBM Cloud Pak for Integration nie jest używany, należy wykonać kroki opisane w sekcji [“Przygotowanie klastra OpenShift dla systemu IBM MQ w systemie OpenShift za pomocą komendy Helm” na stronie 23](#).

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

## Procedura

1. Upewnij się, że dodałeś repozytorium IBM Helm do lokalnej kopii Helm.

Na przykład można uruchomić następującą komendę:

```
helm repo add ibm-entitled-charts https://raw.githubusercontent.com/IBM/charts/master/repo/entitled
```

2. Wdróż menedżera kolejek.

W przypadku korzystania z produktu IBM Cloud File Storage zwykle są wyświetlane najlepsze wyniki z użyciem klasy pamięci masowej `ibmc-file-gold-gid`. Ta klasa pamięci masowej umożliwia przechowywanie danych, które mogą być zapisywane przez użytkowników w poprawnej grupie systemów plików.

Na przykład, aby zainstalować podstawowy menedżer kolejek w produkcie IBM Cloud Pak for Integration, należy uruchomić następującą komendę:

```
helm install \  
--tls \  
--name my-release \  
ibm-entitled-charts/ibm-mqadvanced-server-integration-prod \  
--set license=accept \  
--set tls.hostname=my.cluster \  
--set tls.generate=true \  

```

```
--set dataPVC.storageClassName=ibmc-file-gold-gid \  
--set security.context.supplementalGroups={99}
```

W polu `tls.hostname` można wprowadzić dowolną nazwę hosta (jest to pole wymagane, ale nie jest ono używane w tym miejscu, ponieważ w tym przykładzie generujemy nowy certyfikat samopodpisany).

Aby zainstalować podstawowy menedżer kolejek niezależnie od produktu IBM Cloud Pak for Integration, można uruchomić następującą komendę:

```
helm install \  
--name my-release \  
ibm-entitled-charts/ibm-mqadvanced-server-prod \  
--set license=accept \  
--set image.pullSecret=ibm-entitlement-key \  
--set dataPVC.storageClassName=ibmc-file-gold-gid \  
--set security.context.supplementalGroups={99}
```

### Zadania pokrewne

[“Nawiązywanie połączenia z menedżerem kolejek wdrożonym w klastrze OpenShift” na stronie 30](#)  
Zestaw przykładów konfiguracji na potrzeby nawiązywania połączenia z menedżerem kolejek wdrożonym w klastrze Red Hat OpenShift.

[“Nawiązywanie połączenia z konsolą IBM MQ Console wdrożoną w klastrze OpenShift” na stronie 32](#)  
W jaki sposób nawiązać połączenie z konsolą IBM MQ Console menedżera kolejek, która została wdrożona w klastrze Red Hat OpenShift Container Platform.

Linux

MQ Adv.

CD

## Wdrażanie wcześniejszych wersji CD produktu IBM MQ w klastrze produktu IBM Cloud Private

W przypadku wersji CD produktu IBM MQ wcześniejszych niż 9.1.4 należy użyć konsoli zarządzania IBM Cloud Private w celu wdrożenia menedżera kolejek w produkcie IBM Cloud Private.

### Zanim rozpocznie



**Ostrzeżenie:** **V 9.1.4** To wdrożenie nie jest obsługiwane w wersjach produktu IBM MQ 9.1.4 i nowszych.

W przypadku tego zadania założono, że został dodany obraz IBM MQ do klastra IBM Cloud Private.

Plik `README.md` wykresu Helm jest dostępny z pozycji katalogu IBM Cloud Private, która jest wyświetlana po zakończeniu tego kroku podrzędnego lub z poziomu wiersza komend, dodając repozytorium **local-charts** IBM Cloud Private jako zdalne repozytorium Helm i uruchamiając następującą komendę:

```
helm inspect readme remote_repo_name/ibm-mqadvanced-server-prod
```

Konieczne jest posiadanie strategii `PodSecurity` lub `SecurityContext` (dla produktu IBM Cloud Private w systemie Red Hat OpenShift), które obsługują wymagany kontekst zabezpieczeń. Szczegółowe informacje, w tym przykłady, można znaleźć w pliku `README.md` wykresu Helm.

Szczegółowe informacje na temat sposobu konfigurowania wersji Helm można znaleźć również w pliku `README.md` wykresu Helm.

### Uwaga:

- W przypadku wdrażania w środowisku IBM Cloud Private, które domyślnie nie obsługuje wymaganych ustawień zabezpieczeń, należy włączyć wdrażanie, postępując zgodnie z instrukcjami w sekcji [Wdrażanie wykresów Helm](#), które wymagają podwyższonych uprawnień w przestrzeni nazw innej niż domyślna w dokumentacji produktu IBM Cloud Private.
- Jeśli używany jest system SELinux, należy spełnić wymagania IBM MQ opisane w sekcji [Obsługa produktu IBM MQ dla systemu SELinux w systemie Red Hat Enterprise Linux](#).

## O tym zadaniu

Produkt IBM Cloud Private oferuje platformę do zarządzania aplikacjami w kontenerach, w kontenerach. Po dodaniu obrazu IBM MQ do klastra IBM Cloud Private można użyć konsoli zarządzania IBM Cloud Private lub wiersza komend w celu wdrożenia menedżera kolejek.

## Procedura

- Korzystanie z konsoli IBM Cloud Private Management Console
  - a) Otwórz konsolę zarządzania IBM Cloud Private w przeglądarce WWW, a następnie kliknij opcję **Katalog**.

Więcej informacji na ten temat zawiera sekcja [Uzyskiwanie dostępu do klastra produktu IBM Cloud Private przy użyciu konsoli zarządzania w dokumentacji produktu IBM Cloud Private](#).
  - b) Wybierz z listy wykres `ibm-mqadvanced-server-prod`.
  - c) Wybierz opcję **Konfiguruj**, a następnie wykonaj następujące kroki konfiguracyjne:
    - a. Wprowadź nazwę wersji.
    - b. Przeczytaj i zaakceptuj umowy licencyjne.
    - c. W sekcji **dataPVC** ustaw wartość **storageclass** na żadaną klasę pamięci masowej. Pozostaw puste pole, aby wybrać domyślną klasę pamięci masowej.
    - d. W sekcji **image** (Obraz) ustaw repozytorium na pełną ścieżkę obrazu. Na przykład:

```
mycluster.icp:8500/namespace_name/ibm-mqadvanced-server-prod
```

- e. W sekcji **image** (obraz) ustaw znacznik na znacznik obrazu. Na przykład:

```
9.1.3.0-r1
```

- f. Jeśli do uzyskania dostępu do rejestru obrazów wymagany jest klucz tajny pobierania Kubernetes, należy dodać go jako **pullSecret**.
- g. W sekcji **queueManager** (Menedżer kolejek) ustaw nazwę menedżera kolejek.

