

9.3

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 225](#).

Niniejsze wydanie publikacji dotyczy wersji 9, wydania 3 produktu IBM® MQ oraz wszystkich jego późniejszych wydań i modyfikacji, aż do odwołania w nowych wydaniach publikacji.

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


IBM MQ w kontenerach i IBM Cloud Pak for Integration.....	5
Planowanie wdrożenia IBM MQ w kontenerach.....	5
Wybór sposobu używania IBM MQ w kontenerach.....	5
Obsługa IBM MQ w kontenerach.....	6
Planowanie licencjonowania produktu IBM MQ w kontenerach.....	14
Zależności operatora IBM MQ Operator.....	19
Uprawnienia o zasięgu klastra wymagane przez IBM MQ Operator.....	21
Uwagi dotyczące pamięci masowej dla IBM MQ Operator.....	21
IBM MQ Advanced for Developers obraz kontenera.....	23
Wysoka dostępność produktu IBM MQ w kontenerach.....	26
Odtwarzanie po awarii w przypadku IBM MQ w kontenerach.....	28
Planowanie zabezpieczenia produktu IBM MQ w kontenerach.....	29
Planowanie skalowalności i wydajności produktu IBM MQ w kontenerach.....	34
Korzystanie z operatora produktu IBM MQ.....	35
Historia wydań operatora IBM MQ Operator.....	36
Weryfikowanie podpisów obrazów.....	81
Migracja produktu IBM MQ do rozwiązania IBM Cloud Pak for Integration.....	81
Instalowanie produktu IBM MQ Operator.....	105
Instalowanie produktu IBM MQ Operator 2.x w środowisku z luką powietrzną.....	112
Wdrażanie menedżera kolejek w klastrze platformy kontenerowej Red Hat OpenShift.....	118
Deinstalowanie produktu IBM MQ Operator.....	121
Aktualizowanie operatora IBM MQ Operator i menedżerów kolejek.....	123
Konfigurowanie menedżerów kolejek za pomocą programu IBM MQ Operator.....	137
Użytkowanie produktu IBM MQ przy wykorzystaniu operatora IBM MQ Operator.....	176
Rozwiązywanie problemów z produktem IBM MQ Operator.....	184
Skorowidz interfejsu API operatora IBM MQ Operator.....	187
Budowanie własnego kontenera IBM MQ i kodu wdrażania.....	212
Planowanie wdrożenia własnego obrazu menedżera kolejek IBM MQ przy użyciu kontenera.....	213
Budowanie przykładowego obrazu kontenera menedżera kolejek IBM MQ.....	213
Uruchamianie aplikacji z powiązaniem lokalnymi w oddzielnych kontenerach.....	216
Tworzenie rodzimej grupy wysokiej dostępności w przypadku tworzenia własnych kontenerów..	218
Uwagi.....	225
Informacje dotyczące interfejsu programistycznego.....	226
Znaki towarowe.....	227

Multi IBM MQ w kontenerach i IBM Cloud Pak for Integration

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

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

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

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

Multi Planowanie wdrożenia IBM MQ w kontenerach

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

O tym zadaniu

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

Procedura

- [“Wybór sposobu używania IBM MQ w kontenerach”](#) na stronie 5.
- [“Obsługa IBM MQ w kontenerach”](#) na stronie 6.
- [“Uwagi dotyczące pamięci masowej dla IBM MQ Operator”](#) na stronie 21.
- [“Wysoka dostępność produktu IBM MQ w kontenerach”](#) na stronie 26.
- [“Odtwarzanie po awarii w przypadku IBM MQ w kontenerach”](#) na stronie 28.
- [“Uwierzytelnianie i autoryzacja użytkowników oprogramowania IBM MQ w kontenerach”](#) na stronie 29.

Wybór sposobu używania IBM MQ w kontenerach

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

Korzystanie z IBM MQ Operator



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

IBM MQ Operator rozszerza interfejs API Red Hat OpenShift Container Platform w celu dodania nowego zasobu niestandardowego `QueueManager`. Element wykonawczy czuwa nad nowymi definicjami menedżera kolejek, a następnie przekształca je w niezbędne zasoby niskopoziomowe, takie jak zasoby `StatefulSet` czy `Service`. W przypadku rodzimej wysokiej dostępności element wykonawczy może

również wykonać złożoną sekwencyjną aktualizację instancji menedżera kolejek. Patrz: [“Uwagi dotyczące wykonywania własnej sekwencyjnej aktualizacji menedżera kolejek rodzimej wysokiej dostępności”](#) na stronie 220

Niektóre funkcje produktu IBM MQ nie są obsługiwane w przypadku korzystania z operatora IBM MQ Operator. Szczegółowe informacje na temat obsługiwanych funkcji IBM MQ Operator zawiera sekcja [“Obsługa IBM MQ w kontenerach”](#) na stronie 6 .

Należy zauważyć, że IBM MQ Operator nie obsługuje instalacji w klastrze OpenShift z komputerami z wieloma architekturą.

Budowanie własnych obrazów i kodu wdrażania

Multi

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

Dostępne są przykłady przydatne przy budowaniu własnych obrazów. Patrz sekcja [“Budowanie własnego kontenera IBM MQ i kodu wdrażania”](#) na stronie 212.

Szczegółowe informacje o tym, co jest obsługiwane podczas budowania własnego obrazu i kodu wdrażania, zawiera sekcja [“Obsługa IBM MQ w kontenerach”](#) na stronie 6 .

Odsyłacze pokrewne

[“Obsługa IBM MQ w kontenerach”](#) na stronie 6

Nie wszystkie funkcje produktu IBM MQ są dostępne i obsługiwane w ten sam sposób w kontenerach.

OpenShift CP4I-LTS CP4I CD Obsługa IBM MQ w kontenerach

Nie wszystkie funkcje produktu IBM MQ są dostępne i obsługiwane w ten sam sposób w kontenerach.

Poniżej znajduje się tabela, w której przedstawiono szczegółowo, w jaki sposób składniki produktu IBM MQ są obsługiwane z produktem IBM MQ Operator, lub podczas budowania własnych kontenerów i kodu wdrażania.

Uwaga: Wstępnie zbudowane obrazy kontenerów IBM MQ w systemie IBM Container Registry (icr.io i cp.icr.io) są obsługiwane i zakwalifikowane do poprawek tylko wtedy, gdy są używane z IBM MQ Operator.

Nie można "zaktualizować" licencji wstępnie zbudowanego obrazu IBM MQ Advanced for Developers do innej licencji. IBM MQ Operator wdroży różne obrazy w zależności od wybranej licencji.

W tej tabeli obowiązują następujące warunki:

"Kod aktywacji kontenera"

Pliki wykonywalne **runmqserver**, **runmqintegrationserver**, **chkmqhealthy**, **chkmqready** i **chkmqstarted**. Ten kod jest udostępniany jako przykład i jest obsługiwany tylko jako część wstępnie zbudowanych kontenerów, jeśli jest używany z produktem IBM MQ Operator.

	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM Cloud Pak for Integration	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM MQ Advanced	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM MQ Advanced for Developers	Wstępnie zbudowany obraz systemu IBM MQ Advanced for Developers	Budowanie własnego kontenera
Obsługiwane platformy	<p>Obsługiwane tylko w systemie Red Hat OpenShift Container Platform . Wersje systemu Red Hat OpenShift Container Platform nie są już obsługiwane przez IBM MQ po Red Hat zatrzymaniu obsługi.</p> <p>Więcej informacji na ten temat zawiera sekcja “Wersja zgodna z operatorem IBM MQ Operator” na stronie 12.</p>		<p>Opcja jest dostępna tylko w systemie Red Hat OpenShift Container Platform , ale nie jest obsługiwana.</p>	<p>Działa na dowolnej platformie Docker, containerd lub cri-o, ale nie jest obsługiwana. Szczegółowe informacje można znaleźć w sekcji Wymagania systemowe produktu IBM MQ.</p>	<p>Dowolna platforma Docker, containerd lub cri-o. Szczegółowe informacje można znaleźć w sekcji Wymagania systemowe produktu IBM MQ. Rodzima funkcja wysokiej dostępności jest obsługiwana tylko w systemach Kubernetes i Red Hat OpenShift Container Platform. Przykładowy obraz kontenera używa Red Hat Universal Base Image (UBI), który zawiera biblioteki Linux® i programy narzędziowe używane przez IBM MQ. System UBI jest obsługiwany przez produkt Red Hat na platformie Red Hat OpenShift. <i>Kod aktywacji kontenera</i> nie jest obsługiwany.</p>

	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM Cloud Pak for Integration	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM MQ Advanced	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM MQ Advanced for Developers	Wstępnie zbudowany obraz systemu IBM MQ Advanced for Developers	Budowanie własnego kontenera
Architektury procesorów	Obsługiwane w systemach amd64, s390x z/Linux i ppc64le Power Systems.		Dostępne w systemach amd64, s390x z/Linux i ppc64le Power Systems, ale nieobsługiwane.		Zgodnie z oprogramowaniem IBM MQ .
Czas trwania wsparcia	<p>IBM Cloud Pak for Integration - Long Term Support lub Continuous Delivery.¹</p> <p>Operator CD i menedżery kolejek są obsługiwane do następnej wersji produktu IBM Cloud Pak for Integration CD lub CP4I-LTS .</p> <p>Operator CP4I-LTS i menedżery kolejek są obsługiwane do następnej wersji produktu IBM Cloud Pak for Integration CP4I-LTS plus okres karencji na aktualizację.</p>	<p>Tylko strumień Continuous Delivery (zarówno dla IBM MQ Operator, jak i dla menedżerów kolejek).</p> <p>Każda wersja produktu IBM MQ Operator i menedżera kolejek jest obsługiwana tylko do następnej wersji produktu CD lub LTS .</p>	Nieobsługiwane		Zgodnie z oprogramowaniem IBM MQ . Patrz Często zadawane pytania dotyczące wydań Long Term Support i Continuous Delivery produktu IBM MQ. Kod aktywacji kontenera nie jest obsługiwany.

¹ Produkt IBM MQ Operator jest obsługiwany jako wersja systemu IBM MQ CD lub jako wersja systemu CP4I-LTS :

- Obrazy kontenera IBM MQ 9.3.0.x wdrożone z produktem IBM MQ Operator 2.0.x, używane jako część produktu IBM Cloud Pak for Integration 2022.2.1, kwalifikują się do wsparcia CP4I-LTS . Najnowsza wersja produktu Long Term Support (LTS) w systemie IBM MQ Operator to 2.0.23, a najnowszy obraz kontenera LTS to 9.3.0.17-r3.
- Obrazy kontenera IBM MQ 9.3.5 wdrożone z produktem IBM MQ Operator 3.1.x, używane jako część produktu IBM Cloud Pak for Integration 2023.4.1, kwalifikują się do wsparcia CD . Najnowsza wersja produktu Continuous Delivery (CD) w systemie IBM MQ Operator to 3.1.3, a najnowszy obraz kontenera CD to 9.3.5.1-r2.

	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM Cloud Pak for Integration	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM MQ Advanced	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM MQ Advanced for Developers	Wstępnie zbudowany obraz systemu IBM MQ Advanced for Developers	Budowanie własnego kontenera
Dostępność poprawek bezpieczeństwa	Poprawki okresowe dostępne jako obrazy kontenerów w systemie IBM Container Registry				Poprawki oprogramowania IBM MQ są dostępne jako oprogramowanie w systemie Fix Central . <i>Kod aktywacji kontenera</i> nie jest obsługiwany.
Dostępność poprawek tymczasowych	Poprawki menedżera kolejek dostępne jako oprogramowanie i konieczne jest utworzenie niestandardowego obrazu. Poprawki IBM MQ Operator nie są dostępne jako poprawki tymczasowe.	Brak dostępnych poprawek tymczasowych.			Poprawki oprogramowania IBM MQ są dostępne jako oprogramowanie w systemie Fix Central lub za pośrednictwem działu wsparcia IBM . <i>Kod aktywacji kontenera</i> nie jest obsługiwany.
Opcja: Advanced Message Security	Obsługiwane. Należy zauważyć, że szyfrowanie po stronie serwera nie jest łatwe, ponieważ IBM MQ Operator nie pozwala bezpośrednio na określenie własnego magazynu kluczy dla Advanced Message Security.	Dostępne, ale nieobsługiwane.			Obsługiwane zgodnie z oprogramowaniem IBM MQ , ale nie są dostępne żadne przykłady.
Opcja: Managed File Transfer	Niedostępne i nieobsługiwane. Można jednak użyć programu IBM MQ Operator , aby udostępnić jeden lub więcej menedżerów kolejek koordynacji, komend lub agentów.			Niedostępne i nieobsługiwane.	Obsługiwane zgodnie z oprogramowaniem IBM MQ , z przykładowym dla agenta.
Opcja: MQTT	Niedostępne i nieobsługiwane.				Obsługiwane zgodnie z oprogramowaniem IBM MQ , ale nie są dostępne żadne przykłady.

	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM Cloud Pak for Integration	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM MQ Advanced	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM MQ Advanced for Developers	Wstępnie zbudowany obraz systemu IBM MQ Advanced for Developers	Budowanie własnego kontenera
Składnik: AMQP	Niedostępne i nieobsługiwane.				Obsługiwane zgodnie z oprogramowaniem IBM MQ , ale nie są dostępne żadne przykłady.
Opcja: REST API	Dostępne i obsługiwane od wersji IBM MQ Operator 3.0 i IBM MQ 9.3.4 . Wcześniej interfejsy REST API nie były obsługiwane.	Dostępne i obsługiwane. Łatwa konfiguracja od wersji IBM MQ Operator 3.0 i IBM MQ 9.3.4 .	Dostępne i obsługiwane od wersji IBM MQ Operator 3.0 i IBM MQ 9.3.4 , ale nie obsługiwane. Wcześniej interfejsy REST API nie były dostępne.	Dostępne i obsługiwane od wersji IBM MQ 9.3.4 , ale nie obsługiwane. Wcześniej produkt REST API nie był dostępny.	Dostępne i obsługiwane zgodnie z oprogramowaniem IBM MQ .
Składnik: Replikowane menedżery kolejek danych	Niedostępne i nieobsługiwane. Menedżery kolejek replikowanych danych są ściśle powiązane z jądrem Linux i nie są obsługiwane w kontenerach.				
Funkcja: Rodzima HA	Dostępne i obsługiwane.		Dostępne, ale nieobsługiwane.		Dostępne tylko w systemach Kubernetes i Red Hat OpenShift Container Platform. Obsługiwane zgodnie z oprogramowaniem IBM MQ .
Funkcja: menedżery kolejek z wieloma instancjami	Dostępne i obsługiwane.		Dostępne, ale nieobsługiwane.		Dostępne i obsługiwane zgodnie z oprogramowaniem IBM MQ .

	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM Cloud Pak for Integration	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM MQ Advanced	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM MQ Advanced for Developers	Wstępnie zbudowany obraz systemu IBM MQ Advanced for Developers	Budowanie własnego kontenera
Opcja: Typy dzienników odtwarzania	Tylko rejestrowanie cykliczne lub replikowane dzienniki. Rejestrowanie liniowe nie jest obsługiwane.				Dostępne i obsługiwane zgodnie z oprogramowaniem IBM MQ . Należy skonfigurować opcje crtmqm .
Składnik: określanie niestandardowych opcji wiersza komend dla komend <code>crtmqdir</code>, <code>crtmqm</code>, <code>strmqm</code> i <code>endmqm</code>	Niedostępne i nieobsługiwane. Większość opcji można skonfigurować za pomocą pliku INI, ale niektórych nie można skonfigurować, na przykład za pomocą rejestrowania liniowego.				Opcjonalnie, w zależności od sposobu implementacji kodu aktywacji kontenera.
Opcja: użytkownicy systemu operacyjnego	Niedostępne i nieobsługiwane.				Możliwe i obsługiwane zgodnie z oprogramowaniem IBM MQ , jeśli produkt IBM MQ jest instalowany przy użyciu pakietów RPM, ale nie jest dostępna żadna próbka. Niezalecane ze względu na zagrożenie bezpieczeństwa .
Opcja: IBM MQ Bridge to blockchain	Niedostępne i nieobsługiwane. Usunięte w całości z programu IBM MQ od wersji IBM MQ 9.3.2 .				

	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM Cloud Pak for Integration	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM MQ Advanced	Korzystanie z produktu IBM MQ Operator i licencji na produkt IBM MQ Advanced for Developers	Wstępnie zbudowany obraz systemu IBM MQ Advanced for Developers	Budowanie własnego kontenera
Opcja: IBM MQ Bridge to Salesforce	Niedostępne i nieobsługiwane.				Obsługiwane zgodnie z oprogramowaniem IBM MQ, ale nieaktualne od wersji IBM MQ 9.3.1.

Uwaga: Fraza "obsługiwane zgodnie z oprogramowaniem IBM MQ" oznacza, że dział wsparcia technicznego IBM jest ograniczony do podstawowego oprogramowania IBM MQ, które działa wewnątrz kontenera.

Pojęcia pokrewne

Często zadawane pytania dotyczące wydań Long Term Support i Continuous Delivery produktu IBM MQ

Odsyłacze pokrewne

IBM Cloud Pak for Integration -dodatek Software Support Lifecycle Addendum



Wersja zgodna z operatorem IBM MQ

Operator

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

Uwaga:

IBM MQ Operator obsługuje tylko wersje Extended Update Support (EUS) OpenShift Container Platform. Informacje o tym, które wersje tego produktu obejmują, można znaleźć w sekcji [Fazy cyklu życia](#) na stronie WWW strategii cyklu życia produktu Red Hat OpenShift Container Platform.

- [“Dostępne wersje produktu IBM MQ” na stronie 12](#)
- [“Kompatybilne wersje środowiska Red Hat OpenShift Container Platform” na stronie 13](#)
- [“Wersja IBM Cloud Pak for Integration” na stronie 14](#)
- [“Dostępne wersje produktu IBM MQ w starszych operatorach” na stronie 14](#)
- [“Zgodne wersje produktu OpenShift Container Platform dla starszych operatorów” na stronie 14](#)

Dostępne wersje produktu IBM MQ

Kanał operat ora	Wersj a operat ora	Wersja IBM MQ									
		9.2.0 EUS	9.2.3	9.2.4	9.2.5	9.3.0	9.3.1	9.3.2	9.3.3	9.3.4	9.3.5
v2.0	2.0	→	⚠	●	●	●					
v2.1	2.1	→	⚠	⚠	⚠	→	●				
v2.2	2.2	→	⚠	⚠	⚠	→	●				
v2.3	2.3	→	⚠	⚠	⚠	→	⚠	●			

Kanał operatora	Wersja operatora	Wersja IBM MQ									
		9.2.0 EUS	9.2.3	9.2.4	9.2.5	9.3.0	9.3.1	9.3.2	9.3.3	9.3.4	9.3.5
v2.4	2.4	→	⚠	⚠	⚠	→	⚠	⚠	●		
v3.0	3.0					→	⚠	⚠	⚠	●	
v3.1	3.1					→	⚠	⚠	⚠	⚠	●

Klucz:

- Dostępna obsługa rozwiązania Continuous Delivery
- Dostępna obsługa rozwiązania IBM Cloud Pak for Integration - Long Term Support
- Dostępne tylko podczas migracji z operandu IBM Cloud Pak for Integration - Long Term Support do operandu Continuous Delivery.



Deprecated W miarę jak wersje systemu IBM MQ przestają być obsługiwane, mogą być nadal konfigurowane w operatorze, ale nie kwalifikują się już do wsparcia i mogą zostać usunięte w przyszłych wersjach.

Sekcja [“Historia wydań operatora IBM MQ Operator”](#) na stronie 36 zawiera szczegółowe informacje na temat każdej wersji, w tym szczegóły dotyczące składników, zmian i poprawek.

Kompatybilne wersje środowiska Red Hat OpenShift Container Platform

Kanał operatora	Wersja operatora	Wersja OpenShift Container Platform ²		
		4.10	4.12	4.14
v2.0	2.0.0-2.0.15	→	□	
	2.0.16		□	
	2.0.17 i nowsze		□	□
v2.1	2.1	→	●	
v2.2	2.2	→	●	
v2.3	2.3	→	●	
v2.4	2.4.0-2.4.3	→	●	
	2.4.4		●	
	2.4.5		●	●
v3.0	3.0.0 i nowsze		●	●
v3.1	3.1.0 i nowsze		●	●

Klucz:

- Dostępna obsługa rozwiązania Continuous Delivery

² Wersje produktu OpenShift Container Platform są objęte własnymi datami wsparcia. Więcej informacji na ten temat zawiera sekcja dotycząca [strategii cyklu życia OpenShift Container Platform](#).



Dostępna obsługa rozwiązywania IBM Cloud Pak for Integration - Long Term Support



Nie jest już obsługiwane. Przeprowadź migrację do nowszej wersji produktu OpenShift Container Platform .

Wersja IBM Cloud Pak for Integration

Obsługiwane jako część systemu IBM Cloud Pak for Integration w wersji 2022.2.1 lub niezależnie:

- IBM MQ Operator 2.0.x
- IBM MQ Operator 2.1.x

Obsługiwane jako część systemu IBM Cloud Pak for Integration w wersji 2022.4.1 lub niezależnie:

- IBM MQ Operator 2.2.x
- IBM MQ Operator 2.3.x

Obsługiwane jako część systemu IBM Cloud Pak for Integration w wersji 2023.2.1 lub niezależnie:

- IBM MQ Operator 2.4.x

Obsługiwane jako część systemu IBM Cloud Pak for Integration w wersji 2023.4.1 lub niezależnie:

- IBM MQ Operator 3.0.x
- IBM MQ Operator 3.1.x

Dostępne wersje produktu IBM MQ w starszych operatorach

Patrz sekcja [Dostępne wersje systemu IBM MQ](#) w dokumentacji produktu IBM MQ 9.2 .

Zgodne wersje produktu OpenShift Container Platform dla starszych operatorów

Patrz sekcja [Kompatybilne wersje OpenShift Container Platform](#) w dokumentacji produktu IBM MQ 9.2 .

Planowanie licencjonowania produktu IBM MQ w kontenerach

Licencjonowanie kontenerów pozwala na licencjonowanie tylko dostępnej mocy obliczeniowej poszczególnych kontenerów IBM MQ , bez konieczności licencjonowania całego serwera, na którym działają kontenery. Aby skorzystać z licencjonowania kontenerów, należy użyć usługi IBM License Service do śledzenia wykorzystania licencji i określenia wymaganych uprawnień.

Informacje pokrewne

[IBM Kontenerowe licencje](#)

[Często zadawane pytania dotyczące licencjonowania kontenerów](#)

[Instalowanie usługi License Service](#)

[Wyświetlanie i śledzenie wykorzystania licencji](#)

Adnotacje licencji podczas budowania własnego obrazu kontenera IBM MQ

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

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

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

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

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

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

Uwaga: Wiele adnotacji zawiera jeden lub oba następujące wiersze:

```
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
```

Przed użyciem adnotacji należy zmodyfikować następujące wiersze:

- W przypadku parametru `productChargedContainers` należy wybrać wartość "All" lub zastąpić rzeczywistą nazwą kontenera.
- Dla parametru `productMetric` należy wybrać jedną z oferowanych wartości.

Adnotacje do użycia z uprawnieniem do produktu IBM MQ

Jeśli masz uprawnienie do produktu IBM MQ, wybierz poniższą adnotację, która jest zgodna z uprawnieniem, które zostało zakupione i ma zostać użyte.

- ["IBM MQ" na stronie 17](#)
- ["IBM MQ Zaawansowane" na stronie 17](#)
- ["IBM MQ dla środowiska pozaprodukcyjnego" na stronie 17](#)
- ["IBM MQ Advanced for Non-Production Environment \(środowisko pozaprodukcyjne\)" na stronie 17](#)
- ["IBM MQ Advanced for Developers" na stronie 17](#)

Adnotacje IBM MQ, które mają być używane w konfiguracjach wysokiej dostępności z wieloma instancjami produktu IBM MQ, są następujące. Patrz także ["Wybieranie poprawnych adnotacji dla konfiguracji wysokiej dostępności"](#) na stronie 16.

- ["IBM MQ Kontener z wieloma instancjami" na stronie 17](#)
- ["IBM MQ Zaawansowany kontener-wiele instancji" na stronie 18](#)
- ["Produkt IBM MQ Container Multi Instance for Non-Production Environment" na stronie 18](#)
- ["IBM MQ Advanced Container Multi Instance for Non-Production Environment" na stronie 18](#)

Adnotacje do użycia z uprawnieniem do produktu CP4I

Jeśli masz upoważnienie IBM Cloud Pak for Integration (CP4I), wybierz poniżej adnotację, która jest zgodna z upoważnieniem, które zostało zakupione i ma zostać użyte.

- ["IBM MQ z upoważnieniem CP4I" na stronie 18](#)
- ["Uprawnienie IBM MQ Zaawansowane z CP4I" na stronie 18](#)
- ["IBM MQ for Non-Production Environment z uprawnieniem CP4I" na stronie 18](#)
- ["IBM MQ Advanced for Non-Production Environment z uprawnieniem CP4I" na stronie 18](#)

Adnotacje CP4I, które mają być używane w konfiguracjach wysokiej dostępności z wieloma instancjami produktu IBM MQ, są następujące. Patrz także ["Wybieranie poprawnych adnotacji dla konfiguracji wysokiej dostępności"](#) na stronie 16.

- ["IBM MQ Container Multi Instance z upoważnieniem CP4I" na stronie 19](#)

- [“IBM MQ Advanced Container Multi Instance z upoważnieniem CP4I” na stronie 19](#)
- [“IBM MQ Container Multi Instance for Non-Production Environment z uprawnieniem CP4I” na stronie 19](#)
- [“IBM MQ Advanced Container Multi Instance for Non-Production Environment z uprawnieniem CP4I” na stronie 19](#)

Wybieranie poprawnych adnotacji dla konfiguracji wysokiej dostępności

IBM MQ -wiele instancji

Podczas wdrażania pary menedżerów kolejek w konfiguracji wysokiej dostępności z wieloma instancjami programu IBM MQ należy użyć tej samej adnotacji w obu instancjach. W zależności od zakupionego uprawnienia należy wybrać jedną z następujących adnotacji:

- Uprawnienie autonomiczne IBM MQ lub IBM MQ Advanced
 - [“IBM MQ Kontener z wieloma instancjami” na stronie 17](#)
 - [“IBM MQ Zaawansowany kontener-wiele instancji” na stronie 18](#)
 - [“Produkt IBM MQ Container Multi Instance for Non-Production Environment” na stronie 18](#)
 - [“IBM MQ Advanced Container Multi Instance for Non-Production Environment” na stronie 18](#)
- IBM Cloud Pak for Integration upoważnienie
 - [“IBM MQ Container Multi Instance z upoważnieniem CP4I” na stronie 19](#)
 - [“IBM MQ Advanced Container Multi Instance z upoważnieniem CP4I” na stronie 19](#)
 - [“IBM MQ Container Multi Instance for Non-Production Environment z uprawnieniem CP4I” na stronie 19](#)
 - [“IBM MQ Advanced Container Multi Instance for Non-Production Environment z uprawnieniem CP4I” na stronie 19](#)

Jeśli ta opcja jest używana z upoważnieniem IBM Cloud Pak for Integration, współczynniki upoważnień w adnotacjach zapewniają, że zarejestrowane jest poprawne wykorzystanie upoważnień. W przypadku korzystania z uprawnień autonomicznych IBM MQ lub IBM MQ Advanced adnotacje raportowane w usłudze License Service dla każdej instancji muszą zostać odwzorowane na części uprawnień IBM MQ w następujący sposób:

- IBM MQ Advanced container wiele instancji
 - 1 x IBM MQ Advanced i 1 x IBM MQ Advanced High Availability Replica **lub**
 - 2 x IBM MQ Advanced³
- Wiele instancji IBM MQ Advanced container dla środowiska pozaprodukcyjnego
 - 1 x IBM MQ Advanced i 1 x IBM MQ Advanced High Availability Replica **lub**
 - 2 x IBM MQ Advanced for Non-Production Environment)³
- IBM MQ Kontener z wieloma instancjami
 - 1 x IBM MQ i 1 x IBM MQ High Availability Replica **lub**
 - 2 x IBM MQ³
- Produkt IBM MQ Container Multi Instance for Non-Production Environment
 - 1 x IBM MQ i 1 x IBM MQ High Availability Replica **lub**
 - 2 x IBM MQ for Non-Production Environment)³

IBM MQ Rodzima HA

³ Ta opcja uprawnienia jest nieoptymalna i powinna być używana tylko wtedy, gdy nie jest dostępne żadne uprawnienie do odpowiedniej części repliki wysokiej dostępności.

W przypadku wdrażania trzech menedżerów kolejek w kworum rodzimej wysokiej dostępności tylko aktywna instancja korzysta z uprawnień. Wszystkie instancje powinny mieć taką samą adnotację. W zależności od zakupionego uprawnienia należy wybrać jedną z następujących opcji:

- Uprawnienie autonomiczne IBM MQ lub IBM MQ Advanced
 - [“IBM MQ Zaawansowane” na stronie 17](#)
 - [“IBM MQ Advanced for Non-Production Environment \(środowisko pozaprodukcyjne\)” na stronie 17](#)
- IBM Cloud Pak for Integration upoważnienie
 - [“Uprawnienie IBM MQ Zaawansowane z CP4I” na stronie 18](#)
 - [“IBM MQ Advanced for Non-Production Environment z uprawnieniem CP4I” na stronie 18](#)

Adnotacje

W dalszej części tego tematu znajduje się treść każdej adnotacji.

IBM MQ

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

IBM MQ Zaawansowane

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

IBM MQ dla środowiska pozaprodukcyjnego

```
productID: "151bec68564a4a47a14e6fa99266deff"  
productName: "IBM MQ for Non-Production Environment"  
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Advanced for Non-Production Environment (środowisko pozaprodukcyjne)

```
productID: "21dfe9a0f00f444f888756d835334909"  
productName: "IBM MQ Advanced for Non-Production Environment"  
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Advanced for Developers

```
productID: "2f886a3eefbe4ccb89b2adb97c78b9cb"  
productName: "IBM MQ Advanced for Developers (Non-Warranted)"  
productMetric: "FREE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Kontener z wieloma instancjami

```
productID: "2dea73b866b648b6b4abe2a85eb76964"  
productName: "IBM MQ Container Multi Instance"
```

```
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Zaawansowany kontener-wiele instancji

```
productID: "bd35bff411bb47c2a3f3a4590f33a8ef"  
productName: "IBM MQ Advanced Container Multi Instance"  
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

Produkt IBM MQ Container Multi Instance for Non-Production Environment

```
productID: "af11b093f16a4a26806013712b860b60"  
productName: "IBM MQ Container Multi Instance for Non-Production Environment"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Advanced Container Multi Instance for Non-Production Environment

```
productID: "31f844f7a96b49749130cd0708fdbb17"  
productName: "IBM MQ Advanced Container Multi Instance for Non-Production Environment"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ z upoważnieniem CP4I

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

Uprawnienie IBM MQ Zaawansowane z CP4I

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

IBM MQ for Non-Production Environment z uprawnieniem CP4I

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

IBM MQ Advanced for Non-Production Environment z uprawnieniem CP4I

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

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

IBM MQ Container Multi Instance z upoważnieniem CP4I

```
productName: "IBM MQ Container Multi Instance"
productID: "2dea73b866b648b6b4abe2a85eb76964"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productCloudpakRatio: "10:3"
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

IBM MQ Advanced Container Multi Instance z upoważnieniem CP4I

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
cloudpakName: "IBM Cloud Pak for Integration"
productID: "bd35bff411bb47c2a3f3a4590f33a8ef"
productName: "IBM MQ Advanced Container Multi Instance"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productCloudpakRatio: "5:3"
```

IBM MQ Container Multi Instance for Non-Production Environment z uprawnieniem CP4I

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
cloudpakName: "IBM Cloud Pak for Integration"
productID: "af11b093f16a4a26806013712b860b60"
productName: "IBM MQ Container Multi Instance for Non-Production Environment"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productCloudpakRatio: "20:3"
```

IBM MQ Advanced Container Multi Instance for Non-Production Environment z uprawnieniem CP4I

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
cloudpakName: "IBM Cloud Pak for Integration"
productID: "31f844f7a96b49749130cd0708fdbb17"
productName: "IBM MQ Advanced Container Multi Instance for Non-Production Environments"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productCloudpakRatio: "10:3"
```

Zależności operatora IBM MQ Operator

Począwszy od wersji IBM MQ Operator 3.0 , podczas instalacji produktu IBM MQ Operatornie są automatycznie instalowane żadne inne operatory. W starszych wersjach systemu IBM MQ Operatoristnieje zależność od operatora IBM Cloud Pak foundational services , który instaluje również operator IBM Operand Deployment Lifecycle Manager (ODLM).

Operator licencjonowania produktu IBM musi być instalowany oddzielnie w celu śledzenia użycia licencji. Patrz sekcja [Wdrażanie usługi License Service](#) w dokumentacji produktu IBM Cloud Pak for Integration .

IBM MQ Operator 3.0 i nowsze



Podczas tworzenia QueueManager za pomocą licencji IBM Cloud Pak for Integration można wybrać, czy ma być używane pojedyncze logowanie w instancji IBM Cloud Pak for Integration Keycloak. Użycie opcji Keycloak jest domyślnie włączone z licencją IBM Cloud Pak for Integration, ale jeśli nie jest ona zainstalowana, QueueManager przejdzie w stan Zablokowany do czasu zainstalowania poprawnych zależności. Więcej informacji na temat zależności zawiera sekcja [“Instalowanie produktu IBM MQ Operator”](#) na stronie 105.

Starsze wersje produktu IBM MQ Operator

Operatory IBM Cloud Pak foundational services zostaną zainstalowane automatycznie podczas instalowania starszych wersji IBM MQ Operator. Te zależne operatory mają niewielki wpływ na procesor i pamięć. Są one używane do wdrażania dodatkowych zasobów w niektórych okolicznościach.

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

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

```
spec:
  license:
    accept: true
    license: L-AMRD-XH6P3Q
    use: "Production"
```

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

```
spec:
  license:
    accept: true
    license: L-RJON-CD3JKX
    use: "Production"
```

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

```
spec:
  license:
    accept: true
    license: L-RJON-CD3JKX
    use: "Production"
  web:
    enabled: false
```

Szczegółowe informacje na temat wymagań sprzętowych i programowych dla operatorów zależnych można znaleźć w sekcji [Wymagania sprzętowe i zalecenia dla usług podstawowych](#).

Można wybrać poziom wykorzystania CPU i pamięci przez menedżery kolejek. Więcej informacji zawiera temat [“.spec.queueManager.resources”](#) na stronie 196.

Odsyłacze pokrewne

[“Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1”](#) na stronie 187

Operator

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

IBM MQ Operator wymaga następujących uprawnień o zasięgu klastra:

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

Uwaga: IBM MQ Operator wymaga również uprawnień o zasięgu przestrzeni nazw. Jeśli IBM MQ Operator jest zainstalowany w zasięgu klastra, uprawnienia o zasięgu przestrzeni nazw są obecne we wszystkich przestrzeniach nazw.

Operator

IBM MQ Operator może działać w dwóch trybach pamięci masowej:

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



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

Efemeryczna pamięć masowa

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

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

Z tego powodu należy rozważyć, czy pamięć efemeryczna jest faktycznie odpowiednim wyborem w scenariuszu produkcyjnym, testowym lub programistycznym. Czy, na przykład, wiadomo, że wszystkie komunikaty są nietrwałe, a menedżer kolejek nie jest elementem klastra produktu MQ? Przy restarcie usuwany jest nie tylko cały stan środowiska przesyłania komunikatów, lecz także konfiguracja menedżera kolejek. Aby można było korzystać z całkowicie efemerycznego kontenera, konfiguracja IBM MQ musi być dodana do samego obrazu kontenera (więcej informacji na ten temat zawiera sekcja [“Budowanie obrazu z niestandardowymi plikami MQSC i INI przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift”](#) na stronie 169). 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 jest zwykle uruchamiany z trwałą pamięcią masową, aby zapewnić, że menedżer kolejek zachowa swoje trwałe komunikaty i konfigurację po restarcie. To jest zachowanie domyślne. Ponieważ istnieją różni dostawcy pamięci masowej, z których każdy obsługuje różne możliwości, często oznacza to, że wymagane jest dostosowanie konfiguracji. W poniższym przykładzie przedstawiono wspólne pola, które dostosowują konfigurację pamięci masowej IBM MQ w interfejsie API v1beta1 :

- Program **spec.queueManager.availability** steruje trybem dostępności. Jeśli używany jest system SingleInstance lub NativeHA, wymagana jest tylko pamięć masowa ReadWriteOnce . W przypadku systemu multiInstance wymagana jest klasa pamięci masowej, która obsługuje system ReadWriteMany z poprawnymi charakterystykami blokowania plików. Produkt IBM MQ udostępnia [instrukcję obsługi](#) i [instrukcję testowania](#). Tryb dostępności wpływa również na układ trwałego woluminu. Więcej informacji na ten temat zawiera sekcja [“Wysoka dostępność produktu IBM MQ w kontenerach”](#) na stronie 26.
- Program **spec.queueManager.storage** steruje poszczególnymi ustawieniami pamięci masowej. Menedżer kolejek można skonfigurować w taki sposób, aby używał od jednego do czterech woluminów trwałych.

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

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

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

```
spec:
  queueManager:
    availability:
      type: MultiInstance
    storage:
      queueManager:
        class: ibmc-file-gold-gid
      persistedData:
        enabled: true
        class: ibmc-file-gold-gid
      recoveryLogs:
        enabled: true
        class: ibmc-file-gold-gid
    securityContext:
      supplementalGroups: [65534] # Change to 99 for clusters with RHEL7 or earlier worker nodes
```

Informacje na temat zagadnień dotyczących pamięci masowej dla rodzimych menedżerów kolejek wysokiej dostępności zawiera sekcja [“Rodzima wysoka dostępność \(Native HA\)”](#) na stronie 147.

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

Pojemność pamięci masowej



Jeśli używana jest konsola IBM MQ Operator, żądana wielkość pamięci masowej jest stała i nie można zmienić jej wielkości po utworzeniu menedżera kolejek. Należy upewnić się, że wolumin jest wystarczająco duży dla potrzeb użytkownika.


Szyfrowanie



IBM MQ nie szyfruje aktywnie danych w spoczynku. Dlatego do szyfrowania wiadomości należy używać pasywnie zaszyfrowanej pamięci masowej, programu IBM MQ Advanced Message Security lub obu tych metod. W systemie IBM Cloud zarówno blokowa, jak i plikowa pamięć masowa są dostępne z pasywnym szyfrowaniem w spoczynku.

IBM MQ Advanced for Developers obraz kontenera

Wstępnie zbudowany obraz kontenera jest dostępny dla produktu IBM MQ Advanced for Developers. Ten obraz jest dostępny z serwisu IBM Container Registry. Ten obraz jest odpowiedni do użycia z Docker, Podman, Kubernetes i innymi środowiskami kontenerowymi.

Uwaga:  Obrazy IBM MQ Advanced for Developers były wcześniej dostępne w produkcji Docker Hub, ale jest to nieaktualne, a w produkcji Docker Hub nie są dostępne żadne dalsze aktualizacje.

Dostępne obrazy

Obrazy IBM MQ są przechowywane w katalogu IBM Container Registry:

- IBM MQ Advanced for Developers 9.3.0.17: `icr.io/ibm-messaging/mq:9.3.0.17-r3`
- IBM MQ Advanced for Developers 9.3.5.1: `icr.io/ibm-messaging/mq:9.3.5.1-r2`

Krótki przegląd

- Licencja:

- [IBM MQ Advanced for Developers](#) i [Apache License 2.0](#). Należy zauważyć, że licencja IBM MQ Advanced for Developers nie zezwala na dalszą dystrybucję, a warunki ograniczają użycie do komputera programisty.
- Gdzie należy zapisać problemy:
 - [GitHub](#)
- Dostępne dla następujących architektur CPU:
 - amd64
 - s390x
 - ppc64le

Użycie

Uruchom program [IBM MQ Advanced for Developers](#) w kontenerze.

Szczegółowe informacje na temat uruchamiania kontenera zawiera [dokumentacja składni](#).

Aby można było używać obrazu, należy zaakceptować warunki licencji IBM MQ, ustawiając zmienną środowiskową **LICENSE**.

Obsługiwane zmienne środowiskowe

lang

Ustaw język, w którym ma być drukowana licencja.

Licencja

Ustaw opcję `accept`, aby zaakceptować warunki licencji IBM MQ Advanced for Developers.

Ustaw `view`, aby wyświetlić warunki licencji.

Deprecated `log_format`

Nieaktualne: zastąpione przez “[MQ 9.3.2 Feb 2023]MQ_LOGGING_CONSOLE_FORMAT,” na stronie [25](#).

Zmień format dzienników, które są zapisywane w położeniu `stdout` kontenera.

Ustaw wartość `basic`, aby użyć prostego formatu czytelnego dla człowieka. Jest to wartość domyślna.

Ustaw wartość `json`, aby użyć formatu JSON (po jednym obiekcie JSON w każdym wierszu).

Deprecated `HASŁO MQ_ADMIN_PASSWORD`

Podaj hasło administratora.

Musi mieć długość co najmniej 8 znaków.

V 9.3.4 Brak domyślnego hasła dla administratora. W przypadku wersji IBM MQ Operator wcześniejszych niż 3.0.0 wartością domyślną jest `passwd`.

V 9.3.4 Od wersji IBM MQ 9.3.4ta zmienna jest nieaktualna. [Przykład YAML w tym temacie](#) przedstawia, w jaki sposób można samodzielnie utworzyć tę zmienną i zabezpieczyć ją za pomocą klucza tajnego.

Deprecated `MQ_APP_PASSWORD`

Podaj hasło użytkownika aplikacji.

Ustawienie tej opcji powoduje, że kanał **DEV.APP.SVRCONN** jest zabezpieczony i zezwala tylko na połączenia, które dostarczają poprawnego identyfikatora użytkownika i hasła.

Musi mieć długość co najmniej 8 znaków.

V 9.3.4 Brak domyślnego hasła dla użytkownika aplikacji. W wersjach produktu IBM MQ Operator starszych niż 3.0.0 wartość domyślna jest pusta (nie jest wymagane hasło) dla klientów IBM MQ i `passwd` dla klientów HTTP.

V 9.3.4 Od wersji IBM MQ 9.3.4ta zmienna jest nieaktualna. Przykład YAML w tym temacie przedstawia, w jaki sposób można samodzielnie utworzyć tę zmienną i zabezpieczyć ją za pomocą klucza tajnego.

MQ_DEV

Aby zatrzymać tworzone obiekty domyślne, należy ustawić wartość `false`.

MQ_ENABLE_METRICS

Aby wygenerować pomiary Prometheus dla menedżera kolejek, należy ustawić wartość `true`.

V 9.3.2 MQ_LOGGING_CONSOLE_SOURCE (źródło połączenia MSMQ)

Podaj rozdzielaną przecinkami listę źródeł dzienników, dla których tworzona jest kopia lustrzana w położeniu `stdout` kontenera.

Poprawne wartości to `qmgr` i `web`.

Wartością domyślną jest `qmgr, web`.

V 9.3.2 MQ_LOGGING_CONSOLE_FORMAT,

Zastępuje łańcuch “[Nieaktualny]log_format” na stronie 24.

Zmień format dzienników, które są zapisywane w położeniu `stdout` kontenera.

Ustaw wartość `basic`, aby użyć prostego formatu czytelnego dla człowieka. Jest to wartość domyślna.

Ustaw wartość `json`, aby użyć formatu JSON (po jednym obiekcie JSON w każdym wierszu).

V 9.3.2 MQ_LOGGING_CONSOLE_EXCLUDE_ID (Identyfikator klastra MSMQ)

Podaj rozdzielaną przecinkami listę identyfikatorów komunikatów dziennika, które są wykluczone.

Komunikaty dziennika nadal pojawiają się w pliku dziennika na dysku, ale nie są zapisywane w położeniu `stdout` kontenera.

Wartość domyślna to `AMQ5041I, AMQ5052I, AMQ5051I, AMQ5037I, AMQ5975I`.

mq_qmgr_name

Ustaw nazwę, z którą ma zostać utworzony menedżer kolejek.

Więcej informacji na temat domyślnej konfiguracji programisty obsługiwanej przez obraz IBM MQ Advanced for Developers zawiera [dokumentacja domyślnej konfiguracji programisty](#).

V 9.3.4 Przykładowy menedżer kolejek YAML opisujący sposób określania haseł dla użytkowników produktów admin i app

W produkcie IBM MQ 9.3.4 identyfikatory użytkowników `admin` i `app` nie mają już domyślnych haseł. Dla tych użytkowników należy podać hasła podczas wdrażania menedżera kolejek przy użyciu licencji `Development`. Poniżej przedstawiono przykładowy plik YAML menedżera kolejek, który przedstawia sposób wykonania tej czynności w programie IBM MQ Operator.

Poniższa komenda tworzy dane niejawne zawierające hasła dla użytkowników systemów `admin` i `app`.

```
oc create secret generic my-mq-dev-passwords --from-literal=dev-admin-password=passw0rd --from-literal=dev-app-password=passw0rd
```

Poniższe YAML używa tych haseł podczas wdrażania menedżera kolejek.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm-dev
spec:
  license:
    accept: false
    license: L-AXAF-JLZ53A
    use: Development
  web:
    enabled: true
  template:
    pod:
```

```

containers:
  - env:
    - name: MQ_DEV
      value: "true"
    - name: MQ_CONNAUTH_USE_HTP
      value: "true"
    - name: MQ_ADMIN_PASSWORD
      valueFrom:
        secretKeyRef:
          name: my-mq-dev-passwords
          key: dev-admin-password
    - name: MQ_APP_PASSWORD
      valueFrom:
        secretKeyRef:
          name: my-mq-dev-passwords
          key: dev-app-password
      name: qmgr
queueManager:
  storage:
    queueManager:
      type: persistent-claim
      name: QUICKSTART
  version: 9.3.5.1-r2

```

OpenShift CP4I Kubernetes Wysoka dostępność produktu IBM MQ

w kontenerach

W przypadku wysokiej dostępności z systemem IBM MQ Operator dostępne są trzy opcje: **Rodzimy menedżer kolejek wysokiej dostępności** (z aktywną repliką i dwiema replikami rezerwowymi), **Menedżer kolejek z wieloma instancjami** (jest to para typu aktywny-rezerwowi, korzystająca ze współużytkowanego, sieciowego systemu plików) lub **Pojedynczy odporny menedżer kolejek** (który oferuje proste podejście do wysokiej dostępności z użyciem sieciowej pamięci masowej). W tych dwóch ostatnich konfiguracjach dostępność odtwarzalnych danych zależy od systemu plików. W związku z tym, gdy nie jest stosowana rodzima wysoka dostępność, dostępność systemu plików jest krytyczna dla dostępności menedżera kolejek. Jeśli odtwarzanie danych jest istotne, system plików powinien zapewnić nadmiarowość przez replikację.

Dostępność **komunikatów i usług** 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.

Menedżer kolejek o rodzimej wysokiej dostępności

CP4I MQ Adv.

Rodzime menedżery kolejek o wysokiej dostępności obejmują **aktywne** i dwa **repliki** Kubernetes Podów, które działają jako część zestawu Kubernetes StatefulSet z dokładnie trzema replikami, z których każda ma własny zestaw woluminów trwałych Kubernetes. Wymagania IBM MQ dotyczące współużytkowanych systemów plików mają także zastosowanie w przypadku używania pojedynczego menedżera kolejek o rodzimej wysokiej dostępności (z wyjątkiem blokowania opartego na dzierżawie),

ale nie jest konieczne korzystanie ze współużytkowanego systemu plików. Można użyć blokowej pamięci masowej z odpowiednim systemem plików. Na przykład *xfs* lub *ext4*. Na czas przywracania sprawności pojedynczego menedżera kolejek o rodzimej wysokiej dostępności wpływają następujące czynniki:

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

Więcej informacji na ten temat zawiera [“Rodzima wysoka dostępność \(Native HA\)”](#) na stronie 147.

Wieloinstancyjny menedżer kolejek



Wieloinstancyjne menedżery kolejek korzystają z **aktywnego i rezerwowego** zasobnika Kubernetes, 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 **rezerwowy** miały współbieżny dostęp do woluminu trwałego. Aby uzyskać taką konfigurację, należy użyć trwałych woluminów Kubernetes z parametrem **access mode** ustawionym na `ReadWriteMany`. Woluminy muszą również spełniać wymagania produktu IBM MQ dotyczące współużytkowanych systemów plików, ponieważ produkt IBM MQ zakłada automatyczne zwalnianie blokad plików, gdy inicjuje przetęczenie awaryjne menedżera kolejek. IBM MQ generuje listę przetestowanych systemów plików.

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

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

Pojedynczy menedżer kolejek elastycznych



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

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

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

1. Czas pracy sondy działania i liczba tolerowanych przez nią niepowodzeń. Ten czynnik jest konfigurowalny.
2. Czas potrzebny programowi planującemu Kubernetes na przeniesienie harmonogramu niesprawnego zasobnika do nowego węzła.
3. Czas potrzebny na pobranie obrazu kontenera do nowego węzła. Jeśli parametr strategii **imagePullPolicy** jest ustawiony na `IfNotPresent`, obraz może już być dostępny w tym węźle.

4. Czas potrzebny na uruchomienie nowej instancji menedżera kolejek.
5. Czas potrzebny na wykrycie gotowości kontenera przez sondę gotowości zasobnika Kubernetes. Ten czynnik jest konfigurowalny.
6. Czas potrzebny na ponowne nawiązanie połączenia przez klienty IBM MQ.

Ważne:

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

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

1. Węzeł wróci do stanu, w którym węzeł główny Kubernetes będzie mógł się z nim komunikować.
2. W węzle 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.

Uwaga: Zmiana szczegółów StatefulSet menedżera kolejek systemu IBM MQ, w tym liczby replik, nie jest obsługiwana, jeśli menedżer kolejek został utworzony za pomocą programu IBM MQ Operator.

Pojęcia pokrewne

[Konfiguracje o wysokiej dostępności](#)

Zadania pokrewne

[“Konfigurowanie wysokiej dostępności na potrzeby menedżerów kolejek przy użyciu rozwiązania IBM MQ Operator” na stronie 147](#)

OpenShift CP4I Kubernetes Odtwarzanie po awarii w przypadku IBM MQ w kontenerach

Należy wziąć pod uwagę rodzaj awarii, na którą chcemy się przygotować. W środowiskach chmurowych korzystanie ze stref dostępności zapewnia pewien poziom tolerancji dla awarii i jest znacznie łatwiejsze w użyciu. Jeśli istnieje nieparzysta liczba centrów przetwarzania danych (dla kworum) i łącze sieciowe o niskim opóźnieniu, można potencjalnie uruchomić jeden klastr Red Hat OpenShift Container Platform lub Kubernetes z wieloma strefami dostępności, każdy w oddzielnym położeniu fizycznym. W tym temacie opisano zagadnienia dotyczące odtwarzania po awarii w przypadku, gdy nie można spełnić tych kryteriów: oznacza to, że istnieje parzysta liczba centrów przetwarzania danych lub łącze sieciowe o dużym opóźnieniu.