- d) Kliknij przycisk **Instaluj**, aby wdrożyć menedżera kolejek jako *Helm*.
- Za pomocą wiersza komend
    - a) Skonfiguruj produkt **cloudctl**, aby uzyskać dostęp do klastra produktu IBM Cloud Private.

Więcej informacji na ten temat zawiera sekcja [Instalowanie interfejsu CLI produktu IBM Cloud Private](#) w dokumentacji produktu IBM Cloud Private.
    - b) Upewnij się, że dodano repozytorium **local-charts** produktu IBM Cloud Private jako zdalne repozytorium Helm.
    - c) Zainstaluj wykres.

Uruchom następującą komendę, podając następujące parametry:

      - a. Nazwa wydania (na przykład `my-release`)
      - b. Nazwa zdalnego repozytorium helm zawierającego wykres `ibm-mqadvanced-server-prod` (na przykład `my-repo`)
      - c. Repozytorium obrazów (na przykład `mycluster.icp:8500/namespace_name/ibm-mqadvanced-server-prod`)
      - d. Znacznik obrazu (na przykład `9.1.3.0-r1`)

```
helm install --name my-release --repo my-repo ibm-mqadvanced-server-prod --set license=accept --set image.repository=mycluster.icp:8500/namespace_name/ibm-mqadvanced-server-prod --set image.tag=9.1.3.0-r1 --tls
```

## Zadania pokrewne

[“Wdrażanie menedżera kolejek przy użyciu interfejsu CLI Helm” na stronie 25](#)

Użyj opcji Helm , aby wdrożyć menedżera kolejek w klastrze platformy kontenera produktu Red Hat OpenShift . To zadanie powinien wykonać administrator projektu.

[“Dodawanie wcześniejszych wersji dysku CD obrazu IBM MQ do klastra IBM Cloud Private” na stronie 29](#)  
W przypadku wersji CD produktu IBM MQ wcześniejszych niż 9.1.4 należy przygotować klaster produktu IBM Cloud Private w celu wdrożenia obrazu gotowego do użytku produkcyjnego dla produktu IBM MQ.

[“Dodawanie wcześniejszych wersji dysku CD obrazu IBM MQ do klastra IBM Cloud Kubernetes Service” na stronie 30](#)

W przypadku wersji CD produktu IBM MQ wcześniejszych niż 9.1.4 należy zaimportować obraz gotowy do użytku produkcyjnego dla produktu IBM MQ do produktu IBM Cloud Kubernetes Service.

Linux

MQ Adv.

CD

## Dodawanie wcześniejszych wersji dysku CD obrazu IBM MQ do klastra IBM Cloud Private

W przypadku wersji CD produktu IBM MQ wcześniejszych niż 9.1.4 należy przygotować klaster produktu IBM Cloud Private w celu wdrożenia obrazu gotowego do użytku produkcyjnego dla produktu IBM MQ.

### O tym zadaniu



**Ostrzeżenie:** **V 9.1.4** Ten import nie jest obsługiwany w wersjach produktu IBM MQ 9.1.4 i nowszych.

Obraz produktu IBM MQ można pobrać z programu Passport Advantage i zaimportować go do kontenera produktu IBM Cloud Private .

### Procedura

1. Pobierz najnowszy obraz IBM MQ z [Serwis WWW Passport Advantage i Passport Advantage Express](#).

Aby uzyskać szczegółowe informacje o dostępnych plikach do pobrania, należy przejść do strony [Pobieranie produktu IBM MQ 9.1](#) , a następnie kliknąć kartę w celu pobrania wersji, która ma zostać pobrana. Nazwa i numer części do pobrania są wymienione w tabeli.

2. Zaimportuj pobrany plik archiwum do programu IBM Cloud Private.

Więcej informacji na ten temat zawiera sekcja [Dodawanie oprogramowania IBM do katalogu IBM Cloud Private](#) w dokumentacji produktu IBM Cloud Private .

### Co dalej

Teraz można przystąpić do [wdrażania menedżera kolejek w produkcie IBM Cloud Private](#).

#### Zadania pokrewne

[“Wdrażanie menedżera kolejek przy użyciu interfejsu CLI Helm” na stronie 25](#)

Użyj opcji Helm , aby wdrożyć menedżera kolejek w klastrze platformy kontenera produktu Red Hat OpenShift . To zadanie powinien wykonać administrator projektu.

[“Wdrażanie wcześniejszych wersji CD produktu IBM MQ w klastrze produktu IBM Cloud Private” na stronie 27](#)

W przypadku wersji CD produktu IBM MQ wcześniejszych niż 9.1.4 należy użyć konsoli zarządzania IBM Cloud Private w celu wdrożenia menedżera kolejek w produkcie IBM Cloud Private.

[“Dodawanie wcześniejszych wersji dysku CD obrazu IBM MQ do klastra IBM Cloud Kubernetes Service” na stronie 30](#)

W przypadku wersji CD produktu IBM MQ wcześniejszych niż 9.1.4 należy zaimportować obraz gotowy do użytku produkcyjnego dla produktu IBM MQ do produktu IBM Cloud Kubernetes Service.

## Dodawanie wcześniejszych wersji dysku CD obrazu IBM MQ do klastra IBM Cloud Kubernetes Service

W przypadku wersji CD produktu IBM MQ wcześniejszych niż 9.1.4 należy zaimportować obraz gotowy do użytku produkcyjnego dla produktu IBM MQ do produktu IBM Cloud Kubernetes Service.

### O tym zadaniu



**Ostrzeżenie:** **V 9.1.4** Ten import nie jest obsługiwany w wersjach produktu IBM MQ 9.1.4 i nowszych.

Obraz IBM MQ można pobrać z programu Passport Advantage i zaimportować go do klastra IBM Cloud Kubernetes Service .

### Procedura

1. Pobierz najnowszy obraz IBM MQ z [Serwis WWW Passport Advantage](#) i [Passport Advantage Express](#).

Aby uzyskać szczegółowe informacje o dostępnych plikach do pobrania, należy przejść do strony [Pobieranie produktu IBM MQ 9.1](#) , a następnie kliknąć kartę w celu pobrania wersji, która ma zostać pobrana. Nazwa i numer części do pobrania są wymienione w tabeli.