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

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

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

Podczas odtwarzania po awarii konieczne będzie ponowne utworzenie instancji menedżera kolejek w nowym klastrze Kubernetes przy użyciu replikowanych danych. Jeśli używasz IBM MQ Operator,

będziesz potrzebować kodu YAML QueueManager, a także kodu YAML dla innych zasobów pomocniczych takich jak ConfigMap czy Secret.

Informacje pokrewne

[ha_for_ctr.dita](#)

Planowanie zabezpieczenia produktu IBM MQ w kontenerach

Zagadnienia związane z bezpieczeństwem podczas planowania IBM MQ w konfiguracji kontenerów.

Procedura

- [“Uwierzytelnianie i autoryzacja użytkowników oprogramowania IBM MQ w kontenerach” na stronie 29](#)
 - [“Ograniczenia bezpieczeństwa dotyczące używania użytkowników systemu operacyjnego w kontenerach” na stronie 30](#)
- [“Uwagi dotyczące ograniczenia ruchu w sieci do IBM MQ w kontenerach” na stronie 30](#)

Uwierzytelnianie i autoryzacja użytkowników oprogramowania IBM MQ w kontenerach

Produkt IBM MQ w kontenerach można skonfigurować do uwierzytelniania użytkowników za pośrednictwem protokołu LDAP, wzajemnego TLS lub niestandardowej wtyczki produktu MQ .

Należy zauważyć, że operator IBM MQ nie zezwala na użycie użytkowników i grup systemu operacyjnego w obrazie kontenera. Więcej informacji na ten temat zawiera [“Ograniczenia bezpieczeństwa dotyczące używania użytkowników systemu operacyjnego w kontenerach” na stronie 30](#).

LDAP

Informacje na temat konfigurowania produktu IBM MQ do korzystania z repozytorium użytkowników LDAP zawiera sekcja [Uwierzytelnianie połączenia: repozytoria użytkowników](#) i sekcja [Autoryzacja LDAP](#).

Wzajemne uwierzytelnianie TLS

Jeśli połączenia przychodzące do menedżera kolejek są konfigurowane w taki sposób, aby wymagały certyfikatu TLS (wzajemnego TLS), można odwzorować nazwę wyróżniającą certyfikatu na nazwę użytkownika. Należy wykonać dwie czynności:

- Skonfiguruj rekord uwierzytelniania kanału, aby utworzyć odwzorowanie na nazwę użytkownika przy użyciu protokołu SSLPEER. Więcej informacji na ten temat zawiera sekcja [Odwzorowywanie nazwy wyróżniającej SSL lub TLS na identyfikator użytkownika MCAUSER](#).
- Skonfiguruj menedżer kolejek, aby umożliwić definiowanie rekordów uprawnień dla nazwy użytkownika, która nie jest znana w systemie. Więcej informacji na ten temat zawiera sekcja [Service pliku qm.ini](#).

Znaczniki WWW JSON

Informacje na temat konfigurowania produktu IBM MQ do używania znaczników JSON Web Token (JWT) zawiera sekcja [Praca ze znacznikami uwierzytelniania](#).

Niestandardowa wtyczka MQ

Jest to zaawansowana technika i wymaga dużo więcej pracy. Więcej informacji na ten temat zawiera sekcja [Korzystanie z niestandardowej usługi autoryzacji](#).

Zadania pokrewne

[“Przykład: konfigurowanie menedżera kolejek z wzajemnym uwierzytelnianiem TLS” na stronie 141](#)


W tym przykładzie menedżer kolejek jest wdrażany w OpenShift Container Platform przy użyciu IBM MQ Operator. Wzajemny protokół TLS jest używany do uwierzytelniania w celu odwzorowania certyfikatu TLS na tożsamość w menedżerze kolejek.

Ograniczenia bezpieczeństwa dotyczące używania użytkowników systemu operacyjnego w kontenerach

Używanie użytkowników systemu operacyjnego w kontenerach nie jest zalecane i jest zabronione w przypadku operatora IBM MQ .

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

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

 Aby zapewnić zgodność z tymi środkami bezpieczeństwa, IBM MQ Operator nie zezwala na użycie identyfikatorów, które są zdefiniowane w bibliotekach systemu operacyjnego wewnątrz kontenera. W kontenerze nie ma zdefiniowanego identyfikatora użytkownika lub grupy mqm.

Uwagi dotyczące ograniczenia ruchu w sieci do IBM MQ w kontenerach

Istnieje możliwość zdefiniowania strategii sieciowych w celu ograniczenia ruchu w klastrze do zasobników w środowisku OpenShift Container Platform i Kubernetes. W tym temacie opisano niektóre zagadnienia dotyczące sposobu, w jaki strategię sieciowe mogą mieć zastosowanie do produktu IBM MQ.

W przypadku sieciowego przyjmowania danych do menedżera kolejek należy rozważyć kilka portów:

- Port 1414 dla ruchu menedżera kolejek
- Port 9414 dla rodzimej wysokiej dostępności
- Port 9157 dla wielkości mierzonych
- Port 9443 dla konsoli WWW i interfejsów API REST

Wyjście sieci jest bardziej złożone. Przykłady wyjścia sieciowego, które można rozważyć:

- DNS-jeśli istnieją kanały lub inne konfiguracje, które używają nazw DNS
- Inne menedżery kolejek
- Protokół OCSP (Online Certificate Status Protocol) i listy odwołań certyfikatów (Certificate Revocation List-CRL)-określone przez dostawcę certyfikatów.
- Dostawcy uwierzytelniania:
 - LDAP
 - Otwórz program ID Connect lub innego skonfigurowanego dostawcy logowania dla serwera WWW IBM MQ . Obejmuje to interfejs użytkownika platformy IBM Cloud Pak Platform i moduł IAM IBM Cloud Pak Foundational Services.
- Dostawcy śledzenia:
 - Instana
 - Cloud Pak for Integration -panel kontrolny operacji⁴

⁴ Panel kontrolny operacji jest nieaktualny od wersji IBM MQ 9.3.0i został usunięty w wersji IBM MQ 9.3.3. Patrz sekcja [“Integracja z panelem kontrolnym IBM Cloud Pak for Integration Operations Dashboard”](#) na stronie 161.

Przykład zasobu Ingress NetworkPolicy

Poniżej przedstawiono przykładową strategię sieciową sterującą ingress dla menedżera kolejek o nazwie "myqm", do użycia w systemie Red Hat OpenShift Container Platform.

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: myqm
spec:
  podSelector:
    matchLabels:
      app.kubernetes.io/instance: myqm
      app.kubernetes.io/name: ibm-mq
  ingress:
    # Allow access to queue manager listener from anywhere
    - ports:
      - protocol: TCP
        port: 1414
    # Allow access to Native HA port from other instances of the same queue manager
    - from:
      - podSelector:
          matchLabels:
            app.kubernetes.io/instance: myqm
            app.kubernetes.io/name: ibm-mq
        ports:
      - protocol: TCP
        port: 9414
    # Allow access to metrics from monitoring project
    - from:
      - namespaceSelector:
          matchLabels:
            network.openshift.io/policy-group: monitoring
        ports:
      - protocol: TCP
        port: 9157
    # Allow access to web server via Route
    - from:
      - namespaceSelector:
          matchLabels:
            network.openshift.io/policy-group: ingress
        ports:
      - protocol: TCP
        port: 9443
```

Zgodność z FIPS dla IBM MQ w kontenerach

Podczas uruchamiania produktu IBM MQ w kontenerach wykrywa, czy system operacyjny, w którym uruchamiany jest kontener, jest zgodny ze standardem FIPS, a jeśli tak, to automatycznie konfiguruje obsługę standardu FIPS. W tym miejscu podano wymagania i ograniczenia.

Standardy FIPS (Federal Information Processing Standards)

Rząd Stanów Zjednoczonych udziela porad technicznych w zakresie systemów informatycznych i bezpieczeństwa, w tym szyfrowania danych. National Institute for Standards and Technology (NIST) jest organem rządowym zajmującym się systemami informatycznymi i bezpieczeństwem. NIST tworzy rekomendacje i standardy, w tym standardy FIPS (Federal Information Processing Standards).

Istotnym standardem FIPS jest FIPS 140-2, który wymaga użycia silnych algorytmów szyfrowania. Standard FIPS 140-2 określa również wymagania dotyczące algorytmów mieszających, które mają być używane do ochrony pakietów przed modyfikacją podczas przesyłania.

IBM MQ obsługuje standard FIPS 140-2, jeśli został w tym celu skonfigurowany.

Uwaga: W systemie AIX, Linux, and Windows IBM MQ zapewnia zgodność ze standardem FIPS 140-2 za pośrednictwem modułu szyfrującego IBM Crypto for C (ICC) . Certyfikat dla tego modułu został przeniesiony do statusu historycznego. Klienci powinni zapoznać się z informacjami w sekcji [Certyfikat IBM Crypto for C \(ICC\)](#) i zapoznać się z poradami NIST. Zastępczy moduł FIPS 140-3 jest obecnie w toku, a jego status można wyświetlić, wyszukując go na liście [Moduły NIST CMVP na liście procesów](#).

Wymagania

Wymagania związane z konfiguracją klastra i inne uwagi można znaleźć w sekcji [FIPS Wall: Current IBM approach to FIPS compliance](#) (Ściana FIPS: Bieżąca zgodność z FIPS).

Produkt IBM MQ w kontenerach może działać w trybie zgodności ze standardem FIPS 140-2. Podczas uruchamiania produkt IBM MQ w kontenerach (9.3.1.0 i nowsze) wykrywa, czy system operacyjny hosta, na którym uruchamiany jest kontener, jest zgodny ze standardem FIPS. Jeśli system operacyjny hosta jest zgodny ze standardem FIPS, a podano klucze prywatne i certyfikaty, kontener IBM MQ skonfiguruje menedżer kolejek, serwer WWW IBM MQ i przesyłanie danych między węzłami we wdrożeniu rodzimej wysokiej dostępności, tak aby działał w trybie zgodności ze standardami FIPS.

Jeśli do wdrażania menedżerów kolejek używany jest program IBM MQ Operator, operator tworzy trasę z typem zakończenia **Passthrough**. Oznacza to, że ruch jest wysyłany bezpośrednio do miejsca docelowego bez routera udostępniającego zakończenie TLS. W tym przypadku miejscami docelowymi są menedżer kolejek IBM MQ i serwer WWW IBM MQ, które już zapewniają bezpieczną komunikację zgodną ze standardem FIPS.

Kluczowe wymagania:

1. Klucz prywatny i certyfikaty udostępnione w danych niejawnych dla menedżera kolejek i serwera WWW, które umożliwiają zewnętrznym klientom bezpieczne nawiązanie połączenia z menedżerem kolejek i serwerem WWW.
2. Klucz prywatny i certyfikaty do przesyłania danych między różnymi węzłami w rodzimej konfiguracji wysokiej dostępności.

Ograniczenia

W przypadku wdrożenia produktu IBM MQ w kontenerach zgodnego ze standardem FIPS należy wziąć pod uwagę następujące kwestie:

- Produkt IBM MQ w kontenerach udostępnia punkt końcowy dla gromadzenia wielkości mierzonych. Obecnie ten punkt końcowy jest tylko punktem końcowym HTTP. Można wyłączyć punkt końcowy wielkości mierzonych, aby pozostała część IBM MQ była zgodna ze standardem FIPS.
- IBM MQ w kontenerach zezwala na niestandardowe nadpisanie obrazów. Oznacza to, że można tworzyć obrazy niestandardowe przy użyciu obrazu kontenera IBM MQ jako obrazu podstawowego. Zgodność ze standardami FIPS może nie mieć zastosowania w przypadku takich dostosowanych obrazów.
- W przypadku śledzenia komunikatów za pomocą IBM Instanakomunikacja między IBM MQ i IBM Instana odbywa się za pośrednictwem protokołu HTTP lub HTTPS bez zgodności ze standardami FIPS.
- Dostęp IBM MQ Operator do usług zarządzania tożsamością i dostępem (IAM) /Zen w systemie IBM nie jest zgodny ze standardem FIPS.

Sposób wykrywania zgodności ze standardami FIPS i automatycznego konfigurowania obsługi FIPS

Jeśli system operacyjny, w którym uruchamiany jest kontener, jest zgodny ze standardem FIPS, obsługa FIPS jest konfigurowana automatycznie.

Uwaga: W systemie AIX, Linux, and Windows IBM MQ zapewnia zgodność ze standardem FIPS 140-2 za pośrednictwem modułu szyfrującego IBM Crypto for C (ICC). Certyfikat dla tego modułu został przeniesiony do statusu historycznego. Klienci powinni zapoznać się z informacjami w sekcji [Certyfikat IBM Crypto for C \(ICC\)](#) i zapoznać się z poradami NIST. Zastępczy moduł FIPS 140-3 jest obecnie w toku, a jego status można wyświetlić, wyszukując go na liście [Moduły NIST CMVP na liście procesów](#).

Podczas uruchamiania produkt IBM MQ w kontenerach wykrywa, czy system operacyjny, w którym uruchamiany jest kontener, jest zgodny ze standardem FIPS. Jeśli tak, to automatycznie wykonywane są następujące działania:

Menedżer kolejek

Jeśli system operacyjny hosta jest zgodny ze standardem FIPS oraz podano klucz prywatny i certyfikaty, atrybut **SSLFIPS** menedżera kolejek jest ustawiony na wartość YES. W przeciwnym razie atrybut **SSLFIPS** jest ustawiony na wartość NO.

IBM MQ serwer WWW

Serwer WWW IBM MQ udostępnia interfejs HTTP/HTTPS do administrowania produktem IBM MQ. Jeśli system operacyjny hosta jest zgodny ze standardem FIPS, opcje maszyny JVM są aktualizowane, aby serwer WWW używał szyfrowania zgodnego ze standardem FIPS. Aby można było korzystać ze standardu FIPS, podczas uruchamiania kontenera należy podać klucz prywatny i certyfikaty.

Rodzima wysoka dostępność (Native HA)

Bezpieczeństwo danych replikowanych między węzłami jest kontrolowane przez sekcję **NativeHALocalInstance** pliku `qm.ini`. Na przykład:

```
NativeHALocalInstance:
  KeyRepository=/run/runmqserver/ha/tls/key.kdb
  CertificateLabel=NHAQM
  CipherSpec=ECDHE_RSA_AES_256_GCM_SHA384
```

Jeśli tryb FIPS jest włączony, atrybut **SSLFipsRequired** jest dodawany do sekcji z wartością Tak:

```
NativeHALocalInstance:
  KeyRepository=/run/runmqserver/ha/tls/key.kdb
  CertificateLabel=NHAQM
  CipherSpec=ECDHE_RSA_AES_256_GCM_SHA384
  SSLFipsRequired=Yes
```

Jeśli kontener działa w klastrze OpenShift bez obsługi FIPS, to menedżer kolejek, serwer WWW IBM MQ i rodzime komponenty wysokiej dostępności nie mają automatycznie włączonej obsługi FIPS. Obecnie tylko architektura x86-64 jest obsługiwana przez platformę OpenShift w trybie FIPS. W architekturach Power i Linux for IBM Z produkt OpenShift nie obsługuje standardu FIPS. Aby jawnie włączyć obsługę FIPS w komponentach IBM MQ dla tych architektur, należy ustawić zmienną środowiskową `MQ_ENABLE_FIPS` na wartość `true` w menedżerze kolejek YAML. Poniższy fragment kodu YAML opisuje użycie zmiennej środowiskowej `MQ_ENABLE_FIPS`:

```
template:
  pod:
    containers:
      - env:
          - name: MQ_ENABLE_FIPS
            value: "true"
        name: qmgr
```

Nadpisywanie automatycznego trybu FIPS dla modułu IBM MQ w kontenerach

Użyj zmiennej środowiskowej `MQ_ENABLE_FIPS`, aby jawnie włączyć lub wyłączyć tryb FIPS dla komponentów IBM MQ w kontenerze.

Zanim rozpoczniesz

Uwaga: W systemie AIX, Linux, and Windows IBM MQ zapewnia zgodność ze standardem FIPS 140-2 za pośrednictwem modułu szyfrującego IBM Crypto for C (ICC). Certyfikat dla tego modułu został przeniesiony do statusu historycznego. Klienci powinni zapoznać się z informacjami w sekcji [Certyfikat IBM Crypto for C \(ICC\)](#) i zapoznać się z poradami NIST. Zastępczy moduł FIPS 140-3 jest obecnie w toku, a jego status można wyświetlić, wyszukując go na liście [Moduły NIST CMVP](#) na liście procesów.

O tym zadaniu

`MQ_ENABLE_FIPS` obsługuje trzy wartości:

automatycznie

Jest to wartość domyślna.

Jeśli w systemie operacyjnym hosta włączono obsługę standardu FIPS, wszystkie komponenty (menedżer kolejek, serwer WWW IBM MQ i rodzima wysoka dostępność) działają w trybie FIPS.

Jeśli system operacyjny hosta nie obsługuje standardu FIPS, wszystkie komponenty nie będą działać w trybie FIPS.

Prawda

Ta wartość włącza tryb FIPS dla wybranych komponentów w kontenerze.

Atrybut menedżera kolejek **SSLFIPS** ma wartość YES, nawet jeśli produkt IBM MQ w kontenerach działa w systemie operacyjnym hosta, który nie jest zgodny ze standardem FIPS. Oznacza to, że jeśli menedżer kolejek produktu IBM MQ, serwer WWW i rodzima konfiguracja wysokiej dostępności są zgodne ze standardem FIPS, ale system operacyjny kontenera nie jest zgodny z tym standardem.

Fałsz

Ta wartość wyłącza zgodność ze standardami FIPS.

Atrybut menedżera kolejek **SSLFIPS** jest ustawiony na wartość NO, nawet jeśli produkt IBM MQ w kontenerach jest uruchomiony na komputerze hosta zgodnym ze standardem FIPS. Jednak program IBM MQ nadal zabezpiecza połączenia, jeśli zostanie podany klucz prywatny i certyfikaty.

Opcje maszyny JVM dla serwera WWW IBM MQ nie są aktualizowane. Jednak serwer WWW IBM MQ nadal uruchamia punkt końcowy HTTPS, jeśli podano klucz prywatny i certyfikaty.

Replikacja danych w rodzimej wysokiej dostępności nie używa szyfrowania FIPS.

Przykład

Poniżej przedstawiono przykładowy plik YAML menedżera kolejek, który opisuje włączanie protokołów TLS i FIPS dla komponentu menedżera kolejek:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  namespace: ibm-mq-fips
  name: ibm-mq-qm-ppcle
spec:
  license:
    accept: true
    license: L-AMRD-XH6P3Q
    use: Production
  queueManager:
    name: PPCLEQM
    storage:
      queueManager:
        type: ephemeral
  template:
    pod:
      containers:
        - env:
            - name: MQ_ENABLE_FIPS
              value: "true"
          name: qmgr
  version: 9.3.5.1-r2
  web:
    enabled: false
  pki:
    keys:
      - name: ibm-mq-tls-certs
        secret:
          secretName: ibm-mq-tls-secret
          items:
            - tls.key
            - tls.crt
```

Multi Planowanie skalowalności i wydajności produktu IBM MQ w kontenerach

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

O tym zadaniu

Podczas planowania skalowalności i wydajności produktu IBM MQ w kontenerach należy wziąć pod uwagę następujące opcje:

Procedura

- **Ogranicz liczbę wątków i procesów.**

Program IBM MQ używa wątków do zarządzania współbieżnością. W produkcie Linux wątki są implementowane jako procesy, dlatego można napotkać limity narzucone przez platformę kontenerową lub system operacyjny na maksymalną liczbę procesów. Od wersji Red Hat OpenShift Container Platform 4.11 obowiązuje domyślny limit 4096 procesów na kontener. W przypadku starszych wersji systemu OpenShift Container Platform limit wynosi 1024 procesy. Informacje na temat kompatybilności wersji IBM MQ Operator z wersjami OpenShift Container Platform zawiera sekcja [“Kompatybilne wersje środowiska Red Hat OpenShift Container Platform”](#) na stronie 13. Jest to odpowiednie rozwiązanie w przypadku większości scenariuszy, ale mogą wystąpić przypadki, w których może to mieć wpływ na liczbę połączeń klienckich dla menedżera kolejek.

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

W konfiguracji programu IBM MQ można również ustawić maksymalną liczbę połączeń klienckich dla menedżera kolejek. Sekcja [Limity kanału połączenia z serwerem](#) zawiera informacje na temat stosowania limitów dla pojedynczego kanału połączenia z serwerem, a sekcja [Atrybut MAXCHANNELS INI](#) zawiera informacje na temat stosowania limitów dla całego menedżera kolejek.

- **Ogranicz liczbę woluminów.**

W systemach chmurowych i kontenerowych powszechnie używane są woluminy sieciowej pamięci masowej. Istnieje ograniczenie liczby woluminów, które można podłączyć do węzłów Linux. Na przykład komenda [AWS EC2](#) ogranicza się do nie więcej niż 30 woluminów na maszynę wirtualną. Red Hat OpenShift Container Platform ma podobny limit, jak Microsoft Azure i Google Cloud Platform.

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

- **Użyj IBM MQ technik skalowania.**

Zamiast małej liczby dużych menedżerów kolejek korzystne może być użycie technik skalowania produktu IBM MQ, takich jak klastry jednolite produktu IBM MQ, w celu uruchomienia wielu menedżerów kolejek z taką samą konfiguracją. Dodatkową korzyścią jest zmniejszenie wpływu restartu pojedynczego kontenera (na przykład w ramach konserwacji platformy kontenerów).

Korzystanie z programu IBM MQ Operator w systemie Red Hat OpenShift

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

Procedura

- [“Historia wydań operatora IBM MQ Operator”](#) na stronie 36.
- [“Migracja produktu IBM MQ do rozwiązania IBM Cloud Pak for Integration”](#) na stronie 81.
- [“Instalowanie produktu IBM MQ Operator”](#) na stronie 105.
- [“Aktualizowanie operatora IBM MQ Operator i menedżerów kolejek”](#) na stronie 123.

- [“Wdrażanie menedżera kolejek w klastrze platformy kontenerowej Red Hat OpenShift” na stronie 118.](#)
- [“Użytkowanie produktu IBM MQ przy wykorzystaniu operatora IBM MQ Operator” na stronie 176.](#)
- [“Skorowidz interfejsu API operatora IBM MQ Operator” na stronie 187.](#)

OpenShift CP4I-LTS CP4I CD **Historia wydań operatora IBM MQ**

Operator

Uwagi:

- Więcej informacji na temat wcześniejszych operatorów IBM MQ zawiera sekcja [Historia wydań produktu IBM MQ Operator](#) w dokumentacji produktu IBM MQ 9.2 .
- Informacje o przyszłych aktualizacjach produktu IBM MQ można znaleźć na stronie ogólnej [IBM MQ planowanych dat wersji serwisowych](#) .

IBM MQ Operator 3.1.3



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.4.1

Kanał operatora

v3.1

Dozwolone wartości dla `.spec.version`

[9.3.5.1-r2](#)

Dozwolone wartości parametru `.spec.version` podczas migracji

9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1, [9.3.0.17-r2](#), 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.3.3-r2, 9.3.4.0-r1, 9.3.4.1-r1, 9.3.5.0-r1, 9.3.5.0-r2, [9.3.5.1-r1](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersji 4.3 lub nowszej (instalacja opcjonalna).

Co się zmieniło

- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 3.1.2



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.4.1

Kanał operatora

v3.1

Dozwolone wartości dla `.spec.version`

[9.3.5.1-r1](#)

Dozwolone wartości parametru `.spec.version` podczas migracji

9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2,

9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.3.3-r2, 9.3.4.0-r1, 9.3.4.1-r1, 9.3.5.0-r1, 9.3.5.0-r2,

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersji 4.3 lub nowszej (instalacja opcjonalna).

Co się zmieniło

- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 3.1.1



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.4.1

Kanał operatora

v3.1

Dozwolone wartości dla .spec.version

[9.3.5.0-r2](#)

Dozwolone wartości parametru .spec.version podczas migracji

9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, [9.3.0.16-r2](#), 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, [9.3.3.3-r2](#), 9.3.4.0-r1, 9.3.4.1-r1, 9.3.5.0-r1

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersji 4.3 lub nowszej (instalacja opcjonalna).

Co się zmieniło

- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 3.1.0



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.4.1

Kanał operatora

v3.1

Dozwolone wartości dla .spec.version

[9.3.5.0-r1](#)

Dozwolone wartości parametru .spec.version podczas migracji

9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, [9.3.0.16-r1](#), 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1,

9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.3.3-r2, 9.3.4.0-r1, 9.3.4.1-r1

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersji 4.3 lub nowszej (instalacja opcjonalna).

Co się zmieniło

- Słabe punkty zabezpieczeń, które są adresowane, są szczegółowo opisane w następujących biuletynie bezpieczeństwa:
 - <https://www.ibm.com/support/pages/node/7126571>.
 - <https://www.ibm.com/support/pages/node/7137570>.

IBM MQ Operator 3.0.1



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.4.1

Kanał operatora

v3.0

Dozwolone wartości dla `.spec.version`

9.3.4.1-r1

Dozwolone wartości parametru `.spec.version` podczas migracji

9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.3.2-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.4.0-r1

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersji 4.3 lub nowszej (instalacja opcjonalna).

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie “IBM MQ Operator 3.0.0” na stronie 38.
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 3.0.0



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.4.1

Kanał operatora

v3.0

Dozwolone wartości dla `.spec.version`

9.3.4.0-r1

Dozwolone wartości parametru `.spec.version` podczas migracji

9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersji 4.3 lub nowszej (instalacja opcjonalna).

Co nowego

- Serwer WWW IBM MQ można skonfigurować, dodając plik `mqwebuser.xml` do obiektu ConfigMap lub Secret za pomocą nowej właściwości `manualConfig` YAML (wymaga produktu IBM MQ 9.3.4 lub nowszej).
- Produkt `administrative` REST API jest teraz obsługiwany. Można to skonfigurować za pomocą komendy ConfigMap lub klucza tajnego (wymaga produktu IBM MQ 9.3.4 lub nowszej). Należy jednak zauważyć, że serwer WWW nadal nie jest uważany za usługę krytyczną dla sondy aktywności, więc jeśli jej działanie nie powiedzie się, kontener nie zostanie automatycznie zrestartowany.
- Pojedyncze logowanie można wyłączyć podczas korzystania z licencji IBM Cloud Pak for Integration, wybierając opcję "ręcznego" uwierzytelniania i autoryzacji (wymaga produktu IBM MQ 9.3.4 lub nowszej).
- W kontenerze można włączyć główny system plików tylko do odczytu. Zwiększa to bezpieczeństwo, uniemożliwiając zapis do większości plików w kontenerze w czasie wykonywania (wymaga produktu IBM MQ 9.3.4 lub nowszej). Opcji `readOnlyRootFilesystem` towarzyszą dodatkowe opcje służące do konfigurowania wielkości woluminów "scratch" i "tmp", które są podłączane w celu umożliwienia zapisu plików tymczasowych. Więcej informacji znajduje się w sekcji "[Uruchamianie kontenera IBM MQ z systemem plików root tylko do odczytu](#)" na stronie 173

Co się zmieniło

- Usunięte (poprzednio nieaktualne) wersje: IBM MQ 9.2.0 EUS, 9.2.3, 9.2.4, 9.2.5. Ważne: przed aktualizacją produktu IBM MQ Operator należy upewnić się, że nie ma menedżerów kolejek dla żadnej z usuniętych wersji. Po aktualizacji nie będzie już można edytować zasobu QueueManager, poza aktualizacją do wersji z obsługą, ponieważ IBM MQ Operator nie rozpoznaje już starszych wersji.
- Instalacja i cykl życia operatora
 - Produkt IBM MQ Operator jest teraz obsługiwany w systemie Red Hat OpenShift Container Platform w wersji 4.14.
 - Produkt IBM MQ Operator nie jest już automatycznie instalowany IBM Cloud Pak foundational services. Jeśli wdrażany jest QueueManager, który używa licencji IBM Cloud Pak for Integration i który konfiguruje pojedyncze logowanie (domyślnie dla menedżerów kolejek z tą licencją), QueueManager przejdzie w stan Zablokowany, jeśli wymagane zależności nie są jeszcze zainstalowane. Żadne inne operatory nie zostaną zainstalowane automatycznie.
- Zmiany dotyczące zabezpieczeń
 - Program IBM Cloud Pak for Integration 2023.4.1 używa Keycloak do pojedynczego logowania i autoryzacji zamiast IBM Cloud Pak Identity and Access Manager.
 - Szablon Szybki start produktu IBM Cloud Pak for Integration nie wyłącza już zabezpieczeń za pomocą komendy `MQSNOAUT`. Należy skonfigurować uwierzytelnianie. Patrz sekcja "[Uwierzytelnianie i autoryzacja użytkowników oprogramowania IBM MQ w kontenerach](#)" na stronie 29.

- Wyłączono domyślnych użytkowników w programie IBM MQ Advanced for Developers z wersji 9.3.4. Domyślnie wyłączeni są użytkownicy domyślni ("admin" i "app") oraz inne konfiguracje dostarczane jako część IBM MQ Advanced for Developers .
- Drobne zmiany w sekcji IBM MQ Operator :
 - IBM MQ Operator nie wdraża już kontenera inicjującego
 - Nazwą kontenera IBM MQ Operator jest teraz *menedżer*
 - Przedrostek zasobnika IBM MQ Operator to *ibm-mq-operator*
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.4.8

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Kanał operatora

v2.4

Dozwolone wartości dla `.spec.version`

[9.3.3.3-r2](#)

Dozwolone wartości parametru `.spec.version` podczas migracji

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, [9.3.0.16-r1](#), 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersjach 3.19 do 3.24 włącznie.

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.4.0”](#) na stronie 44.
- Słabe punkty zabezpieczeń, które są adresowane, są szczegółowo opisane w następujących biuletynie bezpieczeństwa:
 - <https://www.ibm.com/support/pages/node/7126571>.
 - <https://www.ibm.com/support/pages/node/7137570>.

IBM MQ Operator 2.4.7

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Kanał operatora

v2.4

Dozwolone wartości dla `.spec.version`

[9.3.3.3-r1](#)

Dozwolone wartości parametru `.spec.version` podczas migracji

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4,

9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersjach 3.19 do 3.24 włącznie.

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.4.0”](#) na stronie 44.
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.4.6



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Kanał operatora

v2.4

Dozwolone wartości dla .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersjach 3.19 do 3.24 włącznie.

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.4.0”](#) na stronie 44.
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.4.5



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Kanał operatora

v2.4

Dozwolone wartości dla .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2

r2, 9.3.0.11-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, [9.3.3.2-r2](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersjach 3.19 do 3.24 włącznie.

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.4.0”](#) na stronie 44.
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.4.4



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Kanał operatora

v2.4

Dozwolone wartości dla .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, [9.3.0.11-r1](#), 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, [9.3.3.2-r1](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersjach 3.19 do 3.24 włącznie.

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.4.0”](#) na stronie 44.
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).
- Produkt IBM MQ Operator nie jest już testowany ani obsługiwany w systemie OpenShift Container Platform 4.10.

IBM MQ Operator 2.4.3



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Kanał operatora

v2.4

Dozwolone wartości dla .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1,

[9.3.0.10-r2](#), [9.3.1.0-r1](#), [9.3.1.0-r2](#), [9.3.1.0-r3](#), [9.3.1.1-r1](#), [9.3.2.0-r1](#), [9.3.2.0-r2](#), [9.3.2.1-r1](#), [9.3.2.1-r2](#), [9.3.3.0-r1](#), [9.3.3.0-r2](#), [9.3.3.1-r1](#), [9.3.3.1-r2](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersjach 3.19 do 3.24 włącznie.

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.4.0”](#) na stronie 44.
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.4.2



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Kanał operatora

v2.4

Dozwolone wartości dla `.spec.version`

[9.2.0.1-r1-eus](#), [9.2.0.2-r1-eus](#), [9.2.0.2-r2-eus](#), [9.2.0.4-r1-eus](#), [9.2.0.5-r1-eus](#), [9.2.0.5-r2-eus](#), [9.2.0.5-r3-eus](#), [9.2.0.6-r1-eus](#), [9.2.0.6-r2-eus](#), [9.2.0.6-r3-eus](#), [9.2.3.0-r1](#), [9.2.4.0-r1](#), [9.2.5.0-r1](#), [9.2.5.0-r2](#), [9.2.5.0-r3](#), [9.3.0.0-r1](#), [9.3.0.0-r2](#), [9.3.0.0-r3](#), [9.3.0.1-r1](#), [9.3.0.1-r2](#), [9.3.0.1-r3](#), [9.3.0.1-r4](#), [9.3.0.3-r1](#), [9.3.0.4-r1](#), [9.3.0.4-r2](#), [9.3.0.5-r1](#), [9.3.0.5-r2](#), [9.3.0.5-r3](#), [9.3.0.6-r1](#), [9.3.0.10-r1](#), [9.3.1.0-r1](#), [9.3.1.0-r2](#), [9.3.1.0-r3](#), [9.3.1.1-r1](#), [9.3.2.0-r1](#), [9.3.2.0-r2](#), [9.3.2.1-r1](#), [9.3.2.1-r2](#), [9.3.3.0-r1](#), [9.3.3.0-r2](#), [9.3.3.1-r1](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersjach 3.19 do 3.24 włącznie.

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.4.0”](#) na stronie 44.
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.4.1



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Kanał operatora

v2.4

Dozwolone wartości dla `.spec.version`

[9.2.0.1-r1-eus](#), [9.2.0.2-r1-eus](#), [9.2.0.2-r2-eus](#), [9.2.0.4-r1-eus](#), [9.2.0.5-r1-eus](#), [9.2.0.5-r2-eus](#), [9.2.0.5-r3-eus](#), [9.2.0.6-r1-eus](#), [9.2.0.6-r2-eus](#), [9.2.0.6-r3-eus](#), [9.2.3.0-r1](#), [9.2.4.0-r1](#), [9.2.5.0-r1](#), [9.2.5.0-r2](#), [9.2.5.0-r3](#), [9.3.0.0-r1](#), [9.3.0.0-r2](#), [9.3.0.0-r3](#), [9.3.0.1-r1](#), [9.3.0.1-r2](#), [9.3.0.1-r3](#), [9.3.0.1-r4](#), [9.3.0.3-r1](#), [9.3.0.4-r1](#), [9.3.0.4-r2](#), [9.3.0.5-r1](#), [9.3.0.5-r2](#), [9.3.0.5-r3](#), [9.3.0.6-r1](#), [9.3.1.0-r1](#), [9.3.1.0-r2](#), [9.3.1.0-r3](#), [9.3.1.1-r1](#), [9.3.2.0-r1](#), [9.3.2.0-r2](#), [9.3.2.1-r1](#), [9.3.2.1-r2](#), [9.3.3.0-r1](#), [9.3.3.0-r2](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersjach 3.19 do 3.24 włącznie.

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.4.0”](#) na stronie 44.
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.4.0



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Kanał operatora

v2.4

Dozwolone wartości dla `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, [9.3.0.5-r3](#), 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, [9.3.3.0-r1](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersjach 3.19 do 3.24 włącznie.

Co nowego

- Usunięto integrację panelu kontrolnego operacji.
- Dodano obsługę IBM MQ Operator dla **LogFilePages**.

Co się zmieniło

- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.3.3



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.4.1

Kanał operatora

v2.3

Dozwolone wartości dla `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, [9.3.0.5-r2](#), 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, [9.3.2.1-r2](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersjach 3.19 do 3.24 włącznie.

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.3.0”](#) na stronie 46
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.3.2



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.4.1

Kanał operatora

v2.3

Dozwolone wartości dla .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, [9.3.0.5-r1](#), 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, [9.3.2.1-r1](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersjach 3.19 do 3.24 włącznie.

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.3.0”](#) na stronie 46
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.3.1



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.4.1

Kanał operatora

v2.3

Dozwolone wartości dla .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, [9.3.0.4-r2](#), 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, [9.3.2.0-r2](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersjach 3.19 do 3.24 włącznie.

Co nowego

- Od marca 2023 r. obrazy kontenerów menedżera kolejek IBM MQ Operator i IBM MQ są podpisane cyfrowo. Obrazy IBM MQ Operator 2.3.1 i IBM MQ 9.3.2.0-r2 zostały podpisane w tej wersji. Patrz sekcja [“Weryfikowanie podpisów obrazów”](#) na stronie 81.

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.3.0”](#) na stronie 46
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.3.0



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.4.1

Kanał operatora

v2.3

Dozwolone wartości dla `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, [9.3.0.4-r1](#), 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, [9.3.2.0-r1](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersjach 3.19 do 3.24 włącznie.

Co nowego

- Począwszy od wersji IBM MQ Operator 2.3.0, można skonfigurować obsługę FIPS 140-2. Patrz [“Zgodność z FIPS dla IBM MQ w kontenerach”](#) na stronie 31.
- Począwszy od wersji IBM MQ Operator 2.3.0 produkt IBM MQ 9.3.1 jest nieaktualny.

Co się zmieniło

- Począwszy od wersji IBM MQ Operator 2.3.0 produkt [“Rodzima wysoka dostępność \(Native HA\)”](#) na stronie 147 jest dostępny na licencji IBM MQ Advanced lub IBM MQ Advanced for Developers .
- Zestaw uprawnień wymaganych przez IBM MQ Operator został zredukowany.
- W wersji IBM MQ Operator 2.3.0 plik **ibm-automation-core** został usunięty z pliku OperandRequest przeznaczonego dla wdrożeń IBM Cloud Pak for Integration .
- W wersji IBM MQ Operator 2.3.0 wdrożenie IBM MQ Operator określa **imagePullPolicy** IfNotPresent.
- Słabe punkty zabezpieczeń, które są adresowane, zostały szczegółowo opisane w następujących biuletynie bezpieczeństwa:
 - [Biuletyn CVE-2022-47629 i CVE-2022-35737](#)
 - [Biuletyn CVE-2023-26284](#)

IBM MQ Operator 2.2.2



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.4.1

Kanał operatora

v2.2

Dozwolone wartości dla `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, [9.3.0.3-r1](#), 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, [9.3.1.1-r1](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersjach 3.19 do 3.24 włącznie.

Co się zmieniło

- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.2.1



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.4.1

Kanał operatora

v2.2

Dozwolone wartości dla `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, [9.3.0.1-r4](#), 9.3.1.0-r1, 9.3.1.0-r2, [9.3.1.0-r3](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersjach 3.19 do 3.24 włącznie.

Co się zmieniło

- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.2.0



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.4.1

Kanał operatora

v2.2

Dozwolone wartości dla `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1,

9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.1.0-r1, 9.3.1.0-r2

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services w wersjach 3.19 do 3.24 włącznie.

Co nowego

- Od wersji IBM MQ Operator 2.2.0 (CD) śledzenie IBM Instana jest obsługiwane w trybie rodzimym. Obsługa jest dostępna w obrazie kontenera menedżera kolejek 9.3.1.0-r2 (CD) IBM MQ . Produkt 9.3.1.0-r2 zawiera program w wersji 2.4.0 (2022.4.0) programu IBM Instana MQ Exit. Aby włączyć śledzenie IBM Instana , patrz sekcja “Integrowanie produktu IBM MQ z funkcją śledzenia produktu IBM Instana” na stronie 162.

Co się zmieniło

- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).
- Od wersji IBM MQ Operator 2.2.0 panel kontrolny operacji jest nieaktualny i nie będzie otrzymywał dalszych aktualizacji. Nie należy uruchamiać nowych zastosowań panelu kontrolnego operacji. IBM MQ 2.0.x Operatory nadal obsługują panel kontrolny operacji.

IBM MQ Operator 2.1.0



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.1

Dozwolone wartości dla .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.1.0-r1

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.X, ale co najmniej 3.19

Co nowego

- Dodaje IBM MQ 9.3.1.
- Dodaje nową opcję, która umożliwia użytkownikom wyłączanie aktualizacji wartości domyślnych w specyfikacji menedżera kolejek.
- Dodaje nowy warunek statusu, który powoduje dezaktualizację wszystkich wersji produktu IBM MQ wcześniejszych niż IBM MQ 9.3.1.
- Dodaje nowy warunek statusu ostrzegający użytkowników, którzy używają operandów LTS z tą wersją CD IBM MQ Operator.

Co się zmieniło

- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.23 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla `.spec.version`

[9.3.0.17-r3](#)

Dozwolone wartości parametru `.spec.version` podczas migracji

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1, 9.3.0.17-r2

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- IBM MQ Obraz katalogu został przeniesiony do plikowego formatu katalogu z formatu bazy danych SQLite .
- Słabe punkty zabezpieczeń, które zostały usunięte, są szczegółowo opisane w tym [biuletynie bezpieczeństwa](#) .

IBM MQ Operator 2.0.22 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla `.spec.version`

[9.3.0.17-r2](#)

Dozwolone wartości parametru `.spec.version` podczas migracji

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60

- Słabe punkty zabezpieczeń, które zostały usunięte, są szczegółowo opisane w tym [biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.21 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla .spec.version

[9.3.0.17-r1](#)

Dozwolone wartości parametru .spec.version podczas migracji

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- Słabe punkty zabezpieczeń, które zostały usunięte, są szczegółowo opisane w tym [biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.20 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla .spec.version

[9.3.0.16-r2](#)

Dozwolone wartości parametru .spec.version podczas migracji

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60

- Słabe punkty zabezpieczeń, które zostały usunięte, są szczegółowo opisane w tym [biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.19 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla `.spec.version`

9.3.0.16-r1

Dozwolone wartości parametru `.spec.version` podczas migracji

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- Słabe punkty zabezpieczeń, które są adresowane, są szczegółowo opisane w następujących biuletynie bezpieczeństwa:
 - <https://www.ibm.com/support/pages/node/7126571>.
 - <https://www.ibm.com/support/pages/node/7137570>.

IBM MQ Operator 2.0.18 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla `.spec.version`

9.3.0.15-r1

Dozwolone wartości parametru `.spec.version` podczas migracji

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.17 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, [9.3.0.11-r2](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).
- Produkt IBM MQ Operator nie jest już testowany ani obsługiwany w systemie OpenShift Container Platform 4.10.

IBM MQ Operator 2.0.16 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, [9.3.0.11-r1](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).
- Produkt IBM MQ Operator nie jest już testowany ani obsługiwany w systemie OpenShift Container Platform 4.10.

IBM MQ Operator 2.0.15 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, [9.3.0.10-r2](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.14 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, [9.3.0.10-r1](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60

- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.13 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, [9.3.0.6-r1](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.12 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, [9.3.0.5-r3](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.11 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, [9.3.0.5-r2](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.10 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, [9.3.0.5-r1](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.9 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co nowego

- Od marca 2023 r. obrazy kontenerów menedżera kolejek IBM MQ Operator i IBM MQ są podpisane cyfrowo. Obrazy produktów IBM MQ Operator 2.0.9 i IBM MQ 9.3.0.4-r2 zostały podpisane w tej wersji. Patrz [“Weryfikowanie podpisów obrazów” na stronie 81](#))

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0” na stronie 60](#)
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.8 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0” na stronie 60](#)
- Słabe punkty zabezpieczeń, które są adresowane, zostały szczegółowo opisane w następujących biuletynie bezpieczeństwa:
 - [Biuletyn CVE-2022-47629 i CVE-2022-35737](#)
 - [Biuletyn CVE-2023-26284](#)

IBM MQ Operator 2.0.7 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, [9.3.0.3-r1](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.6 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, [9.3.0.1-r4](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.5 (LTS)

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, [9.3.0.1-r3](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.4

CP4I-LTS

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, [9.3.0.1-r2](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.3

CP4I-LTS CD

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, [9.2.0.6-r3-eus](#), 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, [9.3.0.1-r1](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.2



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, [9.2.0.6-r2-eus](#), 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, [9.3.0.0-r3](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.1



Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, [9.2.0.6-r1-eus](#), 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, [9.3.0.0-r2](#)

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co się zmieniło

- Aktualizacja tylko zabezpieczeń oparta na produkcie [“IBM MQ Operator 2.0.0”](#) na stronie 60
- Informacje o usuwanych słabych punktach zabezpieczeń opisano szczegółowo w tym [Biuletynie bezpieczeństwa](#).

IBM MQ Operator 2.0.0

CP4I-LTS > CD

Wersja IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Kanał operatora

v2.0

Dozwolone wartości dla .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1

Wersja Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 i nowsze. **Uwaga:** Obsługiwane są tylko wydania OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o parzystych numerach, na przykład 4.10 i 4.12.

Wersja IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

Co nowego

- Dodaje IBM MQ 9.3.0.
- Dodaje obsługę procesorów POWER (ppc64le).

Co się zmieniło

- Wymagana jest teraz wersja Red Hat OpenShift Container Platform 4.10 . Patrz [“Wersja zgodna z operatorem IBM MQ Operator”](#) na stronie 12.
- **Deprecated** Nieaktualne wersje: IBM MQ 9.2.3. Te wersje mogą nie być uzgadniane z kolejnymi wersjami produktu IBM MQ Operator.
- **Removed** Usunięte (poprzednio nieaktualne) wersje ciągłego dostarczania: IBM MQ 9.1.5, 9.2.0 CD, 9.2.1, 9.2.2
- Hak WWW sprawdzania poprawności IBM MQ Operator jest teraz instalowany przez program Operator Lifecycle Manager (OLM). OLM zarządza teraz certyfikatem haka WWW.
- Naprawiono błąd, który poprzednio generował ostrzeżenia dotyczące preferencji użytkownika w dzienniku IBM MQ Console .
- Słabe punkty zabezpieczeń, które są adresowane, są szczegółowo opisane w następujących biuletynie bezpieczeństwa:
 - <https://www.ibm.com/support/pages/node/6602255>
 - <https://www.ibm.com/support/pages/node/6602259>

OpenShift > CP4I-LTS > CP4I > CD

Historia wersji dla obrazów

kontenera menedżera kolejek do użycia z produktem IBM MQ Operator

Uwaga: Informacje na temat wcześniejszych obrazów kontenera menedżera kolejek zawiera sekcja [Historia wydań produktu IBM MQ Operator](#) w dokumentacji produktu IBM MQ 9.2 .

9.3.5.1-r2

CD

Wymagana wersja operatora

3.1.3 lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.5.1-r2](https://icr.io/cp/ibm-mqadvanced-server-integration:9.3.5.1-r2)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.5.1-r2](https://icr.io/cp/ibm-mqadvanced-server:9.3.5.1-r2)
- icr.io/ibm-messaging/mq:9.3.5.1-r2

Co nowego

- [Co nowego w produkcji IBM MQ 9.3.5](#)

Co się zmieniło

- [Co zostało zmienione w produkcji IBM MQ 9.3.5](#)
- Na podstawie [Red Hat Universal Base Image 8.9-1161.1715068733](#)
- Biblioteka golang.org/x/net została zaktualizowana w celu naprawienia zgłoszonego słabego punktu zabezpieczeń

9.3.5.1-r1



Wymagana wersja operatora

[3.1.2](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.5.1-r1](https://icr.io/cp/ibm-mqadvanced-server-integration:9.3.5.1-r1)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.5.1-r1](https://icr.io/cp/ibm-mqadvanced-server:9.3.5.1-r1)
- icr.io/ibm-messaging/mq:9.3.5.1-r1

Co nowego

- [Co nowego w produkcji IBM MQ 9.3.5](#)

Co się zmieniło

- [Co zostało zmienione w produkcji IBM MQ 9.3.5](#)
- Na podstawie [Red Hat Universal Base Image 8.9-1161](#)
- "dependabot" zgłoszone słabe punkty zabezpieczeń zostały usunięte

9.3.5.0-r2



Wymagana wersja operatora

[3.1.1](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.5.0-r2](https://icr.io/cp/ibm-mqadvanced-server-integration:9.3.5.0-r2)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.5.0-r2](https://icr.io/cp/ibm-mqadvanced-server:9.3.5.0-r2)
- icr.io/ibm-messaging/mq:9.3.5.0-r2

Co nowego

- [Co nowego w produkcji IBM MQ 9.3.5](#)

Co się zmieniło

- [Co zostało zmienione w produkcji IBM MQ 9.3.5](#)

- Na podstawie [Red Hat Universal Base Image 8.9-1137](#)
- Nowy obraz 9.3.5.0-r2 należy odebrać tylko wtedy, gdy włączony jest panel kontrolny operacji.

9.3.5.0-r1

CD

Wymagana wersja operatora

[3.1.0](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.5.0-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.5.0-r1](#)
- [icr.io/ibm-messaging/mq:9.3.5.0-r1](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.5](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.5](#)
- Na podstawie [Red Hat Universal Base Image 8.9-1137](#)
- Odsyłacz symboliczny jest udostępniany w systemie `/var/mam`, do którego kopiowane są niezasyfrowane informacje autoryzacyjne w systemie `mqwebuser.xml`.
- Biblioteka [golang.org/x/crypto](#) została zaktualizowana w celu naprawienia słabego punktu zabezpieczeń CVE-2023-48795.
- Do utworzenia certyfikatu samopodpisanego w magazynie kluczy WWW używany jest bezpieczniejszy algorytm SHA512 zamiast algorytmu SHA256.
- Magazyn kluczy PKCS#12 do użytku z serwerem WWW IBM MQ jest teraz generowany za pomocą funkcji **Pkcs12.Modern.Encode**, która używa szyfrowania SHA-2 (wcześniej generowanego przy użyciu wcześniejszego szyfrowania SHA-1).
- Słaby punkt zabezpieczeń zgłoszony w przypadku użycia metody **PathTraversal** został naprawiony.

9.3.4.1-r1

CD

Wymagana wersja operatora

[3.0.1](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.4.1-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.4.1-r1](#)
- [icr.io/ibm-messaging/mq:9.3.4.1-r1](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.4](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.4](#)
- Na podstawie [Red Hat Universal Base Image 8.9-1108](#)

9.3.4.0-r1

CD

Wymagana wersja operatora

[3.0.0](#) lub nowsza

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.4.0-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.4.0-r1](#)
- [icr.io/ibm-messaging/mq:9.3.4.0-r1](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.4](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.4](#)
- Na podstawie [Red Hat Universal Base Image 8.9-1029](#)
- Ulepszona obsługa serwera WWW IBM MQ -dziennik serwera WWW IBM MQ jest teraz domyślnie wyświetlany w dzienniku kontenera. Plik `messages.log` serwera WWW jest teraz automatycznie kopiowany do danych wyjściowych dziennika kontenera. W ramach tej zmiany plik `messages.log` zapisany na dysku jest teraz zawsze w formacie JSON, mimo że dziennik kontenera jest nadal dostępny w formacie JSON lub czytelnym dla człowieka formacie "basic".
- Naprawiono obsługę sygnału wewnątrz obrazu kontenera menedżera kolejek, tak aby poprawnie przetwarzała sygnały sterujące, jeśli kontener został zakończony przez program Red Hat OpenShift Container Platform przed zakończeniem uruchamiania.