2. Zaimportuj pobrany plik archiwum do programu IBM Cloud Kubernetes Service.

Patrz [Uruchamianie obrazów IBM Cloud Private w publicznych kontenerach Kubernetes](#).

### Zadania pokrewne

“[Wdrażanie menedżera kolejek przy użyciu interfejsu CLI Helm](#)” na stronie 25

Użyj opcji Helm , aby wdrożyć menedżera kolejek w klastrze platformy kontenera produktu Red Hat OpenShift . To zadanie powinien wykonać administrator projektu.

“[Wdrażanie wcześniejszych wersji CD produktu IBM MQ w klastrze produktu IBM Cloud Private](#)” na stronie 27

W przypadku wersji CD produktu IBM MQ wcześniejszych niż 9.1.4 należy użyć konsoli zarządzania IBM Cloud Private w celu wdrożenia menedżera kolejek w produkcie IBM Cloud Private.

“[Dodawanie wcześniejszych wersji dysku CD obrazu IBM MQ do klastra IBM Cloud Private](#)” na stronie 29

W przypadku wersji CD produktu IBM MQ wcześniejszych niż 9.1.4 należy przygotować klaster produktu IBM Cloud Private w celu wdrożenia obrazu gotowego do użytku produkcyjnego dla produktu IBM MQ.

## Nawiązywanie połączenia z menedżerem kolejek wdrożonym w klastrze OpenShift

Zestaw przykładów konfiguracji na potrzeby nawiązywania połączenia z menedżerem kolejek wdrożonym w klastrze Red Hat OpenShift.

### O tym zadaniu

Aby połączyć aplikację z menedżerem kolejek produktu IBM MQ z poziomu klastra Red Hat OpenShift , należy użyć konsoli [Trasa OpenShift](#) .

Należy włączyć obsługę protokołu TLS w menedżerze kolejek IBM MQ i aplikacji klienckiej, ponieważ wskaźnik Server Name Indication (SNI) jest dostępny tylko w protokole TLS. Router Red Hat OpenShift Container Platform używa wskaźnika SNI do kierowania żądań do menedżera kolejek IBM MQ.

Wymagana konfiguracja trasy OpenShift zależy od tego, jak aplikacja kliencka traktuje wskaźnik SNI.

Aby ustawić nagłówek SNI jako TLS 1.2 lub nowszy, do komunikacji TLS musi być używana specyfikacja CipherSpec lub CipherSuite .

Wartość SNI jest ustawiana na kanał MQ , jeśli spełnione są następujące warunki:

- Klient IBM MQ C ma wartość V8 lub nowszą.
- Klient Java/JMS Client to V9.1.1 lub nowszy, a instalacja Java obsługuje klasę `javax.net.ssl.SNIHostName`.
- Klient .NET jest w trybie niezarządzanym.

Wartość SNI jest ustawiana na nazwę hosta, jeśli nazwa hosta jest podana jako nazwa połączenia, a spełnione są następujące warunki:

- Klient .NET jest w trybie zarządzanym.
- Używany jest klient AMQP lub XR.
- Klienci Java/JMS są używane z produktem **AllowOutboundSNI** ustawionym na wartość NO.

Wartość SNI nie jest ustawiona i jest pusta, jeśli spełnione są następujące warunki:

- Klient IBM MQ C to V7.5 lub wcześniejsza.
- Klient IBM MQ C jest używany z produktem **AllowOutboundSNI** ustawionym na wartość NO.
- Klienci Java/JMS są używane z instalacją Java, która nie obsługuje klasy `javax.net.ssl.SNIHostName`.

## Przykład

### **Nazwa hosta oparta na nazwie OpenShift Routes: Dla aplikacji klienckich, które ustawiają SNI na nazwę hosta**

Następujące wykresy Helm automatycznie tworzą grupę OpenShift opartą na nazwie hosta dla połączenia aplikacji z menedżerem kolejek produktu IBM MQ. Aplikacje klienckie, które ustawiają SNI na nazwę hosta, mogą używać tej trasy OpenShift.

- `ibm-mqadvanced-server-dev`
- `ibm-mqadvanced-server-prod`
- `ibm-mqadvanced-server-integration-prod` w IBM Cloud Pak for Integration.

Jeśli te wykresy nie są używane i konieczne jest utworzenie własnej nazwy hosta OpenShift Route, można zastosować następujące `yaml` w klastrze:

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: <provide a unique name for the Route>
  namespace: <namespace of your MQ deployment>
spec:
  to:
    kind: Service
    name: <name of the Kubernetes Service for your MQ deployment (for example "<Helm Release>-ibm-mq")>
  port:
    targetPort: 1414
  tls:
    termination: passthrough
```

### **Routy OpenShift oparte na kanale MQ : Dla aplikacji klienckich, które ustawiają SNI na kanał MQ**

Aplikacje klienckie, które ustawiają SNI na kanał MQ, wymagają utworzenia nowej trasy OpenShift dla każdego kanału, z którym ma zostać nawiązane połączenie. Konieczne jest również użycie unikalnych nazw kanałów w klastrze Red Hat OpenShift, aby umożliwić kierowanie do właściwego menedżera kolejek.

Aby ustalić wymaganą nazwę hosta dla każdej nowej trasy OpenShift, należy odwzorować każdą nazwę kanału na adres SNI, zgodnie z opisem zamieszczonym tutaj: <https://www.ibm.com/support/pages/ibm-websphere-mq-how-does-mq-provide-multiple-certificates-certlabl-capability>

Następnie należy utworzyć nową trasę OpenShift (dla każdego kanału), stosując następujący kod yam1 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 "<Helm Release>-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 "<Helm Release>-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 OpenShift Container Platform (OCP)-zwykle 443.

### **Zadania pokrewne**

[“Wdrażanie menedżera kolejek przy użyciu interfejsu CLI Helm” na stronie 25](#)

Użyj opcji Helm , aby wdrożyć menedżera kolejek w klastrze platformy kontenera produktu Red Hat OpenShift . To zadanie powinien wykonać administrator projektu.

[“Nawiązywanie połączenia z konsolą IBM MQ Console wdrożoną w klastrze OpenShift” na stronie 32](#)

W jaki sposób nawiązać połączenie z konsolą IBM MQ Console menedżera kolejek, która została wdrożona w klastrze Red Hat OpenShift Container Platform.

Linux

MQ Adv.

CD

V 9.1.4

## **Nawiązywanie połączenia z konsolą IBM MQ Console wdrożoną w klastrze 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**

Jeśli używany jest operator IBM MQ , adres URL produktu IBM MQ Console można znaleźć na stronie szczegółów produktu QueueManager w konsoli WWW OpenShift lub w konsoli IBM Cloud Pak for Integration Platform Navigator. Inną możliwością jest użycie interfejsu wiersza komend OpenShift i uruchomienie następującej komendy:

```
oc get queuemanager <QueueManager Name> -n <namespace of your MQ deployment> --output
jsonpath='{.status.adminUiUrl}'
```

### **Przykład**

Następujące wykresy Helm automatycznie tworzą trasę OpenShift w celu uzyskania dostępu do konsoli IBM MQ Console .

- ibm-mqadvanced-server-dev
- ibm-mqadvanced-server-integration-prod w IBM Cloud Pak for Integration.

Nazwę hosta trasy OpenShift można uzyskać, uruchamiając następującą komendę:



```
oc get route <Route Name (for example "<Helm Release>-ibm-mq-web")>
-n <namespace of your MQ deployment> --output jsonpath='{.spec.host}'
```

Dostęp do produktu IBM MQ Console można uzyskać, korzystając z następującego adresu URL:

```
https://<Route Hostname>/ibmmq/console
```

### Zadania pokrewne

“Wdrażanie menedżera kolejek przy użyciu interfejsu CLI Helm” na stronie 25

Użyj opcji Helm , aby wdrożyć menedżera kolejek w klastrze platformy kontenera produktu Red Hat OpenShift . To zadanie powinien wykonać administrator projektu.

“Nawiązywanie połączenia z menedżerem kolejek wdrożonym w klastrze OpenShift” na stronie 30

Zestaw przykładów konfiguracji na potrzeby nawiązywania połączenia z menedżerem kolejek wdrożonym w klastrze Red Hat OpenShift.

Linux

MQ Adv.

CD

## Tworzenie i odtwarzanie kopii zapasowej konfiguracji menedżera kolejek za pomocą interfejsu wiersza komend 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** .

- a) Uzyskaj nazwę zasobnika dla menedżera kolejek.

Na przykład, jeśli używany jest Operator, 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
```

Na przykład, jeśli używana jest komenda Helm, można uruchomić następującą komendę, gdzie *release\_name* jest nazwą wydania Helm .

```
oc get pods --selector release=release_name
```

- b) 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.

- a) Uzyskaj nazwę zasobnika dla menedżera kolejek.

Na przykład, jeśli używany jest Operator, 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
```

Na przykład, jeśli używana jest komenda Helm, można uruchomić następującą komendę, gdzie `release_name` jest nazwą wydania Helm .

```
oc get pods --selector release=release_name
```

b) Uruchom komendę `runmqsc` w zasobniku, kierując do niej zawartość pliku `backup.mqsc`.

```
oc exec -i pod_name -- runmqsc < backup.mqsc
```

## Budowanie własnego kontenera IBM MQ

Zaprojektuj samodzielnie zbudowany kontener, zwany wcześniej "obrazem kontenera Docker". 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.

### Procedura

- Informacje na temat budowania obrazu kontenera IBM MQ przy użyciu środowiska Docker można znaleźć w następujących podtematach:
  -  ["Tworzenie własnych obrazów kontenerów i wykresów IBM MQ" na stronie 8](#)
  - ["Planowanie wdrożenia własnego obrazu menedzera kolejek IBM MQ przy użyciu kontenera" na stronie 34](#)
  - ["Budowanie przykładowego obrazu menedzera kolejek produktu IBM MQ przy użyciu programu Docker" na stronie 35](#)
  - ["Uruchamianie aplikacji z powiązaniem lokalnymi w oddzielnych kontenerach" na stronie 38](#)

### Pojęcia pokrewne

[IBM MQ w kontenerach](#)

## Planowanie wdrożenia własnego obrazu menedzera 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 zappełnić produkt /var/mqm po raz pierwszy.

## Budowanie przykładowego obrazu menedżera kolejek produktu IBM MQ przy użyciu programu Docker

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 należy uruchomić kontener, który używa tego obrazu jako systemu plików, z zawartością /var/mqm udostępnioną przez wolumin specyficzny 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 35](#)
  - [“Budowanie przykładowego skonfigurowanego obrazu menedżera kolejek IBM MQ” na stronie 36](#)

## 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.

## Wyniki

Mamy teraz podstawowy obraz kontenera z zainstalowanym produktem IBM MQ.

## 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 rozpoczniesz

W przypadku obrazu IBM MQ 9.1 nie można skonfigurować bezpiecznego dostępu przy użyciu ograniczenia kontekstu zabezpieczeń Red Hat OpenShift Container Platform "zastrzeżona" (SCC). SCC "zastrzeżony" korzysta z losowych identyfikatorów użytkowników i uniemożliwia eskalację uprawnień, zmieniając się na innego użytkownika. Instalator oparty na RPM IBM MQ 9.1 opiera się na użytkownikach i grupach mqm , a także na bitach setuid w programach wykonywalnych.

To ograniczenie jest usuwane w produkcie IBM MQ 9.2.

### Procedura

1. Utwórz nowy katalog i dodaj do niego plik o nazwie `config.mqsc` o następującej zawartości:

```
DEFINE CHANNEL(PASSWORD.SVRCONN) CHLTYPE(SVRCONN)
SET CHLAUTH(PASSWORD.SVRCONN) TYPE(BLOCKUSER) USERLIST('nobody') +
DESCR('Allow privileged users on this channel')
SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS) DESCR('BackStop rule')
SET CHLAUTH(PASSWORD.SVRCONN) TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(CHANNEL) CHCKCLNT(REQUIRED)
ALTER AUTHINFO(SYSTEM.DEFAULT.IDPWOS) AUTHTYPE(IDPWOS) ADOPTCTX(YES)
REFRESH SECURITY TYPE(CONNAUTH)
```

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
RUN useradd johndoe -G mqm && \
    echo johndoe:passw0rd | chpasswd
COPY config.mqsc /etc/mqm/
```

gdzie:

- johndoe to identyfikator użytkownika, który ma zostać dodany
- passw0rd jest oryginalnym hasłem

3. Zbuduj niestandardowy obraz kontenera, używając następującej komendy:

```
sudo 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.

Komenda **RUN** dodaje użytkownika o nazwie johndoe z hasłem passw0rd , a komenda **COPY** doda plik `config.mqsc` do konkretnego miejsca, o którym wiadomo, że jest to obraz nadrzędny.

**Uwaga:** W systemie Red Hat Enterprise Linux (RHEL) stosuje się komendę **docker** (RHEL V7) lub **podman** (RHEL V7 lub RHEL V8). W przypadku produktu **podmana** początku tej komendy nie jest potrzebna **sudo** .

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:

```
sudo 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**

Został zbudowany skonfigurowany obraz kontenera i możliwe jest wyświetlanie działających kontenerów za pomocą komendy `docker ps`. Procesy IBM MQ działające w kontenerze można wyświetlić za pomocą komendy `docker top`.



#### **Ostrzeżenie:**

Dzienniki kontenera można przeglądać za pomocą komendy `docker logs ${CONTAINER_ID}`.

## **Co dalej**

- Jeśli kontener nie jest wyświetlany, gdy używana jest komenda `docker ps`, kontener mógł się nie powiodł. Kontenery, których przetwarzanie nie powiodło się, można wyświetlić za pomocą komendy `docker ps -a`.

- Podczas korzystania z komendy `docker ps -a` wyświetlany jest identyfikator kontenera. Ten identyfikator został również wydrukowany podczas wydawania komendy `docker run`.
- Dzienniki kontenera można wyświetlić za pomocą komendy `docker logs ${CONTAINER_ID}`.
- Maksymalną liczbę otwartych plików można ustawić za pomocą komendy `sysctl fs.file-max=524288`.

## V 9.1.0 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 `dspmq` 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 **dspmq**, **amqsput**, **amqsget** i **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 39.

## V 9.1.0 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 38.

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 **dspmq** 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 zwraca 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]  
lpiObtainQMDetails --> 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 dlrmqm QM1  
AMQ7002: An error occurred manipulating a file.
```

```
docker exec container strmqweb  
<file path>/mqrc.c[1152]  
lpiObtainQMDetails --> 545261715
```

## V 9.1.5 Linux MQ Adv. CD Skorowidz interfejsu API operatora IBM MQ

IBM MQ udostępnia operator Kubernetes, który zapewnia rodzimą integrację z platformą OpenShift Container Platform.

## V 9.1.5 Linux MQ Adv. CD Skorowidz interfejsu API mq.ibm.com/v1beta1

Interfejs API v1beta1 może być używany do tworzenia zasobów menedżera kolejek i zarządzania nimi.

## V 9.1.5 Linux MQ Adv. CD Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1

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  |
|---|---------------------------------------|--|
| L-RJON-BN7PN3                             | Production lub NonProduction          | <a href="#">IBM Cloud Pak for Integration 2020.2</a>                 |
| L-RJON-BPHL2Y                             |                                       | <a href="#">IBM Cloud Pak for Integration Limited Edition 2020.2</a> |
| L-APIG-BJAKBF                             | Production lub Development            | <a href="#">IBM MQ Advanced 9.1.5</a>                                |
| L-APIG-BM7GDH                             | Development                           | <a href="#">IBM MQ Advanced for Developers 9.1.5</a>                 |

Należy zauważyć, że określana jest wersja (*version*) licencji, która nie zawsze jest identyczna z wersją produktu IBM MQ.



## kolejek (mq.ibm.com/v1beta1)

## QueueManager

Menedżer kolejek (QueueManager) jest serwerem IBM MQ, który udostępnia aplikacjom usługi kolejki 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 wysłał 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 <a href="#">“Specyfikacja QueueManager” na stronie 45</a> | Pożądaný stan menedżera kolejek.   |
| status <a href="#">“Status QueueManager” na stronie 46</a>     | Obserwowany stan menedżera kolejek.  |

## Dostępność

Ustawienia dostępności dla menedżera kolejek, takie jak informacja o tym, czy ma być używana para aktywny-rezerwowý.

Występuje w:

- [“Konfiguracja QueueManager” na stronie 43](#)

| Pole        | Opis   |
|-------------|--|
| type string | Typ dostępności, który ma być używany. Użyj opcji "SingleInstance" dla pojedynczego poda, który zostanie automatycznie zrestartowany (w niektórych przypadkach) przez Kubernetes. Użyj opcji "MultiInstance" dla pary Pods, z których jeden jest "aktywnym" menedżerem kolejek, a drugi z nich jest rezerwowym. Więcej informacji na ten temat zawiera sekcja <a href="#">Wysoka dostępność dla produktu IBM MQ w kontenerach</a> w najnowszej wersji produktu IBM MQ. |

## Licencja

Ustawienia sterujące akceptacją licencji i wyborem metryk licencji, które mają być używane.

Występuje w:

- [“Specyfikacja QueueManager” na stronie 45](#)

| Pole           | Opis  |
|----------------|---|
| use string     | Ustawienie określające zastosowanie oprogramowania, gdy licencja dopuszcza więcej niż jedno zastosowanie. Poprawne wartości można znaleźć pod adresem <a href="https://ibm.biz/BdqvCF">https://ibm.biz/BdqvCF</a> .       |
| 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="https://ibm.biz/BdqvCF">https://ibm.biz/BdqvCF</a> . |
| metric string  | Ustawienie określające, która metryka licencji ma być używana. Na przykład: "ProcessorValueUnit", "VirtualProcessorCore" lub "ManagedVirtualServer".  |

## Granice

QueueManagerResourceList definiuje ustawienia procesora i pamięci.

Występuje w:

- [“Zasoby” na stronie 48](#)

| Pole   | Opis |
|--------|------|
| cpu    |      |
| memory |      |

## Odwwołanie do obiektu LocalObject

LocalObjectReference zawiera informacje wystarczające do odszukania przywoływanego obiektu w tej samej przestrzeni nazw.

Występuje w:

- [“Specyfikacja QueueManager” na stronie 45](#)

| 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> DO ZROBIENIA: Dodać inne przydatne pola. apiVersion, kind, uid?. |

## 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:

- [“Specyfikacja QueueManager” na stronie 45](#)

| Pole  | Opis   |
|---|--|
| Macierz keys <a href="#">“PKISource” na stronie 43</a>  | Klucze prywatne do dodania do repozytorium kluczy menedżera kolejek. |
| Macierz trust <a href="#">“PKISource” na stronie 43</a> | Certyfikaty do dodania do repozytorium kluczy menedżera kolejek.     |

## PKISource

PKISource definiuje źródło informacji o infrastrukturze klucza publicznego, takich jak klucze lub certyfikaty.

Występuje w:

- [“PKI” na stronie 42](#)

| Pole   | Opis   |
|--|--|
| name string  | Nazwa jest używana jako etykieta klucza lub certyfikatu. Musi być łańcuchem alfanumerycznym zapisanym małymi literami. |
| secret <a href="#">“Dane niejawne” na stronie 49</a> | Podaj klucz, używając sekretu Kubernetes.  |

## Konfiguracja QueueManager

QueueManagerKonfiguracja definiuje ustawienia dla kontenera menedżera kolejek i bazowego menedżera kolejek.