9.3.3.3-r2

Wymagana wersja operatora

[2.4.8](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.3-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.3-r2](#)
- [icr.io/ibm-messaging/mq:9.3.3.3-r2](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.3](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.3](#)
- Na podstawie [Red Hat Universal Base Image 8.9-1137](#)
- Biblioteka [golang.org/x/crypto](#) została zaktualizowana w celu naprawienia słabego punktu zabezpieczeń [CVE-2023-48795](#).
- Do utworzenia certyfikatu samopodpisanego w magazynie kluczy WWW używany jest bezpieczniejszy algorytm SHA512 zamiast algorytmu SHA256.
- Magazyn kluczy PKCS#12 do użytku z serwerem WWW IBM MQ jest teraz generowany za pomocą funkcji **Pkcs12.Modern.Encode**, która używa szyfrowania SHA-2 (wcześniej generowanego przy użyciu wcześniejszego szyfrowania SHA-1).

- Słaby punkt zabezpieczeń zgłoszony w przypadku użycia metody **PathTraversal** został naprawiony.

9.3.3.3-r1

Wymagana wersja operatora

[2.4.7](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.3-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.3-r1](#)
- [icr.io/ibm-messaging/mq:9.3.3.3-r1](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.3](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.3](#)
- Na podstawie [Red Hat Universal Base Image 8.9-1108](#)

Uwzględnione raporty APAR IBM MQ

- IT44961
- IT44821
- IT44954

9.3.3.2-r3



Wymagana wersja operatora

[2.4.6](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.2-r3](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.2-r3](#)
- [icr.io/ibm-messaging/mq:9.3.3.2-r3](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.3](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.3](#)
- Na podstawie [Red Hat Universal Base Image 8.9-1029](#)

9.3.3.2-r2



Wymagana wersja operatora

[2.4.5](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.2-r2](https://icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.2-r2)
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.2-r2
- icr.io/ibm-messaging/mq:9.3.3.2-r2

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.3](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.3](#)
- Na podstawie [Red Hat Universal Base Image 8.8-1072.1697626218](#)
- IBM MQ Obraz kontenera menedżera kolejek 9.3.3.2-r2 zawiera wersję 3.1.7 (2023.4.0) [Instana MQ Exit](#).

9.3.3.2-r1



Wymagana wersja operatora

[2.4.4](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.2-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.2-r1
- icr.io/ibm-messaging/mq:9.3.3.2-r1

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.3](#)

Co się zmieniło

⁵

- [Co zostało zmienione w produkcie IBM MQ 9.3.3](#)
- Na podstawie [Red Hat Universal Base Image 8.8-1072.1697626218](#)
- Aktualizuje poziom biblioteki libcurl do wersji 8.4.0.

Uwzględnione raporty APAR IBM MQ

- IT41871
- IT44585
- IT44623
- IT44762

9.3.3.1-r2



Wymagana wersja operatora

[2.4.3](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

⁵ W poprzedniej wersji tego tematu niepoprawnie stwierdzono, że IBM MQ obraz kontenera menedżera kolejek 9.3.3.2-r1 zawiera wersję 3.1.7 (2023.4.0) [wyjścia Instana MQ](#).

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.1-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.1-r2
- icr.io/ibm-messaging/mq:9.3.3.1-r2

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.3](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.3](#)
- Aktualizacja tylko zabezpieczeń oparta na produkcie [IBM MQ 9.3.3.1-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.8-1037](#)

9.3.3.1-r1



Wymagana wersja operatora

[2.4.2](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.1-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.1-r1
- icr.io/ibm-messaging/mq:9.3.3.1-r1

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.3](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.3](#)
- Na podstawie [Red Hat Universal Base Image 8.8-1037](#).

9.3.3.0-r2



Wymagana wersja operatora

[2.4.1](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.0-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.0-r2
- icr.io/ibm-messaging/mq:9.3.3.0-r2

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.3](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.3](#)
- Na podstawie [Red Hat Universal Base Image 8.8-1014](#).

9.3.3.0-r1

CD

Wymagana wersja operatora

[2.4.0](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.0-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.0-r1](#)
- [icr.io/ibm-messaging/mq:9.3.3.0-r1](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.3](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.3](#)
- Na podstawie [Red Hat Universal Base Image 8.8-860](#).
- IBM MQ Obraz kontenera menedżera kolejek 9.3.3.0-r1 zawiera [wersja 3.1.2 \(2023.2.0\) programu zewnętrznego Instana MQ](#).

9.3.2.1-r2

CD

Wymagana wersja operatora

[2.3.3](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.2.1-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.2.1-r2](#)
- [icr.io/ibm-messaging/mq:9.3.2.1-r2](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.2](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.2](#)
- Na podstawie [Red Hat Universal Base Image 8.7-1107](#).

9.3.2.1-r1

CD

Wymagana wersja operatora

[2.3.2](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.2.1-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.2.1-r1](#)

- icr.io/ibm-messaging/mq:9.3.2.1-r1

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.2](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.2](#)
- Na podstawie [Red Hat Universal Base Image 8.7-1107](#).

9.3.2.0-r2



Wymagana wersja operatora

[2.3.1](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.2.0-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.2.0-r2
- icr.io/ibm-messaging/mq:9.3.2.0-r2

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.2](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.2](#)
- Na podstawie [Red Hat Universal Base Image 8.7-1085](#).

9.3.2.0-r1



Wymagana wersja operatora

[2.3.0](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.2.0-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.2.0-r1
- icr.io/ibm-messaging/mq:9.3.2.0-r1

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.2](#)
- Zmienna środowiskowa `MQ_LOGGING_CONSOLE_FORMAT` jest teraz ustawiona, co zastępuje nieaktualną zmienną `LOG_FORMAT`.

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.2](#)
- Certyfikaty menedżera kolejek z taką samą nazwą wyróżniającą (DN) jak certyfikat wystawcy (CA) nie są obsługiwane. Certyfikat musi mieć unikalną nazwę wyróżniającą podmiotu.
- Na podstawie [Red Hat Universal Base Image 8.7-1049.1675784874](#).

9.3.1.1-r1

CD

Wymagana wersja operatora

[2.2.2](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.1.1-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.1.1-r1](#)
- [icr.io/ibm-messaging/mq:9.3.1.1-r1](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.1](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.1](#)
- Na podstawie [Red Hat Universal Base Image 8.7-1031](#).
- IBM MQ Obraz kontenera menedżera kolejek 9.3.1.1-r1 zawiera [wersję 2.4.3 \(2022.4.3\) programu zewnętrznego produktu IBM Instana MQ](#).

9.3.1.0-r3

CD

Wymagana wersja operatora

[2.2.1](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.1.0-r3](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.1.0-r3](#)
- [icr.io/ibm-messaging/mq:9.3.1.0-r3](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.1](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.1](#)
- Na podstawie [Red Hat Universal Base Image 8.7-923.1669829893](#).
- IBM MQ Obraz kontenera menedżera kolejek 9.3.1.0-r3 zawiera [wersję 2.4.3 \(2022.4.3\) wyjęcia produktu IBM Instana MQ](#).

9.3.1.0-r2

CD

Wymagana wersja operatora

[2.2.0](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.1.0-r2](#)

- cp.icr.io/cp/ibm-mqadvanced-server:9.3.1.0-r2
- icr.io/ibm-messaging/mq:9.3.1.0-r2

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.1](#)
- Z obrazu kontenera menedżera kolejek produktu 9.3.1.0-r2 (CD) IBM MQ śledzenie IBM Instana jest obsługiwane w trybie rodzimym. Produkt IBM MQ w wersji 9.3.1.0-r2 zawiera produkt w wersji 2.4.0 (2022.4.0) programu IBM Instana MQ Exit. Aby włączyć śledzenie IBM Instana, patrz sekcja [“Integrowanie produktu IBM MQ z funkcją śledzenia produktu IBM Instana” na stronie 162.](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.1](#)
- Na podstawie Red Hat Universal Base Image 8.7-923.
- Jeśli klucz i certyfikat nie zostaną podane, atrybut menedżera kolejek **SSLKEYR** zostanie ustawiony na wartość pustą, a nie na wartość `"/run/runmqserver/tls/key"`.

9.3.1.0-r1



Wymagana wersja operatora

[2.1.0](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.1.0-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.1.0-r1
- icr.io/ibm-messaging/mq:9.3.1.0-r1

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.1](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.1](#)
- Na podstawie Red Hat Universal Base Image 8.6-941.

9.3.0.17-r3



Wymagana wersja operatora

[2.0.22](#) lub nowsza wersja

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.17-r3
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.17-r3
- icr.io/ibm-messaging/mq:9.3.0.17-r3

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.0](#)

- Aktualizacja tylko zabezpieczeń oparta na produkcie IBM MQ 9.3.0.0-r1
- Na podstawie [Red Hat Universal Base Image 9.4-949.1716471857](#)

9.3.0.17-r2

CP4I-LTS

Wymagana wersja operatora

[2.0.22](#) lub nowsza wersja

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.17-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.17-r2](#)
- [icr.io/ibm-messaging/mq:9.3.0.17-r2](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.0](#)
- Aktualizacja zabezpieczeń utworzona w produkcie IBM MQ 9.3.0.0-r1
- Na podstawie [Red Hat Universal Base Image 8.9-1161.1715068733](#)
- Biblioteka [golang.org/x/net](#) została zaktualizowana w celu naprawienia zgłoszonego słabego punktu zabezpieczeń

9.3.0.17-r1

CP4I-LTS

Wymagana wersja operatora

[2.0.21](#) lub nowsza wersja

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.17-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.17-r1](#)
- [icr.io/ibm-messaging/mq:9.3.0.17-r1](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.0](#)
- Aktualizacja zabezpieczeń utworzona w produkcie IBM MQ 9.3.0.0-r1
- Na podstawie [Red Hat Universal Base Image 8.9-1161](#)
- "dependabot" poinformowano o słabych punktach zabezpieczeń.

9.3.0.16-r2

CP4I-LTS

Wymagana wersja operatora

[2.0.20](#) lub nowsza wersja

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.16-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.16-r2
- icr.io/ibm-messaging/mq:9.3.0.16-r2

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.0](#)
- Aktualizacja zabezpieczeń utworzona w produkcie IBM MQ 9.3.0.0-r1
- Na podstawie [Red Hat Universal Base Image 8.9-1137](#)
- Nowy obraz 9.3.0.16-r2 należy odebrać tylko wtedy, gdy włączony jest panel kontrolny operacji.

9.3.0.16-r1

CP4I-LTS

Wymagana wersja operatora

[2.0.19](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.16-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.16-r1
- icr.io/ibm-messaging/mq:9.3.0.16-r1

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.0](#)
- Aktualizacja zabezpieczeń utworzona w produkcie IBM MQ 9.3.0.0-r1
- Na podstawie [Red Hat Universal Base Image 8.9-1137](#)
- Biblioteka golang.org/x/crypto została zaktualizowana w celu naprawienia słabego punktu zabezpieczeń CVE-2023-48795 .
- Do utworzenia certyfikatu samopodpisanego w magazynie kluczy WWW używany jest bezpieczniejszy algorytm SHA512 zamiast algorytmu SHA256 .
- Magazyn kluczy PKCS#12 do użytku z serwerem WWW IBM MQ jest teraz generowany za pomocą funkcji **Pkcs12.Modern.Encode** , która używa szyfrowania SHA-2 (wcześniej generowanego przy użyciu wcześniejszego szyfrowania SHA-1).
- Słaby punkt zabezpieczeń zgłoszony w przypadku użycia metody **PathTraversal** został naprawiony.

9.3.0.15-r1

CP4I-LTS

Wymagana wersja operatora

[2.0.18](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.15-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.15-r1
- icr.io/ibm-messaging/mq:9.3.0.15-r1

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.0](#)
- Aktualizacja tylko zabezpieczeń oparta na produkcie [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.9-1108](#)

9.3.0.11-r2

CP4I-LTS

Wymagana wersja operatora

[2.0.17](#) lub nowsza wersja

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.11-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.11-r2
- icr.io/ibm-messaging/mq:9.3.0.11-r2

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.0](#)
- Aktualizacja tylko zabezpieczeń oparta na produkcie [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.9-1029](#).

9.3.0.11-r1

CP4I-LTS

Wymagana wersja operatora

[2.0.16](#) lub nowsza wersja

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.11-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.11-r1
- icr.io/ibm-messaging/mq:9.3.0.11-r1

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- Co zostało zmienione w produkcie [IBM MQ 9.3.0](#)
- Aktualizacja tylko zabezpieczeń oparta na produkcie [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.8-1072.1697626218](#).
- Aktualizuje poziom biblioteki libcurl do wersji 8.4.0

9.3.0.10-r2

CP4I-LTS

Wymagana wersja operatora

[2.0.15](#) lub nowsza wersja

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.10-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.10-r2](#)
- [icr.io/ibm-messaging/mq:9.3.0.10-r2](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- Co zostało zmienione w produkcie [IBM MQ 9.3.0](#)
- Aktualizacja tylko zabezpieczeń oparta na produkcie [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.8-1037](#).

9.3.0.10-r1

CP4I-LTS

Wymagana wersja operatora

[2.0.14](#) lub nowsza wersja

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.10-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.10-r1](#)
- [icr.io/ibm-messaging/mq:9.3.0.10-r1](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- Co zostało zmienione w produkcie [IBM MQ 9.3.0](#)
- Aktualizacja tylko zabezpieczeń oparta na produkcie [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.8-1037](#).

9.3.0.6-r1

CP4I-LTS

Wymagana wersja operatora

[2.0.13](#) lub nowsza wersja

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.6-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.6-r1](#)
- [icr.io/ibm-messaging/mq:9.3.0.6-r1](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.0](#)
- Aktualizacja tylko zabezpieczeń oparta na produkcie [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.8-1014](#).

9.3.0.5-r3

CP4I-LTS

Wymagana wersja operatora

[2.0.12](#) lub nowsza wersja

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.5-r3](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.5-r3](#)
- [icr.io/ibm-messaging/mq:9.3.0.5-r3](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.0](#)
- Aktualizacja tylko zabezpieczeń oparta na produkcie [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.8-860](#).

9.3.0.5-r2

CP4I-LTS

Wymagana wersja operatora

[2.0.11](#) lub nowsza wersja

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.5-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.5-r2](#)
- [icr.io/ibm-messaging/mq:9.3.0.5-r2](#)

Co nowego

- [Co nowego w produkcji IBM MQ 9.3.0](#)

Co się zmieniło

- Co zostało zmienione w produkcji IBM MQ 9.3.0
- Aktualizacja tylko zabezpieczeń oparta na produkcji [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.7-1107](#).

Ważne: Dla użytkowników panelu kontrolnego operacji w IBM MQ LTS obrazie kontenera menedżera kolejek 9.3.0.5-r2

Gdy panel kontrolny Operations Dashboard jest włączony, IBM MQ LTS obraz kontenera menedżera kolejek 9.3.0.5-r2 wdraża obrazy agenta panelu kontrolnego Operations Dashboard i kolektora, które nie zawierają najnowszych poprawek bezpieczeństwa dostępnych w momencie ogólnej dostępności.

Mitygacja: aktualizacja do co najmniej 9.3.0.5-r3 wszystkich obrazów IBM MQ LTS kontenera menedżera kolejek 9.3.0.5-r2 z włączonym panelem kontrolnym operacji. Patrz sekcja [“Aktualizowanie menedżera kolejek systemu IBM MQ przy użyciu programu Red Hat OpenShift”](#) na stronie 134.

9.3.0.5-r1

CP4I-LTS

Wymagana wersja operatora

[2.0.10](#) lub nowsza wersja

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.5-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.5-r1](#)
- [icr.io/ibm-messaging/mq:9.3.0.5-r1](#)

Co nowego

- [Co nowego w produkcji IBM MQ 9.3.0](#)

Co się zmieniło

- Co zostało zmienione w produkcji IBM MQ 9.3.0
- Aktualizacja tylko zabezpieczeń oparta na produkcji [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.7-1107](#).

Ważne: Dla użytkowników panelu kontrolnego operacji w IBM MQ LTS obrazie kontenera menedżera kolejek 9.3.0.5-r1

Gdy panel kontrolny Operations Dashboard jest włączony, IBM MQ LTS obraz kontenera menedżera kolejek 9.3.0.5-r1 wdraża obrazy agenta panelu kontrolnego Operations Dashboard i kolektora, które nie zawierają najnowszych poprawek bezpieczeństwa dostępnych w momencie ogólnej dostępności.

Mitygacja: zaktualizuj co najmniej 9.3.0.5-r3 wszystkie IBM MQ obrazy LTS kontenera menedżera kolejek 9.3.0.5-r1 z włączonym panelem kontrolnym operacji. Patrz sekcja [“Aktualizowanie menedżera kolejek systemu IBM MQ przy użyciu programu Red Hat OpenShift”](#) na stronie 134.

9.3.0.4-r2

CP4I-LTS

Wymagana wersja operatora

[2.0.9](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.4-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.4-r2
- icr.io/ibm-messaging/mq:9.3.0.4-r2

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.0](#)
- Aktualizacja tylko zabezpieczeń oparta na produkcie [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.7-1085](#).

9.3.0.4-r1

CP4I-LTS

Wymagana wersja operatora

[2.0.8](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.4-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.4-r1
- icr.io/ibm-messaging/mq:9.3.0.4-r1

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.0](#)
- Aktualizacja tylko zabezpieczeń oparta na produkcie [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.7-1049.1675784874](#).

9.3.0.3-r1

CP4I-LTS

Wymagana wersja operatora

[2.0.7](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.3-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.3-r1
- icr.io/ibm-messaging/mq:9.3.0.3-r1

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- Co zostało zmienione w produkcie [IBM MQ 9.3.0](#)
- Aktualizacja tylko zabezpieczeń oparta na produkcie [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.7-1031](#).

9.3.0.1-r4

CP4I-LTS

Wymagana wersja operatora

[2.0.6](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.1-r4](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.1-r4](#)
- [icr.io/ibm-messaging/mq:9.3.0.1-r4](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- Co zostało zmienione w produkcie [IBM MQ 9.3.0](#)
- Aktualizacja tylko zabezpieczeń oparta na produkcie [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.7-923.1669829893](#).

9.3.0.1-r3

CP4I-LTS

Wymagana wersja operatora

[2.0.5](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.1-r3](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.1-r3](#)
- [icr.io/ibm-messaging/mq:9.3.0.1-r3](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- Co zostało zmienione w produkcie [IBM MQ 9.3.0](#)
- Aktualizacja tylko zabezpieczeń oparta na produkcie [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.7-923](#).

9.3.0.1-r2

CP4I-LTS

Wymagana wersja operatora

[2.0.4](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.1-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.1-r2
- icr.io/ibm-messaging/mq:9.3.0.1-r2

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.0](#)
- Aktualizacja tylko zabezpieczeń oparta na produkcie [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.6-941](#).

9.3.0.1-r1



Wymagana wersja operatora

[2.0.3](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.1-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.1-r1
- icr.io/ibm-messaging/mq:9.3.0.1-r1

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- [Co zostało zmienione w produkcie IBM MQ 9.3.0](#)
- Aktualizacja tylko zabezpieczeń oparta na produkcie [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.6-941](#).

9.3.0.0-r3



Wymagana wersja operatora

[2.0.2](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.0-r3
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.0-r3
- icr.io/ibm-messaging/mq:9.3.0.0-r3

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- Co zostało zmienione w produkcie IBM MQ 9.3.0
- Aktualizacja tylko zabezpieczeń oparta na produkcie [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.6-902](#).

9.3.0.0-r2



Wymagana wersja operatora

[2.0.1](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.0-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.0-r2](#)
- [icr.io/ibm-messaging/mq:9.3.0.0-r2](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- Co zostało zmienione w produkcie IBM MQ 9.3.0
- Aktualizacja tylko zabezpieczeń oparta na produkcie [IBM MQ 9.3.0.0-r1](#)
- Na podstawie [Red Hat Universal Base Image 8.6-854](#).

9.3.0.0-r1



Wymagana wersja operatora

[2.0.0](#) lub nowszy

Obsługiwane architektury

amd64, s390x, ppc64le

Obrazy

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.0-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.0-r1](#)
- [icr.io/ibm-messaging/mq:9.3.0.0-r1](#)

Co nowego

- [Co nowego w produkcie IBM MQ 9.3.0](#)

Co się zmieniło

- Co zostało zmienione w produkcie IBM MQ 9.3.0
- Domyślna konfiguracja programisty w obrazie produktu MQ Advanced for Developers używa teraz pliku ANY_TLS12_OR_HIGHER.
- Naprawiono problem z serwerem WWW IBM MQ , który spowodował błąd w dzienniku z powodu braku preferencji Java .
- Na podstawie [Red Hat Universal Base Image 8.6-751.1655117800](#).

Od marca 2023 r. obrazy kontenerów menedżera kolejek IBM MQ Operator i IBM MQ są podpisane cyfrowo.

Pierwsze IBM MQ operatory do podpisania:

- 2.3.1 (dysk CD)
- 2.0.9 (LTS)

Pierwsze obrazy kontenera menedżera kolejek IBM MQ do podpisania:

- 9.3.2.0-r2 (dysk CD)
- 9.3.0.4-r2 (LTS)

O tym zadaniu

Podpisy cyfrowe stanowią dla konsumentów treści sposób, aby zapewnić, że to, co pobierają, jest zarówno autentyczne (pochodzi z oczekiwanego źródła) i ma integralność (to jest to, czego oczekujemy, że będzie).

Procedura

- Sprawdź sygnatury obrazów kontenerów menedżera kolejek IBM MQ Operator i IBM MQ :
 - **Operator3.0.0** **V 9.3.4** W przypadku produktu IBM MQ Operator w wersji 3.0.0 lub nowszej albo obrazu kontenera menedżera kolejek systemu IBM MQ w wersji 9.3.4.0-r1 lub nowszej należy zapoznać się z sekcją [Weryfikowanie podpisów obrazów](#) w dokumentacji produktu IBM Cloud Pak for Integration (CP4I) 2023.4 .
 - **Operator2.4.0** W przypadku produktu IBM MQ Operator w wersji 2.4.x lub obrazu kontenera menedżera kolejek systemu IBM MQ w wersji 9.3.3.x należy zapoznać się z sekcją [Weryfikowanie podpisów obrazów](#) w dokumentacji produktu CP4I 2023.2 .
 - W przypadku produktu IBM MQ Operator w wersji wcześniejszej niż 2.4.0 lub obrazu kontenera menedżera kolejek systemu IBM MQ w wersji wcześniejszej niż 9.3.3.0-r1 należy zapoznać się z sekcją [Weryfikowanie podpisów obrazów](#) w dokumentacji produktu CP4I 2022.4 .

Cloud Pak for Integration

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

O tym zadaniu

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

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

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

Ten zestaw tematów zawiera opis scenariusza 3 i kluczowych etapów migracji istniejącego menedżera kolejek IBM MQ do środowiska kontenerowego z operatorem IBM MQ Operator. Ze względu na elastyczność i szerokie zastosowanie produktu IBM MQ istnieje kilka kroków opcjonalnych. Każdy z nich

zawiera sekcję "Czy muszę to zrobić?". Właściwe zweryfikowanie potrzeb pozwoli zaoszczędzić czas podczas migracji.

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

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

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

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

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

Procedura

1. ["Sprawdzanie, czy wymagane funkcje są dostępne" na stronie 82](#)
2. ["Wyodrębnianie konfiguracji menedżera kolejek" na stronie 83](#)
3. Opcjonalnie: ["Opcjonalnie: wyodrębnianie i pozyskanie kluczy i certyfikatów menedżera kolejek" na stronie 84](#)
4. Opcjonalnie: ["Opcjonalnie: konfigurowanie protokołu LDAP" na stronie 86](#)
5. Opcjonalnie: ["Opcjonalnie: zmiana adresów IP i nazw hostów w konfiguracji produktu IBM MQ" na stronie 93](#)
6. ["Aktualizowanie konfiguracji menedżera kolejek dla środowiska kontenera" na stronie 95](#)
7. ["Wybieranie docelowej architektury wysokiej dostępności dla programu IBM MQ działającego w kontenerach" na stronie 97](#)
8. ["Tworzenie zasobów dla menedżera kolejek" na stronie 98](#)
9. ["Tworzenie nowego menedżera kolejek w środowisku Red Hat OpenShift" na stronie 99](#)
10. ["Sprawdzanie poprawności nowego wdrożenia w kontenerze" na stronie 103](#)

Sprawdzanie, czy wymagane funkcje są dostępne

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

Zanim rozpocziesz

Jest to pierwszy krok w procedurze ["Migracja produktu IBM MQ do rozwiązania IBM Cloud Pak for Integration" na stronie 81.](#)

Procedura

1. Sprawdź, czy docelowy obraz kontenera zawiera wszystkie wymagane funkcje.

Więcej informacji na ten temat zawiera sekcja ["Wybór sposobu używania IBM MQ w kontenerach" na stronie 5.](#)

2. IBM MQ Operator ma jeden port obsługujący ruch IBM MQ, nazywany portem nasłuchującym. Jeśli istnieje wiele portów nasłuchujących, należy uprościć konfigurację, używając jednego portu nasłuchującego w kontenerze. Ponieważ nie jest to typowy scenariusz, ta modyfikacja nie została tutaj szczegółowo udokumentowana.
3. Jeśli używane są procedury zewnętrzne IBM MQ, zmigruj je do kontenera, dodając odpowiednią warstwę w plikach binarnych IBM MQ. Jest to zaawansowany scenariusz migracji i w związku z tym nie został tutaj opisany. Ogólny schemat kroków zawiera sekcja [“Budowanie obrazu z niestandardowymi plikami MQSC i INI przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift” na stronie 169.](#)
4. Jeśli w systemie IBM MQ używany jest mechanizm wysokiej dostępności, przeanalizuj dostępne opcje. Patrz sekcja [“Wysoka dostępność produktu IBM MQ w kontenerach” na stronie 26.](#)

Co dalej

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

OpenShift > CP4I-LTS > CD Wyodrębnianie konfiguracji menedżera kolejek

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

Zanim rozpoczniesz

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

Procedura

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

Uruchom następującą komendę:

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

Uwagi dotyczące użycia tej komendy:

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

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

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

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

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

```
startmqm QMGR_NAME
```

Co dalej

Teraz można [wyodrębnić i uzyskać klucze i certyfikaty menedżera kolejek](#).

kluczy i certyfikatów menedżera kolejek

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

Zanim rozpocznie

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

O tym zadaniu

Czy muszę to zrobić?

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

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

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

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

Uwaga: **V9.3.2** Certyfikaty menedżera kolejek z taką samą nazwą wyróżniającą (DN) jak certyfikat wystawcy (CA) nie są obsługiwane. Certyfikat musi mieć unikalną nazwę wyróżniającą podmiotu. Produkt sprawdza teraz, czy nazwy wyróżniające nie są takie same.

Procedura

1. Wyodrębnij wszystkie zaufane certyfikaty z istniejącej składnicy.

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

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

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

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

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

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

- a. Zidentyfikuj położenie bazy TLS:

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

Przykładowy wynik:

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

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

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

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

Przykładowy wynik:

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

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

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

W przykładach, które były wcześniej przedstawione, oznacza to:

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

2. Uzyskiwanie nowego klucza i certyfikatu dla menedżera kolejek

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

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

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

Tworzone są dwa nowe pliki:

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

Co dalej

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

OpenShift CP4I-LTS CD Opcjonalnie: konfigurowanie protokołu LDAP

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

Zanim rozpoczniesz

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

O tym zadaniu

Czy muszę to zrobić?

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

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

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

Przykładowy wynik:

```
DEFINE AUTHINFO('USE.LDAP') +
  AUTHTYPE(IDPWLDAP) +
  ADOPTCTX(YES) +
  CONNNAME('ldap-service.ldap(389)') +
  CHCKCLNT(REQUIRED) +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
* LDAPPWD('*****') +
  SHORTUSR('uid') +
  GRPFIELD('cn') +
  USRFIELD('uid') +
  AUTHORMD(SEARCHGRP) +
* ALTDATA(2020-11-26) +
* ALTTIME(15.44.38) +
  REPLACE
```

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

AUTHTYPE

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

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

AUTHORMD

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

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

Procedura

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

MQ dla serwera LDAP

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

Zanim rozpoczniesz

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

O tym zadaniu

Czy muszę to zrobić?

Jeśli do uwierzytelniania i autoryzacji jest już używany protokół LDAP, zmiany nie są wymagane. Jeśli nie masz pewności, czy używany jest protokół LDAP, patrz sekcja [“Opcjonalnie: konfigurowanie protokołu LDAP”](#) na stronie 86.

Konfigurowanie serwera LDAP przebiega w dwóch etapach:

1. [Zdefiniowanie konfiguracji LDAP](#).
2. [Powiązanie konfiguracji LDAP z definicją menedżera kolejek](#).

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

- [Przegląd repozytorium użytkowników](#)
- [Skorowidz komendy AUTHINFO](#)

Procedura

1. Zdefiniuj konfigurację LDAP.

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

```
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
  SECCOMM(NO) +
  USRFIELD('uid') +
  SHORTUSR('uid') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  AUTHORMD(SEARCHGRP) +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  GRPFIELD('cn') +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember')
REPLACE
```

where

- **CONNAME** to nazwa hosta i port odpowiadające serwerowi LDAP. Jeśli dla zapewnienia odporności stosowanych jest wiele adresów, można je skonfigurować w formie listy rozdzielanej przecinkami.
- **LDAPUSER** jest nazwą wyróżniającą odpowiadającą użytkownikowi, który jest używany przez produkt IBM MQ podczas nawiązywania połączenia z serwerem LDAP w celu wykonywania zapytań o rekordy użytkowników.
- **LDAPPWD** jest hasłem użytkownika **LDAPUSER**.
- **SECCOM** określa, czy w komunikacji z serwerem LDAP powinna być używany protokół TLS. Możliwe wartości:
 - YES: używany jest protokół TLS, a serwer IBM MQ legitymuje się certyfikatem.

- ANON: protokół TLS jest używany bez legitymowania się przez serwer IBM MQ certyfikatem.
- NO: protokół TLS nie jest używany podczas nawiązywania połączenia.
- **USRFIELD** określa pole w rekordzie LDAP, do którego będzie dopasowywana nazwa użytkownika.
- **SHORTUSR** to pole w obrębie rekordu LDAP, którego długość nie może przekraczać 12 znaków. Wartość w tym polu jest zakładaną tożsamością, jeśli uwierzytelnianie powiodło się.
- **BASEDNU** to podstawowa nazwa wyróżniająca, która powinna być używana do przeszukiwania katalogu LDAP.
- **BASEDNG** jest podstawową nazwą wyróżniającą dla grup w katalogu LDAP.
- **AUTHORMD** definiuje mechanizm używany do rozstrzygnięcia przypisania użytkowników do grup. Dostępne są cztery opcje:
 - OS: kieruje do systemu operacyjnego zapytania o grupy powiązane z krótką nazwą.
 - SEARCHGRP: służy do wyszukiwania w katalogu LDAP grup uwierzytelnionego użytkownika.
 - SEARCHUSR: umożliwia wyszukiwanie informacji o uwierzytelnionym użytkowniku w celu uzyskania informacji o przypisaniach do grup.
 - SRCHGRPSN: służy do wyszukiwania pozycji grupy w katalogu LDAP na podstawie krótkiej nazwy użytkownika uwierzytelnionego (zdefiniowanej przez pole SHORTUSR).
- **GRPFIELD** jest atrybutem w rekordzie grupy LDAP, który odpowiada prostej nazwie. Jeśli ten parametr jest określony, można go używać do definiowania rekordów autoryzacji.
- **CLASSUSR** to klasa obiektów LDAP, która odpowiada użytkownikowi.
- **CLASSGRP** to klasa obiektu LDAP, która odpowiada grupie.
- **FINDGRP** jest atrybutem w rekordzie LDAP, który odpowiada przynależności do grupy.

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


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

2. Powiąż konfigurację LDAP z definicją menedżera kolejek.

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

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

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

```

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

```

Co dalej

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

OpenShift > CP4I-LTS > CD **LDAP, część 2: aktualizowanie kopii zapasowej IBM**

MQ z uwzględnieniem informacji o autoryzacji LDAP

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

Zanim rozpoczniesz

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

O tym zadaniu

Czy muszę to zrobić?

Jeśli do uwierzytelniania i autoryzacji jest już używany protokół LDAP, zmiany nie są wymagane. Jeśli nie masz pewności, czy używany jest protokół LDAP, patrz sekcja “Opcjonalnie: konfigurowanie protokołu LDAP” na stronie 86.

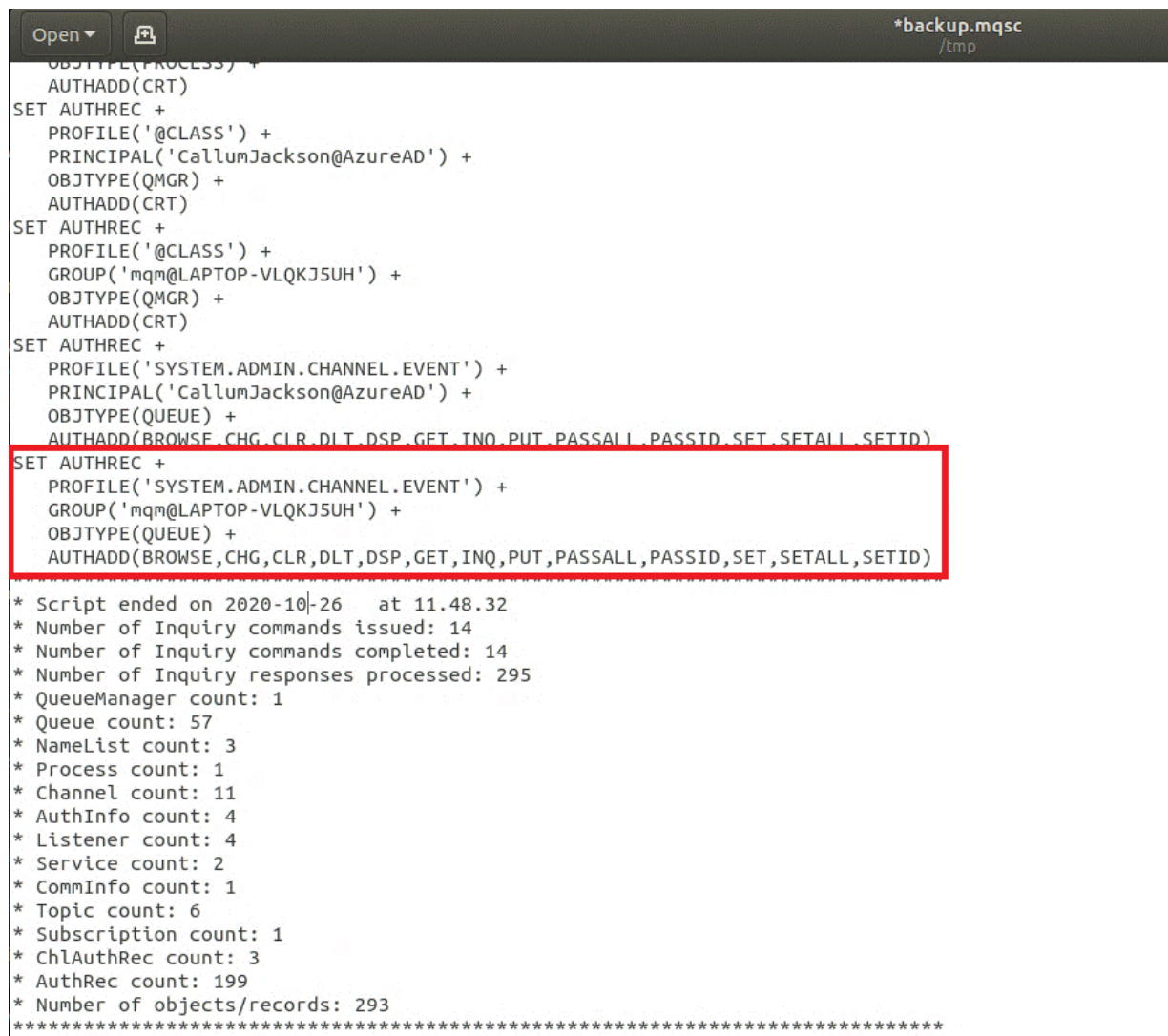
Aby zaktualizować informacje o autoryzacji LDAP, należy wykonać dwa etapy:

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

Procedura

1. Usunięcie wszystkich istniejących uprawnień z pliku

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



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

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

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

2. Zdefiniowanie nowych informacji o autoryzacji dla protokołu LDAP

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

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

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

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

```
SET AUTHREC PROFILE('*') OBJTYPE(Queue) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Topic) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Channel) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(ClntConn) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(AuthInfo) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Listener) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(NameList) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Process) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Service) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(QMGR) GROUP('admins') AUTHADD(ALL)
```

Co dalej

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

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

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

Zanim rozpoczniesz

W tym zadaniu przyjęto założenie, że użytkownik ma [skonfigurowany protokół LDAP](#).

O tym zadaniu

Czy muszę to zrobić?

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

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

Przykładowy wynik:

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

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

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

Procedura

1. Należy właściwie zinterpretować zwrócone wyniki.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Co dalej

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

Aktualizowanie konfiguracji menedżera kolejek dla środowiska kontenera

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

Zanim rozpocznie

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

O tym zadaniu

W kontenerze zdefiniowane są następujące aspekty konfiguracji:

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

Dlatego konieczne jest zaktualizowanie wyeksportowanej konfiguracji:

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

Procedura

1. Usuń wszystkie definicje nasłuchiwanie.

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

```

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

```

2. Zdefiniuj położenie repozytorium kluczy TLS.

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

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

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


```

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

```

Co dalej

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

OpenShift > CP4I-LTS > CD Wybieranie docelowej architektury wysokiej dostępności dla programu IBM MQ działającego w kontenerach

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

Zanim rozpoczniesz

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

O tym zadaniu

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

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

W tym zadaniu wybierana jest tylko docelowa architektura wysokiej dostępności. Czynności związane z konfigurowaniem wybranej architektury są opisane w kolejnym zadaniu w tym scenariuszu ("Tworzenie nowego menedżera kolejek w środowisku Red Hat OpenShift" na stronie 99).

Procedura

1. Porównaj obie dostępne opcje.

Ich obszerny opis znajduje się w sekcji "Wysoka dostępność produktu IBM MQ w kontenerach" na stronie 26.

2. Wybierz docelową architekturę wysokiej dostępności.

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

Co dalej

Teraz można przystąpić do tworzenia zasobów menedżera kolejek.



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

Zanim rozpoczniesz

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

O tym zadaniu

W poprzednich sekcjach wyodrębniono, zaktualizowano i zdefiniowano dwa zasoby:

- IBM MQ Konfiguracja
- Certyfikaty TLS i klucze

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

Procedura

1. Zaimportuj konfigurację IBM MQ do Red Hat OpenShift.

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

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

Uruchom następującą komendę:

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

- c) Sprawdź, czy plik został załadowany pomyślnie.

Uruchom następującą komendę:

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

2. Importowanie zasobów TLS produktu IBM MQ

Tak, jak opisano to w sekcji [“Opcjonalnie: wyodrębnianie i pozyskanie kluczy i certyfikatów menedżera kolejek”](#) na stronie 84, dla wdrożenia menedżera kolejek może być wymagany protokół TLS. W takim przypadku należy mieć już kilka plików kończących się na `.crt` i `.key`. Należy dodać je do kluczy tajnych programu Kubernetes, aby menedżer kolejek mógł się do nich odwoływać podczas wdrażania.

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

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

Aby zaimportować te pliki, uruchom następującą komendę:

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

Program Kubernetes udostępnia ten pomocny program narzędziowy podczas importowania zgodnego klucza publicznego i prywatnego. Jeśli istnieją dodatkowe certyfikaty, które można dodać, na przykład do magazynu zaufanych certyfikatów menedżera kolejek, uruchom następującą komendę:

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

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

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

Co dalej

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

Tworzenie nowego menedżera kolejek w środowisku Red Hat OpenShift

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

Zanim rozpocznieš

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

O tym zadaniu

Tak, jak wyjaśniono w sekcji [“Wybieranie docelowej architektury wysokiej dostępności dla programu IBM MQ działającego w kontenerach”](#) na stronie 97, istnieją dwie możliwe topologie wdrażania. W związku z tym w niniejszym temacie przedstawiono dwa różne szablony postępowania:

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

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

Procedura

- Wdrażanie menedżera kolejek z jedną instancją.

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

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1
spec:
  version: 9.3.5.1-r2
  license:
    accept: true
    license: L-VTPK-22YZPK
    use: "Production"
  pki:
    keys:
      - name: default
        secret:
          secretName: my-tls-migration
          items:
            - tls.key
            - tls.crt
  web:
    enabled: true
  queueManager:
    name: QM1
  mqsc:
    - configMap:
        name: my-mqsc-migrated
        items:
          - backup.mqsc
```

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

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

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

```
spec:
  version: 9.3.5.1-r2
  license:
    accept: true
    license: L-VTPK-22YZPK
    use: "Production"
```

Te dane odpowiadają informacjom o wersji i licencji właściwym dla wdrożenia. Jeśli chcesz je zmienić, skorzystaj z informacji zamieszczonych w sekcji [“Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1”](https://mq.ibm.com/v1beta1) na stronie 187.

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

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

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

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

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

```
web:
  enabled: true
```

Umożliwia włączenie konsoli WWW dla wdrożenia.

```
queueManager:
  name: QM1
```

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

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

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

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

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

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

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

```
oc get storageclass
```

Poniżej znajdują się przykładowe dane zwracane przez tę komendę:

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

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

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

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

- Wdrażanie menedżera kolejek z wieloma instancjami.

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

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1mi
spec:
  version: 9.3.5.1-r2
  license:
    accept: true
    license: L-VTPK-22YZPK
    use: "Production"
  pki:
    keys:
      - name: default
        secret:
          secretName: my-tls-migration
        items:
          - tls.key
          - tls.crt
  web:
    enabled: true
  queueManager:
    name: QM1
    availability: MultiInstance
  storage:
    defaultClass: aws-efs
    persistedData:
      enabled: true
    queueManager:
      enabled: true
    recoveryLogs:
      enabled: true
  mqsc:
    - configMap:
        name: my-mqsc-migrated
      items:
        - backup.mqsc
```

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

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

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

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

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

```
oc get storageclass
```

Poniżej znajdują się przykładowe dane zwracane przez tę komendę:

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

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

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

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

Sprawdzanie poprawności nowego wdrożenia w kontenerze

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

Zanim rozpocznie

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

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

O tym zadaniu

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

Wymagane są następujące informacje:

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

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

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

Procedura

1. Wykonaj komendę Exec względem działającego kontenera IBM MQ.

Użyj następującej komendy:

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

gdzie `qm1-ibm-mq-0` to produkt, który wdrożyliśmy w systemie [“Tworzenie nowego menedżera kolejek w środowisku Red Hat OpenShift” na stronie 99](#). Jeśli wdrożenie ma inną nazwę, należy zmienić tę wartość.

2. Wyślij komunikat.

Wykonaj następujące komendy:

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

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

3. Sprawdź, czy komunikat został odebrany pomyślnie.

Uruchom przykład GET:

```
./amqsgetc Q1 QM1
```

Wyniki

Procedura [“Migracja produktu IBM MQ do rozwiązania IBM Cloud Pak for Integration” na stronie 81](#) została zakończona.

Co dalej

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

Migracja komunikatów z kolejek

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

Nawiązywanie połączenia z produktem IBM MQ spoza środowiska Red Hat OpenShift

Wdrożony menedżer kolejek może być narażony na działanie klientów IBM MQ i menedżerów kolejek spoza środowiska Red Hat OpenShift. Proces jest zależny od wersji produktu IBM MQ łączącego się ze środowiskiem Red Hat OpenShift. Patrz sekcja ["Konfigurowanie trasy w celu nawiązania połączenia z menedżerem kolejek spoza klastra produktu Red Hat OpenShift"](#) na stronie 159.

OpenShift

CP4I

Instalowanie produktu IBM MQ Operator

Produkt IBM MQ Operator można zainstalować w systemie Red Hat OpenShift za pomocą konsoli OpenShift lub interfejsu wiersza komend (CLI).

Zanim rozpoczniesz

Aby upewnić się, że instalacja przebiega tak sprawnie, jak to możliwe, przed rozpoczęciem instalacji należy zapoznać się ze wszystkimi wymaganiami wstępnymi i wymaganiami. Patrz sekcja ["Planowanie wdrożenia IBM MQ w kontenerach"](#) na stronie 5.

Ważne: **V9.3.4** Przed zainstalowaniem produktu IBM MQ Operator należy zapoznać się ze wskazówkami dotyczącymi [tworzenia struktury wdrożenia](#).

O tym zadaniu

Poniższe kroki przedstawiają typowy przepływ zadań podczas instalowania produktu IBM MQ Operator:

1. [Zainstaluj produkt Red Hat OpenShift Container Platform](#).
2. [Skonfiguruj pamięć masową](#).
3. [Obrazy lustrzane \(tylko przerwa powietrzna\)](#).
4. [Dodaj katalog operatora IBM i przygotuj klastr](#).
5. [Zainstaluj IBM MQ Operator](#).
6. [Utwórz klucz tajny uprawnienia \(tylko w przypadku instalacji w trybie z połączeniem\)](#).
7. [Opcjonalnie: zainstaluj produkt IBM Cloud Pak for Integration \(CP4I\) i jego zależności](#).
8. [Przeprowadź wdrożenie usługi License Service](#).
9. [Wdrażanie menedżera kolejek](#).

Procedura

1. Zainstaluj produkt Red Hat OpenShift Container Platform.

Szczegółowe informacje na temat instalowania systemu OpenShift zawiera sekcja [Instalowanie oprogramowania Red Hat 4.6 lub nowszego](#).

Ważne: Upewnij się, że zainstalowana jest obsługiwana wersja produktu OpenShift Container Platform. Na przykład, aby używać produktu IBM MQ Operator 2.0 lub nowszej, należy zainstalować produkt OpenShift Container Platform 4.12 lub nowszej. Należy również zauważyć, że obsługiwane są tylko wersje produktu OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16. Więcej informacji na ten temat zawiera dokumentacja [IBM Cloud Pak and Red Hat OpenShift Container Platform compatibility](#).

Dla wszystkich kroków wykonywanych za pomocą interfejsu CLI w systemie Red Hat OpenShift Container Platform należy zalogować się do klastra OpenShift za pomocą komendy `oc login`. Aby zainstalować interfejs CLI, patrz sekcja [Pierwsze kroki z interfejsem CLI programu OpenShift](#).

Po zainstalowaniu produktu OpenShift można sprawdzić i uzyskać dostęp do oprogramowania kontenera, korzystając z klucza upoważnienia IBM utworzonego w sekcji [Utwórz klucz tajny upoważnienia](#).

2. Skonfiguruj pamięć masową.

Należy zdefiniować klasy pamięci masowej w programie Red Hat OpenShift Container Platform i ustawić konfigurację pamięci masowej, aby spełnić wymagania dotyczące wielkości.

Ważne: Menedżery kolejek z jedną instancją IBM MQ i rodzime menedżery kolejek o wysokiej dostępności mogą używać trybu dostępu RWO, podczas gdy menedżery kolejek z wieloma instancjami wymagają RWX zgodnie z opisem w sekcji [“Uwagi dotyczące pamięci masowej dla IBM MQ Operator”](#) na stronie 21. Menedżery kolejek z wieloma instancjami IBM MQ wymagają konkretnych parametrów systemu plików, które można zweryfikować przy użyciu instrukcji zawartych w sekcji [Testowanie współużytkowanego systemu plików dla produktu IBM MQ](#).

Listę znanych zgodnych i niezgodnych systemów plików oraz uwagi dotyczące innych ograniczeń można znaleźć w sekcji [Instrukcja testowania dla systemów plików IBM MQ](#).

Zalecanych dostawców pamięci masowej można znaleźć na stronie CP4I [Uwagi dotyczące pamięci masowej](#).

3. **V 9.3.4**

Obrazy lustrzane (tylko szczelina powietrzna).

Jeśli klaster znajduje się w ograniczonym środowisku sieciowym (z ograniczeniem powietrza), należy utworzyć kopię lustrzaną obrazów IBM MQ. W zależności od konfiguracji może być konieczne utworzenie kopii lustrzanej niektórych dodatkowych komponentów. Przeczytaj poniższe informacje, a następnie utwórz kopię lustrzaną obrazów zgodnie z wymaganiami.

- Należy utworzyć kopię lustrzaną obrazów IBM MQ. Użyj następujących wartości:

```
export OPERATOR_PACKAGE_NAME=ibm-mq
export OPERATOR_VERSION=3.1.3
```

- Należy również utworzyć kopię lustrzaną niektórych dodatkowych wymaganych komponentów, jeśli planowane jest wdrożenie co najmniej jednego menedżera kolejek, w którym **wszystkie** następujące instrukcje są prawdziwe:

- Używany jest produkt IBM MQ 9.3.4 lub nowszej.
- Używana jest licencja na produkt CP4I.
- IBM MQ Console jest włączona.
- Usługa IBM Cloud Pak for Integration Keycloak jest używana do uwierzytelniania i autoryzacji pojedynczego logowania (SSO) w systemie IBM MQ Console (domyślnie).

Jeśli poprzednie instrukcje mają wartość true, funkcja pojedynczego logowania jest udostępniana przez produkt Keycloak i należy utworzyć kopię lustrzaną każdego z następujących komponentów:

- IBM Cloud Pak foundational services
- Certificate Manager. Jeśli zainstalowano wersję operatora IBM Cloud Pak foundational services wcześniejszą niż wersja 4.4, należy utworzyć kopię lustrzaną Certificate Manager.⁶
- IBM Cloud Pak for Integration
- Keycloak (operatorRed Hat OpenShift)

Aby utworzyć obrazy lustrzane, należy zapoznać się z sekcją [Mirroring images for a air-glimitowany cluster](#).

4. Dodaj źródło katalogu IBM MQ Operator.

Dodaj źródło katalogu, które udostępnia operatory w klastrze. Patrz sekcja [“Dodawanie źródła katalogu IBM MQ Operator”](#) na stronie 108.

⁶ Poczynając od wersji 4.4 produktu IBM Cloud Pak foundational services ten zapis lustrzany nie jest już wymagany.

5. Zainstaluj operator IBM MQ Operator.

Wybierz jedną z następujących dwóch opcji (użyj konsoli lub interfejsu CLI):

- Opcja 1: zainstaluj IBM MQ Operator za pomocą konsoli OpenShift.
- Opcja 2: zainstaluj IBM MQ Operator za pomocą interfejsu OpenShift CLI.

6. Utwórz klucz tajny uprawnienia (tylko w przypadku instalacji w trybie z połączeniem).

Program IBM MQ Operator wdraża obrazy menedżera kolejek pobrane z rejestru kontenera, który wykonuje sprawdzenie uprawnień licencyjnych. To sprawdzenie wymaga klucza uprawnień, który jest przechowywany w secrecie pobierania `docker-registry`. Jeśli w przestrzeni nazw, w której będą instalowane menedżery kolejek, nie ma jeszcze klucza uprawnień, należy postępować zgodnie z poniższymi instrukcjami, aby uzyskać klucz uprawnień i utworzyć klucz tajny pobierania.

Uwaga: Klucz uprawnień nie jest wymagany, jeśli zostaną wdrożone tylko menedżery kolejek IBM MQ Advanced for Developers (bez gwarancji).

Klucz tajny upoważnienia można utworzyć za pomocą konsoli OpenShift lub interfejsu CLI. W poniższym przykładzie użyto interfejsu CLI:

- a. Uzyskaj klucz upoważnienia, który jest przypisany do identyfikatora IBM . 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.
- c. W interfejsie wiersza komend systemu OpenShift uruchom następującą komendę, aby utworzyć klucz tajny pobierania obrazu o nazwie `ibm-entitlement-key`.

```
oc create secret docker-registry ibm-entitlement-key \
--docker-server=cp.icr.io \
--docker-username=cp \
--docker-password=<entitlement-key> \
--docker-email=<user-email> \
--namespace=<namespace>
```

Gdzie `<entitlement-key>` jest kluczem upoważnienia skopiowanym w kroku b, `<user-email>` jest identyfikatorem IBM powiązany z uprawnionym oprogramowaniem, a `<namespace>` jest przestrzenią nazw, w której zainstalowano IBM MQ Operator .

7. Opcjonalne: Zainstaluj produkt CP4I i jego zależności.

Podczas wdrażania co najmniej jednego menedżera kolejek z **wszystkimi** następującymi instrukcjami są wymagane dodatkowe komponenty:

- Używany jest produkt IBM MQ 9.3.4 lub nowszej.
- Używana jest licencja na produkt CP4I .
- IBM MQ Console jest włączona.
- Usługa CP4I Keycloak jest używana do uwierzytelniania i autoryzacji pojedynczego logowania (SSO) w systemie IBM MQ Console (domyślnie).

Jeśli wszystkie powyższe instrukcje są prawdziwe, funkcja pojedynczego logowania jest udostępniana przez Keycloak i należy wykonać następujące dodatkowe kroki:

- Operator IBM Cloud Pak foundational services należy zainstalować w tym samym trybie co operator CP4I . Informacje o obsługiwanych wersjach zawiera sekcja [Wersje kanału operatora dla tej wersji](#) .
- Jeśli zainstalowano operator IBM Cloud Pak foundational services w wersji wcześniejszej niż 4.4, zainstaluj operator menedżera certyfikatów dla Red Hat OpenShift Container Platform.⁷
- Zainstaluj operatora CP4I.
- Opcjonalnie: wdróż interfejs użytkownika platformy.

⁷ Od wersji 4.4 programu IBM Cloud Pak foundational services nie jest to już wymagane.

- a. Utwórz przestrzeń nazw `ibm-common-services`. Po zalogowaniu się do klastra OpenShift za pomocą interfejsu CLI uruchom następującą komendę:

```
oc new-project ibm-common-services
```

- b. Wdróż interfejs użytkownika platformy.

8. Wdróż License Service.

Jest to wymagane do monitorowania użycia licencji menedżerów kolejek. Postępuj zgodnie z instrukcjami podanymi w sekcji Wdrażanie usługi License Service.

9. Wdróż menedżera kolejek.

Instrukcje dotyczące wdrażania przykładowego menedżera kolejek "Szybki start" zawiera sekcja "Wdrażanie menedżera kolejek w klastrze platformy kontenerowej Red Hat OpenShift" na stronie 118.

Zadania pokrewne

"Deinstalowanie produktu IBM MQ Operator" na stronie 121

Do zdeinstalowania IBM MQ Operator z Red Hat OpenShift można użyć konsoli Red Hat OpenShift lub interfejsu CLI.

Odsyłacze pokrewne

"Instalowanie produktu IBM MQ Operator 2.x w środowisku z luką powietrzną" na stronie 112

Ten kurs prowadzi użytkownika przez proces instalowania produktu IBM MQ Operator 2.x w klastrze Red Hat OpenShift, który nie ma połączenia z Internetem. Produkt IBM MQ Operator można zainstalować w środowisku z luką powietrzną przy użyciu przenośnego urządzenia pamięci masowej lub urządzenia bastionowego.

Dodawanie źródła katalogu IBM MQ Operator

Dodanie źródła katalogu do klastra OpenShift powoduje dodanie operatorów IBM do listy operatorów, które można zainstalować.

Zanim rozpoczniesz

W tym zadaniu przyjęto, że zostały wykonane 3 pierwsze kroki procedury "Instalowanie produktu IBM MQ Operator" na stronie 105.

To zadanie musi zostać wykonane przez administratora klastra.

O tym zadaniu

Katalog IBM MQ Operator jest indeksem operatorów dostępnym w celu rozszerzenia funkcji API klastra Red Hat OpenShift Container Platform w celu włączenia produktów oprogramowania IBM.

Wypełnij pole **Opcja A: Air-gap** lub **Opcja B: Internet**, w zależności od tego, czy klaster jest w ograniczonym środowisku sieciowym, czy też klaster ma dostęp do Internetu.

Procedura

V 9.3.4

Opcja A: Przerwa powietrzna Dodaj źródło katalogu w środowisku sieciowym z separacją powietrzną.

- a) Dodaj źródło katalogu IBM MQ Operator.

Należy postępować zgodnie z instrukcjami zawartymi w sekcji Dodawanie źródeł katalogu do klastra.

Uwaga: Ponieważ krok instalacji operatora Obrazy lustrzane (tylko przerwa powietrzna) został już zakończony, wystarczy wykonać krok, który dotyczy źródła katalogu. Na przykład:

```
oc apply -f ~/.ibm-pak/data/mirror/${OPERATOR_PACKAGE_NAME}/${OPERATOR_VERSION}/catalog-sources.yaml
```

b) Dodaj źródło katalogu dla dodatkowych wymaganych komponentów.

Podczas wdrażania co najmniej jednego menedżera kolejek z **wszystkimi** następującymi instrukcjami są wymagane dodatkowe komponenty:

- Używany jest produkt IBM MQ 9.3.4 lub nowszej.
- Używana jest licencja IBM Cloud Pak for Integration .
- IBM MQ Console jest włączona.
- Usługa IBM Cloud Pak for Integration Keycloak jest używana do uwierzytelniania i autoryzacji pojedynczego logowania (SSO) w systemie IBM MQ Console (domyślnie).

Jeśli wszystkie poprzednie instrukcje mają wartość true, funkcja pojedynczego logowania jest udostępniana przez produkt Keycloak. Dlatego, podobnie jak w przypadku źródła katalogu IBM MQ Operator , należy również wykonać kroki opisane w sekcji Dodawanie źródeł katalogu do klastra dla każdego z następujących dodatkowych wymaganych komponentów:

- IBM Cloud Pak foundational services
- Certificate Manager. Jeśli zainstalowano wersję operatora IBM Cloud Pak foundational services wcześniejszą niż wersja 4.4, należy utworzyć kopię lustrzaną Certificate Manager.⁸
- IBM Cloud Pak for Integration

- **Opcja B: Internet** Dodaj źródło katalogu w środowisku, które ma dostęp do Internetu.

Utwórz CatalogSource za pomocą interfejsu CLI OpenShift .

Dodaj katalog, stosując następujący plik YAML do klastra Red Hat OpenShift Container Platform .

a) Utwórz YAML CatalogSource .

Zapisz następującą definicję zasobu jako plik o nazwie `catalog_source.yaml`.

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

b) Zastosuj CatalogSource YAML.

Można to zrobić z poziomu konsoli WWW Red Hat OpenShift Container Platform , klikając przycisk "+" lub korzystając z wiersza komend.

Na przykład zastosuj plik, uruchamiając następującą komendę:

```
oc apply -f catalog_source.yaml -n openshift-marketplace
```

c) Pomyślnie sprawdzaj poprawność utworzenia pliku CatalogSource

Uruchom następującą komendę:

```
oc get CatalogSources ibm-operator-catalog -n openshift-marketplace
```

Dane wyjściowe są wyświetlane po pomyślnym zakończeniu operacji:

NAME	DISPLAY	TYPE	PUBLISHER	AGE
ibm-operator-catalog	IBM operator Catalog	grpc	IBM	28s

⁸ Poczawszy od wersji 4.4 produktu IBM Cloud Pak foundational services ten zapis lustrzany nie jest już wymagany.

Wyniki

Teraz można wykonać [krok 5 instalowania IBM MQ Operator](#).

Instalowanie IBM MQ Operator za pomocą konsoli OpenShift

Produkt IBM MQ Operator można zainstalować w systemie Red Hat OpenShift za pomocą serwisu OperatorHub.

Zanim rozpoczniesz

W tym zadaniu przyjęto, że zostały wykonane kroki 1-4 procedury [“Instalowanie produktu IBM MQ Operator”](#) na stronie 105.

Procedura

1. Zaloguj się do konsoli klastra Red Hat OpenShift .
2. Na panelu nawigacyjnym kliknij opcję **Operators** > **OperatorHub**.
Zostanie wyświetlona strona OperatorHub.
3. W polu **All Items** wprowadź „IBM MQ”.
Zostanie wyświetlony wpis IBM MQ w katalogu.
4. Wybierz wpis **IBM MQ**.
Zostanie wyświetlone okno IBM MQ.
5. Kliknij opcję **Install**.
Zostanie wyświetlona strona operatora instalacji.
6. Wprowadź następujące wartości:
 - a) Ustaw opcję **Kanał** na wybraną wersję.
Zapoznaj się z sekcją [“Wersja zgodna z operatorem IBM MQ Operator”](#) na stronie 12, aby określić, który kanał operatora ma zostać wybrany.
 - b) W polu **Tryb instalacji** ustaw wartość "konkretna przestrzeń nazw w klastrze" (którą można utworzyć w następnym kroku) lub zasięg całego klastra.
Zaleca się wybranie zasięgu dla całego klastra, ponieważ instalowanie różnych wersji operatora w różnych przestrzeniach nazw może prowadzić do problemów. Operatorzy są zaprojektowani tak, aby były przedłużeniami płaszczyzny sterowania.
 - c) Opcjonalne: Jeśli wybrano opcję "konkretna przestrzeń nazw w klastrze", w polu **Przestrzeń nazw** ustaw wartość projektu (przestrzeń nazw), w której ma zostać zainstalowany operator.
Uwaga: Podczas instalowania operatora za pomocą konsoli można użyć istniejącej przestrzeni nazw, domyślnej przestrzeni nazw udostępnianej przez operatora lub utworzyć nową przestrzeń nazw. Aby utworzyć nową przestrzeń nazw, można utworzyć ją na podstawie tego formularza w następujący sposób: na panelu nawigacyjnym kliknij opcję **Strona główna** > **Projekty**, wybierz opcję **Utwórz projekt**, podaj **Nazwę** projektu (przestrzeni nazw), który ma zostać utworzony, a następnie kliknij opcję **Utwórz**.
 - d) Ustaw opcję **Strategia zatwierdzania** na wartość Automatyczna.
7. Kliknij przycisk **Instaluj** i poczekaj, aż operator zainstaluje produkt.
Po zakończeniu instalacji zostanie wyświetlone potwierdzenie.
Aby zweryfikować instalację, przejdź do opcji **Operatory** > **Zainstalowane operatory** i wybierz projekt z listy rozwijanej **Projekty** . Po zakończeniu instalacji status operatora zmienia się na Powodzenie.

Co dalej

Teraz można przystąpić do tworzenia klucza tajnego upoważnienia ([krok 6 procedury “Instalowanie produktu IBM MQ Operator”](#) na stronie 105).

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

Produkt IBM MQ Operator można zainstalować w systemie Red Hat OpenShift za pomocą interfejsu wiersza komend (CLI).

Zanim rozpocznie

W tym zadaniu przyjęto, że zostały wykonane kroki 1-4 procedury [“Instalowanie produktu IBM MQ Operator”](#) na stronie 105.

Procedura

1. Zaloguj się do interfejsu wiersza komend produktu Red Hat OpenShift przy użyciu programu **oc login**.

2. Opcjonalne: Utwórz przestrzeń nazw, która ma być używana dla IBM MQ Operator.

Operator IBM MQ Operator może być zainstalowany w zasięgu jednej przestrzeni nazw lub wszystkich przestrzeni nazw. Ten krok jest wymagany tylko wtedy, gdy instalacja ma zostać przeprowadzona w określonej przestrzeni nazw, która jeszcze nie istnieje.

Aby utworzyć nową przestrzeń nazw w interfejsie CLI, uruchom następującą komendę:

```
oc create namespace <namespace_name>
```

Gdzie *< nazwa_przestrzeni_nazw >* jest nazwą przestrzeni nazw, która ma zostać utworzona.

3. Wyświetl listę operatorów dostępnych dla klastra w serwisie OperatorHub:

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

4. Sprawdź plik IBM MQ Operator , aby upewnić się, że jest obsługiwany plik **InstallModes** i dostępny plik **Channels**.

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

5. Opcjonalne: Utwórz **OperatorGroup**.

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

Przestrzeń nazw, do której subskrybujesz element wykonawczy, musi mieć składnik **OperatorGroup** odpowiadający składnikowi **InstallMode** elementu wykonawczego, albo **AllNamespaces**, albo **SingleNamespace**.

Jeśli operator, który ma zostać zainstalowany, używa trybu **AllNamespaces** , to przestrzeń nazw **openshift-operators** ma już odpowiednią wartość **OperatorGroup** i można pominąć ten krok.

Jeśli operator używa trybu **SingleNamespace** i nie ma jeszcze odpowiedniego **OperatorGroup** , utwórz go, uruchamiając następującą komendę:

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: <operatorgroup_name>
  namespace: <namespace_name>
spec:
  targetNamespaces:
  - <namespace_name>
EOF
```

6. Zapoznaj się z sekcją [“Wersja zgodna z operatorem IBM MQ Operator”](#) na stronie 12, aby określić, który kanał operatora ma zostać wybrany.
7. Zainstaluj operatora.

Użyj następującej komendy, zmieniając wartość `<ibm-mq-operator-channel>` na zgodną z kanałem dla wersji operatora IBM MQ, która ma zostać zainstalowana, oraz zmieniając wartość `<nazwa_przestrzeni_nazw>` na **openshift-operators**, jeśli używany jest tryb "AllNamespaces", lub na przestrzeń nazw, w której ma zostać wdrożony operator IBM MQ, jeśli używany jest tryb "SingleNamespace".

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-mq
  namespace: <namespace_name>
spec:
  channel: <ibm-mq-operator-channel>
  installPlanApproval: Automatic
  name: ibm-mq
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

8. Po kilku minutach operator jest instalowany. Uruchom następującą komendę, aby sprawdzić, czy wszystkie komponenty są w stanie Powodzenie:

```
oc get csv -n <namespace_name> | grep ibm-mq
```

Gdzie `<nazwa_przestrzeni_nazw>` to **openshift-operators**, jeśli używany jest tryb "AllNamespaces", lub nazwa projektu (przestrzeni nazw), jeśli używany jest tryb "SingleNamespace".

Co dalej

Teraz można przystąpić do tworzenia klucza tajnego upoważnienia (krok 6 procedury "Instalowanie produktu IBM MQ Operator" na stronie 105).

OpenShift CP4I Linux Instalowanie produktu IBM MQ Operator 2.x w środowisku z luką powietrzną

Ten kurs prowadzi użytkownika przez proces instalowania produktu IBM MQ Operator 2.x w klastrze Red Hat OpenShift, który nie ma połączenia z Internetem. Produkt IBM MQ Operator można zainstalować w środowisku z luką powietrzną przy użyciu przenośnego urządzenia pamięci masowej lub urządzenia bastionowego.

Zanim zaczniesz

Te instrukcje dotyczą instalowania wersji 2.x produktu IBM MQ Operator w środowisku z luką powietrzną. Aby zainstalować produkt IBM MQ Operator 3.0.0 lub nowszej, należy zapoznać się z sekcją "Instalowanie produktu IBM MQ Operator" na stronie 105, zwracając szczególną uwagę na specyficzne kroki związane z luką powietrzną.

Instalowanie IBM MQ Operator w środowisku z luką powietrzną przy użyciu przenośnego urządzenia pamięci masowej

Kroki instalacji opisano w sekcji Tworzenie kopii lustrzanych obrazów z użyciem przenośnego urządzenia pamięci masowej w dokumentacji produktu IBM Cloud Pak for Integration. Jeśli instalowany jest tylko produkt IBM MQ, zastąp wszystkie wystąpienia następujących zmiennych środowiskowych wartościami podanymi w tym miejscu:

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

gdzie `numer_wersji` to wersja przypadku, który ma być używany do przeprowadzenia instalacji z luką powietrzną. Listę dostępnych wersji przypadków można znaleźć pod adresem <https://github.com/IBM/>

[cloud-pak/tree/master/repo/case/ibm-mq](#). Zapoznaj się z sekcją [“Wersja zgodna z operatorem IBM MQ Operator”](#) na stronie 12, aby określić, który kanał operatora ma zostać wybrany.

Instalowanie IBM MQ Operator w środowisku z luką powietrzną przy użyciu maszyny bastionowej

1. [“Wymagania wstępne”](#) na stronie 113
2. [“Przygotowanie rejestru Docker”](#) na stronie 113
3. [“Przygotowanie hosta bastionowego”](#) na stronie 114
4. [“Tworzenie zmiennych środowiskowych dla instalatora i spisu obrazów”](#) na stronie 115
5. [“Pobieranie instalatora IBM MQ i spisu obrazów”](#) na stronie 115
6. [“Logowanie się do klastra OpenShift Container Platform w roli administratora klastra”](#) na stronie 115
7. [“Utworzenie przestrzeni nazw Kubernetes dla operatora IBM MQ Operator”](#) na stronie 115
8. [“Tworzenie kopii lustrzanych obrazów i konfigurowanie klastra”](#) na stronie 115
9. [“Zainstaluj operator IBM MQ Operator.”](#) na stronie 117
10. [“Wdrażanie menedżera kolejek IBM MQ”](#) na stronie 118

Wymagania wstępne

1. Musi być zainstalowany klaster OpenShift Container Platform . Aby uzyskać informacje na temat obsługiwanych wersji OpenShift Container Platform, patrz [“Wersja zgodna z operatorem IBM MQ Operator”](#) na stronie 12.
2. Musi być dostępny rejestr Docker. Więcej informacji na ten temat zawiera sekcja [“Przygotowanie rejestru Docker”](#) na stronie 113.
3. Musi być skonfigurowany serwer bastionowy. Więcej informacji na ten temat zawiera [“Przygotowanie hosta bastionowego”](#) na stronie 114.

Przygotowanie rejestru Docker

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

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

Po utworzeniu rejestru Docker należy go skonfigurować:

- Przykład prostego rejestru znajduje się w sekcji [Tworzenie rejestru lustrzanego dla instalacji w sieci zastrzeżonej](#) w dokumentacji produktu Red Hat OpenShift .
- Upewnij się, że każda przestrzeń nazw spełnia następujące wymagania:
 - Obsługuje automatyczne tworzenie repozytorium.
 - Ma referencje użytkownika z uprawnieniami do zapisu w repozytoriach i tworzenia repozytoriów. Host bastionowy używa tych referencji.
 - Ma referencje użytkownika z uprawnieniami do odczytu wszystkich repozytoriów. Klaster OpenShift Container Platform używa tych referencji.

Przygotowanie hosta bastionowego

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

Wykonaj następujące kroki w węźle bastionowym:

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

- Aby zainstalować program Docker, uruchom następujące komendy:

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

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

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

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

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

- a. Pobierz plik binarny.

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

Na przykład:

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

- b. Wyodrębnij plik binarny.

```
tar -xif binary-file-name
```

- c. Uruchom następujące komendy, aby zmodyfikować i przenieść plik

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

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

```
cloudctl --help
```

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

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

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

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

```
mkdir $HOME/offline
```

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

Tworzenie zmiennych środowiskowych dla instalatora i spisu obrazów

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

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

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

Pobieranie instalatora IBM MQ i spisu obrazów

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

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

Logowanie się do klastra OpenShift Container Platform w roli administratora klastra

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

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

Utworzenie przestrzeni nazw Kubernetes dla operatora IBM MQ Operator

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

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

Tworzenie kopii lustrzanych obrazów i konfigurowanie klastra

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

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

1. Zapisz referencje uwierzytelniania dla wszystkich źródłowych rejestrów Docker.

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

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

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

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

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

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

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

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

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

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

3. Skonfiguruj sekret uwierzytelniania dla lokalnego rejestru Docker.

Uwaga: Ten krok trzeba wykonać tylko jeden raz.

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

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

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

- a. Sprawdź, czy wymagany jest restart węzła.

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

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

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

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

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

```
oc get imageContentSourcePolicy
```

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

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

7. Sprawdź status węzła klastra.

```
oc get nodes
```

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

8. Utwórz kopię lustrzaną obrazów w rejestrze lokalnym.

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

Zainstaluj operator IBM MQ Operator.

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

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

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

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

4. Zainstaluj operator IBM MQ Operator przy użyciu OLM.

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

Zostanie wyświetlona strona **OperatorHub**.

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

Zostanie wyświetlony wpis w katalogu **IBM MQ**.

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

Zostanie wyświetlone okno **IBM MQ**.

- d. Kliknij opcję **Install**.

Zostanie wyświetlona strona **Create Operator Subscription**.

- e. Zapoznaj się z sekcją [“Wersja zgodna z operatorem IBM MQ Operator”](#) na stronie 12, aby określić, który kanał operatora ma zostać wybrany.
- f. Ustaw opcję **Installation Mode** na konkretną utworzoną przestrzeń nazw lub zasięg całego klastra.
- g. Kliknij opcję **Subscribe**.

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

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

Wdrażanie menedżera kolejek IBM MQ

Aby uzyskać instrukcję tworzenia nowego menedżera kolejek w ramach zainstalowanego operatora, patrz [“Wdrażanie menedżera kolejek w klastrze platformy kontenerowej Red Hat OpenShift”](#) na stronie 118.

Zadania pokrewne

[“\[Nieaktualny\]Przygotowanie do aktualizacji do najnowszej wersji produktu IBM MQ 2.x Operator lub menedżer kolejek w środowisku z luką powietrzną”](#) na stronie 129

W klastrze Red Hat OpenShift, który nie ma połączenia z Internetem, należy wykonać kroki przygotowawcze przed aktualizacją operatora lub menedżera kolejek produktu IBM MQ 2.x.

Wdrażanie menedżera kolejek w klastrze platformy kontenerowej Red Hat OpenShift

W tym przykładzie wdrażany jest menedżer kolejek "Szybki start", który używa pamięci efemerycznej (nietrwałej) i wyłącza zabezpieczenia produktu IBM MQ. Komunikaty nie są zachowywane po restarcie menedżera kolejek. Można zmienić konfigurację, aby zmodyfikować różne ustawienia menedżera kolejek.

O tym zadaniu

To zadanie udostępnia 3 opcje wdrażania menedżera kolejek na serwerze OpenShift:

1. [Wdrażanie menedżera kolejek za pomocą OpenShift konsoli](#).
2. [Wdrażanie menedżera kolejek za pomocą interfejsu OpenShift CLI](#).
3. [Wdrażanie menedżera kolejek za pomocą programu IBM Cloud Pak for Integration Platform UI](#).

Procedura

• Opcja 1: wdrażanie menedżera kolejek za pomocą konsoli OpenShift .

- a) Wdróż menedżera kolejek.
 - a. Zaloguj się do konsoli OpenShift, używając informacji autoryzacyjnych administratora klastra Red Hat OpenShift Container Platform.
 - b. Zmień wartość w polu **Projekt** na przestrzeń nazw, w której zainstalowano produkt IBM MQ Operator. Wybierz przestrzeń nazw z listy rozwijanej **Projekt**.
 - c. W panelu nawigacyjnym kliknij opcję **Operatory > Zainstalowane operatory**.
 - d. Na liście znajdującej się na panelu zainstalowanych operatorów znajdź i kliknij pozycję **IBM MQ**.
 - e. Kliknij kartę **Queue Manager**.
 - f. Kliknij przycisk **Create QueueManager**. Zostanie wyświetlony panel tworzenia instancji zawierający dwie metody konfigurowania zasobu: **Widok formularza** i **widok YAML**. Domyślnie wybrany jest **widok formularza**.

- b) Skonfiguruj menedżer kolejek.

Krok 2 Opcja 1: konfigurowanie w **widoku formularza**.

W widoku **Formularz** zostanie otwarty formularz, którego można użyć do wyświetlenia lub zmodyfikowania konfiguracji zasobu.

- a. Kliknij strzałkę znajdującą się obok pola **Licencja**, aby rozwinąć sekcję akceptowania licencji.
- b. Ustaw opcję **Akceptuj licencję** na wartość **true** , jeśli akceptujesz umowę licencyjną.
- c. Kliknij strzałkę, aby otworzyć listę rozwijaną i wybierz licencję. 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 187. Aby wdrożyć menedżer kolejek, należy zaakceptować licencję.
- d. Kliknij opcję **Create**. Zostanie teraz wyświetlona lista menedżerów kolejek w bieżącym projekcie (przestrzeni nazw). Nowy zasób QueueManager powinien być w stanie Pending.

Krok 2 Opcja 2: konfigurowanie w **widoku YAML**.

W widoku **YAML** zostanie otwarty edytor zawierający przykładowy plik YAML dla QueueManager. Zaktualizuj wartości w pliku, wykonując poniższe kroki.

- a. Zmień wartość metadata . namespace na nazwę projektu (przestrzeń nazw).
 - b. Zmień wartość spec . license . license na łańcuch licencji, który jest zgodny z wymaganiami. Szczegółowe informacje na temat licencji zawiera sekcja [“Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1”](#) na stronie 187 .
 - c. Po zaakceptowaniu umowy licencyjnej zmień wartość spec . license . accept na true .
 - d. Kliknij opcję **Create**. Zostanie teraz wyświetlona lista menedżerów kolejek w bieżącym projekcie (przestrzeni nazw). Nowy zasób QueueManager powinien być w stanie Pending.
- c) Sprawdź poprawność utworzenia menedżera kolejek.
- Aby sprawdzić, czy menedżer kolejek został utworzony, wykonaj następujące kroki:
- a. Upewnij się, że jesteś w przestrzeni nazw, w której utworzono IBM MQ Operator .
 - b. Na ekranie **Strona główna** kliknij opcję **Operatory** > **Zainstalowane operatory**, a następnie wybierz zainstalowany IBM MQ Operator , dla którego został utworzony menedżer kolejek.
 - c. Kliknij kartę **Queue Manager**. Status Running zasobu QueueManager oznacza, że tworzenie zostało zakończone.

• **Opcja 2: wdrażanie menedżera kolejek za pomocą interfejsu CLI produktu OpenShift .**

- 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: 9.3.5.1-r2
  license:
    accept: false
    license: L-VTPK-22YZPK
    use: NonProduction
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
    storage:
      queueManager:
        type: ephemeral
```

Ważne: Jeśli akceptujesz umowę licencyjną, zmień accept: false na accept: true. Aby uzyskać szczegółowe informacje na temat licencji, patrz [“Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1”](#) na stronie 187.

Ten przykład obejmuje również serwer WWW wdrożony z menedżerem kolejek z włączoną konsolą WWW z pojedynczym logowaniem w produkcie IBM Cloud Pak for Integration. **V 9.3.4** Od wersji IBM Cloud Pak for Integration 2023.4.1, aby funkcja pojedynczego logowania działała, należy najpierw zainstalować inne komponenty IBM Cloud Pak for Integration.

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: 9.3.5.1-r2
  license:
    accept: false
    license: L-AMRD-XH6P3Q
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
    storage:
      queueManager:
        type: ephemeral
```

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 187.

b) Utwórz obiekt QueueManager .

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

c) Sprawdź poprawność utworzenia menedżera kolejek.

Sprawdź, czy menedżer kolejek został utworzony, wykonując następujące kroki:

a. Sprawdź poprawność wdrożenia:

```
oc describe queuemanager <QueueManagerResourceName>
```

b. Sprawdź status:

```
oc describe queuemanager quickstart
```

- **Opcja 3: wdrażanie menedżera kolejek za pomocą programu IBM Cloud Pak for Integration Platform UI.**

a) W przeglądarce uruchom IBM Cloud Pak for Integration Platform UI.

b) W IBM Cloud Pak for Integration Platform UI kliknij opcję **Utwórz instancję**.

c) Wybierz opcję **Przesyłanie komunikatów**, a następnie kliknij przycisk **Dalej**.

Zostanie wyświetlony panel tworzenia instancji zawierający dwie metody konfigurowania zasobu: **Widok formularza** i **widok YAML**. Domyślnie wybrany jest **widok formularza** .

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 uzyskać szczegółowe informacje na temat licencji, patrz [“Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1”](#) na stronie 187. Aby wdrożyć menedżer kolejek, należy zaakceptować licencję.

f) W sekcji **Queue Manager** sprawdź lub zaktualizuj w polu **Name** nazwę bazowego menedżera kolejek. W starszych wersjach IBM Cloud Pak for Integration Platform UI użyj sekcji **Queue Manager Config**.

Domyślnie nazwa menedżera kolejek używanego przez aplikacje klienckie IBM MQ jest taka sama jak nazwa QueueManager, ale z usuniętymi niepoprawnymi znakami (na przykład myślnikami).

g) Kliknij opcję **Create**.

Zostanie teraz wyświetlona lista menedżerów kolejek w bieżącym projekcie (przestrzeni nazw). Nowy zasób QueueManager powinien mieć status Pending.

h) Sprawdź poprawność utworzenia menedżera kolejek.

Status Running zasobu QueueManager oznacza, że tworzenie zostało zakończone.

Zadania pokrewne

[“Konfigurowanie trasy w celu nawiązania połączenia z menedżerem kolejek spoza klastra produktu Red Hat OpenShift” na stronie 159](#)

Do połączenia aplikacji z menedżerem kolejek IBM MQ spoza klastra Red Hat OpenShift potrzebna jest trasa Red Hat OpenShift. Należy włączyć protokół TLS w menedżerze kolejek i aplikacji klienckiej IBM MQ, ponieważ protokół SNI jest dostępny w protokole TLS tylko wtedy, gdy używany jest protokół TLS w wersji 1.2 lub nowszej. Red Hat OpenShift Container Platform Router używa wskaźnika SNI do kierowania żądań do menedżera kolejek IBM MQ.

[“Nawiązywanie połączenia z rozwiązaniem IBM MQ Console wdrożonym w klastrze Red Hat OpenShift” na stronie 176](#)

W tej sekcji opisano nawiązywanie połączenia z bazą danych IBM MQ Console menedżera kolejek, który został wdrożony w klastrze Red Hat OpenShift Container Platform.

[“Przykłady konfigurowania menedżera kolejek” na stronie 137](#)

Menedżer kolejek można skonfigurować, modyfikując zawartość niestandardowego zasobu QueueManager.

OpenShift

CP4I

Deinstalowanie produktu IBM MQ Operator

Do zdeinstalowania IBM MQ Operator z Red Hat OpenShift można użyć konsoli Red Hat OpenShift lub interfejsu CLI.

Procedura

- Opcja 1: zdeinstaluj program IBM MQ Operator za pomocą konsoli OpenShift.

Uwaga: Jeśli program IBM MQ Operator jest zainstalowany we wszystkich projektach/przestrzeniach nazw w klastrze, powtórz kroki 2-6 poniższej procedury dla każdego projektu, w którym mają zostać usunięte menedżery kolejek.

- a) Zaloguj się do konsoli WWW Red Hat OpenShift Container Platform, używając informacji autoryzacyjnych administratora klastra Red Hat OpenShift Container Platform.
- b) Zmień wartość w polu **Projekt** na przestrzeń nazw, z której ma zostać zdeinstalowany produkt IBM MQ Operator. Wybierz przestrzeń nazw z listy rozwijanej **Projekt**.
- c) W panelu nawigacyjnym kliknij opcję **Operatory > Zainstalowane operatory**.
- d) Kliknij operator **IBM MQ**.
- e) Kliknij kartę **Queue Managers**, aby wyświetlić menedżery kolejek zarządzane przez ten operator IBM MQ Operator.
- f) Usuń jeden lub więcej menedżerów kolejek.
Należy zauważyć, że pomimo tego, że menedżery kolejek nadal działają, mogą nie działać zgodnie z oczekiwaniami bez operatora IBM MQ Operator.
- g) Opcjonalnie: W razie potrzeby powtórz kroki od 2 do 6 dla każdego projektu, w którym mają zostać usunięte menedżery kolejek.
- h) Wróć do opcji **Operators > Installed Operators**.
 - i) Obok operatora **IBM MQ** kliknij wielokropek i wybierz opcję **Uninstall Operator**.

- Opcja 2: zdeinstaluj program IBM MQ Operator za pomocą interfejsu CLI programu OpenShift

- a) Zaloguj się do klastra produktu Red Hat OpenShift przy użyciu programu `oc login`.
- b) Jeśli IBM MQ Operator jest zainstalowany w jednej przestrzeni nazw, wykonaj następujące kroki podrzędne:
- a. Upewnij się, że znajdujesz się w projekcie zawierającym plik IBM MQ Operator , który ma zostać zdeinstalowany:

```
oc project <project_name>
```

- b. Wyświetl menedżery kolejek zainstalowane w projekcie:

```
oc get qmgr
```

- c. Usuń jeden lub więcej menedżerów kolejek:

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

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

- d. Wyświetl instancje **ClusterServiceVersion**:

```
oc get csv
```

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

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

- f. Wyświetl subskrypcje:

```
oc get subscription
```

- g. Usuń wszystkie subskrypcje:

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

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

i) Zdeinstaluj operator usług wspólnych, postępując zgodnie z instrukcjami zawartymi w sekcji [Deinstalowanie podstawowych usług w dokumentacji produktu IBM Cloud Pak foundational services](#) .

- ii) Wyświetl grupę operatorów:

```
oc get operatorgroup
```

- iii) Usuń grupę operatorów:

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

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

- a. Wyświetl wszystkie zainstalowane menedżery kolejek:

```
oc get qmgr -A
```

- b. Usuń jeden lub więcej menedżerów kolejek:

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

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

c. Wyświetl instancje **ClusterServiceVersion**:

```
oc get csv -A
```

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

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

e. Wyświetl subskrypcje:

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

f. Usuń subskrypcje:

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

g. Opcjonalnie: jeśli nic innego nie używa wspólnych usług, można zdeinstalować operatora wspólnych usług. W tym celu należy postępować zgodnie z instrukcjami zawartymi w sekcji [Deinstalowanie usług podstawowych](#) w dokumentacji produktu IBM Cloud Pak foundational services .

Operator 2.0.0 OpenShift CP4I Aktualizowanie operatora IBM MQ Operator i menedżerów kolejek

Istnieją różne procesy aktualizacji dla wersji Continuous Delivery (CD) i Long Term Support (LTS) produktu IBM MQ Operator. Wykonaj krok aktualizacji dla danego typu wdrożenia.

O tym zadaniu

Aby zaktualizować program IBM MQ Operator i menedżery kolejek, wykonaj jeden z następujących kroków:

Procedura

- Opcja 1: **Zaktualizuj wdrożenia do najnowszej wersji w bieżącym kanale operatora** .

Aby zaktualizować wdrożenia produktu IBM MQ Operator do najnowszej wersji w bieżącym kanale operatora, należy zapoznać się z sekcją [“Aktualizacja do najnowszej wersji zabezpieczeń kanału IBM MQ Operator”](#) na stronie 123.

- Opcja 2: **Upgrade CD deployments** (Zaktualizuj wdrożenia produktu).

Aby zaktualizować wcześniejsze wdrożenia produktu CD IBM MQ Operator do najnowszej wersji produktu CD IBM MQ Operator (wersja 3.1.3), należy zapoznać się z sekcją [“Migracja do bieżącego kanału CD IBM MQ Operator”](#) na stronie 125.

Uwaga:

Wersja 2.0.x została wydana zarówno w wersji CD , jak i LTS , dlatego można użyć procedury opisanej w sekcji [“Migracja do bieżącego kanału CD IBM MQ Operator”](#) na stronie 125 , aby przeprowadzić aktualizację z dowolnej wersji 2.0.x IBM MQ Operator do najnowszej wersji produktu CD IBM MQ Operator.

OpenShift CP4I Aktualizacja do najnowszej wersji zabezpieczeń kanału IBM MQ Operator

Aktualizacja programu IBM MQ Operator umożliwia zaktualizowanie menedżerów kolejek.

Zanim rozpoczniesz

Ważne: Ten temat dotyczy aktualizacji wdrożeń IBM MQ Operator do najnowszej wersji zabezpieczeń w kanale wdrożenia. Jeśli nie dotyczy to danego wdrożenia, należy zapoznać się z alternatywnymi

ścieżkami aktualizacji opisanymi w sekcji [“Aktualizowanie operatora IBM MQ Operator i menedżerów kolejek”](#) na stronie 123.

W przypadku wdrożeń IBM MQ Operator w klastrze Red Hat OpenShift , który nie ma połączenia z Internetem, należy wykonać procedurę opisaną w sekcji [“\[Nieaktualny\]Przygotowanie do aktualizacji do najnowszej wersji produktu IBM MQ 2.x Operator lub menedżer kolejek w środowisku z luką powietrzną”](#) na stronie 129.

Procedura

1. Zaktualizuj produkt IBM MQ Operator do nowszej wersji.



Jeśli ustawiono automatyczne aktualizacje, to po wydaniu nowego wydania zabezpieczeń program IBM MQ Operator wykona aktualizację.

Jeśli nie masz ustawionych automatycznych aktualizacji, ręcznie zatwierdź aktualizację IBM MQ Operator :

- Jeśli dostępna jest aktualizacja, **Upgrade Status** może mieć wartość "Dostępna aktualizacja".
- W takim przypadku może istnieć dostępna kontrola, której można użyć do zatwierdzenia **InstallPlan** , który aktualizuje IBM MQ Operator.

2. Aktualizacja menedżera kolejek systemu IBM MQ do nowszej wersji.

W poniższej tabeli opisano najnowszą wersję menedżera kolejek IBM MQ dla każdego aktywnego kanału operatora. Korzystając z odpowiedniej wersji, wykonaj procedurę opisaną w sekcji [“Aktualizowanie menedżera kolejek systemu IBM MQ przy użyciu programu Red Hat OpenShift”](#) na stronie 134.

Kanał operatora	Najnowszy menedżer kolejek produktu IBM MQ
 v2.0 (LTS)	9.3.0.17-r3
 v3.1 (dysk CD)	9.3.5.1-r2

Migracja do kanału LTS produktu IBM MQ Operator

Aktualizacja programu IBM MQ Operator umożliwia zaktualizowanie menedżerów kolejek.

Zanim rozpoczniesz

Ważne: Ten temat dotyczy aktualizacji wdrożeń 1.3.x Long Term Support (LTS) IBM MQ Operator do LTS strumienia IBM MQ Operator 2.0.x **tylko**. Jeśli nie dotyczy to danego wdrożenia, należy zapoznać się z alternatywnymi ścieżkami aktualizacji opisanymi w sekcji [Aktualizowanie produktu IBM MQ Operator i menedżerów kolejek do nowej wersji](#).

W przypadku wdrożeń IBM MQ Operator w klastrze Red Hat OpenShift , który nie ma połączenia z Internetem, należy wykonać procedurę opisaną w sekcji [“\[Nieaktualny\]Przygotowanie do aktualizacji do najnowszej wersji produktu IBM MQ 2.x Operator lub menedżer kolejek w środowisku z luką powietrzną”](#) na stronie 129.

Ważne: Produkt IBM MQ Operator 2.0.x wymaga:

- Red Hat OpenShift Container Platform 4.12.

Aby przeprowadzić aktualizację, należy wykonać procedurę opisaną w sekcji [Aktualizowanie produktu Red Hat OpenShift](#).

- IBM Cloud Pak foundational services 3.19.x

W przypadku aktualizacji z wersji IBM MQ Operator 1.3.x (2020.4) *obie* instancje menedżera kolejek z wieloma instancjami są restartowane równocześnie. Dzieje się tak po zmianie wersji produktu IBM MQ na 9.2.0.5-r3-eus. Podczas aktualizacji z wersji IBM MQ Operator 1.3.x do wersji 2.0.x istnieje sekwencyjna aktualizacja menedżera kolejek produktu IBM MQ . Jeśli zainstalowano produkt IBM Cloud Pak for Integration Platform UI , po zmianie wersji IBM Cloud Pak for Integration Platform UI na 2020.4.1-8-eusi na 2022.2.1-0 następuje dodatkowy restart produktu IBM MQ .

Procedura

1. Przed skorzystaniem z odsyłacza w kroku 2 należy zapoznać się z następującymi podstawowymi informacjami dotyczącymi aktualizacji:
 - Należy pominąć wszystkie kroki podrzędne dla komponentów, które nie zostały zainstalowane. Dotyczy to również IBM Cloud Pak for Integration Platform UI , jeśli nie jest on zainstalowany.
 - Krok 2 prowadzi do dokumentacji produktu IBM Cloud Pak for Integration . Podczas procesu aktualizacji zostanie ponownie omówiony następujący temat IBM MQ w celu zaktualizowania operacji IBM MQ do nowej wersji: [Aktualizowanie menedżera kolejek systemu IBM MQ](#).
 - Zaleca się, aby wszyscy użytkownicy programu IBM MQ wykonali co najmniej następujące czynności, korzystając z instrukcji podanych w kroku 2, a także z instrukcji dotyczących danego środowiska:
 - Poprawka IBM MQ Operator i operand (Patching 2020.4):
 - Zaktualizuj program IBM MQ Operator do co najmniej wersji 1.3.5 w kanale operatora v1.3-eus .
 - Zaktualizuj program IBM MQ Operand (obraz kontenera menedżera kolejek) do wersji co najmniej 9.2.0.5-r3-eus.
 - **Uwaga:** Zaleca się zaktualizowanie operatora IBM MQ przynajmniej do tej wersji, ale nie jest to obowiązkowe.
 - Zależności aktualizacji:
 - Zaktualizuj komponent IBM Cloud Pak foundational services.
 - Zaktualizuj komponent OpenShift Container Platform.
 - Zaktualizuj operatorów:
 - Zaktualizuj IBM MQ Operator do wersji 2.0.23.
 - Zaktualizuj możliwości:
 - Zaktualizuj program IBM MQ Operand (obraz kontenera menedżera kolejek) do najnowszej wersji 9.3.0 (9.3.0.17-r3), aby uzyskać najnowsze poprawki zabezpieczeń.
2. Zaktualizuj program IBM MQ Operator i menedżery kolejek, wykonując [aktualizację z produktu IBM MQ Operator 1.3-eus \(IBM Cloud Pak for Integration 2020.4\)](#).

Zadania pokrewne

[“\[Nieaktualny\]Przygotowanie do aktualizacji do najnowszej wersji produktu IBM MQ 2.x Operator lub menedżer kolejek w środowisku z luką powietrzną” na stronie 129](#)

W klastrze Red Hat OpenShift , który nie ma połączenia z Internetem, należy wykonać kroki przygotowawcze przed aktualizacją operatora lub menedżera kolejek produktu IBM MQ 2.x .

[“Aktualizowanie IBM MQ Operator za pomocą programu Red Hat OpenShift” na stronie 132](#)
IBM MQ Operator można zaktualizować za pomocą konsoli WWW Red Hat OpenShift lub interfejsu CLI.

[“Aktualizowanie menedżera kolejek systemu IBM MQ przy użyciu programu Red Hat OpenShift” na stronie 134](#)

Migracja do bieżącego kanału CD IBM MQ Operator

Aktualizacja z wcześniejszej wersji produktu IBM MQ Operator do wersji 3.1.3. Aktualizacja operatora umożliwia aktualizację menedżerów kolejek.

Zanim rozpoczniesz

Ten temat dotyczy aktualizowania wdrożeń Continuous Delivery (CD) produktu IBM MQ Operator w wersji wcześniejszej niż 3.1.0 do wersji 3.1.3 **Tylko**. Jeśli nie dotyczy to danego wdrożenia, należy zapoznać się z alternatywnymi ścieżkami aktualizacji opisanymi w sekcji [Aktualizowanie produktu IBM MQ Operator i menedżerów kolejek do nowej wersji](#).

Aby wykonać aktualizację do produktu IBM MQ Operator 3.1.3 , należy uruchomić program Red Hat OpenShift Container Platform 4.12 lub nowszej. Aby zaktualizować platformę, należy zapoznać się z sekcją [Aktualizowanie produktu Red Hat OpenShift](#).

Uwaga: Obsługiwane są tylko wersje systemu OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

Procedura

1. Opcjonalne: **Zaktualizuj produkt IBM MQ Operator , który jest obecnie w wersji CD wcześniejszej niż 2.0.0.**

Jeśli IBM MQ Operator jest obecnie w wersji 1.x CD , najpierw wykonaj procedurę opisaną w sekcji [“Migrowanie produktu 1.x CD IBM MQ Operator do wersji 2.0.x”](#) na stronie 127, a następnie wróć do tego miejsca, aby przeprowadzić aktualizację do najnowszej wersji CD .

2. Opcjonalne: **Zaktualizuj serwer IBM MQ Operator w wersji 2.2.x lub 2.3.x do wersji 2.4.x.**

Jeśli serwer IBM MQ Operator jest obecnie w wersji 2.2.x lub 2.3.x , wykonaj odpowiednie kroki opisane w sekcji [“Migracja do kanału v2.4 produktu IBM MQ Operator”](#) na stronie 128, a następnie wróć do tego miejsca, aby zaktualizować serwer CD do najnowszej wersji. Należy zauważyć, że jest to obowiązkowy krok wymagań wstępnych przed aktualizacją do wersji 3.1.3.

3. **Zaktualizuj komponenty.**

Wybierz jedną z następujących opcji:

- **Opcja 1:** jeśli jesteś użytkownikiem programu CP4I lub wdrożono co najmniej jeden z menedżerów kolejek przy użyciu licencji produktu CP4I , wykonaj odpowiednie kroki w celu **zaktualizowania wszystkich komponentów**, w tym programu IBM MQ Operator i menedżerów kolejek, zgodnie z wygenerowanym planem aktualizacji:
 - Aby przeprowadzić aktualizację z wersji 2023.2, należy zapoznać się z sekcją [Aktualizowanie z wersji 2023.2](#) , generując plan aktualizacji.
 - Aby przeprowadzić aktualizację z wersji 2022.2, należy zapoznać się z sekcją [Aktualizowanie z wersji 2022.2](#) , generując plan aktualizacji.
- **Opcja 2:** dla wszystkich pozostałych użytkowników:
 - a. **Obrazy lustrzane (tylko przerwa powietrzna).**

Należy utworzyć kopię lustrzaną obrazów IBM MQ . Wykonaj kroki podane w następującym odsyłaczu, używając tylko tych wartości:

```
export OPERATOR_PACKAGE_NAME=ibm-mq
export OPERATOR_VERSION=3.1.3
```

Należy pominąć sekcję 3.5 "Konfigurowanie klastra", ponieważ połączenie z rejestrem obrazów powinno zostać skonfigurowane podczas poprzednich instalacji lub aktualizacji.

Odsyłacz: [Tworzenie kopii lustrzanych obrazów dla klastra z łączem powietrznym](#).

- b. **Zaktualizuj serwer IBM MQ Operator do wersji 3.1.3.**

Patrz sekcja [“Aktualizowanie IBM MQ Operator za pomocą programu Red Hat OpenShift”](#) na stronie 132.

- c. **Zaktualizuj instancje.**

Aby uzyskać najnowsze funkcje i poprawki zabezpieczeń, należy zaktualizować program IBM MQ Operand (obraz kontenera menedżera kolejek) do najnowszej wersji programu CD (9.3.5.1-r2). Patrz [“Aktualizowanie menedżera kolejek systemu IBM MQ przy użyciu programu Red Hat OpenShift”](#) na stronie 134.

4. Opcjonalne: **Zaktualizuj produkt Red Hat OpenShift Container Platform 4.12 do wersji 4.14.**

Od wersji IBM MQ Operator 3.0.0wymagana jest wersja Red Hat OpenShift Container Platform 4.12 . Należy pamiętać, że opcjonalnie można wybrać opcję aktualizacji do wersji Red Hat OpenShift 4.14. Aby sprawdzić kompatybilne wersje dla każdego kanału IBM MQ Operator , należy zapoznać się z sekcją [“Kompatybilne wersje środowiska Red Hat OpenShift Container Platform”](#) na stronie 13. Aby przeprowadzić aktualizację, należy zapoznać się z sekcją [Aktualizowanie produktu Red Hat OpenShift](#).

5. Opcjonalne: **Przypnij konkretne źródło katalogu dla IBM MQ Operator.**

Jeśli aktualizowana instalacja korzysta z katalogu IBM MQ Operator , należy przypiąć konkretne źródło katalogu dla IBM MQ Operator. Patrz sekcja [Przenoszenie do konkretnych źródeł katalogu dla każdego operatora](#).

Zadania pokrewne

[“\[Nieaktualny\]Przygotowanie do aktualizacji do najnowszej wersji produktu IBM MQ 2.x Operator lub menedżer kolejek w środowisku z luką powietrzną”](#) na stronie 129

W klastrze Red Hat OpenShift , który nie ma połączenia z Internetem, należy wykonać kroki przygotowawcze przed aktualizacją operatora lub menedżera kolejek produktu IBM MQ 2.x .

Migrowanie produktu 1.x CD IBM MQ Operator do wersji 2.0.x

Aktualizacja programu IBM MQ Operator umożliwia zaktualizowanie menedżerów kolejek.

Zanim rozpocznie

Ważne: Ten temat dotyczy aktualizacji **tylko** Continuous Delivery (CD) wdrożeń produktu IBM MQ Operator w wersji wcześniejszej niż 2.0.x do wersji 2.0.x . Jeśli nie dotyczy to wdrożenia, należy zapoznać się z alternatywnymi ścieżkami aktualizacji opisanymi w sekcji [Aktualizowanie produktu IBM MQ Operator i menedżerów kolejek do nowej wersji](#).

W przypadku wdrożeń IBM MQ Operator w klastrze Red Hat OpenShift , który nie ma połączenia z Internetem, należy wykonać procedurę opisaną w sekcji [“\[Nieaktualny\]Przygotowanie do aktualizacji do najnowszej wersji produktu IBM MQ 2.x Operator lub menedżer kolejek w środowisku z luką powietrzną”](#) na stronie 129.

Aby można było wykonać tę aktualizację, muszą być spełnione następujące wymagania dotyczące produktu IBM MQ Operator 2.0.0 :

- Red Hat OpenShift Container Platform 4.12.

Aby przeprowadzić aktualizację, należy wykonać procedurę opisaną w sekcji [Aktualizowanie produktu Red Hat OpenShift](#).

- IBM Cloud Pak foundational services 3.19

Procedura

1. Przed skorzystaniem z odsyłacza w kroku 2 należy zapoznać się z następującymi podstawowymi informacjami dotyczącymi aktualizacji:
 - Pomiń wszystkie kroki podrzędne dla komponentów, które nie zostały zainstalowane. Dotyczy to również IBM Cloud Pak for Integration Platform UI , jeśli nie jest on zainstalowany.
 - Krok 2 prowadzi do dokumentacji produktu IBM Cloud Pak for Integration . Podczas procesu aktualizacji zostanie ponownie omówiony następujący temat IBM MQ w celu zaktualizowania operacji IBM MQ do nowej wersji: [Aktualizowanie menedżera kolejek systemu IBM MQ](#).

- Zaleca się, aby wszyscy użytkownicy programu IBM MQ wykonali co najmniej następujące czynności, korzystając z instrukcji podanych w kroku 2, a także z instrukcji dotyczących danego środowiska:
 - Poprawka IBM MQ Operator i operand (Patching 2021.4):
 - Zaktualizuj produkt IBM MQ Operator do wersji 1.8.0 w kanale operatora v1.8 .
 - Zaktualizuj program IBM MQ Operand (obraz kontenera menedżera kolejek) do co najmniej wersji 9.2.5.0-r3.

Uwaga: Zaleca się zaktualizowanie operatora IBM MQ do bieżącej wersji (9.3.0.17-r3), ale nie jest to obowiązkowe.
 - Zależności aktualizacji:
 - Zaktualizuj komponent IBM Cloud Pak foundational services.
 - Zaktualizuj komponent OpenShift Container Platform.
 - Zaktualizuj operatorów:
 - Zaktualizuj IBM MQ Operator do wersji 2.0.23.
 - Zaktualizuj możliwości:
 - Zaktualizuj program IBM MQ Operand (obraz kontenera menedżera kolejek) do najnowszej wersji 9.3.0 (9.3.0.17-r3), aby uzyskać najnowsze poprawki zabezpieczeń.
- 2. Zaktualizuj program IBM MQ Operator i menedżery kolejek, wykonując czynności [Aktualizacja z operatora IBM MQ 1.8 \(IBM Cloud Pak for Integration 2021.4\)](#) lub wcześniejszej wersji produktu CD IBM MQ Operator .

Co dalej

Teraz można zaktualizować program IBM MQ Operator i menedżery kolejek do najnowszej wersji programu CD (3.1.3). Patrz [“Migracja do bieżącego kanału CD IBM MQ Operator”](#) na stronie 125.

Migracja do kanału v2.4 produktu IBM MQ Operator

Aktualizacja programu IBM MQ Operator umożliwia zaktualizowanie menedżerów kolejek.

Zanim rozpocznie

Ważne: Ten temat dotyczy aktualizowania wdrożeń Continuous Delivery (CD) produktu IBM MQ Operator w wersji wcześniejszej niż 2.4.0 do wersji 2.4.8 **tylko**. Jest to krok pośredni do aktualizacji do najnowszej wersji produktu CD IBM MQ Operator; kanał v2.4 nie otrzymuje aktualizacji zabezpieczeń. Jeśli nie dotyczy to danego wdrożenia, należy zapoznać się z alternatywnymi ścieżkami aktualizacji opisanymi w sekcji [Aktualizowanie produktu IBM MQ Operator i menedżerów kolejek do nowej wersji](#).

W przypadku wdrożeń IBM MQ Operator w klastrze Red Hat OpenShift , który nie ma połączenia z Internetem, należy wykonać procedurę opisaną w sekcji [“\[Nieaktualny\]Przygotowanie do aktualizacji do najnowszej wersji produktu IBM MQ 2.x Operator lub menedżer kolejek w środowisku z luką powietrzną”](#) na stronie 129.

Aby zakończyć tę aktualizację, muszą być spełnione następujące wymagania dotyczące produktu IBM MQ Operator 2.4.8 :

- Red Hat OpenShift Container Platform 4.12.

Aby przeprowadzić aktualizację, należy wykonać procedurę opisaną w sekcji [Aktualizowanie produktu Red Hat OpenShift](#).

Uwaga: Obsługiwane są tylko wersje systemu OpenShift Container Platform Extended Update Support (EUS), które są wersjami podrzędnymi o numerach parzystych, na przykład 4.14 i 4.16.

- IBM Cloud Pak foundational services 3.19 do 3.24 włącznie.

Procedura

1. Opcjonalne: **Aktualizacja produktu IBM MQ Operator , który jest obecnie w wersji CD wcześniejszej niż 2.0.0**

Jeśli IBM MQ Operator jest obecnie w wersji 1.x CD , najpierw wykonaj procedurę opisaną w sekcji [“Migrowanie produktu 1.x CD IBM MQ Operator do wersji 2.0.x”](#) na stronie 127, a następnie wróć do tej sekcji, aby przeprowadzić aktualizację do najnowszej wersji 2.4 .

2. **Zaktualizuj IBM MQ Operator w wersji CD 2.x.x do najnowszej wersji 2.4 (2.4.8).**

Wykonaj procedurę opisaną w sekcji [“Aktualizowanie IBM MQ Operator za pomocą programu Red Hat OpenShift”](#) na stronie 132.

3. Opcjonalne: **Zaktualizuj inne komponenty produktu IBM Cloud Pak for Integration.**

Jeśli jesteś użytkownikiem IBM Cloud Pak for Integration , możesz mieć inne komponenty, które chcesz zaktualizować. Kroki aktualizacji innych komponentów można wykonać w zależności od wdrożenia, wykonując następujące kroki:

- Opcja 1: [Upgrade from IBM MQ Operator 2.0.x/2.1.x \(IBM Cloud Pak for Integration 2022.2\)](#).
- Opcja 2: [Upgrade from IBM MQ Operator 2.2.x/2.3.x \(IBM Cloud Pak for Integration 2022.4\)](#).

4. Opcjonalne: **Zaktualizuj IBM Cloud Pak foundational services.**

Jeśli użytkownik jest użytkownikiem produktu IBM Cloud Pak for Integration , może chcieć zaktualizować produkt IBM Cloud Pak foundational services z wersji 3.19.x do wersji 3.24.x. Informacje o krokach, które należy wykonać, aby wykonać tę aktualizację, zawiera sekcja [Aktualizowanie produktu IBM Cloud Pak foundational services](#).

Zadania pokrewne

[“\[Nieaktualny\]Przygotowanie do aktualizacji do najnowszej wersji produktu IBM MQ 2.x Operator lub menedżer kolejek w środowisku z luką powietrzną”](#) na stronie 129

W klastrze Red Hat OpenShift , który nie ma połączenia z Internetem, należy wykonać kroki przygotowawcze przed aktualizacją operatora lub menedżera kolejek produktu IBM MQ 2.x .

[“Aktualizowanie IBM MQ Operator za pomocą programu Red Hat OpenShift”](#) na stronie 132

IBM MQ Operator można zaktualizować za pomocą konsoli WWW Red Hat OpenShift lub interfejsu CLI.

[“Aktualizowanie menedżera kolejek systemu IBM MQ przy użyciu programu Red Hat OpenShift”](#) na stronie 134

Przygotowanie do aktualizacji do najnowszej wersji produktu IBM MQ 2.x Operator lub menedżer kolejek w środowisku z luką powietrzną

W klastrze Red Hat OpenShift , który nie ma połączenia z Internetem, należy wykonać kroki przygotowawcze przed aktualizacją operatora lub menedżera kolejek produktu IBM MQ 2.x .

Zanim rozpoczniesz

Uwaga: Te instrukcje dotyczą aktualizacji produktu IBM MQ Operator do wersji 2.x w środowisku z luką powietrzną. Aby przeprowadzić aktualizację do wersji IBM MQ Operator 3.0.0 lub nowszej, należy zapoznać się z sekcją [“Aktualizowanie operatora IBM MQ Operator i menedżerów kolejek”](#) na stronie 123, zwracając szczególną uwagę na specyficzne kroki związane z luką powietrzną.

W tym temacie przyjęto założenie, że skonfigurowano już lokalny rejestr obrazów, w którym są tworzone kopie lustrzane poprzednio wydanych obrazów IBM Cloud Pak for Integration .

O tym zadaniu

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

Należy zauważyć, że pierwsze cztery kroki w tej czynności są takie same, jak kroki wykonywane podczas wykonywania czynności [“Instalowanie produktu IBM MQ Operator 2.x w środowisku z luką powietrzną”](#) na stronie 112.

Procedura

1. Utwórz zmienne środowiskowe dla instalatora i spisu obrazów.

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

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

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

2. Pobierz instalator IBM MQ i spis obrazów.

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

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

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

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

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

4. Utwórz kopię lustrzaną obrazów i skonfiguruj klaster.

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

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

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

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

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

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

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

```
cloudctl case launch \  
  --case $HOME/offline/CASE_ARCHIVE \  
  --inventory CASE_INVENTORY \  
  --action configure-creds-airgap \  
  --namespace CASE_NAMESPACE \  
  --registry CASE_REGISTRY \  
  --username CASE_USERNAME \  
  --password CASE_PASSWORD
```

```
--args "--registry registry --user registry_userid --pass registry_password --inputDir $HOME/offline"
```

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

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

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

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

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

- c. Skonfiguruj sekret uwierzytelniania dla lokalnego rejestru Docker.

Uwaga: Ten krok trzeba wykonać tylko jeden raz.

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

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

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

- i) Sprawdź, czy wymagany jest restart węzła.

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

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

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

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

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

```
oc get imageContentSourcePolicy
```

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

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

g. Sprawdź status węzła klastra.

```
oc get nodes
```

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

h. Utwórz kopię lustrzaną obrazów w rejestrze lokalnym.

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

5. Zaktualizuj źródło katalogu.

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

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

Co dalej

Aby zakończyć aktualizację IBM Cloud Pak for Integration, może być konieczne powrót do dokumentacji IBM Cloud Pak for Integration.

W przeciwnym razie można przystąpić do aktualizacji programu IBM MQ Operator i menedżera kolejek, wykonując jedną z następujących czynności:

- [“Aktualizowanie IBM MQ Operator za pomocą programu Red Hat OpenShift” na stronie 132](#)
- [“Aktualizowanie menedżera kolejek systemu IBM MQ przy użyciu programu Red Hat OpenShift” na stronie 134](#)

Aktualizowanie IBM MQ Operator za pomocą programu Red Hat OpenShift

IBM MQ Operator można zaktualizować za pomocą konsoli WWW Red Hat OpenShift lub interfejsu CLI.

Procedura

Aby zaktualizować IBM MQ Operator za pomocą Red Hat OpenShift, wykonaj jedną z następujących czynności:

- [“Aktualizowanie produktu IBM MQ Operator przy użyciu konsoli WWW Red Hat OpenShift” na stronie 133](#)
- [“Aktualizowanie produktu IBM MQ Operator przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift” na stronie 133](#)

WWW Red Hat OpenShift

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

Zanim rozpoczniesz

Uwaga: Najnowszą wersją programu IBM MQ Operator w systemie CD jest 3.1.3. Najnowszą wersją programu IBM MQ Operator w systemie LTS jest 2.0.23. Najnowsze uwagi do wydania systemu IBM MQ Operator można znaleźć pod adresem [“Historia wydań operatora IBM MQ Operator”](#) na stronie 36.

Zaloguj się do konsoli WWW klastra produktu Red Hat OpenShift.

Procedura

1. Zapoznaj się z sekcją [“Wersja zgodna z operatorem IBM MQ Operator”](#) na stronie 12, aby określić, który kanał operatora wybrać do aktualizacji.
2. Zastosuj najnowsze źródło katalogu.

Jeśli używane jest źródło katalogu specyficzne dla produktu IBM MQ (wszystkie instalacje Air-gap), a nie produkt `ibm-operator-catalog`, należy zastosować źródło katalogu dla używanej wersji produktu IBM MQ .

Należy postępować zgodnie z instrukcjami zawartymi w sekcji [Dodawanie źródeł katalogu do klastra](#).

Uwaga: Jeśli wykonano już krok instalacji operatora dla [obrazów kopii lustrzanych](#) (tylko dla luk powietrznych), należy tylko wykonać krok, który dotyczy źródła katalogu. Na przykład:

```
oc apply -f ~/.ibm-pak/data/mirror/${OPERATOR_PACKAGE_NAME}/${OPERATOR_VERSION}/catalog-sources.yaml
```

3. Zaktualizuj IBM MQ Operator. Nowe główne/podrzędne wersje operatora IBM MQ Operator są dostarczane za pośrednictwem nowych kanałów subskrypcji. Aby zaktualizować operator do nowej wersji głównej/podrzędnej, należy zaktualizować wybrany kanał w subskrypcji operatora IBM MQ Operator.
 - a) Na panelu nawigacyjnym kliknij opcję **Operators > Installed Operators**.
Zostaną wyświetlone wszystkie operatory zainstalowane w określonym projekcie.
 - b) Wybierz pozycję **IBM MQ Operator**
 - c) Przejdź do karty **Subskrypcja**
 - d) Kliknij opcję **Channel**
Zostanie wyświetlone okno **Change Subscription Update Channel**.
 - e) Wybierz żądany kanał i kliknij opcję **Save**.
Operator zostanie zaktualizowany do najnowszej wersji dostępnej dla nowego kanału. Patrz sekcja [“Wersja zgodna z operatorem IBM MQ Operator”](#) na stronie 12.

interfejsu wiersza komend produktu Red Hat OpenShift

Produkt IBM MQ Operator można zaktualizować z poziomu wiersza komend.

Zanim rozpoczniesz

Uwaga: Najnowszą wersją programu IBM MQ Operator w systemie CD jest 3.1.3. Najnowszą wersją programu IBM MQ Operator w systemie LTS jest 2.0.23. Najnowsze uwagi do wydania systemu IBM MQ Operator można znaleźć pod adresem [“Historia wydań operatora IBM MQ Operator”](#) na stronie 36.

Zaloguj się do klastra przy użyciu komendy **oc login**.

Zanim będzie można zaktualizować IBM MQ Operator w środowisku z luką powietrzną, należy utworzyć kopię lustrzaną najnowszych obrazów IBM Cloud Pak for Integration . W przypadku aktualizacji do

wersji IBM MQ Operator 3.0 lub nowszej opcja [Migrowanie do bieżącego CD kanału IBM MQ Operator](#) obejmuje kroki specyficzne dla szczeliny powietrznej. Informacje o aktualizacji do wcześniejszych wersji operatora produktu IBM MQ zawiera sekcja **Deprecated** [Przygotowanie do aktualizacji do najnowszej wersji produktu IBM MQ 2.x Operator lub menedżer kolejek w środowisku z luką powietrzną](#).

Procedura

1. Zapoznaj się z sekcją [“Wersja zgodna z operatorem IBM MQ Operator”](#) na stronie 12, aby określić, który kanał operatora wybrać do aktualizacji.
2. Zastosuj najnowsze źródło katalogu.

Jeśli używane jest źródło katalogu specyficzne dla produktu IBM MQ (wszystkie instalacje Air-gap), a nie produkt `ibm-operator-catalog`, należy zastosować źródło katalogu dla używanej wersji produktu IBM MQ.

Należy postępować zgodnie z instrukcjami zawartymi w sekcji [Dodawanie źródeł katalogu do klastra](#).

Uwaga: Jeśli wykonano już krok instalacji operatora dla obrazów kopii lustrzanych (tylko dla luk powietrznych), należy tylko wykonać krok, który dotyczy źródła katalogu. Na przykład:

```
oc apply -f ~/.ibm-pak/data/mirror/${OPERATOR_PACKAGE_NAME}/${OPERATOR_VERSION}/catalog-sources.yaml
```

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

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

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

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

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

Aktualizowanie menedżera kolejek systemu IBM MQ przy użyciu programu Red Hat OpenShift

Zanim rozpoczniesz

W ramach procesu aktualizowania menedżerów kolejek systemu IBM MQ użytkownik mógł zostać wysłany do tego tematu z dokumentacji produktu IBM Cloud Pak for Integration .

Procedura

Aby zaktualizować menedżer kolejek produktu IBM MQ za pomocą programu Red Hat OpenShift, wykonaj jedną z następujących czynności:

- [“Aktualizowanie menedżera kolejek produktu IBM MQ przy użyciu konsoli WWW Red Hat OpenShift”](#) na stronie 135
- [“Wdrażanie menedżera kolejek produktu IBM MQ przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift”](#) na stronie 135
- [“Aktualizowanie menedżera kolejek produktu IBM MQ w produkcie Red Hat OpenShift przy użyciu interfejsu użytkownika platformy”](#) na stronie 136

Co dalej

Aby zakończyć aktualizację programu IBM Cloud Pak for Integration , może być konieczne powrót do dokumentacji IBM Cloud Pak for Integration .

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

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

Zanim rozpoczniesz

Uwaga: The latest CD version of the IBM MQ queue manager is 9.3.5.1-r2. The latest LTS version of the IBM MQ queue manager is 9.3.0.17-r3. Najnowsze uwagi do wydania menedżera kolejek systemu IBM MQ zawiera sekcja [“Historia wersji dla obrazów kontenera menedżera kolejek do użycia z produktem IBM MQ Operator” na stronie 60.](#)

- Zaloguj się do konsoli WWW klastra produktu Red Hat OpenShift.
- Upewnij się, że IBM MQ Operator używa żądanego kanału aktualizacji. Patrz [“Aktualizowanie IBM MQ Operator za pomocą programu Red Hat OpenShift” na stronie 132.](#)

Przed zaktualizowaniem menedżera kolejek w środowisku z luką powietrzną należy utworzyć kopię lustrzaną najnowszych obrazów produktu IBM Cloud Pak for Integration . W przypadku aktualizacji do wersji IBM MQ Operator 3.0 lub nowszej opcja [Migrowanie do bieżącego CD kanału IBM MQ Operator](#) obejmuje kroki specyficzne dla szczeliny powietrznej. Informacje o aktualizacji do wcześniejszych wersji

operatora produktu IBM MQ zawiera sekcja  [Przygotowanie do aktualizacji do najnowszej wersji produktu IBM MQ 2.x Operator lub menedżer kolejek w środowisku z luką powietrzną.](#)

Procedura

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

Informacje na temat odwzorowywania wersji produktu IBM MQ Operator i obrazów kontenerów menedżera kolejek IBM MQ zawiera sekcja [“Historia wersji dla obrazów kontenera menedżera kolejek do użycia z produktem IBM MQ Operator” na stronie 60 .](#)

7. Zapisz zmieniony kod YAML menedżera kolejek.

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

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

Zanim rozpocznesz

Uwaga: The latest CD version of the IBM MQ queue manager is 9.3.5.1-r2. The latest LTS version of the IBM MQ queue manager is 9.3.0.17-r3. Najnowsze uwagi do wydania menedżera kolejek systemu IBM MQ zawiera sekcja “[Historia wersji dla obrazów kontenera menedżera kolejek do użycia z produktem IBM MQ Operator](#)” na stronie 60.

Aby wykonać te kroki, należy być administratorem klastra.

- Zaloguj się do interfejsu wiersza komend produktu Red Hat OpenShift przy użyciu programu `oc login`.
- Upewnij się, że IBM MQ Operator używa żądanego kanału aktualizacji. Patrz “[Aktualizowanie operatora IBM MQ Operator i menedżerów kolejek](#)” na stronie 123.

Przed zaktualizowaniem menedżera kolejek w środowisku z luką powietrzną należy utworzyć kopię lustrzaną najnowszych obrazów produktu IBM Cloud Pak for Integration . W przypadku aktualizacji do wersji IBM MQ Operator 3.0 lub nowszej opcja Migrowanie do bieżącego CD kanału IBM MQ Operator obejmuje kroki specyficzne dla szczeliny powietrznej. Informacje o aktualizacji do wcześniejszych wersji operatora produktu IBM MQ zawiera sekcja **Deprecated** [Przygotowanie do aktualizacji do najnowszej wersji produktu IBM MQ 2.x Operator lub menedżer kolejek w środowisku z luką powietrzną](#).

Procedura

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

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

Informacje na temat odwzorowania kanałów na wersje “[Wersja zgodna z operatorem IBM MQ Operator](#)” na stronie 12 i wersje menedżera kolejek IBM MQ Operator można znaleźć w sekcji IBM MQ.

Użyj następującej komendy:

```
oc edit queuemanager my_qmgr
```

gdzie `mój_qmgr` to nazwa zasobu QueueManager, który ma zostać zaktualizowany.

CP4I Aktualizowanie menedżera kolejek produktu IBM MQ w produkcie Red Hat OpenShift przy użyciu interfejsu użytkownika platformy

Menedżer kolejek produktu IBM MQ wdrożony za pomocą rozwiązania IBM MQ Operator może zostać zaktualizowany w produkcie Red Hat OpenShift przy użyciu rozwiązania IBM Cloud Pak for Integration Platform UI (previously the Platform Navigator).

Zanim rozpocznesz

Uwaga: The latest CD version of the IBM MQ queue manager is 9.3.5.1-r2. The latest LTS version of the IBM MQ queue manager is 9.3.0.17-r3. Najnowsze uwagi do wydania menedżera kolejek systemu IBM MQ zawiera sekcja “[Historia wersji dla obrazów kontenera menedżera kolejek do użycia z produktem IBM MQ Operator](#)” na stronie 60.

- Zaloguj się do programu IBM Cloud Pak for Integration Platform UI w przestrzeni nazw zawierającej menedżer kolejek, który ma zostać zaktualizowany.
- Upewnij się, że IBM MQ Operator używa żądanego kanału aktualizacji. Patrz “[Aktualizowanie operatora IBM MQ Operator i menedżerów kolejek](#)” na stronie 123.

Przed zaktualizowaniem menedżera kolejek w środowisku z luką powietrzną należy utworzyć kopię lustrzaną najnowszych obrazów produktu IBM Cloud Pak for Integration . W przypadku aktualizacji do wersji IBM MQ Operator 3.0 lub nowszej opcja Migrowanie do bieżącego CD kanału IBM MQ Operator obejmuje kroki specyficzne dla szczeliny powietrznej. Informacje o aktualizacji do wcześniejszych wersji

operatora produktu IBM MQ zawiera sekcja **Deprecated** [Przygotowanie do aktualizacji do najnowszej wersji produktu IBM MQ 2.x Operator lub menedżer kolejek w środowisku z luką powietrzną.](#)

Procedura

1. Na stronie głównej programu IBM Cloud Pak for Integration Platform UI (previously the Platform Navigator) kliknij kartę **Runtimes**.
2. Menedżery kolejek z dostępnymi aktualizacjami są oznaczone niebieskimi ikonami **i** obok pozycji **Version**. Kliknij ikonę **i**, aby wyświetlić opcję **New version available**.
3. Kliknij trzy kropki po prawej stronie menedżera kolejek, który ma zostać zaktualizowany, a następnie kliknij opcję **Change version**.
4. W sekcji **Select a new channel or version** wybierz wymaganą wersję aktualizacji.
5. Kliknij opcję **Change version**.

Wyniki

Menedżer kolejek zostanie zaktualizowany do nowej wersji.

OpenShift Konfigurowanie menedżerów kolejek za pomocą programu IBM MQ Operator

Przykłady konfiguracji; konfigurowanie wysokiej dostępności; nawiązywanie połączenia spoza klastra OpenShift ; integrowanie z panelem kontrolnym CP4i ; integrowanie z funkcją śledzenia Instana; budowanie obrazu z niestandardowymi plikami MQSC i INI; dodawanie niestandardowych adnotacji i etykiet.

O tym zadaniu

Procedura

- [“Przykłady konfigurowania menedżera kolejek” na stronie 137.](#)
- [“Konfigurowanie wysokiej dostępności na potrzeby menedżerów kolejek przy użyciu rozwiązania IBM MQ Operator” na stronie 147.](#)
- [“Konfigurowanie trasy w celu nawiązania połączenia z menedżerem kolejek spoza klastra produktu Red Hat OpenShift” na stronie 159.](#)
- [“Integracja z panelem kontrolnym IBM Cloud Pak for Integration Operations Dashboard” na stronie 161.](#)
- [“Integrowanie produktu IBM MQ z funkcją śledzenia produktu IBM Instana” na stronie 162.](#)
- [“Budowanie obrazu z niestandardowymi plikami MQSC i INI przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift” na stronie 169.](#)
- [“Dodawanie niestandardowych adnotacji i etykiet do zasobów menedżera kolejek” na stronie 171.](#)
- [“Wyłączanie kontroli elementu webhook w czasie wykonywania” na stronie 172.](#)
- [“Wyłączanie aktualizacji wartości domyślnych w specyfikacji menedżera kolejek” na stronie 172.](#)

OpenShift **CP4I** Przykłady konfigurowania menedżera kolejek

Menedżer kolejek można skonfigurować, modyfikując zawartość niestandardowego zasobu QueueManager.

O tym zadaniu

Poniższe przykłady ilustrują konfigurowanie menedżera kolejek przy użyciu pliku YAML zasobu QueueManager.

Procedura

- [“Przykład: Udostępnianie plików MQSC i INI” na stronie 138](#)
- [“Przykład: konfigurowanie menedżera kolejek z wzajemnym uwierzytelnianiem TLS” na stronie 141](#)

Przykład: Udostępnianie plików MQSC i INI

Ten przykład tworzy mapę Kubernetes ConfigMap, która zawiera dwa pliki MQSC i jeden plik INI. Następnie wdrażany jest menedżer kolejek, który przetwarza te pliki MQSC i INI.

O tym zadaniu

Pliki [MQSC](#) i [INI](#) mogą być udostępniane na etapie wdrażania menedżera kolejek. Dane MQSC i INI muszą być zdefiniowane w jednym lub wielu zasobach Kubernetes [ConfigMap](#) i [Secret](#). Te zasoby muszą zostać utworzone w przestrzeni nazw (projekcie), w której zostanie wdrożony menedżer kolejek.

Uwaga: Gdy plik MQSC lub INI zawiera dane poufne, należy stosować klucz tajny Kubernetes.

Przykład

Poniższy przykład tworzy mapę Kubernetes ConfigMap, która zawiera dwa pliki MQSC i jeden plik INI. Następnie wdrażany jest menedżer kolejek, który przetwarza te pliki MQSC i INI.

Przykładowa mapa ConfigMap — zastosuj następujący kod YAML w klastrze:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: mqsc-ini-example
data:
  example1.mqsc: |
    DEFINE QLOCAL('DEV.QUEUE.1') REPLACE
    DEFINE QLOCAL('DEV.QUEUE.2') REPLACE
  example2.mqsc: |
    DEFINE QLOCAL('DEV.DEAD.LETTER.QUEUE') REPLACE
  example.ini: |
    Channels:
      MQIBindType=FASTPATH
```

Przykład QueueManager -wdrażanie menedżera kolejek z następującą konfiguracją przy użyciu wiersza komend lub konsoli WWW programu Red Hat OpenShift Container Platform :

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mqsc-ini-qm
spec:
  version: 9.3.5.1-r2
  license:
    accept: false
    license: L-AMRD-XH6P3Q
    use: Production
  web:
    enabled: true
  queueManager:
    name: "MQSCINI"
  mqsc:
    - configMap:
        name: mqsc-ini-example
        items:
          - example1.mqsc
          - example2.mqsc
  ini:
    - configMap:
        name: mqsc-ini-example
        items:
          - example.ini
  storage:
    queueManager:
      type: ephemeral
```

Ważne: Jeśli akceptujesz umowę licencyjną produktu IBM MQ Advanced, zmień wartość `accept` : `false` na `accept` : `true`. Szczegółowe informacje na temat licencji można znaleźć w sekcji [Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1](#).

Informacje dodatkowe:

- Menedżer kolejek można skonfigurować w taki sposób, aby używał pojedynczego Kubernetes ConfigMap lub klucza tajnego (jak pokazano w tym przykładzie) lub wielu ConfigMaps i danych niejawnych.
- Można wybrać użycie wszystkich danych MQSC i INI z zasobu Kubernetes ConfigMap lub Secret (tak jak w tym przykładzie) lub skonfigurować każdy menedżer kolejek tak, aby używał tylko podzbioru dostępnych plików.
- Pliki MQSC i INI są przetwarzane w porządku alfabetycznym na podstawie ich klucza. Dlatego `example1.mqsc` będzie zawsze przetwarzany przed `example2.mqsc` niezależnie od kolejności ich występowania w konfiguracji menedżera kolejek.
- Jeśli więcej niż jeden plik MQSC lub INI ma ten sam klucz w wielu zasobach Kubernetes ConfigMap lub Secret, ten zestaw plików jest przetwarzany na podstawie kolejności, w jakiej pliki są zdefiniowane w konfiguracji menedżera kolejek.
- Jeśli zasobnik menedżera kolejek jest uruchomiony, wszystkie zmiany w Kubernetes ConfigMap nie są pobierane, ponieważ IBM MQ Operator nie wie o tej zmianie. W przypadku wprowadzenia zmian w ConfigMap, na przykład zmian w komendach MQSC lub w plikach INI, należy ręcznie zrestartować menedżery kolejek, aby wprowadzić te zmiany. W przypadku menedżerów kolejek z jedną instancją usuń zasobnik, aby wyzwolić wymagany restart. W przypadku rodzimych wdrożeń wysokiej dostępności należy najpierw zrestartować zasobniki rezerwowe, usuwając je. Gdy ponownie znajdują się w stanie działania, usuń aktywny zasobnik, aby go zrestartować. Taka kolejność restartowania zapewnia minimalny czas przestoju menedżera kolejek.




Tworzenie samopodpisanej infrastruktury PKI przy użyciu produktu OpenSSL

Produkt IBM MQ umożliwia użycie wzajemnego protokołu TLS na potrzeby uwierzytelniania, w którym oba końce połączenia dostarczają certyfikatu, a szczegóły w certyfikacie są używane do ustanowienia tożsamości z menedżerem kolejek. W tym temacie przedstawiono sposób tworzenia przykładowej infrastruktury klucza publicznego (PKI) przy użyciu narzędzia wiersza komend OpenSSL, tworzenia dwóch certyfikatów, które mogą być używane w innych przykładach.

Zanim rozpocznieś

Upewnij się, że narzędzie wiersza komend OpenSSL jest zainstalowane.

Zainstaluj produkt IBM MQ client, a następnie dodaj `samp/bin` i `bin` do elementu `PATH`. Potrzebna jest komenda `runmqicred`, którą można zainstalować w ramach programu IBM MQ client w następujący sposób:

-   W systemach Windows i Linux: zainstaluj klienta redystrybucyjnego IBM MQ dla używanego systemu operacyjnego z serwisu <https://ibm.biz/mq93redistclients>
-  W systemie Mac: pobierz i skonfiguruj IBM MQ MacOS Toolkit: <https://developer.ibm.com/tutorials/mq-macos-dev/>

O tym zadaniu

Ważne: Opisane tutaj przykłady nie są odpowiednie dla środowiska produkcyjnego i są przeznaczone wyłącznie jako przykłady do szybkiego działania. Zarządzanie certyfikatami jest złożonym tematem dla zaawansowanych użytkowników. W przypadku produkcji należy wziąć pod uwagę takie kwestie, jak rotacja, cofanie, długość klucza, odtwarzanie po awarii i wiele innych.

Te kroki zostały przetestowane przy użyciu programu OpenSSL 3.1.4.

Procedura

1. Utwórz klucz prywatny, który będzie używany przez wewnętrzny ośrodek certyfikacji

```
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:4096 -out ca.key
```

Klucz prywatny dla wewnętrznego ośrodka certyfikacji jest tworzony w pliku o nazwie *ca.key*. Plik ten powinien być bezpieczny i tajny-będzie używany do podpisywania certyfikatów dla wewnętrznego ośrodka certyfikacji.

2. Wystawianie certyfikatu samopodpisanego dla wewnętrznego ośrodka certyfikacji

```
openssl req -x509 -new -nodes -key ca.key -sha512 -days 30 -subj "/CN=example-selfsigned-ca" -out ca.crt
```

Parametr `-days` określa liczbę dni, przez które certyfikat głównego ośrodka CA będzie ważny.

Certyfikat jest tworzony w pliku o nazwie *ca.crt*. Ten certyfikat zawiera informacje publiczne o wewnętrznym ośrodku certyfikacji i jest swobodnie współużytkowany.

3. Utwórz klucz prywatny i certyfikat dla menedżera kolejek

- a) Utwórz klucz prywatny i żądanie podpisania certyfikatu dla menedżera kolejek

```
openssl req -new -nodes -out example-qm.csr -newkey rsa:4096 -keyout example-qm.key -subj '/CN=example-qm'
```

Klucz prywatny jest tworzony w pliku o nazwie *example-qm.key*, a żądanie podpisania certyfikatu jest tworzony w pliku o nazwie *example-qm.csr*

- b) Podpisz klucz menedżera kolejek przy użyciu wewnętrznego ośrodka certyfikacji

```
openssl x509 -req -in example-qm.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out example-qm.crt -days 7 -sha512
```

Parametr `-days` określa liczbę dni, przez które certyfikat będzie ważny.

Podpisany certyfikat jest tworzony w pliku o nazwie *example-qm.crt*

- c) Utwórz klucz tajny Kubernetes z kluczem i certyfikatem menedżera kolejek

```
oc create secret generic example-qm-tls --type="kubernetes.io/tls" --from-file=tls.key=example-qm.key --from-file=tls.crt=example-qm.crt --from-file=ca.crt
```

Zostanie utworzony klucz tajny Kubernetes o nazwie *example-qm-tls*. Ten klucz tajny zawiera klucz prywatny dla menedżera kolejek, certyfikat publiczny i certyfikat ośrodka CA.

4. Utwórz klucz prywatny i certyfikat dla aplikacji

- a) Utwórz klucz prywatny i żądanie podpisania certyfikatu dla aplikacji

```
openssl req -new -nodes -out example-app1.csr -newkey rsa:4096 -keyout example-app1.key -subj '/CN=example-app1'
```

Klucz prywatny jest tworzony w pliku o nazwie *example-app1.key*, a żądanie podpisania certyfikatu jest tworzony w pliku o nazwie *example-app1.csr*

- b) Podpisz klucz menedżera kolejek przy użyciu wewnętrznego ośrodka certyfikacji

```
openssl x509 -req -in example-app1.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out example-app1.crt -days 7 -sha512
```

Parametr `-days` określa liczbę dni, przez które certyfikat będzie ważny.

Podpisany certyfikat jest tworzony w pliku o nazwie *example-app1.crt*

- c) Utwórz magazyn kluczy PKCS#12 z kluczem i certyfikatem aplikacji.

Produkt IBM MQ używa bazy danych kluczy, a nie pojedynczych plików kluczy. Skonteneryzowany menedżer kolejek utworzy bazę danych kluczy dla menedżera kolejek na podstawie klucza tajnego, ale w przypadku aplikacji klienckich należy ręcznie utworzyć bazę danych kluczy.

```
openssl pkcs12 -export -in "example-app1.crt" -name "example-app1" -certfile "ca.crt"
-inkey "example-app1.key" -out "example-app1.p12" -passout pass:<PASSWORD>
```

Zastąp parametr `<PASSWORD>` własnym hasłem.

Magazyn kluczy jest tworzony w pliku o nazwie `example-app1.p12`. Klucz i certyfikat aplikacji są przechowywane wewnątrz, z etykietą lub nazwą przyjazną `example-app1`, a także z certyfikatem ośrodka CA.

- d) Jeśli używany jest system Apple Mac arm64, należy skonfigurować dodatkowy plik łączący certyfikaty aplikacji i ośrodka CA.

Na przykład:

```
cat example-app1.crt ca.crt > example-app1-chain.crt
```

Zadania pokrewne

“Przykład: konfigurowanie menedżera kolejek z wzajemnym uwierzytelnianiem TLS” na stronie 141

W tym przykładzie menedżer kolejek jest wdrażany w OpenShift Container Platform przy użyciu IBM MQ Operator. Wzajemny protokół TLS jest używany do uwierzytelniania w celu odwzorowania certyfikatu TLS na tożsamość w menedżerze kolejek.

“Przykład: konfigurowanie rodzimej wysokiej dostępności przy użyciu IBM MQ Operator” na stronie 150

W tym przykładzie menedżer kolejek jest wdrażany przy użyciu rodzimej funkcji wysokiej dostępności w programie OpenShift Container Platform za pomocą programu IBM MQ Operator. Wzajemny protokół TLS jest używany do uwierzytelniania w celu odwzorowania certyfikatu TLS na tożsamość w menedżerze kolejek.

“Konfigurowanie menedżera kolejek z wieloma instancjami przy użyciu programu IBM MQ Operator” na stronie 156

W tym przykładzie wdrażany jest menedżer kolejek z wieloma instancjami przy użyciu programu OpenShift Container Platform i programu IBM MQ Operator. Wzajemny protokół TLS jest używany do uwierzytelniania w celu odwzorowania certyfikatu TLS na tożsamość w menedżerze kolejek.

Przykład: konfigurowanie menedżera kolejek z wzajemnym uwierzytelnianiem TLS

W tym przykładzie menedżer kolejek jest wdrażany w OpenShift Container Platform przy użyciu IBM MQ Operator. Wzajemny protokół TLS jest używany do uwierzytelniania w celu odwzorowania certyfikatu TLS na tożsamość w menedżerze kolejek.

Zanim rozpoczniesz

Aby zrealizować ten przykład, należy najpierw spełnić następujące wymagania wstępne:

- Utwórz projekt/przestrzeń nazw OpenShift Container Platform (OCP) dla tego przykładu.
- W wierszu komend zaloguj się do klastra OCP i przełącz się na powyższą przestrzeń nazw.
- Upewnij się, że operator IBM MQ Operator jest zainstalowany i dostępny w powyższej przestrzeni nazw.

O tym zadaniu

W tym przykładzie udostępniono niestandardowy zasób YAML definiujący menedżer kolejek, który ma zostać wdrożony w środowisku OpenShift Container Platform. Opisano tu również szczegółowe kroki wymagane do wdrożenia menedżera kolejek z włączonym protokołem TLS.

Procedura

1. Utwórz parę certyfikatów zgodnie z opisem w sekcji “Tworzenie samopodpisanej infrastruktury PKI przy użyciu produktu OpenSSL” na stronie 139.

2. Utwórz mapę konfiguracji zawierającą komendy MQSC i plik INI

Utwórz obiekt Kubernetes ConfigMap zawierający komendy MQSC służące do tworzenia nowej kolejki i kanału SVRCONN oraz do dodawania rekordu uwierzytelniania kanału, który umożliwia dostęp do kanału.

Upewnij się, że jesteś w przestrzeni nazw utworzonej wcześniej (patrz sekcja [Zanim rozpoczniesz](#)), a następnie wprowadź następujący kod YAML w konsoli WWW OCP lub użyj wiersza komend.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-tls-configmap
data:
  example-tls.mqsc: |
    DEFINE CHANNEL('MTLS.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
    SSLCIPH('ANY_TLS13_OR_HIGHER') REPLACE
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
    ACTION(REPLACE)
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=example-app1') USERSRC(MAP)
    MCAUSER('app1') ACTION(REPLACE)
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(CONNECT,INQ)
    DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
    SET AUTHREC PROFILE('EXAMPLE.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
    AUTHADD(BROWSE,PUT,GET,INQ)
  example-tls.ini: |
    Service:
      Name=AuthorizationService
      EntryPoints=14
      SecurityPolicy=UserExternal
```

Komenda MQSC definiuje kanał o nazwie *MTLS.SVRCONN* i kolejkę o nazwie *EXAMPLE.QUEUE*. Kanał jest skonfigurowany w taki sposób, aby zezwalał tylko na dostęp do klientów, którzy prezentują certyfikat z nazwą zwykłą ("common name") o nazwie *example-app1*. Jest to nazwa zwykła używana w jednym z certyfikatów utworzonych w kroku "1" na stronie 141. Połączenia w tym kanale o tej nazwie zwykłej są odwzorowywane na identyfikator użytkownika *app1*, który jest autoryzowany do nawiązywania połączenia z menedżerem kolejek i uzyskiwania dostępu do przykładowej kolejki. Plik INI włącza strategię bezpieczeństwa, która oznacza, że ID użytkownika *app1* nie musi istnieć w zewnętrznym rejestrze użytkowników-istnieje tylko jako nazwa w tej konfiguracji.

3. Wdrażanie menedżera kolejek

Utwórz nowy menedżer kolejek przy użyciu następującego zasobu niestandardowego YAML. Upewnij się, że znajdujesz się w przestrzeni nazw utworzonej przed rozpoczęciem tego zadania, a następnie wprowadź następujący kod YAML w konsoli WWW OCP lub użyj wiersza komend. Sprawdź, czy wskazano poprawną licencję, a następnie zaakceptuj licencję, zmieniając wartość *false* na *true*.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-AMRD-XH6P3Q
    use: Production
  queueManager:
    name: EXAMPLEQM
  mqsc:
    - configMap:
        name: example-tls-configmap
        items:
          - example-tls.mqsc
  ini:
    - configMap:
        name: example-tls-configmap
        items:
          - example-tls.ini
  storage:
    queueManager:
      type: ephemeral
  version: 9.3.5.1-r2
  pki:
    keys:
      - name: default
```

```
secret:  
  secretName: example-qm-tls  
  items:  
    - tls.key  
    - tls.crt  
    - ca.crt
```

Należy zauważyć, że w kroku “1” na stronie 141 utworzono klucz tajny *example-qm-tls*, a w kroku “2” na stronie 142 utworzono obiekt ConfigMap *example-tls-configmap*.

4. Upewnij się, że menedżer kolejek jest uruchomiony

Menedżer kolejek jest obecnie wdrażany. Przed kontynuowaniem upewnij się, że znajduje się on w stanie Running. Na przykład:

```
oc get qmgr exampleqm
```

5. Testowanie połączenia z menedżerem kolejek

Aby potwierdzić, że menedżer kolejek jest skonfigurowany pod kątem wzajemnej komunikacji TLS, wykonaj kroki opisane w sekcji “Testowanie wzajemnego połączenia TLS z menedżerem kolejek z komputera laptop” na stronie 143.

Wyniki

Gratulacje, pomyślnie wdrożono menedżer kolejek z włączoną obsługą protokołu TLS, który używa szczegółów podanych w certyfikacie TLS do uwierzytelniania w menedżerze kolejek i udostępniania tożsamości.

OpenShift CP4I Linux **Testowanie wzajemnego połączenia TLS z menedżerem kolejek z komputera laptop**

Po utworzeniu menedżera kolejek przy użyciu programu IBM MQ Operator można przetestować, czy działa, nawiązując połączenie z tym menedżerem kolejek oraz umieszczając i pobierając komunikat. W tym zadaniu opisano, w jaki sposób można nawiązać połączenie przy użyciu przykładowych programów IBM MQ, uruchamiając je na komputerze poza klastrem Kubernetes, takim jak komputer laptop.

Zanim rozpocznie

Aby zrealizować ten przykład, należy najpierw spełnić następujące wymagania wstępne:

- Zainstaluj operator IBM MQ client. Potrzebne są komendy **amqspu** i **amqsgetc**, które można zainstalować jako część IBM MQ client w następujący sposób:
 - **Windows** **Linux** W systemach Windows i Linux: zainstaluj klienta redystrybucyjnego IBM MQ dla używanego systemu operacyjnego z serwisu <https://ibm.biz/mq93redistclients>
 - **mac OS** W systemie Mac: pobierz i skonfiguruj IBM MQ MacOS Toolkit: <https://developer.ibm.com/tutorials/mq-macos-dev/>
- Upewnij się, że niezbędne pliki kluczy i certyfikatów zostały pobrane do katalogu na komputerze i że znasz hasło magazynu kluczy. Na przykład następujące pliki są tworzone w katalogu “Tworzenie samopodpisanej infrastruktury PKI przy użyciu produktu OpenSSL” na stronie 139:
 - `example-app1.p12`
 - `example-app1-chain.crt` (tylko jeśli używany jest system arm64 Apple Mac)
- Wdróż menedżera kolejek skonfigurowanego z protokołem TLS w klastrze OCP, na przykład wykonując kroki opisane w sekcji “Przykład: konfigurowanie menedżera kolejek z wzajemnym uwierzytelnianiem TLS” na stronie 141.

O tym zadaniu

W tym przykładzie użyto przykładowych programów IBM MQ działających na komputerze poza klastrem Kubernetes, takim jak komputer laptop, do nawiązania połączenia z QueueManager skonfigurowanym z protokołem TLS oraz do umieszczania i pobierania komunikatów.

Procedura

1. Upewnij się, że menedżer kolejek jest uruchomiony

Menedżer kolejek jest obecnie wdrażany. Przed kontynuowaniem upewnij się, że znajduje się on w stanie Running. Na przykład:

```
oc get qmgr exampleqm
```

2. Znajdowanie nazwy hosta menedżera kolejek

Użyj następującej komendy, aby znaleźć pełną nazwę hosta menedżera kolejek spoza klastra OCP przy użyciu trasy, która jest tworzona automatycznie: `exampleqm-ibm-mq-qm`:

```
oc get route exampleqm-ibm-mq-qm --template="{{.spec.host}}"
```

3. Tworzenie tabeli definicji kanału klienta (CCDT) IBM MQ

Utwórz plik o nazwie `ccdt.json` z następującą zawartością:

```
{
  "channel": [
    {
      "name": "MTLS.SVRCONN",
      "clientConnection": {
        "connection": [
          {
            "host": "<hostname from previous step>",
            "port": 443
          }
        ],
        "queueManager": "EXAMPLEQM"
      },
      "transmissionSecurity": {
        "cipherSpecification": "ANY_TLS13",
        "certificateLabel": "example-app1"
      },
      "type": "clientConnection"
    }
  ]
}
```

Połączenie używa portu 443, ponieważ jest to port, na którym nasłuchuje router Red Hat OpenShift Container Platform. Ruch zostanie przekazany do menedżera kolejek na porcie 1414.

Jeśli użyto innej nazwy kanału, należy ją również dostosować. Wzajemne przykłady TLS używają kanału o nazwie `MTLS.SVRCONN`

Więcej informacji na ten temat zawiera sekcja [Konfigurowanie tabeli definicji kanału klienta w formacie JSON](#).

4. Utwórz plik INI klienta, aby skonfigurować szczegóły połączenia

Utwórz plik o nazwie `mqclient.ini` w katalogu bieżącym. Ten plik zostanie odczytany przez **amqspu** i **amqsgetc**.

```
Channels:
  ChannelDefinitionDirectory=.
  ChannelDefinitionFile=ccdt.json
SSL:
  OutboundSNI=HOSTNAME
```



```
SSLKeyRepository=example-app1.p12
SSLKeyRepositoryPassword=<password you used when creating the p12 file>
```

Upewnij się, że hasło *SSLKeyRepository* zostało zmienione na hasło wybrane podczas tworzenia pliku PKCS#12. Istnieją inne sposoby ustawiania hasła magazynu kluczy, w tym użycie zaszyfowanego hasła. Więcej informacji na ten temat zawiera sekcja [Podawanie hasła repozytorium kluczy dla IBM MQ MQI client w systemie AIX, Linux, and Windows](#)

Należy zauważyć, że Red Hat OpenShift Container Platform Router używa SNI do kierowania żądań do menedżera kolejek IBM MQ. Atrybut *OutboundSNI= HOSTNAME* zapewnia, że klient IBM MQ zawiera informacje niezbędne do pracy routera z domyślną trasą skonfigurowaną przez IBM MQ Operator. Więcej informacji na ten temat zawiera sekcja ["Konfigurowanie trasy w celu nawiązania połączenia z menedżerem kolejek spoza klastra produktu Red Hat OpenShift"](#) na stronie 159.

5. Jeśli używany jest Apple Mac arm64, należy skonfigurować dodatkową zmienną środowiskową.

```
export MQSSLTRUSTSTORE=example-app1-chain.crt
```

Ten plik zawiera pełny łańcuch certyfikatów, w tym certyfikaty aplikacji i ośrodka CA.

6. Wstawianie komunikatów do kolejki

Uruchom następującą komendę:

```
/opt/mqm/samp/bin/amqspc EXAMPLE.QUEUE EXAMPLEQM
```

Jeśli nawiązanie połączenia z menedżerem kolejek powiedzie się, zostanie wygenerowana następująca odpowiedź:

```
target queue is EXAMPLE.QUEUE
```

Umieść kilka komunikatów w kolejce, wprowadzając tekst i naciskając za każdym razem klawisz **Enter**.

Aby zakończyć, naciśnij klawisz **Enter** dwa razy.

7. Pobieranie komunikatów z kolejki

Uruchom następującą komendę:

```
/opt/mqm/samp/bin/amqgetc EXAMPLE.QUEUE EXAMPLEQM
```

Komunikaty dodane w poprzednim kroku zostaną skonsumowane i wyświetlone. Po kilku sekundach komenda kończy działanie..

Wyniki

Gratulacje, pomyślnie przetestowano połączenie z menedżerem kolejek z włączoną obsługą protokołu TLS i pokazano, że można bezpiecznie umieścić komunikaty w menedżerze kolejek i pobrać je z klienta.

Przykład: dostosowywanie adnotacji usługi licencji

Produkt IBM MQ Operator automatycznie dodaje adnotacje produktu IBM License Service do wdrożonych zasobów. Są one monitorowane przez produkt IBM License Service. Generowane są raporty, które odpowiadają wymaganemu uprawnieniu.

O tym zadaniu

Adnotacje dodane przez produkt IBM MQ Operator są tymi, które są oczekiwane w sytuacjach standardowych. Są one oparte na wartościach licencji wybranych podczas wdrażania menedżera kolejek.

Przykład

Jeśli produkt **License** jest ustawiony na wartość `L-RJON-BZFQU2` (IBM Cloud Pak for Integration 2021.2.1), a produkt **Use** jest ustawiony na wartość `NonProduction`, stosowane są następujące adnotacje:

- `cloudpakId: c8b82d189e7545f0892db9ef2731b90d`

- cloudpakName: IBM Cloud Pak for Integration
- productChargedContainers: qmgr
- productCloudpakRatio: '4:1'
- productID: 21dfe9a0f00f444f888756d835334909
- productName: IBM MQ Advanced for Non-Production
- productMetric: VIRTUAL_PROCESSOR_CORE
- productVersion: 9.2.3.0

W ramach produktu IBM Cloud Pak for Integration wdrożenia produktu IBM App Connect Enterprise obejmują ograniczone uprawnienia dla produktu IBM MQ. W takich sytuacjach adnotacje te muszą zostać przestonięte, aby zapewnić przechwytywanie przez IBM License Service poprawnego użycia. Aby to zrobić, należy użyć podejścia opisanego w sekcji [“Dodawanie niestandardowych adnotacji i etykiet do zasobów menedżera kolejek”](#) na stronie 171.

Jeśli na przykład produkt IBM MQ zostanie wdrożony w ramach uprawnienia IBM App Connect Enterprise, należy użyć metody przedstawionej w następującym fragmencie kodu:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mq4ace
  namespace: cp4i
spec:
  annotations:
    productMetric: FREE
```

Istnieją dwa inne typowe powody, dla których adnotacje licencji mogą wymagać modyfikacji:

1. IBM MQ Advanced jest uwzględniany w uprawnieniu innego produktu IBM.
 - W takiej sytuacji należy użyć metody opisanej wcześniej dla produktu IBM App Connect Enterprise.
2. Produkt IBM MQ jest wdrażany na licencji IBM Cloud Pak for Integration.
 - Jeśli użytkownik posiada licencję na produkt IBM Cloud Pak for Integration, może podjąć decyzję o wdrożeniu menedżera kolejek w ramach współczynnika IBM MQ lub IBM MQ Advanced. W przypadku wdrożenia w oparciu o współczynnik IBM MQ należy się upewnić, że nie są używane zaawansowane możliwości, takie jak rodzima wysoka dostępność lub Advanced Message Security.
 - W tej sytuacji należy użyć następujących adnotacji do użycia w środowisku produkcyjnym:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mq4ace
  namespace: cp4i
spec:
  annotations:
    productID: c661609261d5471fb4ff8970a36bccea
    productCloudpakRatio: '4:1'
    productName: IBM MQ for Production
    productMetric: VIRTUAL_PROCESSOR_CORE
```

- W przypadku środowiska innego niż produkcyjne należy używać następujących adnotacji:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mq4ace
  namespace: cp4i
spec:
  annotations:
    productID: 151bec68564a4a47a14e6fa99266def
    productCloudpakRatio: '8:1'
    productName: IBM MQ for Non-Production
    productMetric: VIRTUAL_PROCESSOR_CORE
```

Konfigurowanie wysokiej dostępności na potrzeby menedżerów kolejek przy użyciu rozwiązania IBM MQ Operator

O tym zadaniu

Procedura

- [“Rodzima wysoka dostępność \(Native HA\)” na stronie 147.](#)
- [“Przykład: konfigurowanie rodzimej wysokiej dostępności przy użyciu IBM MQ Operator” na stronie 150.](#)
- [“Konfigurowanie menedżera kolejek z wieloma instancjami przy użyciu programu IBM MQ Operator” na stronie 156.](#)

Rodzima wysoka dostępność (Native HA)

Rodzima wysoka dostępność (Native HA) jest rodzimym (wbudowanym) rozwiązaniem zapewniającym wysoką dostępność środowisk IBM MQ, odpowiednim do użycia z blokową pamięcią masową w chmurze.