Występuje w:

- [“Specyfikacja QueueManager” na stronie 45](#)

| Pole  | Opis  |
|---|---|
| logFormat string  | Format dziennika, który ma być używany dla tego kontenera. W przypadku dzienników formatowanych JSON z kontenera użyj formatu JSON. Użyj języka "Basic" dla komunikatów sformatowanych w formie tekstowym.                                    |
| metrics <a href="#">“Wielkości mierzone QueueManager” na stronie 44</a>   | Ustawienia dotyczące metryk typu Prometheus.  |
| readinessProbe <a href="#">“QueueManagerReadinessProbe” na stronie 45</a> | Ustawienia sterujące sondą gotowości.   |
| resources <a href="#">“Zasoby” na stronie 48</a>                          | Ustawienia, które sterują wymaganiami co do zasobów.  |
| storage <a href="#">“Pamięć masowa QueueManager” na stronie 47</a>        | Ustawienia pamięci masowej służące do sterowania używaniem woluminów trwałych i klas pamięci przez menedżer kolejek.  |
| availability <a href="#">“Dostępność” na stronie 41</a>                   | Ustawienia dostępności dla menedżera kolejek, takie jak informacja o tym, czy ma być używana para aktywny-rezerwowo.  |
| imagePullPolicy string  | Ustawienie wpływające na to, kiedy kubelet próbuje pobrać określony obraz.  |
| livenessProbe <a href="#">“QueueManagerLivenessProbe” na stronie 44</a>   | Ustawienia sterujące sondą działania.   |
| debug boolean   | Określa, czy komunikaty debugowania z kodu właściwego dla kontenera mają być rejestrowane w dzienniku kontenera.  |
| image string  | Obraz kontenera, który będzie używany.  |
| name string   | Nazwa bazowego menedżera kolejek MQ, jeśli jest inna niż nazwa metadata.name. Tego pola należy użyć, jeśli nazwa menedżera kolejek nie jest zgodna z regułami nazewnictwa w środowisku Kubernetes (na przykład nazwa zawiera wielkie litery). |

## QueueManagerLivenessProbe

Ustawienia sterujące sondą działania.

Występuje w:

- [“Konfiguracja QueueManager” na stronie 43](#)

| Pole                        | Opis   |
|-----------------------------|--|
| failureThreshold integer    | Minimalna liczba następujących po sobie niepowodzeń, poprzedzonych powodzeniem, po której przyjmuje się, że doszło do niepowodzenia sondy.   |
| initialDelaySeconds integer | Liczba sekund od uruchomienia kontenera, po której inicjowane są sondy działania. 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.  |
| successThreshold integer    | Minimalna liczba następujących po sobie powodzeń, poprzedzonych niepowodzeniem, po której przyjmuje się, że sonda zadziałała pomyślnie.  |
| timeoutSeconds integer      | Limit czasu sondy w sekundach. 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> .  |

## Wielkości mierzone QueueManager

Ustawienia dotyczące metryk typu Prometheus.

Występuje w:

- [“Konfiguracja QueueManager” na stronie 43](#)

| Pole            | Opis   |
|-----------------|--|
| enabled boolean | Określa, czy ma być włączony punkt końcowy metryk zgodny z Prometheus. |

## QueueManagerOptionalVolume

Szczegóły woluminu trwałego (persistentVolume) dla dzienników odtwarzania MQ. Wymagane w przypadku korzystania z menedżera kolejek z wieloma instancjami.

Występuje w:

- [“Pamięć masowa QueueManager” na stronie 47](#)

| Pole             | Opis   |
|------------------|--|
| class string     | Klasa pamięci masowej, która ma być używana dla tego woluminu. Poprawne tylko wtedy, gdy "type" ma wartość "persistent-claim".   |
| enabled boolean  | Niezależnie od tego, czy wolumin ten powinien być włączony jako oddzielny wolumin, czy też nie, należy go umieścić w domyślnym woluminie "queueManager".   |
| size string      | Wielkość woluminu PersistentVolume w celu przejścia do sieci Kubernetes. Poprawne tylko wtedy, gdy "type" ma wartość "persistent-claim".   |
| sizeLimit string | Limit wielkości podczas korzystania z woluminu "efemerycznego". Pliki są nadal zapisywane do katalogu tymczasowego, więc można użyć tej opcji, aby ograniczyć wielkość. Poprawne tylko wtedy, gdy type jest "efemeryczny". |
| type string      | Typ woluminu, który ma być używany. Wybierz opcję ephemeral, aby utworzyć nietrwały wolumin "emptyDir", lub persistent-claim, aby użyć trwałego woluminu.  |

## QueueManagerReadinessProbe

Ustawienia sterujące sondą gotowości.

Występuje w:

- [“Konfiguracja QueueManager” na stronie 43](#)

| Pole                        | Opis   |
|-----------------------------|--|
| failureThreshold integer    | Minimalna liczba następujących po sobie niepowodzeń, poprzedzonych powodzeniem, po której przyjmuje się, że doszło do niepowodzenia sondy.   |
| initialDelaySeconds integer | Liczba sekund od uruchomienia kontenera, po której inicjowane są sondy działania. 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.  |
| successThreshold integer    | Minimalna liczba następujących po sobie powodzeń, poprzedzonych niepowodzeniem, po której przyjmuje się, że sonda zadziałała pomyślnie.  |
| timeoutSeconds integer      | Limit czasu sondy w sekundach. 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> .  |

## Specyfikacja QueueManager

Pożądaný stan menedżera kolejek.

Występuje w:

- [“QueueManager” na stronie 41](#)

| Pole   | Opis   |
|--|--|
| license <a href="#">“Licencja” na stronie 41</a>                       | Ustawienia sterujące akceptacją licencji i wyborem metryk licencji, które mają być używane.  |
| pki <a href="#">“PKI” na stronie 42</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="#">“Konfiguracja QueueManager” na stronie 43</a> | QueueManagerKonfiguracja definiuje ustawienia dla kontenera menedżera kolejek i bazowego menedżera kolejek.  |
| securityContext <a href="#">“SecurityContext” na stronie 49</a>        | Ustawienia zabezpieczeń, które mają zostać dodane do elementu securityContext zasobnika menedżera kolejek.   |
| tracing <a href="#">“TracingConfig” na stronie 50</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" określiłby wersję MQ 9.1.5.0, korzystając z drugiej korekty obrazu kontenera. W korektach wprowadzane są często poprawki specyficzne dla kontenera, np. poprawki do obrazu podstawowego. |
| 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> .                 |

| Pole   | Opis  |
|--|---|
| Macierz <code>imagePullSecrets</code> “ <a href="#">Odwołanie do obiektu LocalObject</a> ” na stronie 42 | 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 <code>DockerConfig</code> . 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> . |
| <code>template</code> “ <a href="#">Szablon</a> ” na stronie 49  | Zaawansowane użycie szablonów zasobów Kubernetes. Szablon umożliwia użytkownikom zmianę sposobu generowania przez IBM MQ bazowych zasobów Kubernetes, takich jak zestawy stanowe ( <code>StatefulSet</code> ), zasobniki ( <code>Pod</code> ) i usługi ( <code>Service</code> ). 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.  |
| <code>terminationGracePeriod</code> 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.  |
| <code>web</code> “ <a href="#">Konfiguracja serwera WebServer</a> ” na stronie 51                        | Ustawienia serwera WWW produktu MQ.   |

## Status QueueManager

Obserwowany stan menedżera kolejek.

Występuje w:

- “[QueueManager](#)” na stronie 41

| Pole  | Opis   |
|---|--|
| Macierz <code>endpoints</code> “ <a href="#">QueueManagerStatusEndpoint</a> ” na stronie 47   | Informacje dotyczące punktów końcowych, które eksponuje ten menedżer kolejek, takich jak punkty końcowe interfejsu API lub interfejsu użytkownika. |
| <code>name</code> string  | Nazwa menedżera kolejek.   |
| <code>versions</code> “ <a href="#">QueueManagerStatusVersion</a> ” na stronie 47             | Używana wersja produktu MQ oraz inne wersje dostępne w rejestrze IBM Entitled Registry.  |
| <code>adminUiUrl</code> string  | Adres URL administracyjnego interfejsu użytkownika.  |
| Macierz <code>conditions</code> “ <a href="#">QueueManagerStatusCondition</a> ” na stronie 46 | Warunki reprezentują najnowsze dostępne obserwacje stanu menedżera kolejek.  |

## QueueManagerStatusCondition

`QueueManagerStatusCondition` definiuje stany menedżera kolejek.

Występuje w:

- “[Status QueueManager](#)” na stronie 46

| Pole                      | Opis   |
|---------------------------|--|
| message string            | Czytelny dla człowieka komunikat opisujący szczegóły ostatniej zmiany statusu. |
| type string               | Typ stanu.   |
| lastTransitionTime string | Czas ostatniego przejścia stanu z jednego statusu do innego.                   |

### QueueManagerStatusEndpoint

QueueManagerStatusEndpoint definiuje punkty końcowe menedżera kolejek.

Występuje w:

- [“Status QueueManager” na stronie 46](#)

| 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.  |

### QueueManagerStatusVersion

Używana wersja produktu MQ oraz inne wersje dostępne w rejestrze IBM Entitled Registry.

Występuje w:

- [“Status QueueManager” na stronie 46](#)

| Pole  | Opis   |
|---|--|
| available<br><a href="#">“QueueManagerStatusVersion Dostępny” na stronie 47</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. |

### QueueManagerStatusVersionDostępny

Inne wersje produktu MQ dostępne w rejestrze IBM Entitled Registry.

Występuje w:

- [“QueueManagerStatusVersion” na stronie 47](#)

| Pole  | Opis   |
|---|--|
| channels array  | Kanały dostępne do automatycznego aktualizowania wersji produktu MQ. |
| Macierz <a href="#">versions “Wersje” na stronie 51</a> | Konkretne wersje produktu MQ, które są dostępne.                     |

### Pamięć masowa QueueManager

Ustawienia pamięci masowej służące do sterowania używaniem woluminów trwałych i klas pamięci przez menedżer kolejek.

Występuje w:

- [“Konfiguracja QueueManager” na stronie 43](#)

| Pole   | Opis   |
|--|--|
| <code>persistedData</code><br><a href="#">“QueueManagerOptionalVolume” na stronie 44</a> | Szczegóły woluminu trwałego ( <code>persistentVolume</code> ) dla utwalonych danych produktu MQ, w tym konfiguracji, kolejek i komunikatów. Wymagane w przypadku korzystania z menedżera kolejek z wieloma instancjami.                        |
| <code>queueManager</code> <a href="#">“Wolumin QueueManager” na stronie 48</a>           | Wartość domyślna <code>PersistentVolume</code> dla wszelkich danych znajdujących się normalnie w woluminie <code>/var/mqm</code> . Jeśli nie zostaną określone inne woluminy, będzie zawierać wszystkie utwalone dane i dzienniki odtwarzania. |
| <code>recoveryLogs</code><br><a href="#">“QueueManagerOptionalVolume” na stronie 44</a>  | Szczegóły woluminu trwałego ( <code>persistentVolume</code> ) dla dzienników odtwarzania MQ. Wymagane w przypadku korzystania z menedżera kolejek z wieloma instancjami.   |

## Wolumin QueueManager

Wartość domyślna `PersistentVolume` dla wszelkich danych znajdujących się normalnie w woluminie `/var/mqm`. Jeśli nie zostaną określone inne woluminy, będzie zawierać wszystkie utwalone dane i dzienniki odtwarzania.

Występuje w:

- [“Pamięć masowa QueueManager” na stronie 47](#)

| Pole                          | Opis  |
|-------------------------------|---|
| <code>class string</code>     | Klasa pamięci masowej, która ma być używana dla tego woluminu. Poprawne tylko wtedy, gdy <code>type</code> ma wartość <code>"persistent-claim"</code> .   |
| <code>size string</code>      | Wielkość woluminu <code>PersistentVolume</code> w celu przejścia do sieci Kubernetes. Poprawne tylko wtedy, gdy <code>type</code> ma wartość <code>"persistent-claim"</code> .  |
| <code>sizeLimit string</code> | Limit wielkości podczas korzystania z woluminu "efemerycznego". Pliki są nadal zapisywane do katalogu tymczasowego, więc można użyć tej opcji, aby ograniczyć wielkość. Poprawne tylko wtedy, gdy <code>type</code> jest "efemeryczny". |
| <code>type string</code>      | Typ woluminu, który ma być używany. Wybierz opcję <code>ephemeral</code> , aby utworzyć nietrwały wolumin <code>"emptyDir"</code> , lub <code>persistent-claim</code> , aby użyć trwałego woluminu.                                     |

## Żądania

`QueueManagerResourceList` definiuje ustawienia procesora i pamięci.

Występuje w:

- [“Zasoby” na stronie 48](#)

| Pole                | Opis |
|---------------------|------|