Konfiguracja rodzimej wysokiej dostępności obejmuje wysoce dostępny menedżer kolejek, w którym odtwarzalne dane produktu MQ (na przykład komunikaty) są replikowane w wielu zestawach pamięci masowej, co zapobiega utracie danych w wyniku awarii pamięci masowej. Menedżer kolejek składa się z wielu działających instancji – jedna jest liderem, a pozostałe są gotowe do szybkiego przejęcia obciążenia w przypadku awarii. Takie rozwiązanie maksymalizuje dostępność menedżera kolejek i jego komunikatów.

Konfiguracja rodzimej wysokiej dostępności składa się z trzech zasobników Kubernetes, z których każdy jest instancją menedżera kolejek. Jedna instancja jest aktywnym menedżerem kolejek, przetwarza komunikaty i tworzy wpisy w dzienniku odtwarzania. Za każdym razem, gdy tworzony jest wpis w dzienniku odtwarzania, aktywny menedżer kolejek wysyła dane do pozostałych dwóch instancji, zwanych replikami. Każda replika tworzy wpisy we własnym dzienniku odtwarzania, potwierdza dane, a następnie aktualizuje własne dane kolejki ze zreplikowanego dziennika odtwarzania. Jeśli w zasobniku aktywnego menedżera kolejek dojdzie do awarii, jedna z replik menedżera kolejek przejmuje rolę aktywną i od tej chwili zawiera bieżące dane do pracy.

Typ dziennika jest nazywany 'replikowanym dziennikiem'. Replikowany dziennik jest w zasadzie liniowym dziennikiem z włączonym automatycznym zarządzaniem dziennikiem i automatycznym obrazem nośnika. Patrz sekcja [Typy rejestrowania](#). Do zarządzania replikowanym dziennikiem używane są te same techniki, co do zarządzania dziennikiem liniowym.

Usługa Kubernetes jest używana do kierowania połączeń klientów TCP/IP do bieżącej aktywnej instancji, która jest identyfikowana jako jedyny zasobnik gotowy do obsługi ruchu sieciowego. Aplikacja kliencka nie musi wiedzieć o istnieniu różnych instancji.

Użycie trzech zasobników znacznie zmniejsza ryzyko rozsynchronizowania instancji. W systemie wysokiej dostępności z dwoma zasobnikami rozsynchronizowanie może wystąpić w razie przerwania połączenia między tymi zasobnikami. W przypadku braku połączenia w obu zasobnikach mogą jednocześnie działać menedżery kolejek zbierające różne dane. Po przywróceniu połączenia dostępne są dwie różne wersje danych i wymagana jest ręczna interwencja w celu podjęcia decyzji, który zestaw danych ma być zachowany, a który ma zostać odrzucony.

Mechanizm rodzimej wysokiej dostępności trzech zasobników i wymaga istnienia kworum, aby uniknąć rozsynchronizowania. Zasobniki, które mogą komunikować się z co najmniej jednym z pozostałych zasobników, tworzą kworum. Menedżer kolejek może stać się aktywną instancją tylko w zasobniku, który ma kworum. Menedżer kolejek nie może być aktywny w zasobniku, który nie ma połączenia z co najmniej jednym innym zasobnikiem, więc nigdy nie mogą istnieć jednocześnie dwie aktywne instancje:

- Jeśli jeden zasobnik ulegnie awarii, menedżer kolejek na jednym z pozostałych dwóch zasobników może przejąć jego funkcję. Jeśli dwa zasobniki ulegną awarii, menedżer kolejek nie może stać się aktywną instancją w pozostałym zasobniku, ponieważ nie ma kworum (ten zasobnik nie może stwierdzić, czy pozostałe dwa zasobniki są w stanie awarii, czy są nadal uruchomione, ale utraciły połączenie).

- Jeśli jeden zasobnik utraci połączenie, menedżer kolejek nie może stać się aktywny w tym zasobniku, ponieważ nie ma kworum. Menedżer kolejek na jednym z pozostałych dwóch zasobników może przejąć aktywność, ponieważ te zasobniki mają kworum. Jeśli wszystkie zasobniki utracą połączenie, menedżer kolejek nie może być aktywny na żadnym z zasobników, ponieważ żaden z zasobników nie ma kworum.

Jeśli aktywny zasobnik ulegnie awarii, a następnie odzyska sprawność, może ponownie dołączyć do grupy w roli repliki.

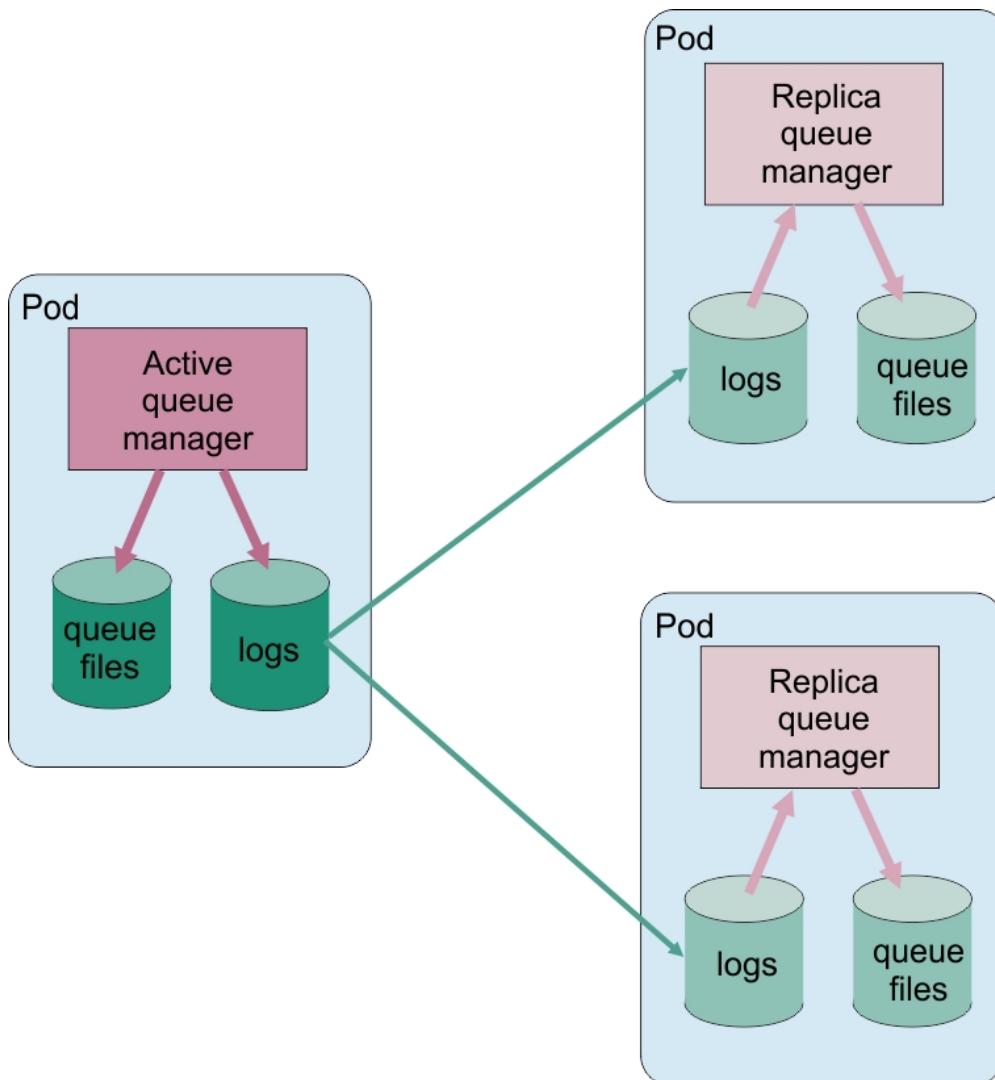
Ze względu na wydajność i niezawodność zaleca się użycie trwałej pamięci masowej RWO (ReadWriteOnce) w konfiguracji o rodzimej wysokiej dostępności. Woluminy RWO od dowolnego dostawcy pamięci masowej są obsługiwane, jeśli spełniają następujące warunki:

- Uzyskane od dostawcy blokowej pamięci masowej.
- Sformatowany jako ext4 lub XFS (co zapewnia zgodność z POSIX).
- Obsługuje dynamiczne udostępnianie woluminów i tryb "volumeBinding: WaitForFirstConsumer".

Następujący dostawcy są wyraźnie zakazani:

- NFS
- GlusterFS
- Inni dostawcy nie będący blokami.

Na poniższym rysunku przedstawiono typowe wdrożenie z trzema instancjami menedżera kolejek wdrożonymi w trzech kontenerach.



Rysunek 1. Przykład konfiguracji rodzimej wysokiej dostępności

OpenShift **MQ Adv.** Konfigurowanie rodzimej wysokiej dostępności przy użyciu operatora IBM MQ Operator

Rodzima wysoka dostępność jest konfigurowana przy użyciu interfejsu API QueueManager, a opcje zaawansowane są dostępne za pośrednictwem pliku INI.

Rodzima wysoka dostępność jest konfigurowana przy użyciu `.spec.queueManager.availability` interfejsu API QueueManager, na przykład:

```

apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: nativeha-example
spec:
  license:
    accept: false
    license: L-AMRD-XH6P3Q
    use: Production
  queueManager:
    availability:
      type: NativeHA
    version: 9.3.5.1-r2

```

Pole `.spec.queueManager.availability.type` musi być ustawione na wartość NativeHA.

W `.spec.queueManager.availability` można również skonfigurować klucz sekretny i szyfry TLS do wykorzystania między instancjami menedżera kolejek podczas replikacji. Jest to zalecane, a instrukcja krok po kroku jest dostępna w sekcji [“Przykład: konfigurowanie rodzimej wysokiej dostępności przy użyciu IBM MQ Operator”](#) na stronie 150.

Zadania pokrewne

[“Przykład: konfigurowanie rodzimej wysokiej dostępności przy użyciu IBM MQ Operator”](#) na stronie 150
W tym przykładzie menedżer kolejek jest wdrażany przy użyciu rodzimej funkcji wysokiej dostępności w programie OpenShift Container Platform za pomocą programu IBM MQ Operator. Wzajemny protokół TLS jest używany do uwierzytelniania w celu odwzorowania certyfikatu TLS na tożsamość w menedżerze kolejek.

 [Przykład: konfigurowanie rodzimej wysokiej dostępności przy użyciu IBM MQ Operator](#)

W tym przykładzie menedżer kolejek jest wdrażany przy użyciu rodzimej funkcji wysokiej dostępności w programie OpenShift Container Platform za pomocą programu IBM MQ Operator. Wzajemny protokół TLS jest używany do uwierzytelniania w celu odwzorowania certyfikatu TLS na tożsamość w menedżerze kolejek.

Zanim rozpoczniesz

Aby zrealizować ten przykład, należy najpierw spełnić następujące wymagania wstępne:

- Utwórz projekt/przestrzeń nazw OpenShift Container Platform (OCP) dla tego przykładu.
- W wierszu komend zaloguj się do klastra OCP i przełącz się na powyższą przestrzeń nazw.
- Upewnij się, że operator IBM MQ Operator jest zainstalowany i dostępny w powyższej przestrzeni nazw.

O tym zadaniu

W tym przykładzie udostępniono niestandardowy zasób YAML definiujący menedżer kolejek, który ma zostać wdrożony w środowisku OpenShift Container Platform. Opisano tu również szczegółowe kroki wymagane do wdrożenia menedżera kolejek z włączonym protokołem TLS.

Procedura

1. Utwórz parę certyfikatów zgodnie z opisem w sekcji [“Tworzenie samopodpisanej infrastruktury PKI przy użyciu produktu OpenSSL”](#) na stronie 139.
2. Utwórz mapę konfiguracji zawierającą komendy MQSC i plik INI

Utwórz obiekt Kubernetes ConfigMap zawierający komendy MQSC służące do tworzenia nowej kolejki i kanału SVRCONN oraz do dodawania rekordu uwierzytelniania kanału, który umożliwia dostęp do kanału.

Upewnij się, że jesteś w przestrzeni nazw utworzonej wcześniej (patrz sekcja [Zanim rozpoczniesz](#)), a następnie wprowadź następujący kod YAML w konsoli WWW OCP lub użyj wiersza komend.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-nativeha-configmap
data:
  example-tls.mqsc: |
    DEFINE CHANNEL('MTLS.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
    SSLCIPH('ANY_TLS13_OR_HIGHER') REPLACE
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
    ACTION(REPLACE)
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=example-app1') USERSRC(MAP)
  MCAUSER('app1') ACTION(REPLACE)
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(CONNECT,INQ)
    DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
    SET AUTHREC PROFILE('EXAMPLE.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
    AUTHADD(BROWSE,PUT,GET,INQ)
  example-tls.ini: |
    Service:
```

```
Name=AuthorizationService
EntryPoints=14
SecurityPolicy=UserExternal
```

Komenda MQSC definiuje kanał o nazwie *MTLS.SVRCONN* i kolejkę o nazwie *EXAMPLE.QUEUE*. Kanał jest skonfigurowany w taki sposób, aby zezwalał tylko na dostęp do klientów, którzy prezentują certyfikat z nazwą zwykłą ("common name") o nazwie *example-app1*. Jest to nazwa zwykła używana w jednym z certyfikatów utworzonych w kroku "1" na stronie 150. Połączenia w tym kanale o tej nazwie zwykłej są odwzorowywane na identyfikator użytkownika *app1*, który jest autoryzowany do nawiązywania połączenia z menedżerem kolejek i uzyskiwania dostępu do przykładowej kolejki. Plik INI włącza strategię bezpieczeństwa, która oznacza, że ID użytkownika *app1* nie musi istnieć w zewnętrznym rejestrze użytkowników-istnieje tylko jako nazwa w tej konfiguracji.

3. Wdrażanie menedżera kolejek

Utwórz nowy menedżer kolejek przy użyciu następującego zasobu niestandardowego YAML. Upewnij się, że znajdujesz się w przestrzeni nazw utworzonej przed rozpoczęciem tego zadania, a następnie wprowadź następujący kod YAML w konsoli WWW OCP lub użyj wiersza komend. Sprawdź, czy wskazano poprawną licencję, a następnie zaakceptuj licencję, zmieniając wartość *false* na *true*.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-AMRD-XH6P3Q
    use: Production
  queueManager:
    name: EXAMPLEQM
    availability:
      type: NativeHA
    tls:
      secretName: example-qm-tls
  mqsc:
    - configMap:
        name: example-nativeha-configmap
        items:
          - example-tls.mqsc
  ini:
    - configMap:
        name: example-nativeha-configmap
        items:
          - example-tls.ini
    storage:
      queueManager:
        type: persistent-claim
  version: 9.3.5.1-r2
  pki:
    keys:
      - name: default
        secret:
          secretName: example-qm-tls
          items:
            - tls.key
            - tls.crt
            - ca.crt
```

Należy zauważyć, że w kroku "1" na stronie 150 utworzono klucz tajny *example-qm-tls*, a w kroku "2" na stronie 150 utworzono ConfigMap *example-nativeha-configmap*.

Typ dostępności jest ustawiony na *NativeHA* i wybrano pamięć masową trwałą. Zostanie użyta domyślna klasa pamięci masowej skonfigurowana w klastrze Kubernetes. Jeśli klasa pamięci masowej nie jest skonfigurowana jako domyślna lub ma być używana inna klasa pamięci masowej, należy dodać `defaultClass: <storage_class_name>` w `spec.queueManager.storage`.

Trzy zasobniki w menedżerze kolejek o rodzimej wysokiej dostępności replikują dane w sieci. Ten odsyłacz nie jest domyślnie szyfrowany, ale w tym przykładzie do szyfrowania ruchu używany jest certyfikat menedżera kolejek. Można określić inny certyfikat dla dodatkowych zabezpieczeń. Rodzimy klucz tajny TLS wysokiej dostępności musi być kluczem tajnym TLS Kubernetes, który ma określoną strukturę (na przykład klucz prywatny musi mieć nazwę *tls.key*).

4. Upewnij się, że menedżer kolejek jest uruchomiony

Menedżer kolejek jest obecnie wdrażany. Przed kontynuowaniem upewnij się, że znajduje się on w stanie Running. Na przykład:

```
oc get qmgr exampleqm
```

5. Testowanie połączenia z menedżerem kolejek

Aby potwierdzić, że menedżer kolejek jest skonfigurowany i dostępny, wykonaj kroki opisane w sekcji [“Testowanie wzajemnego połączenia TLS z menedżerem kolejek z komputera laptop”](#) na stronie 143.

6. Wymuszanie awarii aktywnego zasobnika

Aby sprawdzić poprawność automatycznego odtwarzania menedżera kolejek, zasymuluj awarię zasobnika:

a) Wyświetlanie aktywnych i rezerwowych zasobników

Uruchom następującą komendę:

```
oc get pods --selector app.kubernetes.io/instance=exampleqm
```

Uwaga: w polu **READY** aktywny zasobnik zwraca wartość 1/1, natomiast zasobniki repliki zwracają wartość 0/1.

b) Usuwanie aktywnego zasobnika

Uruchom następującą komendę, podając pełną nazwę aktywnego zasobnika:

```
oc delete pod exampleqm-ibm-mq-<value>
```

c) Wyświetl ponownie status zasobnika

Uruchom następującą komendę:

```
oc get pods --selector app.kubernetes.io/instance=exampleqm
```

d) Wyświetlanie statusu menedżera kolejek

Uruchom następującą komendę, określając pełną nazwę jednego z pozostałych zasobników:

```
oc exec -t Pod -- dspmq -o nativeha -x -m EXAMPLEQM
```

Powinien zostać wyświetlony status, który wskazuje, że aktywna instancja została zmieniona, na przykład:

```
QMNAME(EXAMPLEQM) ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

e) Ponownie przetestuj połączenie z menedżerem kolejek

Aby potwierdzić, że menedżer kolejek został odzyskany, wykonaj kroki opisane w sekcji [“Testowanie wzajemnego połączenia TLS z menedżerem kolejek z komputera laptop”](#) na stronie 143.

Wyniki

Gratulacje, pomyślnie wdrożono menedżer kolejek z rodzimą wysoką dostępnością i wzajemnym uwierzytelnianiem TLS oraz sprawdzono, czy jest on automatycznie odzyskiwane w przypadku awarii aktywnego zasobnika.

MQ

W przypadku kontenerów IBM MQ można wyświetlić status rodzimych instancji wysokiej dostępności, uruchamiając komendę **dspmq** w jednym z działających zasobników.

O tym zadaniu

Aby wyświetlić status działania instancji menedżera kolejek, można użyć komendy **dspmq** w jednym z uruchomionych zasobników. Zwrócone informacje zależą od tego, czy instancja jest aktywna, czy jest repliką. Informacje dostarczone przez aktywną instancję są definitywne, a informacje z węzłów replik mogą być nieaktualne.

Można wykonać następujące czynności:

- Sprawdź, czy instancja menedżera kolejek w bieżącym węźle jest aktywna, czy jest repliką.
- Wyświetl status działania rodzimej wysokiej dostępności dla instancji w bieżącym węźle.
- Wyświetl status działania wszystkich trzech instancji w konfiguracji rodzimej wysokiej dostępności.

Następujące pola statusu są używane do raportowania statusu konfiguracji rodzimej wysokiej dostępności:

ROLE

Określa bieżącą rolę instancji i jest jedną z następujących wartości: Active, Replica lub Unknown.

INSTANCE

Nazwa podana dla tej instancji menedżera kolejek, gdy była ona tworzona przy użyciu opcji **-lr** komendy **crtmqm**.

INSYNC

Wskazuje, czy w razie potrzeby instancja jest w stanie przejąć rolę aktywnej instancji.

QUORUM

Raportuje status kworum w formacie *liczba zsynchronizowanych instancji/liczba skonfigurowanych instancji*.

REPLADDR

Adres replikacji instancji menedżera kolejek.

CONNECTV

Wskazuje, czy węzeł jest połączony z aktywną instancją.

BACKLOG

Określa zaległość instancji w kB.

CONNINST

Informuje, czy instancja o podanej nazwie jest połączona z tą instancją.

ALTDATE

Wskazuje datę ostatniej aktualizacji tych informacji (wartość pusta, jeśli nigdy nie zostały zaktualizowane).

ALTTIME

Wskazuje godzinę ostatniej aktualizacji tych informacji (wartość pusta, jeśli nigdy nie zostały zaktualizowane).

Procedura

- Znajdź zasobniki, które są częścią menedżera kolejek.

```
oc get pod --selector app.kubernetes.io/instance=nativeha-qm
```

- Uruchom dspmq w jednym z zasobników

```
oc exec -t Pod dspmq
```

```
oc rsh Pod
```

dla interaktywnej powłoki, w której można bezpośrednio uruchomić produkt dspmq.

- Aby określić, czy instancja menedżera kolejek działa jako aktywna instancja, czy jako replika:

```
oc exec -t Pod dspmq -o status -m QMgrName
```

Aktywna instancja menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          STATUS(Running)
```

Instancja-replika menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          STATUS(Replica)
```

Nieaktywna instancja zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          STATUS(Ended Immediately)
```

- Aby sprawdzić status działania rodzimej wysokiej dostępności instancji w podanym zasobniku:

```
oc exec -t Pod dspmq -o nativeha -m QMgrName
```

Aktywna instancja menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

Instancja-replika menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```

Nieaktywna menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- Aby określić status działania rodzimej wysokiej dostępności dla wszystkich instancji w konfiguracji o rodzimej wysokiej dostępności:

```
oc exec -t Pod dspmq -o nativeha -x -m QMgrName
```

Jeśli ta komenda zostanie wydana w węźle, w którym działa aktywna instancja menedżera kolejek BOB, może zostać wyświetlony następujący status:

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
  INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
  INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
  INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

Jeśli ta komenda zostanie wydana w węźle, w którym uruchomiona jest instancja-replika menedżera kolejek BOB, może zostać wyświetlony następujący status, który wskazuje, że jedna z replik ma zaległości:

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
  INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
  INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
  INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No)  BACKLOG(435)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

Jeśli ta komenda zostanie wydana w węźle, w którym uruchomiona jest nieaktywna instancja menedżera kolejek BOB, może zostać wyświetlony następujący status:

```
QMNAME(BOB)                ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
INSTANCE(inst1)            ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown)          CONNINST(No) ALTDATE() ALTTIME()
INSTANCE(inst2)            ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown)          CONNINST(No) ALTDATE() ALTTIME()
INSTANCE(inst3)            ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown)          CONNINST(No) ALTDATE() ALTTIME()
```

W przypadku wydania komendy w momencie, gdy instancje nadal negocjują, która będzie aktywna, a które będą replikami, zwrócony zostanie następujący status:

```
QMNAME(BOB)                STATUS(Negotiating)
```

Zadania pokrewne

[“Przykład: konfigurowanie rodzimej wysokiej dostępności przy użyciu IBM MQ Operator” na stronie 150](#)
W tym przykładzie menedżer kolejek jest wdrażany przy użyciu rodzimej funkcji wysokiej dostępności w programie OpenShift Container Platform za pomocą programu IBM MQ Operator. Wzajemny protokół TLS jest używany do uwierzytelniania w celu odwzorowania certyfikatu TLS na tożsamość w menedżerze kolejek.

Odsyłacze pokrewne

[Komenda dspmq \(wyświetl menedżery kolejek\)](#)

  *Strojenie zaawansowane rodzimej wysokiej dostępności*

Ustawienia zaawansowane służące do strojenia liczników czasu i interwałów. Nie powinno być konieczne używanie tych ustawień, chyba że wiadomo, iż wartości domyślne nie są zgodne z wymaganiami systemu.

Podstawowe opcje konfigurowania rodzimej wysokiej dostępności są obsługiwane za pomocą interfejsu API QueueManager, którego IBM MQ Operator używa do konfigurowania bazowych plików INI menedżera kolejek dla użytkownika. Istnieją bardziej zaawansowane opcje, które można konfigurować tylko za pomocą pliku INI w sekcji NativeHALocalInstancja. Więcej informacji na temat konfigurowania pliku INI zawiera sekcja [“Przykład: Udostępnianie plików MQSC i INI” na stronie 138](#).

HeartbeatInterval

Interwał pulsu definiuje, co ile milisekund aktywna instancja menedżera kolejek o rodzimej wysokiej dostępności wysyła puls sieciowy. Poprawny zakres wartości interwału pulsu wynosi od 500 (0,5 sekundy) do 60000 (1 minuta), a wartość spoza tego zakresu powoduje, że uruchomienie menedżera kolejek nie powiedzie się. Jeśli ten atrybut zostanie pominięty, zostanie użyta wartość domyślna 5000 (5 sekund). Każda instancja musi używać tego samego interwału pulsu.

HeartbeatTimeout

Limit czasu pulsu definiuje, jak długo instancja-replika menedżera kolejek o rodzimej wysokiej dostępności oczekuje przed podjęciem decyzji, że aktywna instancja nie odpowiada. Poprawny zakres limitu czasu interwału pulsu wynosi od 500 (0,5 sekundy) do 120000 (2 minuty). Wartość limitu czasu pulsu musi być większa niż lub równa wartości interwału pulsu.

Niepoprawna wartość powoduje, że uruchomienie menedżera kolejek nie powiedzie się. Jeśli ten atrybut zostanie pominięty, replika oczekuje na 2 x HeartbeatInterval przed rozpoczęciem procesu w celu wybrania nowej aktywnej instancji. Każda instancja musi używać tego samego limitu czasu pulsu.

RetryInterval

Interwał ponowienia definiuje, co ile milisekund menedżer kolejek o rodzimej wysokiej dostępności powinien ponawiać próbę nawiązania połączenia przez niesprawne łącze replikacji. Poprawny zakres interwału ponowienia wynosi od 500 (0,5 sekundy) do 120000 (2 minuty). Jeśli ten atrybut zostanie pominięty, replika oczekuje na 2 x HeartbeatInterval przed ponowną próbą utworzenia łącza replikacji, które zakończyło się niepowodzeniem.

Za pomocą komendy `endmqm` można zakończyć działanie aktywnego menedżera kolejek lub menedżera kolejek repliki, który jest częścią rodzimej grupy wysokiej dostępności.

Procedura

- Aby zakończyć aktywną instancję menedżera kolejek, należy zapoznać się z sekcją [Kończenie działania rodzimych menedżerów kolejek o wysokiej dostępności](#) w sekcji Konfigurowanie tej dokumentacji.

OpenShift → CP4I → MQ Adv. → Kubernetes **Konfigurowanie menedżera kolejek z wieloma instancjami przy użyciu programu IBM MQ Operator**

W tym przykładzie wdrażany jest menedżer kolejek z wieloma instancjami przy użyciu programu OpenShift Container Platform i programu IBM MQ Operator. Wzajemny protokół TLS jest używany do uwierzytelniania w celu odwzorowania certyfikatu TLS na tożsamość w menedżerze kolejek.

Zanim rozpoczniesz

Aby zrealizować ten przykład, należy najpierw spełnić następujące wymagania wstępne:

- Utwórz projekt/przestrzeń nazw OpenShift Container Platform (OCP) dla tego przykładu.
- W wierszu komend zaloguj się do klastra OCP i przełącz się na powyższą przestrzeń nazw.
- Upewnij się, że operator IBM MQ Operator jest zainstalowany i dostępny w powyższej przestrzeni nazw.

O tym zadaniu

W tym przykładzie udostępniono niestandardowy zasób YAML definiujący menedżer kolejek, który ma zostać wdrożony w środowisku OpenShift Container Platform. Opisano tu również szczegółowe kroki wymagane do wdrożenia menedżera kolejek z włączonym protokołem TLS.

Procedura

1. Określanie odpowiedniej klasy pamięci masowej

Dostęp do pamięci masowej w klastrze Kubernetes można uzyskać za pomocą wielu trybów dostępu do woluminu trwałego. Menedżer kolejek z wieloma instancjami tworzy wiele woluminów trwałych: jeden dla każdego menedżera kolejek i co najmniej jeden wolumin współużytkowany. Wolumin współużytkowany dla menedżera kolejek z wieloma instancjami musi używać klasy pamięci masowej `ReadWriteMany`. Domyślna klasa pamięci masowej w klastrze Kubernetes jest zwykle przeznaczona dla klasy pamięci masowej `ReadWriteOnce` (blokowa pamięć masowa). Na przykład, jeśli używany jest system Red Hat OpenShift Data Foundation, klasa pamięci masowej `ocs-storagecluster-cephfs` udostępnia odpowiedni współużytkowany system plików. Wybór systemu plików jest bardzo ważny, ponieważ nie wszystkie współużytkowane systemy plików obsługują blokowanie plików w ten sam sposób. Patrz sekcja [Planowanie obsługi systemu plików w środowisku wieloplatformowym](#) i [Instrukcja testowania dla systemów plików menedżera kolejek z wieloma instancjami produktu IBM MQ](#).

2. Utwórz parę certyfikatów zgodnie z opisem w sekcji ["Tworzenie samopodpisanej infrastruktury PKI przy użyciu produktu OpenSSL"](#) na stronie 139.

3. Utwórz mapę konfiguracji zawierającą komendy MQSC i plik INI

Utwórz obiekt Kubernetes ConfigMap zawierający komendy MQSC służące do tworzenia nowej kolejki i kanału SVRCONN oraz do dodawania rekordu uwierzytelniania kanału, który umożliwia dostęp do kanału.

Upewnij się, że jesteś w przestrzeni nazw utworzonej wcześniej (patrz sekcja [Zanim rozpoczniesz](#)), a następnie wprowadź następujący kod YAML w konsoli WWW OCP lub użyj wiersza komend.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-miqm-configmap
```

```

data:
  example-tls.mqsc: |
    DEFINE CHANNEL('MTLS.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
    SSLCIPH('ANY_TLS13_OR_HIGHER') REPLACE
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
    ACTION(REPLACE)
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=example-app1') USERSRC(MAP)
    MCAUSER('app1') ACTION(REPLACE)
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(CONNECT,INQ)
    DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
    SET AUTHREC PROFILE('EXAMPLE.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
    AUTHADD(BROWSE,PUT,GET,INQ)
  example-tls.ini: |
    Service:
      Name=AuthorizationService
      EntryPoints=14
      SecurityPolicy=UserExternal

```

Komenda MQSC definiuje kanał o nazwie *MTLS.SVRCONN* i kolejkę o nazwie *EXAMPLE.QUEUE*. Kanał jest skonfigurowany w taki sposób, aby zezwalał tylko na dostęp do klientów, którzy prezentują certyfikat z nazwą zwykłą ("common name") o nazwie *example-app1*. Jest to nazwa zwykła używana w jednym z certyfikatów utworzonych w kroku "2" na stronie 156. Połączenia w tym kanale o tej nazwie zwykłej są odwzorowywane na identyfikator użytkownika *app1*, który jest autoryzowany do nawiązywania połączenia z menedżerem kolejek i uzyskiwania dostępu do przykładowej kolejki. Plik INI włącza strategię bezpieczeństwa, która oznacza, że ID użytkownika *app1* nie musi istnieć w zewnętrznym rejestrze użytkowników-istnieje tylko jako nazwa w tej konfiguracji.

4. Wdrażanie menedżera kolejek

Utwórz nowy menedżer kolejek przy użyciu następującego zasobu niestandardowego YAML. Upewnij się, że znajdujesz się w przestrzeni nazw utworzonej przed rozpoczęciem tego zadania, a następnie wprowadź następujący kod YAML w konsoli WWW OCP lub użyj wiersza komend. Sprawdź, czy wskazano poprawną licencję, a następnie zaakceptuj licencję, zmieniając wartość *false* na *true*.

```

apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-AMRD-XH6P3Q
    use: Production
  queueManager:
    name: EXAMPLEQM
    availability:
      type: MultiInstance
  mqsc:
    - configMap:
      name: example-miqm-configmap
      items:
        - example-tls.mqsc
  ini:
    - configMap:
      name: example-miqm-configmap
      items:
        - example-tls.ini
  storage:
    defaultClass: <STORAGE CLASS>
  version: 9.3.5.1-r2
  pki:
    keys:
      - name: default
        secret:
          secretName: example-qm-tls
          items:
            - tls.key
            - tls.crt
            - ca.crt

```

Zmień *< STORAGE CLASS >* na klasę pamięci masowej określoną w kroku "1" na stronie 156.

Należy zauważyć, że klucz tajny *example-qm-tls* został utworzony w kroku "2" na stronie 156, a ConfigMap *example-miqm-configmap* został utworzony w kroku "3" na stronie 156.

Typ dostępności jest ustawiony na *MultiInstance*, co powoduje automatyczne wybranie trwałej pamięci masowej.

5. Upewnij się, że menedżer kolejek jest uruchomiony

Menedżer kolejek jest obecnie wdrażany. Przed kontynuowaniem upewnij się, że znajduje się on w stanie *Running*. Na przykład:

```
oc get qmgr exampleqm
```

6. Testowanie połączenia z menedżerem kolejek

Aby potwierdzić, że menedżer kolejek jest skonfigurowany i dostępny, wykonaj kroki opisane w sekcji [“Testowanie wzajemnego połączenia TLS z menedżerem kolejek z komputera laptop”](#) na stronie 143.

7. Wymuszanie awarii aktywnego zasobnika

Aby sprawdzić poprawność automatycznego odtwarzania menedżera kolejek, zasymuluj awarię zasobnika:

a) Wyświetlanie aktywnych i rezerwowych zasobników

Uruchom następującą komendę:

```
oc get pods --selector app.kubernetes.io/instance=exampleqm
```

Uwaga: w polu **READY** aktywny zasobnik zwraca wartość 1/1, natomiast zasobnik rezerwowy zwraca wartość 0/1.

b) Usuwanie aktywnego zasobnika

Uruchom następującą komendę, podając pełną nazwę aktywnego zasobnika:

```
oc delete pod exampleqm-ibm-mq-<value>
```

c) Wyświetl ponownie status zasobnika

Uruchom następującą komendę:

```
oc get pods --selector app.kubernetes.io/instance=exampleqm
```

d) Wyświetlanie statusu menedżera kolejek

Uruchom następującą komendę, podając pełną nazwę drugiego zasobnika:

```
oc exec -t Pod -- dspmq -x
```

Powinien zostać wyświetlony status, który wskazuje, że aktywna instancja została zmieniona, na przykład:

```
QMNAME(EXAMPLEQM)                                STATUS(Running as standby)
  INSTANCE(exampleqm-ibm-mq-1) MODE(Active)
  INSTANCE(exampleqm-ibm-mq-0) MODE(Standby)
```

e) Ponownie przetestuj połączenie z menedżerem kolejek

Aby potwierdzić, że menedżer kolejek został odzyskany, wykonaj kroki opisane w sekcji [“Testowanie wzajemnego połączenia TLS z menedżerem kolejek z komputera laptop”](#) na stronie 143.

Wyniki

Gratulacje, pomyślnie wdrożono menedżer kolejek z wieloma instancjami z wzajemnym uwierzytelnianiem TLS i sprawdzono, że jest on automatycznie odzyskiwane w przypadku niepowodzenia aktywnego zasobnika.

nawiązania połączenia z menedżerem kolejek spoza klastra produktu Red Hat OpenShift

Do połączenia aplikacji z menedżerem kolejek IBM MQ spoza klastra Red Hat OpenShift potrzebna jest trasa Red Hat OpenShift. Należy włączyć protokół TLS w menedżerze kolejek i aplikacji klienckiej IBM MQ, ponieważ protokół SNI jest dostępny w protokole TLS tylko wtedy, gdy używany jest protokół TLS w wersji 1.2 lub nowszej. Red Hat OpenShift Container Platform Router używa wskaźnika SNI do kierowania żądań do menedżera kolejek IBM MQ.

O tym zadaniu

Wymagana konfiguracja Red Hat OpenShift Route zależy od zachowania aplikacji klienckiej Server Name Indication (SNI). IBM MQ obsługuje dwa różne ustawienia nagłówka SNI w zależności od konfiguracji i typu klienta. Nagłówek SNI jest ustawiony na nazwę hosta miejsca docelowego klienta lub na nazwę kanału IBM MQ. Informacje na temat odwzorowania nazwy kanału na nazwę hosta przez program IBM MQ można znaleźć w sekcji Jak produkt IBM MQ obsługuje więcej niż jeden certyfikat.

Określa, czy nagłówek SNI jest ustawiony na nazwę kanału IBM MQ, czy na nazwę hosta, za pomocą atrybutu **OutboundSNI**. Możliwe wartości to `OutboundSNI=CHANNEL` (wartość domyślna) lub `OutboundSNI=HOSTNAME`. Więcej informacji na ten temat zawiera sekcja SSL pliku konfiguracyjnego klienta. Należy zauważyć, że `CHANNEL` i `HOSTNAME` są dokładnymi wartościami, które są używane; nie są to nazwy zmiennych, które są zastępowane rzeczywistą nazwą kanału lub nazwą hosta.

Zachowania klienta z różnymi ustawieniami **OutboundSNI**

Jeśli parametr **OutboundSNI** jest ustawiony na `HOSTNAME`, to następujące klienty ustawiają nazwę hosta SNI pod warunkiem, że w nazwie połączenia jest podana nazwa hosta:

- Klienty C
- Klienty .NET w trybie niezarządzanym
- Java/klienty JMS

Jeśli parametr **OutboundSNI** jest ustawiony na `HOSTNAME`, a w nazwie połączenia używany jest adres IP, to następujące klienty wysyłają pusty nagłówek SNI:

- Klienty C
- Klienty .NET w trybie niezarządzanym
- Klienty Java/JMS (które nie mogą wykonać odwrotnego wyszukiwania DNS nazwy hosta)

Jeśli parametr **OutboundSNI** jest ustawiony na `CHANNEL` lub nie jest ustawiony, zamiast tego jest używana nazwa kanału IBM MQ, która jest zawsze wysyłana, niezależnie od tego, czy używana jest nazwa hosta, czy nazwa połączenia z adresem IP.

Następujące typy klientów nie obsługują ustawiania nagłówka SNI na nazwę kanału IBM MQ, więc zawsze należy ustawić nagłówek SNI na nazwę hosta niezależnie od ustawienia **OutboundSNI**:

- Klienty AMQP
- Klienty XR
- Klienty .NET w trybie zarządzanym (przed wersją IBM MQ 9.3.0)

W produkcie IBM MQ 9.3.0 IBM MQ zarządzany klient .NET został zaktualizowany i ustawił wartość `SERVERNAME` na odpowiednią nazwę hosta, jeśli właściwość **OutboundSNI** ma wartość `HOSTNAME`, co umożliwi klientowi IBM MQ zarządzanemu .NET nawiązanie połączenia z menedżerem kolejek przy użyciu tras Red Hat OpenShift.

Jeśli aplikacja kliencka łączy się z menedżerem kolejek wdrożonym w klastrze Red Hat OpenShift za pomocą programu IBM MQ Internet Pass-Thru (MQIPT), produkt MQIPT można skonfigurować w taki sposób, aby ustawić SNI na nazwę hosta, używając właściwości SSLClientOutboundSNI w definicji trasy.

OutboundSNI, wiele certyfikatów i trasy Red Hat OpenShift

Produkt IBM MQ korzysta z nagłówka SNI w celu udostępnienia wielu funkcji certyfikatów. Jeśli aplikacja łączy się z kanałem IBM MQ skonfigurowanym do używania innego certyfikatu za pomocą pola CERTLABL, aplikacja musi połączyć się z ustawieniem **OutboundSNI** o wartości CHANNEL.

Jeśli konfiguracja trasy systemu Red Hat OpenShift wymaga nazwy hosta SNI, nie można użyć funkcji obsługi wielu certyfikatów programu IBM MQ i nie można ustawić ustawienia CERTLABL dla żadnego obiektu kanału IBM MQ .

Jeśli aplikacja z ustawieniem **OutboundSNI** innym niż CHANNEL nawiąże połączenie z kanałem ze skonfigurowaną etykietą certyfikatu, aplikacja zostanie odrzucona z błędem MQRC_SSL_INITIALIZATION_ERROR i w dziennikach błędów menedżera kolejek zostanie zapisany komunikat AMQ9673 .

Więcej informacji na temat sposobu, w jaki program IBM MQ udostępnia funkcje obsługi wielu certyfikatów, zawiera sekcja [W jaki sposób program IBM MQ udostępnia możliwość obsługi wielu certyfikatów](#) .

Przykład

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

Ważne jest, aby nazwy kanałów produktu MQ nie kończyły się małą literą, ponieważ produkt IBM MQ odwzorowuje nazwy kanałów na nagłówki SNI.

Aby określić wymaganą nazwę hosta dla każdej z nowych tras produktu Red Hat OpenShift, należy odwzorować każdą nazwę kanału na adres SNI. Więcej informacji na ten temat zawiera sekcja [Jak produkt IBM MQ obsługuje więcej niż jeden certyfikat](#).

Następnie należy utworzyć nową trasę Red Hat OpenShift dla każdego kanału. W tym celu należy zastosować następującą komendę 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 "<Queue Manager Name>-ibm-mq")>
  port:
    targetPort: 1414
  tls:
    termination: passthrough
```

Konfigurowanie szczegółów połączenia aplikacji klienckiej

Nazwę hosta, która ma być używana dla połączenia z klientem, można określić, uruchamiając następującą komendę:

```
oc get route <Name of hostname based Route (for example "<Queue Manager Name>-ibm-mq-qm")>
-n <namespace of your MQ deployment> -o jsonpath="{.spec.host}"
```

Port dla połączenia klienckiego powinien być ustawiony na port używany przez router Red Hat OpenShift Container Platform — zwykle 443.

Zadania pokrewne

[“Nawiązywanie połączenia z rozwiązaniem IBM MQ Console wdrożonym w klastrze Red Hat OpenShift” na stronie 176](#)

W tej sekcji opisano nawiązywanie połączenia z bazą danych IBM MQ Console menedżera kolejek, który został wdrożony w klastrze Red Hat OpenShift Container Platform .




Deprecated CP4I Removed 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.

Zanim rozpocznesz



Ostrzeżenie:

   Od wersji IBM MQ Operator 2.0.0 panel kontrolny operacji jest nieaktualny i nie będzie otrzymywać dalszych aktualizacji. Nie należy tworzyć nowych zastosowań panelu kontrolnego operacji.

  Z wersji IBM MQ Operator 2.4.0 panel kontrolny operacji został usunięty. Należy zauważyć, że panel kontrolny operacji może być nadal używany dla istniejących menedżerów kolejek, które są starsze niż 9.3.3.0-r1 , jeśli w systemie IBM MQ Operator , który obsługuje ten obraz kontenera menedżera kolejek. Informacje na temat obsługi wersji dla IBM MQ Operator zawiera sekcja [“Dostępne wersje produktu IBM MQ”](#) na stronie 12.

Obsługa panelu kontrolnego operacji kończy się 30 czerwca 2024 roku. Więcej informacji na ten temat zawiera sekcja [Wycofywanie oprogramowania i/lub wycofywanie wsparcia](#).

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 UI (previously the 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. Więcej informacji na temat wdrażania menedżera kolejek zawiera sekcja [Wdrażanie menedżera kolejek za pomocą programu IBM Cloud Pak for Integration Platform UI](#)

W przypadku wdrażania przy użyciu interfejsu wiersza komend Red Hat OpenShift lub konsoli WWW Red Hat OpenShift można włączyć śledzenie za pomocą następującego fragmentu kodu YAML:

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

Ważne: Menedżer kolejek nie zostanie uruchomiony, dopóki produkt MQ nie zostanie zarejestrowany w panelu kontrolnym Operations Dashboard (patrz następny krok).

Należy zwrócić uwagę, że gdy ta funkcja jest włączona, oprócz kontenera z menedżerem kolejek działać będą także dwa kontenery pomocnicze („Agent” i „Collector”). Obrazy tych kontenerów pomocniczych będą dostępne w tym samym rejestrze, co główny obraz MQ, i będą używać tej

samej strategii pobierania oraz sekretu pobierania. Dostępne są dodatkowe ustawienia służące do konfigurowania limitów procesora i pamięci.

2. Jeśli menedżer kolejek z integracją z panelem kontrolnym Operations Dashboard został po raz pierwszy wdrożony w tej przestrzeni nazw, wówczas należy dokonać rejestracji w panelu kontrolnym Operations Dashboard.

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

Integrowanie produktu IBM MQ z funkcją śledzenia produktu IBM Instana

Program IBM Instana może być używany do śledzenia transakcji w programie IBM Cloud Pak for Integration.

Zanim rozpocznie

Ten dokument obejmuje śledzenie IBM Instana, które jest procesem śledzenia komunikatów w systemie. Nie obejmuje on monitorowania IBM Instana, w którym pobierane są szczegółowe informacje o stanie menedżera kolejek produktu IBM MQ. Informacje dotyczące monitorowania produktu IBM MQ przez produkt IBM Instana zawiera sekcja Monitorowanie produktu IBM MQ. Szczegółowe instrukcje dotyczące uwierzytelnionego monitorowania zawiera sekcja “Konfigurowanie uwierzytelnionego monitorowania IBM Instana za pomocą protokołu TLS” na stronie 164.

Uwaga:

- Ta funkcja może być używana tylko z operatorem IBM MQ w wersji 2.2.0 lub nowszej. Ta funkcja jest obsługiwana tylko na operandach produktu IBM MQ w wersji 9.3.1.0-r2 lub nowszej.
- Śledzenie IBM Instana można uruchomić w poprzednich wersjach operatora i menedżera kolejek produktu IBM MQ, ale nie w trybie rodzimym. Więcej informacji na ten temat zawiera sekcja Konfigurowanie śledzenia produktu IBM MQ w dokumentacji produktu IBM Instana.

Zanim będzie można wykonać śledzenie IBM Instana za pomocą operatora IBM MQ, należy wdrożyć zarówno agenty IBM Instana zaplecza, jak i IBM Instana. Domyślnie menedżer kolejek produktu IBM MQ komunikuje się z agentem IBM Instana wdrożonym w tym samym węźle co zasobnik menedżera kolejek.

O tym zadaniu

Włączenie integracji z produktem IBM Instana powoduje zainstalowanie w menedżerze kolejek wyjścia funkcji API języka IBM MQ. Wyjście funkcji API wysyła do agentów IBM Instana dane śledzenia dotyczące komunikatów przepływających przez menedżer kolejek.

Wyjście funkcji API dodaje nagłówki RFH2 do każdego komunikatu. Te nagłówki zawierają informacje o śledzeniu.

Agenty IBM Instana są odpowiedzialne za wysyłanie danych śledzenia do zaplecza IBM Instana.

Informacje na temat wdrażania zaplecza IBM Instana i agentów IBM Instana zawiera sekcja Włączanie monitorowania IBM Instana w interfejsie użytkownika platformy CP4I w dokumentacji produktu IBM Instana.

Procedura

Wdrożenie standardowe

- Wdróż menedżera kolejek z włączonym śledzeniem IBM Instana.

Domyślnie śledzenie IBM Instana jest wyłączone.

Jeśli używana jest konsola WWW IBM Cloud Pak for Integration Platform UI (previously the Platform Navigator) lub OpenShift :

1. Kliknij opcję **Telemetry > Śledzenie > Instana**.

2. Ustaw przełącznik **Włącz śledzenie na Instana** na wartość `true`.

W przypadku wdrażania za pomocą języka YAML należy użyć następującego fragmentu kodu:

```
spec:
  telemetry:
    tracing:
      instana:
        enabled: true
```

Wdrażanie zaawansowane

- Komunikacja z agentem IBM Instana za pośrednictwem protokołu HTTPS.

Domyślnie wyjście IBM Instana dla IBM MQ komunikuje się z agentem IBM Instana za pośrednictwem protokołu `http`. Adres hosta agenta jest ustawiony na adres IP węzła, na którym działa menedżer kolejek. Jest to zgodne z konfiguracją opisaną w sekcji [Włączanie monitorowania IBM Instana](#) w dokumentacji produktu IBM Instana, w której agenty IBM Instana są wdrażane przez operatora agenta IBM Instana jako zestaw demonów.

Obecnie komunikacja między wyjściem programu IBM Instana dla IBM MQ i agentem IBM Instana obsługuje protokoły `http` lub `https`. Aby używać protokołu HTTPS, należy najpierw skonfigurować agenta IBM Instana do używania szyfrowania TLS. Więcej informacji na ten temat zawiera sekcja [Konfigurowanie szyfrowania TLS dla punktu końcowego agenta](#) w dokumentacji produktu IBM Instana. Protokół można następnie ustawić na `https` w następujący sposób:

Jeśli używana jest konsola WWW OpenShift :

1. Kliknij opcję **Telemetria > Instana**.
2. Rozwiń listę **Konfiguracja zaawansowana**.
3. Ustaw opcję **Instana agent communication protocol** (Protokół komunikacyjny agenta Instana) na wartość `https`.

W przypadku wdrażania za pomocą języka YAML należy użyć następującego fragmentu kodu:

```
spec:
  telemetry:
    instana:
      enabled: true
      protocol: https
```

- Ustaw **agentHost**

Jeśli agenty IBM Instana nie zostały wdrożone jako zestaw demonów w klastrze Openshift, w którym działa menedżer kolejek, należy ustawić wartość **agentHost** na nazwę hosta lub adres IP, w którym działa agent IBM Instana. Wartość **agentHost** nie powinna zawierać protokołu ani portu.

Jeśli używana jest konsola WWW OpenShift :

1. Kliknij opcję **Telemetria > Instana**.
2. Rozwiń listę **Konfiguracja zaawansowana**.
3. Wpisz nazwę hosta w polu tekstowym **Host agenta Instana**.

W przypadku wdrażania za pomocą języka YAML należy użyć następującego fragmentu kodu:

```
spec:
  telemetry:
    instana:
      enabled: true
      agentHost: 9.9.9.9
```

Co dalej

Patrz także [“Wdrażanie menedżera kolejek w klastrze platformy kontenerowej Red Hat OpenShift”](#) na stronie 118.

Konfigurowanie uwierzytelnionego monitorowania IBM Instana za pomocą protokołu TLS

Aby można było monitorować menedżer kolejek za pomocą agenta IBM Instana, należy skonfigurować zarówno agenta, jak i menedżera kolejek.

Zanim rozpocznie

Sekcja "Configuration" (Konfiguracja) publikacji "Monitoring IBM MQ" w dokumentacji IBM Instana zawiera ogólne informacje dotyczące konfiguracji monitorowania w systemie IBM Instana. Nie zawiera on jednak szczegółów dotyczących konfigurowania menedżera kolejek.

Zanim będzie można wykonać śledzenie IBM Instana za pomocą operatora IBM MQ, należy wdrożyć zarówno agenta IBM Instana zaplecza, jak i IBM Instana. W tym celu należy zapoznać się z sekcją [Włączanie monitorowania IBM Instana w interfejsie użytkownika platformy CP4I](#) w dokumentacji produktu IBM Instana.

Procedura

1. [Wygeneruj certyfikaty](#).
2. [Skonfiguruj agenty IBM Instana](#).
3. [Skonfiguruj menedżer kolejek](#).
4. [Sprawdź i debuguj](#).

Zadania pokrewne

"Integrowanie produktu IBM MQ z funkcją śledzenia produktu IBM Instana" na stronie 162 Program IBM Instana może być używany do śledzenia transakcji w programie IBM Cloud Pak for Integration.

Generowanie certyfikatu i klucza dla agenta IBM Instana i menedżera kolejek

W przypadku komunikacji TLS między agentem IBM Instana i menedżerem kolejek oba te protokoły muszą mieć certyfikat i odpowiadający mu klucz prywatny.

Zanim rozpocznie

Jest to pierwsze z czterech zadań [konfigurowania uwierzytelnionego monitorowania IBM Instana za pomocą protokołu TLS](#).

Uwaga: Wartości używane podczas generowania tych certyfikatów służą do celów demonstracyjnych. Podczas wdrażania w środowisku produkcyjnym należy upewnić się, że temat i termin ważności certyfikatu są odpowiednie.

Procedura

IBM MQ Menedżer kolejek

Aby można było komunikować się z agentem IBM Instana za pomocą protokołu TLS, menedżer kolejek musi mieć certyfikat i odpowiadający mu klucz prywatny. Jeśli już je masz, pomiń tę sekcję.

1. Wygeneruj certyfikat i klucz prywatny dla menedżera kolejek.

Uruchom następującą komendę:

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

agentIBM Instana

Aby agent mógł wykonać komunikację TLS z menedżerem kolejek produktu IBM MQ , agent musi mieć certyfikat i odpowiadający mu klucz prywatny. Jeśli w magazynie kluczy JKS znajduje się już klucz prywatny i certyfikat, który ma być używany, należy pominąć tę sekcję.

2. Wygeneruj certyfikat i klucz prywatny dla agenta IBM Instana .

Uruchom następującą komendę:

```
openssl req \
  -newkey rsa:2048 -nodes -keyout application.key \
  -subj "/CN=instana-agent/OU=app team1" \
  -x509 -days 3650 -out application.crt
```

3. Zapisz certyfikat i klucz prywatny w magazynie kluczy PKCS12 .

Uruchom następującą komendę, zastępując *hasło_użytkownika* hasłem, które ma być używane do zabezpieczenia magazynu kluczy. Wykonaj tę wymianę we wszystkich kolejnych krokach.

```
openssl pkcs12 -export -out application.p12 -inkey application.key -in application.crt
-passout pass:your_password
```

4. Przekształć magazyn kluczy PKCS12 w magazyn kluczy JKS.

Uruchom następującą komendę:

```
keytool -importkeystore \
  -srckeystore application.p12 \
  -srcstoretype pkcs12 \
  -destkeystore application.jks \
  -deststoretype JKS \
  -srcstorepass your_password \
  -deststorepass your_password \
  -noprompt
```

5. Oznacz certyfikat etykietą.

Uruchom następującą komendę:

```
keytool -changealias -alias "1" -destalias "instana" -keypass your_password -keystore
application.jks -storepass your_password -noprompt
```

6. Zaimportuj certyfikat menedżera kolejek do magazynu kluczy.

Uruchom następującą komendę:

```
keytool -importcert -file server.crt -keystore application.jks -storepass your_password
-alias myca -noprompt
```

Co dalej

Teraz można skonfigurować agenta do monitorowania IBM Instana.

Monitorowanie Instana: konfigurowanie agentów

Podłącz magazyn kluczy do agentów IBM Instana , a następnie skonfiguruj monitorowanie dla konkretnego menedżera kolejek.

Zanim rozpocznie

W tym zadaniu przyjęto, że użytkownik wygenerował certyfikat i klucz dla agentów IBM Instana i menedżera kolejek.

Procedura

Podłączanie magazynu kluczy do agentów IBM Instana

1. Utwórz klucz tajny z magazynu kluczy JKS w przestrzeni nazw agenta IBM Instana .

Uruchom następującą komendę, zastępując zmienną `keystore_secret_name` nazwą, która ma być używana. Wykonaj tę wymianę we wszystkich kolejnych krokach.

```
oc create secret generic keystore_secret_name --from-file=./application.jks -n instana-agent
```

2. W przestrzeni nazw `instana-agent` użyj komendy `oc edit daemonset instana-agent`, aby zmodyfikować zestaw demonów `instana-agent` w celu uwzględnienia następujących dodatkowych wartości `volumeMount` i `woluminu`:

```
volumeMounts:
- name: mq-key-jks-name
  subPath: application.jks
  mountPath: /opt/instana/agent/etc/application.jks
volumes:
- name: mq-key-jks-name
  secret:
    secretName: keystore_secret_name
```

Konfigurowanie monitorowania dla konkretnego menedżera kolejek

3. W przestrzeni nazw `instana-agent` użyj komendy `oc edit configmap instana-agent`, aby zmodyfikować mapę konfiguracji `instana-agent`.
4. Dodaj następującą sekcję w sekcji `configuration.yaml`: |. Jeśli ta sekcja została już zdefiniowana, wystarczy dodać nowy menedżer kolejek do listy.

```
com.instana.plugin.ibmmq:
  enabled: true
  poll_rate: 60
  queueManagers:
    QUEUE_MANAGER_NAME:
      channel: 'INSTANA.A.SVRCONN'
      keystorePassword: 'your_password'
      keystore: '/opt/instana/agent/etc/application.jks'
      cipherSuite: 'TLS_RSA_WITH_AES_256_CBC_SHA256'
```

where

- `hasło_użytkownika` to hasło do magazynu kluczy JKS.
- `QUEUE_MANAGER_NAME` to nazwa bazowego menedżera kolejek produktu IBM MQ, który ma zostać wdrożony, a nie nazwa operandu menedżera kolejek.

Uwaga: Jeśli parametr `QUEUE_MANAGER_NAME` nie jest ustawiony na bazową nazwę menedżera kolejek, a zamiast tego na Operand, monitorowanie nie będzie działać. Nazwa bazowa jest zdefiniowana w pliku `spec.queuemanager.name` dla operacji menedżera kolejek.

5. Usuń zasobniki `instana-agent` w przestrzeni nazw `instana-agent`. Spowoduje to ich zrestartowanie i rozpoczęcie monitorowania z nowymi ustawieniami.

Co dalej

Teraz można [skonfigurować menedżer kolejek na potrzeby IBM Instana monitorowania](#).

Monitorowanie instana: konfigurowanie menedżera kolejek

Skonfiguruj menedżera kolejek, który używa protokołu TLS do komunikacji z agentem IBM Instana. Uwierzytelnianie dla tego połączenia jest wykonywane za pomocą komendy [SSLPEERMAP](#).

Zanim rozpocznie

W tym zadaniu przyjęto, że [skonfigurowano agenty dla IBM Instana monitorowania](#).

Procedura

1. Skonfiguruj menedżer kolejek za pomocą komend `MQSC` i `INI`.

MQSC jest używany do konfigurowania nowego kanału z włączoną obsługą protokołu TLS, a następnie do konfigurowania tego kanału w celu uwierzytelniania łączącego się agenta IBM Instana, jeśli ma on certyfikat z wymaganymi polami. W takim przypadku każdy klient łączący się z certyfikatem zawierającym pola CN=instana-agent, OU=app team1 zostanie odwzorowany na użytkownika app1. Następnie komenda MQSC nadaje użytkownikowi app1 uprawnienie do wykonywania wymaganych operacji na potrzeby monitorowania produktu IBM Instana.

Plik INI służy do nadawania uprawnień użytkownikowi zewnętrznemu app1.

Poniższa komenda configmap zawiera wymagane ustawienia MQSC i INI. Wdróż go w przestrzeni nazw menedżera kolejek.

```
apiVersion: v1
data:
  channel.mqsc: |-
    DEFINE CHANNEL('INSTANA.A.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
    SSLCIPH('ANY_TLS12_OR_HIGHER')
    ALTER QMGR CONNAUTH(' ')
    REFRESH SECURITY
    SET CHLAUTH('INSTANA.A.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
  ACTION(REPLACE)
    SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS) ACTION(REPLACE)
    SET CHLAUTH('INSTANA.A.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=instana-agent,OU=app
team1') USERSRC(MAP) MCAUSER('app1')
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(ALL)
    SET AUTHREC PROFILE('SYSTEM.ADMIN.COMMAND.QUEUE') PRINCIPAL('app1') OBJTYPE(Queue)
  AUTHADD(PUT,INQ,DSP,CHG)
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(Topic) AUTHADD(DSP)
    SET AUTHREC PROFILE('*') PRINCIPAL('app1') OBJTYPE(Topic) AUTHADD(DSP)
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(Queue) AUTHADD(DSP,CHG,GET)
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(Listener) AUTHADD(DSP)
    SET AUTHREC PROFILE('AMQ.*') PRINCIPAL('app1') OBJTYPE(Queue) AUTHADD(DSP,CHG)
    REFRESH SECURITY TYPE(CONNAUTH)
  auth.ini: |-
    Service:
      Name=AuthorizationService
      EntryPoints=14
      SecurityPolicy=UserExternal
kind: ConfigMap
metadata:
  namespace: your-queue-manager-namespace
  name: qmgr-monitoring-config
```

gdzie *przestrzeń_nazw-menedżera-kolejek* jest przestrzenią nazw, w której zostanie wdrożony menedżer kolejek.

Uwaga: Jeśli monitorowane są kolejki zdefiniowane przez użytkownika, należy dodać dodatkowe wiersze do pliku configmap MQSC, nadając uprawnienia DSP, CHG i GET do tych kolejek. Na przykład:

```
SET AUTHREC PROFILE('MYQUEUE') PRINCIPAL('app1') OBJTYPE(Queue) AUTHADD(DSP,CHG,GET).
```

W tym przykładzie użyto odwzorowania configmap dla danych MQSC i INI, ale można użyć klucza tajnego, jeśli jakiegokolwiek dodane dane są poufne. Ogólne informacje na temat wdrażania za pomocą komend MQSC i INI zawiera sekcja “Przykład: Udostępnianie plików MQSC i INI” na stronie 138.

2. Aby można było nawiązać połączenie TLS, menedżer kolejek musi ufać certyfikatowi agenta IBM Instana. Aby to osiągnąć, należy utworzyć klucz tajny zawierający tylko certyfikat agenta IBM Instana:

```
oc create secret generic instana-certificate-secret --from-file=./application.crt -n your-queue-manager-namespace
```

3. Menedżer kolejek musi przedstawić własny certyfikat na potrzeby uzgadniania TLS i wymaga dostępu do powiązanego klucza prywatnego. Wdróż klucz tajny zawierający klucz i certyfikat, który został utworzony wcześniej lub jest już w posiadaniu:

```
oc create secret tls qm-tls-secret --cert server.crt --key server.key -n your-queue-manager-namespace
```

Po utworzeniu mapy konfiguracji i klucza tajnego można utworzyć sam menedżer kolejek.

4. Upewnij się, że menedżer kolejek YAML nie ustawia zmiennej środowiskowej **MQSNOAUT** w kontenerze menedżera kolejek.

W przeciwnym razie po włączeniu mechanizmu uwierzytelniania nie będzie on działać. Usunięcie zmiennej po wdrożeniu nie powoduje ponownego włączenia mechanizmu i konieczne jest ponowne utworzenie menedżera kolejek.

5. Dodaj następujące sekcje do definicji menedżera kolejek, gdzie *MYQM* jest nazwą menedżera kolejek:

```
spec:
  queueManager:
    name: MYQM #(a)
    ini: #(b)
    - configMap:
        items:
          - auth.ini
        name: qmgr-monitoring-config
    mqsc: #(c)
    - configMap:
        items:
          - channel.mqsc
        name: qmgr-monitoring-config
  pki:
    keys: #(d)
    - name: default
      secret:
        items:
          - tls.key
          - tls.crt
        secretName: qm-tls-secret
    trust: #(e)
    - name: app
      secret:
        items:
          - application.crt
        secretName: instana-certificate-secret
```

Oznaczone sekcje specyfikacji są opisane w następujący sposób:

- a. Upewnij się, że bazowa nazwa menedżera kolejek jest unikalna. Jeśli bazowy menedżer kolejek nie ma unikalnej nazwy, monitorowanie może nie działać zgodnie z przeznaczeniem. Ta nazwa musi być zgodna z nazwą w pliku configmap agenta IBM Instana, który był edytowany wcześniej.
 - b. Informacje INI, które zostały zapisane w mapie konfiguracji, są dodawane do menedżera kolejek.
 - c. Informacje MQSC, które zostały zapisane w mapie konfiguracji, zostaną dodane do menedżera kolejek.
 - d. Certyfikat menedżera kolejek i klucz prywatny są dodawane do magazynu kluczy menedżera kolejek.
 - e. Certyfikat agenta IBM Instana zostanie dodany do magazynu zaufanych certyfikatów menedżera kolejek.
6. Opcjonalne: Włącz śledzenie IBM Instana w monitorowanym menedżerze kolejek.
Aby to zrobić, zapoznaj się z sekcją [“Integrowanie produktu IBM MQ z funkcją śledzenia produktu IBM Instana” na stronie 162.](#)
 7. Wdróż menedżer kolejek.

Co dalej

Teraz można zweryfikować i debugować IBM Instana monitorowanie.

   **Monitorowanie instana: weryfikowanie**

i debugowanie

Aby można było monitorować menedżer kolejek za pomocą agenta IBM Instana, należy skonfigurować zarówno agenta, jak i menedżera kolejek.

Zanim rozpocznie

W tym zadaniu przyjęto, że użytkownik [skonfigurował menedżer kolejek na potrzeby IBM Instana monitorowania](#).

Procedura

Weryfikowanie

1. Aby sprawdzić, czy wdrożenie zakończyło się pomyślnie, należy wyświetlić menedżer kolejek w panelu kontrolnym produktu IBM Instana .

Menedżer kolejek powinien być widoczny w sekcji usług strony aplikacji, a także w widoku infrastruktury.

Debugowanie

Uwaga: W tych krokach debugowania przyjęto wdrożenie Openshift agenta IBM Instana działającego jako zestaw demonów.

Jeśli menedżer kolejek nie jest widoczny na panelu kontrolnym produktu IBM Instana , oznacza to, że menedżer kolejek został niepoprawnie skonfigurowany. Wykonaj następujące kroki, aby zbadać problem.

2. Zidentyfikuj węzeł, w którym działa aktywny zasobnik menedżera kolejek.

Uruchom następującą komendę w przestrzeni nazw menedżera kolejek:

```
oc get pods -o wide -n your-queue-manager-namespace
```

3. Aby określić, który zasobnik agenta IBM Instana jest uruchomiony w tym samym węźle co menedżer kolejek, uruchom tę samą komendę w przestrzeni nazw instana-agent:

```
oc get pods -o wide -n instana-agent-namespace
```

4. Aby ułatwić sobie zrozumienie problemów występujących po stronie agenta IBM Instana , należy pobrać dzienniki zasobnika agenta IBM Instana i wyszukać pozycje związane z 'mq' lub nazwą menedżera kolejek.

Uruchom następującą komendę:

```
oc logs instana-agent-pod -c instana-agent -n instana-agent
```

5. Sprawdź dzienniki menedżera kolejek.

Jeśli agent podjął próbę nawiązania połączenia z menedżerem kolejek, dzienniki menedżera kolejek powinny wskazywać przyczynę niepowodzenia połączenia. Uruchom następującą komendę:

```
oc logs your-queue-manager-name -n your-queue-manager-namespace
```

Wyniki

Wykonano wszystkie cztery zadania związane z [konfigurowaniem uwierzytelnionego IBM Instana monitorowania przy użyciu protokołu TLS](#).

Budowanie obrazu z niestandardowymi plikami MQSC i INI przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift

Używając potoku Red Hat OpenShift Container Platform, można utworzyć nowy obraz kontenera IBM MQ z plikami MQSC i INI, które mają być zastosowane do menedżerów kolejek korzystających z tego obrazu. To zadanie powinien wykonać administrator projektu.

Zanim rozpocznie

Konieczne będzie zainstalowanie [interfejsu wiersza komend Red Hat OpenShift Container Platform](#).

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

Jeśli w projekcie Red Hat OpenShift nie ma klucza tajnego Red Hat OpenShift dla rejestru IBM Entitled Registry, należy wykonać kroki opisane w sekcji [Create the entitlement key secret](#) (Utwórz klucz tajny uprawnień).

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 kompilacje 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:9.3.5.1-r2
      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:9.3.5.1-r2"
      pullSecret:
        name: ibm-entitlement-key
  output:
    to:
      kind: ImageStreamTag
      name: 'mymq:latest-amd64'
```

Aby wskazać poprawny obraz bazowy IBM MQ, który ma być używany, należy zastąpić nazwy w dwóch miejscach, w których jest wymieniony bazowy obraz (aby uzyskać szczegółowe informacje, patrz [“Historia wydań operatora IBM MQ Operator”](#) na stronie 36). 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 procedurą opisaną w sekcji [“Wdrażanie menedżera kolejek w klastrze platformy kontenerowej Red Hat OpenShift”](#) na stronie 118, dodając nowy obraz niestandardowy do kodu YAML.

Można dodać następujący fragment kodu YAML do normalnego kodu YAML QueueManager, gdzie *my-namespace* to projekt / przestrzeń nazw Red Hat OpenShift użytkownika, a *image* to nazwa utworzonego wcześniej obrazu (na przykład "mymq:latest-amd64"):

```
spec:
  queueManager:
    image: image-registry.openshift-image-registry.svc:5000/my-namespace/my-image
```

Zadania pokrewne

[“Wdrażanie menedżera kolejek w klastrze platformy kontenerowej Red Hat OpenShift”](#) na stronie 118

W tym przykładzie wdrażany jest menedżer kolejek "Szybki start", który używa pamięci efemerycznej (nietrwałej) i wyłącza zabezpieczenia produktu IBM MQ. Komunikaty nie są zachowywane po restarcie menedżera kolejek. Można zmienić konfigurację, aby zmodyfikować różne ustawienia menedżera kolejek.

Dodawanie niestandardowych adnotacji i etykiet do zasobów menedżera kolejek

Można dodawać niestandardowe adnotacje i etykiety do metadanych menedżera kolejek.

O tym zadaniu

Niestandardowe adnotacje i etykiety są dodawane do wszystkich zasobów z wyjątkiem rezerwacji PVC. Jeśli adnotacja niestandardowa lub etykieta jest zgodna z istniejącym kluczem, obowiązuje wartość ustawiona przez parametr IBM MQ Operator.

Procedura

- Dodaj niestandardowe adnotacje.

Aby dodać niestandardowe adnotacje do zasobów menedżera kolejek, w tym do zasobnika, dodaj adnotacje do składnika metadata. Na przykład:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    annotationKey: "value"
```

- Dodaj niestandardowe etykiety.

Aby dodać etykiety niestandardowe do zasobów menedżera kolejek, w tym do zasobnika, dodaj etykiety w sekcji metadata. Na przykład:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  labels:
    labelKey: "value"
```

Wyłączanie kontroli elementu webhook w czasie wykonywania

Kontrole elementu webhook w czasie wykonywania zapewniają, że klasy pamięci masowej są użyteczne dla menedżera kolejek. Można wyłączyć te kontrole, aby zwiększyć wydajność, lub jeśli nie są one poprawne w danym środowisku.

O tym zadaniu

Kontrole elementów webhook w czasie wykonywania odbywają się w ramach konfiguracji menedżera kolejek. Sprawdzają one, czy klasy pamięci masowej są odpowiednie dla wybranego typu menedżera kolejek.

Można wyłączyć te kontrole, aby zmniejszyć czas potrzebny na utworzenie menedżera kolejek lub jeśli kontrole nie są poprawne w konkretnym środowisku.

Uwaga: Po wyłączeniu kontroli elementu webhook w czasie wykonywania wszystkie wartości klas pamięci masowej są dozwolone. Może to spowodować uszkodzenie menedżera kolejek.

Procedura

- Wyłącz kontrole elementu webhook w czasie wykonywania

Dodaj następującą adnotację w sekcji metadata. Na przykład:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    "com.ibm.cp4i/disable-webhook-runtime-checks" : "true"
```

Wyłączanie aktualizacji wartości domyślnych w specyfikacji menedżera kolejek

Program IBM MQ Operator aktualizuje nieokreślone wartości w specyfikacji menedżera kolejek, używając ich wartości domyślnych. To zachowanie można wyłączyć, aby uniknąć modyfikacji specyfikacji menedżera kolejek. Pola statusu menedżera kolejek są nadal aktualizowane.

Procedura

- Wyłącz aktualizacje wartości domyślnych menedżera kolejek.

Dodaj następującą adnotację w sekcji metadata. Na przykład:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    "com.ibm.mq/write-defaults-spec" : "false"
```

Uwaga: Ta funkcja może być używana tylko z opcją MQ Operator 2.1.0 i nowszymi wersjami. Począwszy od wersji IBM MQ Operator 2.1.0, w przykładach szybkiego startu adnotacja ta jest stosowana domyślnie.

V 9.3.4 Uruchamianie kontenera IBM MQ z systemem plików root tylko do odczytu

W produkcie IBM MQ Operator 3.0.0 i kontenerze IBM MQ 9.3.4.0 można skonfigurować kontener IBM MQ do działania z systemem plików root w trybie tylko do odczytu. Uniemożliwia to atakującym kopiowanie i uruchamianie szkodliwego kodu w kontenerze.

O tym zadaniu

Włączenie systemu plików root w trybie tylko do odczytu powoduje, że pliki kontenera stają się niezmiennalne. Oznacza to, że w systemie plików kontenera można wyświetlać pliki, ale nie można ich modyfikować i nie można tworzyć nowych plików. Pliki mogą być modyfikowane lub tworzone tylko w podłączonym systemie plików.

Jeśli włączony jest główny system plików tylko do odczytu, tworzone są dwa efemeryczne woluminy Scratch i Tmp, które są podłączane odpowiednio w katalogach /run i /tmp w kontenerze.

- Wolumin Scratch zawiera pliki, magazyny kluczy i inne pliki używane do konfigurowania menedżera kolejek.
- Wolumin Tmp zawiera pliki diagnostyczne, na przykład pliki RAS menedżera kolejek.

Ponieważ te woluminy są krótkotrwałe, pliki na tych woluminach są tracone podczas restartowania zasobnika.

Typ woluminu utworzonego dla danych menedżera kolejek zależy od typu pamięci masowej. Domyślnie podłączony jest wolumin trwały. Jeśli typem pamięci masowej jest ephemeral, podłączany jest wolumin krótkotrwały. Jeśli wielkość danych w woluminie przekracza wartość określoną dla właściwości **sizeLimit**, produkt Kubernetes może wysunąć kontener i utworzyć nowy. W wersjach wcześniejszych niż IBM MQ Operator 3.0.0 limit wielkości nie był wymuszany podczas używania pamięci efemerycznej dla danych menedżera kolejek.

System plików root w trybie tylko do odczytu nie jest domyślnie włączony. Aby ją włączyć, wykonaj następujące kroki:

Procedura

1. Użyj funkcji `API spec . securityContext`, aby włączyć system plików root tylko do odczytu.

Dla menedżera kolejek ustaw właściwość **readOnlyRootFilesystem** w pliku [“.spec.securityContext” na stronie 201](#) na wartość `true`.

IBM MQ Operator tworzy dwa woluminy efemeryczne: Scratch i Tmp.

2. Opcjonalne: Ustaw lub zmień typ pamięci masowej danych menedżera kolejek.

Domyślnie opcja Persistent Volume Claim jest podłączona w katalogu `/mnt/mqm`. Jeśli właściwość **type** jest ustawiona na wartość `ephemeral` w pliku [“.spec.queueManager.storage.queueManager” na stronie 199](#), wolumin krótkotrwały jest tworzony i podłączany.

3. Dla każdego woluminu efemerycznego należy dokładnie rozważyć, jak bardzo dane mogą się powiększać. Ustaw odpowiednio wartość właściwości **sizeLimit**, w tym jednostki SI.

- Dla woluminu efemerycznego Scratch ustaw właściwość **sizeLimit** w pliku [“.spec.queueManager.storage.scratch” na stronie 201](#). Wartością domyślną jest "100M".
- Dla woluminu efemerycznego Tmp ustaw właściwość **sizeLimit** w pliku [“.spec.queueManager.storage.tmp” na stronie 201](#). Wartością domyślną jest "2Gi".

- Jeśli właściwość **type** woluminu menedżera kolejek jest ustawiona na wartość `ephemeral`, należy ustawić właściwość **sizeLimit** w pliku `spec.queueManager.storage.queueManager` na stronie 199. Wartością domyślną jest "2Gi".

V 9.3.4

OpenShift

Konfigurowanie serwera IBM MQ Console z podstawowym rejestrem za pomocą konsoli IBM MQ Operator

Aby zalogować się do programu IBM MQ Console, można podać własną konfigurację w menedżerze kolejek.

Zanim rozpoczniesz

W przypadku wdrażania menedżera kolejek z licencją produktu IBM MQ Advanced for Developers wbudowana jest prosta konfiguracja. Patrz "[MQ 9.3.4 gru 2023]Przykładowy menedżer kolejek YAML opisujący sposób określania haseł dla użytkowników produktów admin i app" na stronie 25.

Jeśli wdrażany jest menedżer kolejek licencji na produkt IBM Cloud Pak for Integration, można włączyć integrację z produktem IBM Cloud Pak for Integration Keycloak, aby zalogować się do produktu IBM MQ Console przy użyciu funkcji pojedynczego logowania. Patrz sekcja "Nawiązywanie połączenia z rozwiązaniem IBM MQ Console wdrożonym w klastrze Red Hat OpenShift" na stronie 176.

Procedura

1. Utwórz hasło i zaszyfruj je za pomocą `securityUtility`.

Do przechowywania referencji używanych w celu uzyskania dostępu do menedżera kolejek używany jest program `ConfigMap`. W celu zwiększenia bezpieczeństwa referencje te należy zakodować za pomocą komendy `securityUtility`.

Alternatywnie można użyć klucza tajnego, który chroni referencje w warstwie Kubernetes. Jednak narzędzia do monitorowania i rozwiązywania problemów mogą ujawnić plik bazowy w sposób niezabezpieczony.

2. Opcjonalne: Zaloguj się do interfejsu wiersza komend (CLI) systemu Red Hat OpenShift.

Jeśli używany jest interfejs CLI programu OpenShift, zaloguj się, używając `oc login`.

Alternatywnie można użyć konsoli OpenShift.

3. Utwórz `ConfigMap` z konfiguracją.

Aby uzyskać pomoc w tworzeniu konfiguracji XML, patrz [IBM MQ Console i REST API zabezpieczenia](#).

W poniższym przykładzie tworzony jest użytkownik w grupie `MQWebAdminGroup`. Członkowie grupy `MQWebAdminGroup` mają przypisaną rolę `MQWebAdmin`. W tym przykładzie:

- Użytkownik **musi** zastąpić `USERNAME` i `PASSWORD` własnymi wartościami. Należy zauważyć, że wartość `USERNAME` jest używana w tym przykładzie dwa razy.

Należy określić `PRZESTRZEŃ NAZW` jako tę, w której wdrożony jest produkt IBM MQ Operator i w której będzie lub jest już wdrożony menedżer kolejek.

a) Użyj konsoli OpenShift lub wiersza komend, aby utworzyć następujące `ConfigMap`:

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: mqwebuserconfigmap
  namespace: NAMESPACE
data:
  mqwebuser.xml: |
    <?xml version="1.0" encoding="UTF-8"?>
    <server>
      <featureManager>
        <feature>appSecurity-2.0</feature>
        <feature>basicAuthenticationMQ-1.0</feature>
      </featureManager>
      <enterpriseApplication id="com.ibm.mq.console">
```

```

<application-bnd>
  <security-role name="MQWebAdmin">
    <group name="MQWebAdminGroup" realm="defaultRealm"/>
  </security-role>
</application-bnd>
</enterpriseApplication>
<basicRegistry id="basic" realm="defaultRealm">
  <user name="USERNAME" password="PASSWORD"/>
  <group name="MQWebAdminGroup">
    <member name="USERNAME"/>
  </group>
</basicRegistry>
<sslDefault sslRef="mqDefaultSSLConfig"/>
</server>

```

b) Opcjonalne: Jeśli używany jest wiersz komend, zastosuj ConfigMap:

```
oc apply -f mqwebuserconfigmap.yaml
```

Dla pozostałych kroków wybierz jedną z następujących opcji:

- Wdróż nowy menedżer kolejek z konfiguracją, aby uzyskać dostęp do produktu IBM MQ Console.
- Zastosuj konfigurację, która zapewnia dostęp programu IBM MQ Console do istniejącego menedżera kolejek.

4. Opcjonalne: **Wdróż nowy menedżer kolejek z konfiguracją, aby uzyskać dostęp do produktu IBM MQ Console.**

a) Utwórz menedżer kolejek.

Ustaw dostawców uwierzytelniania i autoryzacji na manual i podaj nowo utworzony ConfigMap mqwebuserconfigmap, stosując jedną z następujących opcji:

- Opcja 1: za pośrednictwem menedżera kolejek YAML

Dodaj następujący kod w sekcji web menedżera kolejek YAML:

```

...
web:
  enabled: true
  console:
    authentication:
      provider: manual
    authorization:
      provider: manual
  manualConfig:
    configMap:
      name: mqwebuserconfigmap

```

- Opcja 2: za pośrednictwem widoku formularza konsoli OpenShift :

- W konsoli OpenShift wybierz opcję **Operatory > Zainstalowane operatory**.
- Wybierz wdrożenie produktu IBM MQ Operator.
- Wybierz opcję **Menedżer kolejek** i kliknij opcję **Utwórz QueueManager**.
- Wybierz odpowiednie opcje dla menedżera kolejek.
- Wybierz opcję **Web** i ustaw opcję **Enable web server** (Włącz serwer WWW) na wartość true.
- Otwórz okno listy **Konfiguracja zaawansowana**.
- W polu listy **Konsola** ustaw **dostawcę** dla opcji **Uwierzytelnianie i Autoryzacja** na wartość ręcznie.
- Otwórz okno listy **Konfiguracja**.
- Otwórz pole listy **ConfigMap** i wybierz opcję ConfigMap mqwebuserconfigmap, która została utworzona w kroku "3" na stronie 174.
- Kliknij opcję **Create**.

Teraz można uzyskać dostęp do pliku IBM MQ Console nowego menedżera kolejek za pomocą referencji określonych w pliku ConfigMap utworzonym w kroku "3" na stronie 174.

5. Opcjonalne: **Zastosuj konfigurację, która włącza IBM MQ Console dla istniejącego menedżera kolejek.**

Edytuj YAML menedżera kolejek, dla którego włączasz IBM MQ Console:

- W konsoli OpenShift wybierz opcję **Operatory > Zainstalowane operatory**.
- Wybierz wdrożenie produktu IBM MQ Operator.
- Wybierz opcję **Menedżer kolejek** i wybierz nazwę menedżera kolejek.
- Wybierz opcję **YAML**.
- Zastąp istniejącą sekcję web w pliku YAML menedżera kolejek następującym kodem:

```
...
web:
  enabled: true
  console:
    authentication:
      provider: manual
    authorization:
      provider: manual
  manualConfig:
    configMap:
      name: mqwebuserconfigmap
```

- Kliknij przycisk **Zapisz**.

Dostęp do pliku IBM MQ Console istniejącego menedżera kolejek można teraz uzyskać za pomocą referencji określonych w pliku ConfigMap utworzonym w kroku [“3”](#) na stronie 174.

Użytkowanie produktu IBM MQ przy wykorzystaniu operatora IBM MQ Operator

Procedura

- “[Nawiązywanie połączenia z rozwiązaniem IBM MQ Console wdrożonym w klastrze Red Hat OpenShift](#)” na stronie 176.
- “[Monitorowanie w przypadku korzystania z operatora IBM MQ Operator](#)” na stronie 178.
- “[Tworzenie i odtwarzanie kopii zapasowej konfiguracji menedżera kolejek przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift](#)” na stronie 183.

Nawiązywanie połączenia z rozwiązaniem IBM MQ Console wdrożonym w klastrze Red Hat OpenShift

W tej sekcji opisano nawiązywanie połączenia z bazą danych IBM MQ Console menedżera kolejek, który został wdrożony w klastrze Red Hat OpenShift Container Platform .

O tym zadaniu

Adres URL IBM MQ Console można znaleźć na stronie szczegółów QueueManager konsoli WWW produktu Red Hat OpenShift lub w IBM Cloud Pak for Integration Platform UI (previously the Platform Navigator). Alternatywnie można go znaleźć w interfejsie wiersza komend produktu Red Hat OpenShift, uruchamiając następującą komendę:

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

Jeśli używana jest licencja na produkt IBM Cloud Pak for Integration :

- W przypadku produktu IBM MQ Operator 3.0.0 i nowszych produkt IBM MQ Console używa produktu Keycloak do zarządzania tożsamością i dostępem. Patrz sekcja [Zarządzanie tożsamością i dostępem](#) w dokumentacji produktu IBM Cloud Pak for Integration .

- W przypadku wdrożeń IBM MQ Operator w wersji wcześniejszej niż 3.0.0 produkt IBM MQ Console używa modułu IBM Cloud Pak Identity and Access Manager (IAM). Być może komponent IAM został już skonfigurowany przez administratora klastra. Jeśli jednak jest to pierwsze użycie IAM w klastrze Red Hat OpenShift, należy pobrać początkowe hasło administratora. Patrz sekcja [Uzyskiwanie początkowego hasła administratora](#).

Jeśli używana jest licencja na produkt IBM MQ, produkt IBM MQ Console nie jest wstępnie skonfigurowany i należy go skonfigurować samodzielnie. Więcej informacji na ten temat zawiera sekcja [Konfigurowanie użytkowników i ról](#). Przykład zawiera sekcja ["Konfigurowanie serwera IBM MQ Console z podstawowym rejestrem za pomocą konsoli IBM MQ Operator"](#) na stronie 174.

Zadania pokrewne

["Konfigurowanie trasy w celu nawiązania połączenia z menedżerem kolejek spoza klastra produktu Red Hat OpenShift"](#) na stronie 159

Do połączenia aplikacji z menedżerem kolejek IBM MQ spoza klastra Red Hat OpenShift potrzebna jest trasa Red Hat OpenShift. Należy włączyć protokół TLS w menedżerze kolejek i aplikacji klienckiej IBM MQ, ponieważ protokół SNI jest dostępny w protokole TLS tylko wtedy, gdy używany jest protokół TLS w wersji 1.2 lub nowszej. Red Hat OpenShift Container Platform Router używa wskaźnika SNI do kierowania żądań do menedżera kolejek IBM MQ.

Nadawanie uprawnień dla IBM MQ Console za pomocą IBM Cloud Pak IAM

Uprawnienia do produktu IBM MQ Console są zarządzane za pomocą produktu IBM Cloud Pak Administration Hub, a nie za pomocą produktu IBM Cloud Pak for Integration Platform UI (previously the Platform Navigator). IBM MQ nie używa uprawnień "Automatyzacja" udostępnianych przez IBM Cloud Pak for Integration, ale używa uprawnień podstawowych włączonych przez produkt IBM Cloud Pak Identity and Access Manager (IAM).

Procedura

1. Otwórz Konsolę administracyjną IBM Cloud Pak.
 - W programie IBM Cloud Pak for Integration Platform UI kliknij przełącznik Cloud Pak (ikona 9-punktowa) w prawym górnym rogu paska narzędzi, a następnie kliknij panel **IBM Cloud Pak Administration**.
2. W menu nawigacyjnym w lewym górnym rogu wybierz opcję **Tożsamość i dostęp**, a następnie wybierz opcję **Identyfikatory zespołów i usług**.
3. Utwórz zespół, a następnie dodaj do niego użytkowników.
 - a) Wybierz opcję **Utwórz zespół**.
 - b) Wprowadź nazwę zespołu, a następnie wybierz domenę zabezpieczeń dla użytkowników, którzy mają być zarządzani.
 - c) Wyszukaj użytkowników.
 - Ci użytkownicy muszą już istnieć w dostawcy tożsamości.
 - d) Po znalezieniu każdego użytkownika nadaj mu rolę. Aby można było administrować produktem IBM MQ za pomocą konsoli IBM MQ Console, musi to być wartość Administrator lub Administrator klastra.
4. Dodaj każdego użytkownika do przestrzeni nazw.
 - a) Wybierz zespół do edycji.
 - b) Wybierz opcję **Zasoby > Zarządzaj zasobami**.
 - c) Wybierz przestrzeń nazw, którymi ten zespół ma administrować. Mogą to być dowolne przestrzenie nazw z menedżerem kolejek.

MQ Operator

Menedżery kolejek zarządzane przez IBM MQ Operator mogą generować metryki kompatybilne z rozwiązaniem Prometheus.

Można wyświetlić te wielkości mierzone za pomocą stosu monitorowania Red Hat OpenShift Container Platform (OCP). Otwórz kartę **Metryki** w programie OCP, a następnie kliknij opcję **Obserwuj > Metryki**. Wielkości mierzone menedżera kolejek są domyślnie włączone, ale można je wyłączyć, ustawiając wartość parametru **.spec.metrics.enabled** na `false`.

Prometheus jest bazą danych szeregów czasowych i mechanizmem oceny reguł dotyczących metryk. Kontenery IBM MQ eksponują punkt końcowy metryk, który może być odpytywany przez narzędzie Prometheus. Metryki generowane na podstawie tematów systemu MQ na potrzeby monitorowania i śledzenia aktywności.

OpenShift Container Platform zawiera wstępnie skonfigurowany, wstępnie zainstalowany i samoaktualizujący się stos monitorowania wykorzystujący serwer Prometheus. Stos monitorowania OpenShift Container Platform musi być skonfigurowany do monitorowania projektów zdefiniowanych przez użytkownika. Więcej informacji na ten temat zawiera sekcja [Włączanie monitorowania projektów zdefiniowanych przez użytkownika](#). IBM MQ Operator tworzy `ServiceMonitor` podczas tworzenia `QueueManager` z włączonymi metrykami, które element wykonawczy programu Prometheus może następnie wykryć.

W starszych wersjach produktu IBM Cloud Pak for Integration można także użyć usługi [IBM Cloud Platform Monitoring](#), aby udostępnić serwer Prometheus.

OpenShift CP4I **Metryki publikowane, gdy używany jest IBM MQ Operator**

Kontenery menedżera kolejek mogą publikować metryki zgodne z programem Red Hat OpenShift Monitoring.

Pomiar	Typ	Opis
<code>ibmmq_qmgr_commit_total</code>	counter	Liczba zatwierdzeń
<code>ibmmq_qmgr_cpu_load_fifteen_minute_average_percentage</code>	gauge	Obciążenie procesora - średnia dla piętnastu minut
<code>ibmmq_qmgr_cpu_load_five_minute_average_percentage</code>	gauge	Obciążenie procesora - średnia dla pięciu minut
<code>ibmmq_qmgr_cpu_load_one_minute_average_percentage</code>	gauge	Obciążenie procesora - średnia dla jednej minuty
<code>ibmmq_qmgr_destructive_get_bytes_total</code>	counter	Łączna liczba bajtów niszczących operacji get w okresie
<code>ibmmq_qmgr_destructive_get_total</code>	counter	Łączna liczba niszczących operacji get w okresie
<code>ibmmq_qmgr_durable_subscription_alter_total</code>	counter	Liczba operacji zmiany trwałej subskrypcji
<code>ibmmq_qmgr_durable_subscription_create_total</code>	counter	Liczba operacji tworzenia trwałej subskrypcji

Pomiar	Typ	Opis
ibmmq_qmgr_durable_subscription_delete_total	counter	Liczba operacji usuwania trwałej subskrypcji
ibmmq_qmgr_durable_subscription_resume_total	counter	Liczba operacji wznowienia trwałej subskrypcji
ibmmq_qmgr_errors_file_system_free_space_percentage	gauge	System plików błędów MQ - wolne miejsce
ibmmq_qmgr_errors_file_system_in_use_bytes	gauge	System plików błędów MQ - bajty w użyciu
ibmmq_qmgr_expired_message_total	counter	Liczba komunikatów, które utraciły ważność
ibmmq_qmgr_failed_browse_total	counter	Liczba operacji przeglądania zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqcb_total	counter	Liczba operacji MQCB zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqclose_total	counter	Liczba operacji MQCLOSE zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqconn_mqconnx_total	counter	Liczba operacji MQCONN lub MQCONNX zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqget_total	counter	Liczba operacji MQGET zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqinq_total	counter	Liczba operacji MQINQ zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqopen_total	counter	Liczba operacji MQOPEN zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqput1_total	counter	Liczba operacji MQPUT1 zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqput_total	counter	Liczba operacji MQPUT zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqset_total	counter	Liczba operacji MQSET zakończonych niepowodzeniem
ibmmq_qmgr_failed_mqsubrq_total	counter	Liczba operacji MQSUBRQ zakończonych niepowodzeniem
ibmmq_qmgr_failed_subscription_create_alter_resume_total	counter	Liczba operacji tworzenia/zmiany/wznowienia subskrypcji zakończonych niepowodzeniem
ibmmq_qmgr_failed_subscription_delete_total	counter	Liczba operacji usuwania subskrypcji zakończonych niepowodzeniem

Pomiar	Typ	Opis
ibmmq_qmgr_failed_topic_mqput_mqput1_total	counter	Liczba operacji MQPUT lub MQPUT1 dla tematów zakończonych niepowodzeniem
ibmmq_qmgr_fdc_files	gauge	Liczba plików FDC produktu MQ
ibmmq_qmgr_log_file_system_in_use_bytes	gauge	System plików dzienników - bajty w użyciu
ibmmq_qmgr_log_file_system_max_bytes	gauge	System plików dzienników - maksymalna liczba bajtów
ibmmq_qmgr_log_in_use_bytes	gauge	Dziennik - bajty w użyciu
ibmmq_qmgr_log_logical_written_bytes_total	counter	Dziennik - liczba zapisanych bajtów logicznych
ibmmq_qmgr_log_max_bytes	gauge	Dziennik - maksymalna liczba bajtów
ibmmq_qmgr_log_occupied_by_reusable_extents_bytes	gauge	Dziennik – liczba bajtów zajmowanych przez zakresy wielokrotnego użytku
ibmmq_qmgr_log_physical_written_bytes_total	counter	Dziennik - liczba zapisanych bajtów fizycznych
ibmmq_qmgr_log_primary_space_in_use_percentage	gauge	Dziennik – bieżące wykorzystanie przestrzeni podstawowej
ibmmq_qmgr_log_required_for_media_recovery_bytes	gauge	Dziennik – liczba bajtów wymaganych do odtwarzania nośnika
ibmmq_qmgr_log workload_primary_space_utilization_percentage	gauge	Dziennik – obciążeniowe wykorzystanie przestrzeni podstawowej
ibmmq_qmgr_log_write_latency_seconds	gauge	Dziennik - opóźnienie zapisu
ibmmq_qmgr_log_write_size_bytes	gauge	Dziennik - wielkość zapisu
ibmmq_qmgr_mqcb_total	counter	Liczba operacji MQCB
ibmmq_qmgr_mqclose_total	counter	Liczba operacji MQCLOSE
ibmmq_qmgr_mqconn_mqconnx_total	counter	Liczba operacji MQCONN lub MQCONNX
ibmmq_qmgr_mqctl_total	counter	Liczba operacji MQCTL

Pomiar	Typ	Opis
ibmmq_qmgr_mqdisc_total	counter	Liczba operacji MQDISC
ibmmq_qmgr_mqinq_total	counter	Liczba operacji MQINQ
ibmmq_qmgr_mqopen_total	counter	Liczba operacji MQOPEN
ibmmq_qmgr_mqput_mqput1_bytes_total	counter	Łączna liczba bajtów operacji MQPUT lub MQPUT1 w okresie
ibmmq_qmgr_mqput_mqput1_total	counter	Łączna liczba operacji MQPUT lub MQPUT1 w okresie
ibmmq_qmgr_mqset_total	counter	Liczba operacji MQSET
ibmmq_qmgr_mqstat_total	counter	Liczba operacji MQSTAT
ibmmq_qmgr_mqsubrq_total	counter	Liczba operacji MQSUBRQ
ibmmq_qmgr_non_durable_subscription_create_total	counter	Liczba operacji tworzenia nietrwalej subskrypcji
ibmmq_qmgr_non_durable_subscription_delete_total	counter	Liczba operacji usuwania nietrwalej subskrypcji
ibmmq_qmgr_non_persistent_message_browse_bytes_total	counter	Przeglądanie komunikatów nietrwalej - liczba bajtów
ibmmq_qmgr_non_persistent_message_browse_total	counter	Przeglądanie komunikatów nietrwalej - liczba
ibmmq_qmgr_non_persistent_message_destructive_get_total	counter	Liczba niszczących operacji get dla komunikatów nietrwalej
ibmmq_qmgr_non_persistent_message_get_bytes_total	counter	Otrzymane komunikaty nietrwale - liczba bajtów
ibmmq_qmgr_non_persistent_message_mqput1_total	counter	Liczba operacji MQPUT1 dla komunikatów nietrwalej
ibmmq_qmgr_non_persistent_message_mqput_total	counter	Liczba operacji MQPUT dla komunikatów nietrwalej
ibmmq_qmgr_non_persistent_message_put_bytes_total	counter	Komunikaty nietrwale operacji put - liczba bajtów

Pomiar	Typ	Opis
ibmmq_qmgr_non_persistent_topic_mqput_mqput1_total	counter	Nietrwałe - liczba operacji MQPUT lub MQPUT1 dla tematów
ibmmq_qmgr_persistent_message_browse_bytes_total	counter	Przeglądanie komunikatów trwałych - liczba bajtów
ibmmq_qmgr_persistent_message_browse_total	counter	Przeglądanie komunikatów trwałych - liczba
ibmmq_qmgr_persistent_message_destructive_get_total	counter	Liczba niszczących operacji get dla komunikatów trwałych
ibmmq_qmgr_persistent_message_get_bytes_total	counter	Otrzymane komunikaty trwałe - liczba bajtów
ibmmq_qmgr_persistent_message_mqput1_total	counter	Liczba operacji MQPUT1 dla komunikatów trwałych
ibmmq_qmgr_persistent_message_mqput_total	counter	Liczba operacji MQPUT dla komunikatów trwałych
ibmmq_qmgr_persistent_message_put_bytes_total	counter	Komunikaty trwałe operacji put - liczba bajtów
ibmmq_qmgr_persistent_topic_mqput_mqput1_total	counter	Trwałe - liczba operacji MQPUT lub MQPUT1 dla tematów
ibmmq_qmgr_published_to_subscribers_bytes_total	counter	Liczba bajtów opublikowanych do subskrybentów
ibmmq_qmgr_published_to_subscribers_message_total	counter	Liczba komunikatów opublikowanych do subskrybentów
ibmmq_qmgr_purged_queue_total	counter	Liczba wyczyszczonych kolejek
ibmmq_qmgr_queue_manager_file_system_free_space_percentage	gauge	System plików menedżera kolejek - wolne miejsce
ibmmq_qmgr_queue_manager_file_system_in_use_bytes	gauge	System plików menedżera kolejek - bajty w użyciu
ibmmq_qmgr_ram_free_percentage	gauge	Procent wolnej pamięci RAM
ibmmq_qmgr_ram_usage_estimate_for_queue_manager_bytes	gauge	Łączna liczba bajtów pamięci RAM - szacunkowa wartość dla menedżera kolejek

Pomiar	Typ	Opis
ibmmq_qmgr_rollback_total	counter	Liczba wycofań zmian
ibmmq_qmgr_system_cpu_time_estimate_for_queue_manager_percentage	gauge	Czas pracy procesora dla systemu - szacunkowy procent dla menedżera kolejek
ibmmq_qmgr_system_cpu_time_percentage	gauge	Procent czasu pracy procesora dla systemu
ibmmq_qmgr_topic_mqput_mqput1_total	counter	Łączna liczba operacji MQPUT lub MQPUT1 dla tematów w okresie
ibmmq_qmgr_topic_put_bytes_total	counter	Łączna liczba bajtów operacji put w okresie
ibmmq_qmgr_trace_file_system_free_space_percentage	gauge	System plików śledzenia MQ - wolne miejsce
ibmmq_qmgr_trace_file_system_in_use_bytes	gauge	System plików śledzenia MQ - bajty w użyciu
ibmmq_qmgr_user_cpu_time_estimate_for_queue_manager_percentage	gauge	Czas pracy procesora dla użytkownika - szacunkowy procent dla menedżera kolejek
ibmmq_qmgr_user_cpu_time_percentage	gauge	Procent czasu pracy procesora dla użytkownika

Informacje pokrewne

Pomiary opublikowane w tematach dotyczących systemu

Tworzenie i odtwarzanie kopii zapasowej konfiguracji menedżera kolejek przy użyciu interfejsu wiersza komend produktu Red Hat OpenShift

Utworzenie kopii zapasowej konfiguracji menedżera kolejek może pomóc w odbudowaniu menedżera na podstawie jego definicji, jeśli konfiguracja zostanie utracona. Ta procedura nie powoduje utworzenia kopii zapasowej danych dziennika menedżera kolejek. Ze względu na przejściowy charakter komunikatów dane w dzienniku historycznym mogą być nieistotne w momencie odtwarzania.

Zanim rozpoczniesz

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

Procedura

- Utwórz kopię zapasową konfiguracji menedżera kolejek.
 - Do wykonania zrzutu konfiguracji menedżera kolejek systemu IBM MQ można użyć komendy **dmpmqcfg**.
 - a) Uzyskaj nazwę zasobnika dla menedżera kolejek.

Na przykład można uruchomić następującą komendę, gdzie *nazwa_menedzera_kolejek* jest nazwą zasobu QueueManager:

```
oc get pods --selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/instance=queue_manager_name
```

b) Uruchom komendę **dmpmqcfig** w zasobniku, kierując wyniki do pliku na komputerze lokalnym.

dmpmqcfig generuje na wyjściu konfigurację MQSC menedzera kolejek.

```
oc exec -it pod_name -- dmpmqcfig > backup.mqsc
```

- Odtwórz konfigurację menedzera kolejek.

Po wykonaniu procedury tworzenia kopii zapasowej opisanej w poprzednim kroku powinien być dostępny plik `backup.mqsc` zawierający konfigurację menedzera kolejek. Konfigurację tę można odtworzyć, stosując ten plik do nowego menedzera kolejek.

a) Uzyskaj nazwę zasobnika dla menedzera kolejek.

Na przykład można uruchomić następującą komendę, gdzie *nazwa_menedzera_kolejek* jest nazwą zasobu QueueManager:

```
oc get pods --selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/instance=queue_manager_name
```

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

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

OpenShift CP4I Rozwiązywanie problemów z produktem IBM MQ

Operator

W przypadku wystąpienia problemów związanych z programem IBM MQ Operator należy użyć opisanych technik w celu znalezienia i rozwiązania tych problemów.

Procedura

- [“Gromadzenie informacji o rozwiązywaniu problemów dla menedżerów kolejek wdrożonych za pomocą programu IBM MQ Operator” na stronie 184](#)
- [“Rozwiązywanie problemów: uzyskiwanie dostępu do danych menedzera kolejek” na stronie 186](#)

OpenShift CP4I Gromadzenie informacji o rozwiązywaniu problemów dla menedżerów kolejek wdrożonych za pomocą programu IBM MQ Operator

Gromadzenie informacji o rozwiązywaniu problemów, które powinny być udostępniane działowi wsparcia IBM podczas tworzenia nowej sprawy dotyczącej wsparcia.

Procedura

1. Zgromadź informacje o dostawcy chmury.

Jest to dostawca chmury, który udostępnia klaster Red Hat OpenShift (na przykład IBM Cloud).

2. Zgromadź informacje o architekturze.

Architektura klastra Red Hat OpenShift może być następująca:

- Linux for x86-64
- Linux on Power Systems (ppc64le)
- Linux for IBM Z

3. Zgromadź informacje o wdrożeniu IBM MQ .

a) Zaloguj się do klastra Red Hat OpenShift , używając powłoki bash/zsh .

b) Ustaw następujące zmienne środowiskowe:

```
export QM=QueueManager_name
export QM_NAMESPACE=QueueManager_namespace
export MQ_OPERATOR_NAMESPACE=mq_operator_namespace
```

Gdzie *nazwa_menedżera_kolejek* jest nazwą zasobu QueueManager , *przestrzeń_nazw* jest przestrzenią nazw, w której jest wdrożony, a *przestrzeń_nazw* jest przestrzenią nazw, w której jest wdrożony IBM MQ Operator . Może być taka sama, jak przestrzeń nazw QueueManager .

c) Uruchoom następujące komendy i przekaż wszystkie wynikowe pliki wyjściowe do działu wsparcia IBM .

```
# OCP / Kubernetes: Version
oc version -o yaml > ocversion.yaml

# QueueManager: YAML
oc get qmgr $QM -n $QM_NAMESPACE -o yaml > "queue-manager-$QM.yaml"

# MQ Queue Manager: Pods
oc get pods -n $QM_NAMESPACE -o wide --selector "app.kubernetes.io/instance=$QM" > "qm-pods-$QM.txt"

# MQ Queue Manager: Pod YAML
oc get pods -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-pods-$QM.yaml"

# MQ Queue Manager: Pod Logs
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/instance=$QM" | cut -d ' ' -f 1); do oc logs -n $QM_NAMESPACE --previous "$p" > "qm-logs-previous-$p.txt"; oc logs -n $QM_NAMESPACE $p > "qm-logs-$p.txt";done

# MQ Web UI: Console Log
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/instance=$QM" | cut -d ' ' -f 1); do oc cp -n $QM_NAMESPACE --retries=10 "$p:var/mqm/web/installations/Installation1/servers/mqweb/logs/console.log" "web-$p-console.log"; done

# MQ Web UI: Messages Log
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/instance=$QM" | cut -d ' ' -f 1); do oc cp -n $QM_NAMESPACE --retries=10 "$p:var/mqm/web/installations/Installation1/servers/mqweb/logs/messages.log" "web-$p-messages.log"; done

# MQ Queue Manager: routes defined by operator
oc get routes -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-routes-$QM.yaml"

# MQ Queue Manager: routes to QM
oc get routes -n $QM_NAMESPACE -o yaml --field-selector "spec.to.name=$QM-ibm-mq" > "qm-routes2-$QM.yaml"

# MQ Queue Manager: stateful set
oc get statefulset -n $QM_NAMESPACE -o yaml ${QM}-ibm-mq > "qm-statefulset-$QM.yaml"

# MQ Queue Manager: services
oc get services -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-services-$QM.yaml"

# MQ Queue Manager: PVCs
oc get pvc -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-pvcs-$QM.yaml"

# MQ Operator: Version
oc get csv -n $QM_NAMESPACE | grep "^ibm-mq\\|NAME" > mq-operator-csv.txt

# Cloud Pak Foundational Services: Version
oc get csv -n $QM_NAMESPACE | grep "^ibm-common-service-operator\\|NAME" > common-services-csv.txt

# Cloud Pak for Integration: Version (if applicable)
oc get csv -n $QM_NAMESPACE | grep "^ibm-integration-platform-navigator\\|NAME" > cp4i-csv.txt

# Output from runmqras (this may take a while to execute)
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/instance=$QM" | cut -d ' ' -f 1); do timestamp=$(TZ=UTC date +"%Y%m%d_%H%M%S");
oc exec -n $QM_NAMESPACE $p -- runmqras -workdirectory "/tmp/runmqras_$timestamp"
-section logger,mqweb,nativeha,trace; oc cp -n $QM_NAMESPACE --retries=10 "$p:tmp/
```

```
runmqras_$timestamp/" .; done

# MQ Operator: Pod Log
oc logs -n $MQ_OPERATOR_NAMESPACE $(oc get pods -n $MQ_OPERATOR_NAMESPACE --no-headers
--selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/managed-by=olm | cut -d ' ' -f
1) > mq-operator-log.txt
```

Uwaga:

Większość tych komend wymaga dostępu do przestrzeni nazw, w której wdrożony jest menedżer kolejek. Jednak gromadzenie dziennika IBM MQ Operator może dodatkowo wymagać dostępu **administratora klastra** , jeśli IBM MQ Operator jest zainstalowany **w zasięgu klastra**.

Zadania pokrewne

[Gromadzenie informacji o rozwiązywaniu problemów dla działu wsparcia IBM](#)

Rozwiązywanie problemów: uzyskiwanie dostępu do danych menedżera kolejek

Użyj narzędzia sprawdzającego PVC, aby uzyskać dostęp do plików w PVC menedżera kolejek, w którym nie można ustanowić zdalnej powłoki dla zasobnika menedżera kolejek. Może to być spowodowane tym, że zasobnik jest w stanie **Error** lub **CrashLoopBackOff** . To narzędzie jest przeznaczone do użycia z menedżerami kolejek wdrożonymi przez program IBM MQ Operator.

Zanim rozpocznie

Aby użyć narzędzia sprawdzającego PVC. Użytkownik musi mieć dostęp do przestrzeni nazw menedżera kolejek.

O tym zadaniu

Aby ułatwić rozwiązywanie problemów, można uzyskać dostęp do danych przechowywanych w PVC (Persistent Volume Claims) powiązanych z danym menedżerem kolejek. W tym celu należy użyć narzędzia do podłączenia PVC do zestawu zasobników inspektora. Następnie można umieścić zdalną powłokę w dowolnym zasobnikach inspektora, aby odczytać pliki.

W zależności od typu wdrożenia tworzone są od jednego do trzech zasobników inspektora. Woluminy specyficzne dla danego zasobnika rodzimego menedżera kolejek wysokiej dostępności lub menedżera kolejek z wieloma instancjami są dostępne w powiązonym zasobniku kontrolera PVC. Współużytkowane woluminy są dostępne dla wszystkich inspektorów. Nazwa zasobnika inspektora zawiera nazwę powiązanego zasobnika menedżera kolejek.

Procedura

1. Pobierz narzędzie inspektora PVC MQ .

Narzędzie jest dostępne pod adresem: <https://github.com/ibm-messaging/mq-pvc-tool>.

2. Upewnij się, że jesteś zalogowany w klastrze.
3. Znajdź nazwę menedżera kolejek i przestrzeń nazw, w której działa menedżer kolejek.
4. Uruchom narzędzie Inspector dla menedżera kolejek.
 - a) Uruchom następującą komendę, podając nazwę menedżera kolejek i jego nazwę przestrzeni nazw.

```
./pvc-tool.sh queue_manager_name queue_manager_namespace_name
```

- b) Po zakończeniu działania narzędzia uruchom następującą komendę, aby wyświetlić tworzone zasobniki inspektora.

```
oc get pods
```

5. Wyświetl pliki podłączone do zasobnika inspektora.

- a) Każdy zasobnik inspektora PVC jest powiązany z zasobnikiem menedżera kolejek, dlatego może istnieć wiele zasobników inspektora. Uzyskaj dostęp do jednego z tych zasobników, uruchamiając następującą komendę:

```
oc rsh pvc-inspector-pod-name
```

Zostanie umieszczony w katalogu zawierającym podłączone katalogi PVC.

- b) Wyświetl listę katalogów PVC, uruchamiając następującą komendę:

```
ls
```

- c) Wyświetl listę kanałów PVC, uruchamiając następującą komendę poza sesją zdalnej powłoki:

```
oc get pvc
```

- d) Wyczyść zasobniki utworzone przez narzędzie, uruchamiając następującą komendę:

```
oc delete pods -l tool=mq-pvc-inspector
```

OpenShift CP4I Skorowidz interfejsu API operatora IBM MQ Operator

Produkt IBM MQ udostępnia operatora Kubernetes, który zapewnia rodzimą integrację z produktem Red Hat OpenShift Container Platform.

OpenShift CP4I Skorowidz interfejsu API mq.ibm.com/v1beta1

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

OpenShift CP4I-LTS CP4I CD Skorowidz informacji dotyczących licencjonowania mq.ibm.com/v1beta1

Bieżące wersje licencji

Pole `spec.license.license` musi zawierać identyfikator licencji dla akceptowanej licencji. Poprawne wartości są następujące:

Wartość <code>spec.license.license</code>	Wartość <code>spec.license.use</code>	Informacje o licencji	Odpowiednie wersje IBM MQ
L-VTPK-22YZPK	Production lub NonProduction	IBM Cloud Pak for Integration 2023.4.1	9.3.4 lub 9.3.5
L-QYQF-8UFZBN	Production lub NonProduction	IBM Cloud Pak for Integration Wydanie ograniczone 2023.4.1	9.3.4 lub 9.3.5
L-AMRD-XH6P3Q	Production	IBM MQ Advanced i IBM MQ Advanced dla środowiska pozaprodukcyjnego 9.3-05/2023	9.3.3, 9.3.4 lub 9.3.5
L-AXAF-JLZ53A	Development	IBM MQ Advanced for Developers (bez gwarancji) 9.3-05/2023	9.3.3, 9.3.4 lub 9.3.5
L-YBXJ-ADJNSM	Production lub NonProduction	IBM Cloud Pak for Integration 2023.2.1	9.3.3
L-PYRA-849GYQ	Production lub NonProduction	IBM Cloud Pak for Integration Wydanie ograniczone 2023.2.1	9.3.3
L-RJON-CJR2RX	Production lub NonProduction	IBM Cloud Pak for Integration 2022.4.1	9.3.1 lub 9.3.2

Wartość spec.license.license	Wartość spec.license.use	Informacje o licencji	Odpowiednie wersje IBM MQ
L-RJON-CJR2TC	Production lub NonProduction	IBM Cloud Pak for Integration Wydanie ograniczone 2022.4.1	9.3.1 lub 9.3.2
L-UPFX-8MW49T	Production	IBM MQ Advanced i IBM MQ Advanced dla środowiska pozaprodukcyjnego 9.3-02/2023	9.3.2
L-APIG-CAUEQC	Development	IBM MQ Advanced for Developers (Brak gwarancji) 9.3	9.3.0, 9.3.1 lub 9.3.2
L-RJON-CD3JKX	Production lub NonProduction	IBM Cloud Pak for Integration 2022.2.1	9.3.0 lub 9.3.1
L-RJON-CD3JJU	Production lub NonProduction	IBM Cloud Pak for Integration Wydanie ograniczone 2022.2.1	9.3.0 lub 9.3.1
L-APIG-CAUEBE	Production	IBM MQ Advanced i IBM MQ Advanced dla środowiska pozaprodukcyjnego 9.3	9.3.0 lub 9.3.1

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

Starsze wersje licencji

Więcej informacji na ten temat zawiera sekcja [Starsze wersje licencji](#) w dokumentacji produktu IBM MQ 9.2.

  **Skorowidz interfejsu API menedżera kolejek (mq.ibm.com/v1beta1)**

QueueManager

Menedżer kolejek (QueueManager) jest serwerem IBM MQ, który udostępnia aplikacjom usługi kolejowania i publikowania/subskrypcji. IBM MQ : <https://ibm.biz/BdPZqj>. Informacje dodatkowe o licencji: <https://ibm.biz/BdPZfq>.

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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources .
kind string	Kind to wartość łańcuchowa reprezentująca zasób REST reprezentowany przez ten obiekt. Serwery mogą wywodzić go z punktu końcowego, do którego klient wysyła żądania. Nie można go zmienić. MieszanaWielkośćLiter. Więcej informacji: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds .
metadata	
spec QueueManagerSpec	Pożądaný stan menedżera kolejek.
status QueueManagerStatus	Obserwowany stan menedżera kolejek.

.spec

Pożądaný stan menedżera kolejek.

Występuje w:

- [“QueueManager” na stronie 188](#)

Pole	Opis
affinity	Standardowe reguły powinowactwa Kubernetes. Aby uzyskać więcej informacji, patrz https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#affinity-v1-core .
annotations <u>Adnotacje</u>	Pole annotations służy do przekazywania adnotacji zasobnika. Użytkownik może dodać dowolną adnotację do tego pola, a zostanie ona zastosowana do zasobnika. Ewentualne adnotacje określone w tym miejscu zastępują domyślne adnotacje. Wymaga operatora MQ w wersji 1.3.0 lub nowszej.
imagePullSecrets <u>LocalObjectReference</u> array	Opcjonalna lista odwołań do danych tajnych w tej samej przestrzeni nazw, które mają być używane do pobierania obrazów używanych przez ten menedżer kolejek. Te dane tajne, jeśli zostaną określone, będą przekazywane do indywidualnych implementacji programów pobierających. Na przykład w przypadku środowiska Docker honorowane są dane tajne typu DockerConfig. Więcej informacji: https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod .
labels <u>Etykiety</u>	Pole labels służy do przekazywania etykiet zasobnika. Użytkownik może dodać dowolną etykietę do tego pola, a zostanie ona zastosowana do zasobnika. Ewentualne etykiety określone w tym miejscu zastępują domyślne etykiety. Wymaga operatora MQ w wersji 1.3.0 lub nowszej.
license <u>License</u>	Ustawienia sterujące akceptacją licencji i wyborem metryk licencji, które mają być używane.
pki <u>PKI</u>	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 <u>QueueManagerConfig</u>	Ustawienia dla kontenera menedżera kolejek i bazowego menedżera kolejek.
securityContext <u>SecurityContext</u>	Ustawienia zabezpieczeń, które mają zostać dodane do elementu securityContext zasobnika menedżera kolejek.
telemetry <u>Telemetria</u>	Ustawienia dla konfiguracji Open Telemetry. Wymaga produktu MQ Operator 2.2.0 lub nowszego.
template <u>Template</u>	Zaawansowane użycie szablonów zasobów Kubernetes. Szablon umożliwia użytkownikom zmianę sposobu generowania przez IBM MQ bazowych zasobów Kubernetes, takich jak zestawy stanowe (StatefulSet), zasobniki (Pod) i usługi (Service). Ta możliwość przeznaczona jest tylko dla zaawansowanych użytkowników, ponieważ jej niepoprawne wykorzystanie może zakłócić normalne działanie produktu MQ. Wszystkie wartości określone w menedżerze kolejek zostaną przestonięte przez odpowiednie ustawienia w szablonie.
terminationGracePeriod <u>Seconds</u> integer	Opcjonalny czas trwania (w sekundach), po którym zasobnik musi prawidłowo zakończyć działanie. Wartość musi być nieujemną liczbą całkowitą. Wartość zero oznacza natychmiastowe usunięcie. Docelowy czas, w którym podejmowane są próby zakończenia działania menedżera kolejek z kolejno eskalowanymi fazami odłączenia aplikacji. W razie potrzeby przerywane są zasadnicze zadania obsługi kontenera. Wartością domyślną jest 30 sekund.

Pole	Opis
tracing TracingConfig	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 spowoduje określenie produktu MQ w wersji 9.1.5.0 przy użyciu drugiej podwersji obrazu kontenera. W korektach wprowadzane są często poprawki specyficzne dla kontenera, np. poprawki do obrazu podstawowego.
web WebServerConfig	Ustawienia serwera WWW produktu MQ.

.spec.annotations

Pole annotations służy do przekazywania adnotacji zasobnika. Użytkownik może dodać dowolną adnotację do tego pola, a zostanie ona zastosowana do zasobnika. Ewentualne adnotacje określone w tym miejscu zastępują domyślne adnotacje. Wymaga operatora MQ w wersji 1.3.0 lub nowszej.

Występuje w:

- [“.spec” na stronie 189](#)

.spec.imagePullSecrets

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

Występuje w:

- [“.spec” na stronie 189](#)

Pole	Opis
name string	Nazwa obiektu przywołującego. Więcej informacji: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names DO ZROBIENIA: Dodać inne przydatne pola. apiVersion, kind, uid?.

.spec.labels

Pole labels służy do przekazywania etykiet zasobnika. Użytkownik może dodać dowolną etykietę do tego pola, a zostanie ona zastosowana do zasobnika. Ewentualne etykiety określone w tym miejscu zastępują domyślne etykiety. Wymaga operatora MQ w wersji 1.3.0 lub nowszej.

Występuje w:

- [“.spec” na stronie 189](#)

.spec.license

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

Występuje w:

- [“.spec” na stronie 189](#)

Pole	Opis
accept boolean	Określa, czy użytkownik akceptuje licencję powiązaną z tym oprogramowaniem (wymagane).

Pole	Opis
license string	Identyfikator akceptowanej licencji. Musi to być poprawny identyfikator licencji na używaną wersję produktu MQ. Poprawne wartości można znaleźć w sekcji https://ibm.biz/BdPZfq .
metric string	Ustawienie określające, która metryka licencji ma być używana. Na przykład ProcessorValueUnit, VirtualProcessorCore lub ManagedVirtualServer. Wartością domyślną jest ProcessorValueUnit w przypadku korzystania z licencji na MQ i VirtualProcessorCore w przypadku korzystania z licencji na Cloud Pak for Integration.
use string	Ustawienie określające zastosowanie oprogramowania, gdy licencja dopuszcza więcej niż jedno zastosowanie. Poprawne wartości można znaleźć w sekcji https://ibm.biz/BdPZfq .

.spec.pki

Ustawienia infrastruktury klucza publicznego, służące do definiowania kluczy i certyfikatów, które mają być używane z protokołem TLS (Transport Layer Security) lub produktem MQ Advanced Message Security (AMS).

Występuje w:

- [“.spec”](#) na stronie 189

Pole	Opis
keys PKISource array	Klucze prywatne do dodania do repozytorium kluczy menedżera kolejek.
ttrust PKISource array	Certyfikaty do dodania do repozytorium kluczy menedżera kolejek.

.spec.pki.keys

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

Występuje w:

- [“.spec.pki”](#) na stronie 191

Pole	Opis
name string	Nazwa jest używana jako etykieta klucza lub certyfikatu. Musi być łańcuchem alfanumerycznym zapisanym małymi literami.
secret Secret	Podaj klucz, używając sekretu Kubernetes.

.spec.pki.keys.secret

Podaj klucz, używając sekretu Kubernetes.

Występuje w:

- [“.spec.pki.keys”](#) na stronie 191

Pole	Opis
items array	Klucze znajdujące się wewnątrz sekretu Kubernetes, które powinny zostać dodane do kontenera menedżera kolejek.
secretName string	Nazwa sekretu Kubernetes.

.spec.pki.trust

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

Występuje w:

- [“.spec.pki” na stronie 191](#)

Pole	Opis
name string	Nazwa jest używana jako etykieta klucza lub certyfikatu. Musi być łańcuchem alfanumerycznym zapisanym małymi literami.
secret Secret	Podaj klucz, używając sekretu Kubernetes.

.spec.pki.trust.secret

Podaj klucz, używając sekretu Kubernetes.

Występuje w:

- [“.spec.pki.trust” na stronie 192](#)

Pole	Opis
items array	Klucze znajdujące się wewnątrz sekretu Kubernetes, które powinny zostać dodane do kontenera menedżera kolejek.
secretName string	Nazwa sekretu Kubernetes.

.spec.queueManager

Ustawienia dla kontenera menedżera kolejek i bazowego menedżera kolejek.

Występuje w:

- [“.spec” na stronie 189](#)

Pole	Opis
availability Availability	Ustawienia dostępności menedżera kolejek, takie jak informacja o tym, czy ma być używana para aktywny-rezerwowo lub rodzimy mechanizm wysokiej dostępności.
debug boolean	Określa, czy komunikaty debugowania z kodu właściwego dla kontenera mają być rejestrowane w dzienniku kontenera. Wartością domyślną jest false.
image string	Obraz kontenera, który będzie używany.
imagePullPolicy string	Ustawienie wpływające na to, kiedy kubelet próbuje pobrać określony obraz. Wartością domyślną jest IfNotPresent.
ini INISource array	Ustawienia udostępniania plików INI dla menedżera kolejek. Wymaga operatora produktu MQ 1.1.0 lub nowszej.
livenessProbe QueueManagerLivenessProbe	Ustawienia sterujące sondą działania.
logFormat string	Format dziennika, który ma być używany dla tego kontenera. Użyj wartości JSON dla dzienników w formacie JSON z kontenera. Użyj formatu Basic w przypadku komunikatów w formacie tekstowym. Wartością domyślną jest Basic.
metrics QueueManagerMetrics	Ustawienia dotyczące metryk typu Prometheus.

Pole	Opis
mjsc MQSCSource array	Ustawienia dotyczące udostępniania MQSC dla menedżera kolejek. Wymaga operatora produktu MQ 1.1.0 lub nowszej.
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 Kubernetes (na przykład nazwa zawierająca wielkie litery).
readinessProbe QueueManagerReadinessProbe	Ustawienia sterujące sondą gotowości.
recoveryLogs RecoveryLogs	Ustawienia dzienników odtwarzania produktu MQ . Wymaga produktu MQ Operator 2.4.0 lub nowszego.
resources Resources	Ustawienia, które sterują wymaganiami co do zasobów.
route Route	Ustawienia trasy menedżera kolejek. Wymaga operatora produktu MQ w wersji 1.4.0 lub nowszej.
startupProbe StartupProbe	Ustawienia sterujące sondą uruchamiania. Ma zastosowanie tylko do wdrożeń MultiInstance i NativeHA. Wymaga operatora MQ w wersji 1.5.0 lub nowszej.
storage QueueManagerStorage	Ustawienia pamięci masowej służące do sterowania używaniem woluminów trwałych i klas pamięci przez menedżer kolejek.

.spec.queueManager.availability

Ustawienia dostępności menedżera kolejek, takie jak informacja o tym, czy ma być używana para aktywny-rezerwowy lub rodzimy mechanizm wysokiej dostępności.

Występuje w:

- [“.spec.queueManager”](#) na stronie 192

Pole	Opis
tls Tls	Opcjonalne ustawienia TLS służące do konfigurowania bezpiecznej komunikacji między replikami mechanizmu NativeHA. Wymaga operatora MQ w wersji 1.5.0 lub nowszej.
type string	Typ dostępności, który ma być używany. Użyj typu SingleInstance dla pojedynczego zasobnika, który zostanie automatycznie zrestartowany (w niektórych przypadkach) przez środowisko Kubernetes. Użyj wartości MultiInstance dla pary zasobników, z których jeden jest menedżerem kolejek produktu active , a drugi jest rezerwowy. Użyj opcji NativeHA do realizacji rodzimej replikacji w celu zapewnienia wysokiej dostępności (wymaga to operatora MQ w wersji 1.5.0 lub nowszej). Wartością domyślną jest SingleInstance. Więcej szczegółowych informacji można znaleźć pod adresem http://ibm.biz/BdqAQa .
updateStrategy string	Strategia aktualizacji menedżerów kolejek MultiInstance i NativeHA. Wartość RollingUpdate włącza aktualizacje sekwencyjne po każdej zmianie konfiguracji menedżera kolejek. Wartość OnDelete wyłącza automatyczne aktualizacje sekwencyjne – zmiany w menedżerze kolejek będą stosowane tylko przy usuwaniu zasobników (również gdy takie usunięcie będzie wywołane czynnikami zewnętrznymi). Wartością domyślną jest RollingUpdate. Wymaga operatora MQ Operator w wersji 1.6.0 lub nowszej.

.spec.queueManager.availability.tls

Opcjonalne ustawienia TLS służące do konfigurowania bezpiecznej komunikacji między replikami mechanizmu NativeHA. Wymaga operatora MQ w wersji 1.5.0 lub nowszej.

Występuje w:

- [“.spec.queueManager.availability” na stronie 193](#)

Pole	Opis
<code>cipherSpec</code> string	Nazwa usługi CipherSpec dla połączenia TLS mechanizmu NativeHA.
<code>secretName</code> string	Nazwa sekretu Kubernetes.

.spec.queueManager.ini

Źródło plików konfiguracyjnych INI.

Występuje w:

- [“.spec.queueManager” na stronie 192](#)

Pole	Opis
<code>configMap</code> <code>ConfigMapINISource</code>	ConfigMap reprezentuje obiekt Kubernetes ConfigMap, który zawiera informacje INI.
<code>secret</code> <code>SecretINISource</code>	Sekret reprezentuje sekret Kubernetes, który zawiera informacje INI.

.spec.queueManager.ini.configMap

ConfigMap reprezentuje obiekt Kubernetes ConfigMap, który zawiera informacje INI.

Występuje w:

- [“.spec.queueManager.ini” na stronie 194](#)

Pole	Opis
<code>items</code> array	Klucze wewnątrz źródła Kubernetes, które powinny być stosowane.
<code>name</code> string	Nazwa źródła Kubernetes.

.spec.queueManager.ini.secret

Sekret reprezentuje sekret Kubernetes, który zawiera informacje INI.

Występuje w:

- [“.spec.queueManager.ini” na stronie 194](#)

Pole	Opis
<code>items</code> array	Klucze wewnątrz źródła Kubernetes, które powinny być stosowane.
<code>name</code> string	Nazwa źródła Kubernetes.

.spec.queueManager.livenessProbe

Ustawienia sterujące sondą działania.

Występuje w:

- [“.spec.queueManager” na stronie 192](#)

Pole	Opis
failureThreshold integer	Minimalna liczba następujących po sobie niepowodzeń, poprzedzonych powodzeniem, po której przyjmuje się, że doszło do niepowodzenia sondy. Wartością domyślną jest 1.
initialDelaySeconds integer	Liczba sekund od uruchomienia kontenera, po której sonda jest inicjowana. Wartość domyślna to 90 sekund dla wdrożenia SingleInstance. Wartość domyślna to 0 sekund dla wdrożeń MultiInstance i NativeHA. Więcej informacji: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds integer	Co ile sekund ma być uruchamiana sonda. Wartością domyślną jest 10 sekund.
successThreshold integer	Minimalna liczba następujących po sobie powodzeń, poprzedzonych niepowodzeniem, po której przyjmuje się, że sonda zadziałała pomyślnie. Wartością domyślną jest 1.
timeoutSeconds integer	Limit czasu sondy w sekundach. Wartością domyślną jest 5 sekund. Więcej informacji: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.queueManager.metrics

Ustawienia dotyczące metryk typu Prometheus.

Występuje w:

- [“.spec.queueManager” na stronie 192](#)

Pole	Opis
enabled boolean	Określa, czy punkt końcowy dla metryk zgodnych z narzędziem Prometheus jest włączony, czy nie. Wartością domyślną jest true.

.spec.queueManager.mqsc

Źródło plików konfiguracyjnych MQSC.

Występuje w:

- [“.spec.queueManager” na stronie 192](#)

Pole	Opis
configMap ConfigMapMQSCSource	ConfigMap reprezentuje obiekt Kubernetes ConfigMap, który zawiera informacje MQSC.
secret SecretMQSCSource	Sekret reprezentuje sekret Kubernetes, który zawiera informacje MQSC.

.spec.queueManager.mqsc.configMap

ConfigMap reprezentuje obiekt Kubernetes ConfigMap, który zawiera informacje MQSC.

Występuje w:

- [“.spec.queueManager.mqsc” na stronie 195](#)

Pole	Opis
items array	Klucze wewnątrz źródła Kubernetes, które powinny być stosowane.
name string	Nazwa źródła Kubernetes.

.spec.queueManager.mqsc.secret

Sekret reprezentuje sekret Kubernetes, który zawiera informacje MQSC.

Występuje w:

- [“.spec.queueManager.mqsc”](#) na stronie 195

Pole	Opis
items array	Klucze wewnątrz źródła Kubernetes, które powinny być stosowane.
name string	Nazwa źródła Kubernetes.

.spec.queueManager.readinessProbe

Ustawienia sterujące sondą gotowości.

Występuje w:

- [“.spec.queueManager”](#) na stronie 192

Pole	Opis
failureThreshold integer	Minimalna liczba następujących po sobie niepowodzeń, poprzedzonych powodzeniem, po której przyjmuje się, że doszło do niepowodzenia sondy. Wartością domyślną jest 1.
initialDelaySeconds integer	Liczba sekund od uruchomienia kontenera, po której sonda jest inicjowana. Wartość domyślna to 10 sekund dla wdrożenia SingleInstance. Wartość domyślna to 0 dla wdrożeń MultiInstance i NativeHA. Więcej informacji: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds integer	Co ile sekund ma być uruchamiana sonda. Wartością domyślną jest 5 sekund.
successThreshold integer	Minimalna liczba następujących po sobie powodzeń, poprzedzonych niepowodzeniem, po której przyjmuje się, że sonda zadziałała pomyślnie. Wartością domyślną jest 1.
timeoutSeconds integer	Limit czasu sondy w sekundach. Wartością domyślną są 3 sekundy. Więcej informacji: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.queueManager.recoveryLogs

Ustawienia dzienników odtwarzania produktu MQ . Wymaga produktu MQ Operator 2.4.0 lub nowszego.

Występuje w:

- [“.spec.queueManager”](#) na stronie 192

Pole	Opis
logFilePages integer	Dane dziennika odtwarzania są przechowywane w szeregu plików. Wielkość pliku dziennika jest określana w stronach o wielkości 4 kB.

.spec.queueManager.resources

Ustawienia, które sterują wymaganiami co do zasobów.

Występuje w:

- [“.spec.queueManager”](#) na stronie 192

Pole	Opis
limits Limits	Ustawienia procesora i pamięci.
requests Requests	Ustawienia procesora i pamięci.

.spec.queueManager.resources.limits

Ustawienia procesora i pamięci.

Występuje w:

- [“.spec.queueManager.resources” na stronie 196](#)

Pole	Opis
cpu	
memory	

.spec.queueManager.resources.requests

Ustawienia procesora i pamięci.

Występuje w:

- [“.spec.queueManager.resources” na stronie 196](#)

Pole	Opis
cpu	
memory	

.spec.queueManager.route

Ustawienia trasy menedżera kolejek. Wymaga operatora produktu MQ w wersji 1.4.0 lub nowszej.

Występuje w:

- [“.spec.queueManager” na stronie 192](#)

Pole	Opis
enabled boolean	Określa, czy trasa ma być włączona. Wartością domyślną jest true.

.spec.queueManager.startupProbe

Ustawienia sterujące sondą uruchamiania. Ma zastosowanie tylko do wdrożeń MultiInstance i NativeHA. Wymaga operatora MQ w wersji 1.5.0 lub nowszej.

Występuje w:

- [“.spec.queueManager” na stronie 192](#)

Pole	Opis
failureThreshold integer	Minimalna liczba następujących kolejno po sobie niepowodzeń, po której przyjmuje się, że doszło do niepowodzenia sondy. Wartością domyślną jest 24.
initialDelaySeconds integer	Liczba sekund od uruchomienia kontenera, po której sonda jest inicjowana. Wartością domyślną jest 0 sekund. Więcej informacji: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds integer	Co ile sekund ma być uruchamiana sonda. Wartością domyślną jest 5 sekund.

Pole	Opis
successThreshold integer	Minimalna liczba następujących kolejno po sobie powodzeń, po której przyjmuje się, że sonda zadziałała pomyślnie. Wartością domyślną jest 1.
timeoutSeconds integer	Limit czasu sondy w sekundach. Wartością domyślną jest 5 sekund. Więcej informacji: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.queueManager.storage

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

Występuje w:

- [“.spec.queueManager” na stronie 192](#)

Pole	Opis
defaultClass string	Klasa pamięci masowej, która ma być domyślnie stosowana do wszystkich trwałych woluminów tego menedżera kolejek. Konkretnie trwałe woluminy mogą definiować swoje własne klasy pamięci masowej, które przestonią to ustawienie domyślnej klasy pamięci masowej. Jeśli parametr type of availability ma wartość SingleInstance lub NativeHA, klasa pamięci masowej może być typu ReadWriteOnce lub ReadWriteMany. Jeśli parametr type of availability ma wartość MultiInstance, wymagana jest klasa pamięci masowej typu ReadWriteMany.
defaultDeleteClaim boolean	Określa, czy przy usuwaniu menedżera kolejek powinny zostać usunięte wszystkie woluminy. Dla konkretnych woluminów trwałych można określić odrębną wartość deleteClaim, która przestoni ustawienie defaultDeleteClaim. Wartością domyślną jest false.
persistedData QueueManagerOptionalVolume	Szczegóły woluminu trwałego (persistentVolume) dla utrwalonych danych produktu MQ, w tym konfiguracji, kolejek i komunikatów. Wymagane w przypadku korzystania z menedżera kolejek z wieloma instancjami.
queueManager QueueManagerVolume	Wartość domyślna PersistentVolume dla wszelkich danych znajdujących się normalnie w woluminie /var/mqm. Jeśli nie zostaną określone inne woluminy, będzie zawierać wszystkie utrwalone dane i dzienniki odtwarzania.
recoveryLogs QueueManagerOptionalVolume	Szczegóły woluminu trwałego dla dzienników odtwarzania produktu MQ. Wymagane w przypadku korzystania z menedżera kolejek z wieloma instancjami.
scratch Zdrapka	Ustawienia tymczasowego woluminu tymczasowego menedżera kolejek. Ten wolumin zostanie podłączony jako folder '/run' w kontenerze. Ma zastosowanie tylko wtedy, gdy główny system plików jest ustawiony jako tylko do odczytu. Wymaga produktu MQ Operator 3.0.0 lub nowszego.
tmp Tmp (Tmp)	Ustawienia tymczasowego woluminu tymczasowego menedżera kolejek. Ten wolumin zostanie podłączony w kontenerze jako folder '/tmp'. W tym woluminie zostaną utworzone pliki danych diagnostycznych, takie jak plik zip utworzony przez komendę runmqras. Ma zastosowanie tylko wtedy, gdy główny system plików jest ustawiony jako tylko do odczytu. Wymaga produktu MQ Operator 3.0.0 lub nowszego.

.spec.queueManager.storage.persistedData

Szczegóły woluminu trwałego (persistentVolume) dla utrwalonych danych produktu MQ, w tym konfiguracji, kolejek i komunikatów. Wymagane w przypadku korzystania z menedżera kolejek z wieloma instancjami.

Występuje w:

- [“.spec.queueManager.storage” na stronie 198](#)

Pole	Opis
class string	Klasa pamięci masowej, która ma być używana dla tego woluminu. Wartość poprawna tylko wtedy, gdy type ma wartość persistent-claim. Jeśli parametr type of availability ma wartość SingleInstance lub NativeHA, klasa pamięci masowej może być typu ReadWriteOnce lub ReadWriteMany. Jeśli parametr type of availability ma wartość MultiInstance, wymagana jest klasa pamięci masowej typu ReadWriteMany.
deleteClaim boolean	Określa, czy ten wolumin ma zostać usunięty przy usuwaniu menedżera kolejek.
enabled boolean	Określa, czy ten wolumin powinien być włączony jako oddzielny wolumin, czy też umieszczony w domyślnym woluminie queueManager. Wartością domyślną jest false.
size string	Wielkość woluminu trwałego (PersistentVolume), która ma być przekazana do środowiska Kubernetes, w tym jednostek SI. Wartość poprawna tylko wtedy, gdy type ma wartość persistent-claim. Na przykład: 2Gi. Wartością domyślną jest 2Gi.
sizeLimit string	Limit wielkości w przypadku używania woluminu ephemeral . Pliki są nadal zapisywane do katalogu tymczasowego, więc można użyć tej opcji, aby ograniczyć wielkość. Poprawna tylko wtedy, gdy type ma wartość ephemeral , a główny system plików jest ustawiony na tylko do odczytu. Wymaga produktu MQ Operator 3.0.0 lub nowszego.
type string	Typ woluminu, który ma być używany. Wybierz opcję ephemeral, aby użyć nietrwałej pamięci masowej, lub persistent-claim, aby użyć trwałego woluminu. Wartością domyślną jest persistent-claim.

.spec.queueManager.storage.queueManager

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

Występuje w:

- [“.spec.queueManager.storage” na stronie 198](#)

Pole	Opis
class string	Klasa pamięci masowej, która ma być używana dla tego woluminu. Wartość poprawna tylko wtedy, gdy type ma wartość persistent-claim. Jeśli parametr type of availability ma wartość SingleInstance lub NativeHA, klasa pamięci masowej może być typu ReadWriteOnce lub ReadWriteMany. Jeśli parametr type of availability ma wartość MultiInstance, wymagana jest klasa pamięci masowej typu ReadWriteMany.

Pole	Opis
deleteClaim boolean	Określa, czy ten wolumin ma zostać usunięty przy usuwaniu menedżera kolejek.
size string	Wielkość woluminu trwałego (PersistentVolume), która ma być przekazana do środowiska Kubernetes, w tym jednostek SI. Wartość poprawna tylko wtedy, gdy type ma wartość persistent-claim. Na przykład: 2Gi. Wartością domyślną jest 2Gi.
sizeLimit string	Limit wielkości w przypadku używania woluminu ephemeral . Pliki są nadal zapisywane do katalogu tymczasowego, więc można użyć tej opcji, aby ograniczyć wielkość. Poprawna tylko wtedy, gdy type ma wartość ephemeral , a główny system plików jest ustawiony na tylko do odczytu. Wymaga produktu MQ Operator 3.0.0 lub nowszego.
type string	Typ woluminu, który ma być używany. Wybierz opcję ephemeral, aby użyć nietrwałej pamięci masowej, lub persistent-claim, aby użyć trwałego woluminu. Wartością domyślną jest persistent-claim.

.spec.queueManager.storage.recoveryLogs

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

Występuje w:

- [“.spec.queueManager.storage”](#) na stronie 198

Pole	Opis
class string	Klasa pamięci masowej, która ma być używana dla tego woluminu. Wartość poprawna tylko wtedy, gdy type ma wartość persistent-claim. Jeśli parametr type of availability ma wartość SingleInstance lub NativeHA, klasa pamięci masowej może być typu ReadWriteOnce lub ReadWriteMany. Jeśli parametr type of availability ma wartość MultiInstance, wymagana jest klasa pamięci masowej typu ReadWriteMany.
deleteClaim boolean	Określa, czy ten wolumin ma zostać usunięty przy usuwaniu menedżera kolejek.
enabled boolean	Określa, czy ten wolumin powinien być włączony jako oddzielny wolumin, czy też umieszczony w domyślnym woluminie queueManager. Wartością domyślną jest false.
size string	Wielkość woluminu trwałego (PersistentVolume), która ma być przekazana do środowiska Kubernetes, w tym jednostek SI. Wartość poprawna tylko wtedy, gdy type ma wartość persistent-claim. Na przykład: 2Gi. Wartością domyślną jest 2Gi.
sizeLimit string	Limit wielkości w przypadku używania woluminu ephemeral . Pliki są nadal zapisywane do katalogu tymczasowego, więc można użyć tej opcji, aby ograniczyć wielkość. Poprawna tylko wtedy, gdy type ma wartość ephemeral , a główny system plików jest ustawiony na tylko do odczytu. Wymaga produktu MQ Operator 3.0.0 lub nowszego.
type string	Typ woluminu, który ma być używany. Wybierz opcję ephemeral, aby użyć nietrwałej pamięci masowej, lub persistent-claim, aby użyć trwałego woluminu. Wartością domyślną jest persistent-claim.

.spec.queueManager.storage.scratch

Ustawienia tymczasowego woluminu tymczasowego menedżera kolejek. Ten wolumin zostanie podłączony jako folder '/run' w kontenerze. Ma zastosowanie tylko wtedy, gdy główny system plików jest ustawiony jako tylko do odczytu. Wymaga produktu MQ Operator 3.0.0 lub nowszego.

Występuje w:

- [“.spec.queueManager.storage” na stronie 198](#)

Pole	Opis
sizeLimit string	Limit wielkości objętości efemerycznej, w tym jednostek SI. Na przykład: 2Gi. Poprawne tylko wtedy, gdy główny system plików jest ustawiony na tylko do odczytu. Wymaga produktu MQ Operator 3.0.0 lub nowszego.

.spec.queueManager.storage.tmp

Ustawienia tymczasowego woluminu tymczasowego menedżera kolejek. Ten wolumin zostanie podłączony w kontenerze jako folder '/tmp'. W tym woluminie zostaną utworzone pliki danych diagnostycznych, takie jak plik zip utworzony przez komendę runmqras. Ma zastosowanie tylko wtedy, gdy główny system plików jest ustawiony jako tylko do odczytu. Wymaga produktu MQ Operator 3.0.0 lub nowszego.

Występuje w:

- [“.spec.queueManager.storage” na stronie 198](#)

Pole	Opis
sizeLimit string	Limit wielkości objętości efemerycznej, w tym jednostek SI. Na przykład: 2Gi. Poprawne tylko wtedy, gdy główny system plików jest ustawiony na tylko do odczytu. Wymaga produktu MQ Operator 3.0.0 lub nowszego.

.spec.securityContext

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

Występuje w:

- [“.spec” na stronie 189](#)

Pole	Opis
fsGroup integer	Specjalna grupa uzupełniająca, która ma zastosowanie do wszystkich kontenerów w zasobniku. Niektóre typy woluminów umożliwiają agentowi Kubelet zmianę właściciela woluminu na zasobnik: 1. Identyfikator GID to FSGroup 2. Bit setgid jest ustawiony (nowe pliki utworzone w woluminie będą własnością FSGroup) 3. Bity uprawnień to OR'd z rw-rw----. Jeśli nie zostaną ustawione, agent Kubelet nie zmodyfikuje prawa własności i uprawnień dla woluminu.

Pole	Opis
<code>initVolumeAsRoot</code> boolean	Ma to wpływ na kontekst zabezpieczeń używany przez kontener, który inicjuje wolumin <code>PersistentVolume</code> . Ustaw na <code>true</code> , jeśli używany jest dostawca pamięci masowej, który wymaga, aby użytkownik <code>root</code> miał dostęp do nowo udostępnianych woluminów. Ustawienie <code>true</code> ma wpływ na to, które obiekty SCC (Security Context Constraints – ograniczenia kontekstu zabezpieczeń) mogą być używane, a menedżer kolejek może nie uruchomić się, jeśli użytkownik nie będzie mieć uprawnień do korzystania z obiektu SCC dopuszczającego użytkownika <code>root</code> . Wartością domyślną jest <code>false</code> . Aby uzyskać więcej informacji, patrz https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html .
<code>readOnlyRootFilesystem</code> boolean	Określa, czy włączyć ustawienia głównego systemu plików tylko do odczytu dla menedżera kolejek. Wartością domyślną jest <code>false</code> . Wymaga produktu MQ Operator 3.0.0 lub nowszego.
<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.

.spec.telemetry

Ustawienia dla konfiguracji Open Telemetry. Wymaga produktu MQ Operator 2.2.0 lub nowszego.

Występuje w:

- [“.spec” na stronie 189](#)

Pole	Opis
<code>tracing</code> Śledzenie	Ustawienia śledzenia Open Telemetry.

.spec.telemetry.tracing

Ustawienia śledzenia Open Telemetry.

Występuje w:

- [“.spec.telemetry” na stronie 202](#)

Pole	Opis
<code>instana</code> Instana	Ustawienia śledzenia Instana.

.spec.telemetry.tracing.instana

Ustawienia śledzenia Instana.

Występuje w:

- [“.spec.telemetry.tracing” na stronie 202](#)

Pole	Opis
<code>agentHost</code> string	Nazwa hosta agenta Instana, do którego mają być wysyłane dane śledzenia. Nie powinno to obejmować protokołu.
<code>enabled</code> boolean	Określa, czy włączyć śledzenie Instana. Wartością domyślną jest <code>false</code> .
<code>protocol</code> string	Protokół, który ma być używany w komunikacji z agentem Instana. Obsługiwane są protokoły <code>http</code> i <code>https</code> .

.spec.template

Zaawansowane użycie szablonów zasobów Kubernetes. Szablon umożliwia użytkownikom zmianę sposobu generowania przez IBM MQ bazowych zasobów Kubernetes, takich jak zestawy stanowe (StatefulSet), zasobniki (Pod) i usługi (Service). Ta możliwość przeznaczona jest tylko dla zaawansowanych użytkowników, ponieważ jej niepoprawne wykorzystanie może zakłócić normalne działanie produktu MQ. Wszystkie wartości określone w menedżerze kolejek zostaną przestonięte przez odpowiednie ustawienia w szablonie.

Występuje w:

- [“.spec” na stronie 189](#)

Pole	Opis
pod	Przestonięcia dla szablonu używanego dla zasobnika. Patrz https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#pods-spec-v1-core .

.spec.tracing

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

Występuje w:

- [“.spec” na stronie 189](#)

Pole	Opis
agent TracingAgent	Tylko w pakiecie Cloud Pak for Integration: można skonfigurować ustawienia opcjonalnego agenta śledzenia.
collector TracingCollector	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. Wartością domyślną jest false.
namespace string	Przestrzeń nazw, w której zainstalowany jest panel kontrolny Cloud Pak for Integration Operations Dashboard.

.spec.tracing.agent

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

Występuje w:

- [“.spec.tracing” na stronie 203](#)

Pole	Opis
image string	Obraz kontenera, który będzie używany.
imagePullPolicy string	Ustawienie wpływające na to, kiedy kubelet próbuje pobrać określony obraz. Wartością domyślną jest <code>IfNotPresent</code> .
livenessProbe TracingProbe	Ustawienia sterujące sondą działania.
readinessProbe TracingProbe	Ustawienia sterujące sondą gotowości.

.spec.tracing.agent.livenessProbe

Ustawienia sterujące sondą działania.

Występuje w:

- [“.spec.tracing.agent” na stronie 203](#)

Pole	Opis
failureThreshold integer	Minimalna liczba następujących po sobie niepowodzeń, poprzedzonych powodzeniem, po której przyjmuje się, że doszło do niepowodzenia sondy. Wartością domyślną jest 1.
initialDelaySeconds integer	Liczba sekund od uruchomienia kontenera, po której inicjowane są sondy działania. Wartością domyślną jest 10 sekund. Więcej informacji: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds integer	Co ile sekund ma być uruchamiana sonda. Wartością domyślną jest 10 sekund.
successThreshold integer	Minimalna liczba następujących po sobie powodzeń, poprzedzonych niepowodzeniem, po której przyjmuje się, że sonda zadziałała pomyślnie. Wartością domyślną jest 1.
timeoutSeconds integer	Limit czasu sondy w sekundach. Wartością domyślną są 2 sekundy. Więcej informacji: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.tracing.agent.readinessProbe

Ustawienia sterujące sondą gotowości.

Występuje w:

- [“.spec.tracing.agent” na stronie 203](#)

Pole	Opis
failureThreshold integer	Minimalna liczba następujących po sobie niepowodzeń, poprzedzonych powodzeniem, po której przyjmuje się, że doszło do niepowodzenia sondy. Wartością domyślną jest 1.
initialDelaySeconds integer	Liczba sekund od uruchomienia kontenera, po której inicjowane są sondy działania. Wartością domyślną jest 10 sekund. Więcej informacji: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds integer	Co ile sekund ma być uruchamiana sonda. Wartością domyślną jest 10 sekund.
successThreshold integer	Minimalna liczba następujących po sobie powodzeń, poprzedzonych niepowodzeniem, po której przyjmuje się, że sonda zadziałała pomyślnie. Wartością domyślną jest 1.
timeoutSeconds integer	Limit czasu sondy w sekundach. Wartością domyślną są 2 sekundy. Więcej informacji: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.tracing.collector

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

Występuje w:

- [“.spec.tracing”](#) na stronie 203

Pole