| <code>memory</code> |      |
| <code>cpu</code>    |      |

## Zasoby

Ustawienia, które sterują wymaganiami co do zasobów.

Występuje w:

- [“Konfiguracja QueueManager” na stronie 43](#)



| Pole  | Opis   |
|---|--|
| <code>limits</code> <a href="#">“Granice” na stronie 42</a>   | QueueManagerResourceList definiuje ustawienia procesora i pamięci. |
| <code>requests</code> <a href="#">“Żądania” na stronie 48</a> | QueueManagerResourceList definiuje ustawienia procesora i pamięci. |

## Dane niejawne

Podaj klucz, używając sekretu Kubernetes.

Występuje w:

- [“PKISource” na stronie 43](#)

| Pole                           | Opis   |
|--------------------------------|--|
| <code>items</code> array       | Klucze znajdujące się wewnątrz sekretu Kubernetes, które powinny zostać dodane do kontenera menedżera kolejek. |
| <code>secretName</code> string | Nazwa sekretu Kubernetes.  |

## SecurityContext

Ustawienia zabezpieczeń, które mają zostać dodane do elementu securityContext zasobnika menedżera kolejek.

Występuje w:

- [“Specyfikacja QueueManager” na stronie 45](#)

| Pole                                  | Opis   |
|---------------------------------------|--|
| <code>supplementalGroups</code> 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.  |
| <code>fsGroup</code> 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.   |
| <code>initVolumeAsRoot</code> boolean | Ma to wpływ na kontekst zabezpieczeń używany przez kontener, który inicjuje wolumin PersistentVolume. Tę wartość należy ustawić na wartość "true", jeśli używany jest dostawca pamięci masowej, który wymaga, aby użytkownik root mógł uzyskać dostęp do nowo udostępnianych woluminów. Ustawienie wartości "true" ma wpływ na to, które obiekty SCC (Security Context Constraints) mogą być używane, a menedżer kolejek może się nie uruchamiać, jeśli użytkownik nie ma uprawnień do korzystania z SCC, który umożliwia użytkownikowi root. 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> . |

## Szablon

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:

- [“Specyfikacja QueueManager” na stronie 45](#)

| Pole | Opis   |
|------|--|
| pod  | Przestonięcia dla szablonu używanego dla zasobnika.<br>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> . |

## TracingAgent

Tylko w pakiecie Cloud Pak for Integration: można skonfigurować ustawienia opcjonalnego agenta śledzenia.

Występuje w:

- [“TracingConfig” na stronie 50](#)

| Pole   | Opis   |
|--|--|
| image string   | Obraz kontenera, który będzie używany.                                     |
| imagePullPolicy string   | Ustawienie wpływające na to, kiedy kubelet próbuje pobrać określony obraz. |
| livenessProbe<br><a href="#">“TracingProbe” na stronie 51</a>  | Ustawienia sterujące sondą działania.                                      |
| readinessProbe<br><a href="#">“TracingProbe” na stronie 51</a> | Ustawienia sterujące sondą gotowości.                                      |

## TracingCollector

Tylko w pakiecie Cloud Pak for Integration: można skonfigurować ustawienia opcjonalnego kolektora śledzenia.

Występuje w:

- [“TracingConfig” na stronie 50](#)

| Pole   | Opis   |
|--|--|
| image string   | Obraz kontenera, który będzie używany.                                     |
| imagePullPolicy string   | Ustawienie wpływające na to, kiedy kubelet próbuje pobrać określony obraz. |
| livenessProbe<br><a href="#">“TracingProbe” na stronie 51</a>  | Ustawienia sterujące sondą działania.                                      |
| readinessProbe<br><a href="#">“TracingProbe” na stronie 51</a> | Ustawienia sterujące sondą gotowości.                                      |

## TracingConfig

Ustawienia integracji śledzenia z panelem kontrolnym Cloud Pak for Integration Operations Dashboard.

Występuje w:

- [“Specyfikacja QueueManager” na stronie 45](#)

| Pole   | Opis  |
|--|---|
| agent <a href="#">“TracingAgent” na stronie 50</a>         | Tylko w pakiecie Cloud Pak for Integration: można skonfigurować ustawienia opcjonalnego agenta śledzenia.                       |
| collector <a href="#">“TracingCollector” na stronie 50</a> | Tylko w pakiecie Cloud Pak for Integration: można skonfigurować ustawienia opcjonalnego kolektora śledzenia.                    |
| enabled boolean  | Określa, czy włączyć integrację z panelem kontrolnym Cloud Pak for Integration Operations Dashboard za pośrednictwem śledzenia. |
| namespace string   | Przestrzeń nazw, w której zainstalowany jest panel kontrolny Cloud Pak for Integration Operations Dashboard.                    |

## TracingProbe

Ustawienia sterujące sondą gotowości.

Występuje w:

- [“TracingCollector” na stronie 50](#)

| Pole                        | Opis   |
|-----------------------------|--|
| failureThreshold integer    | Minimalna liczba następujących po sobie niepowodzeń, poprzedzonych powodzeniem, po której przyjmuje się, że doszło do niepowodzenia sondy.   |
| initialDelaySeconds integer | Liczba sekund od uruchomienia kontenera, po której inicjowane są sondy działania. 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.  |
| successThreshold integer    | Minimalna liczba następujących po sobie powodzeń, poprzedzonych niepowodzeniem, po której przyjmuje się, że sonda zadziałała pomyślnie.  |
| timeoutSeconds integer      | Limit czasu sondy w sekundach. 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> .  |

## Wersje

QueueManagerStatusVersion definiuje wersję produktu MQ.

Występuje w:

- [“QueueManagerStatusVersionDostępny” na stronie 47](#)

| Pole        | Opis  |
|-------------|---|
| name string | Wersja "name" dla tej wersji QueueManager. Są to poprawne wartości pola <code>spec.version</code> . |

## Konfiguracja serwera WebServer

Ustawienia serwera WWW produktu MQ.

Występuje w:

- [“Specyfikacja QueueManager” na stronie 45](#)

| Pole            | Opis                                     |
|-----------------|--|
| enabled boolean | Określa, czy serwer WWW ma być włączony. |



## 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 (<http://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: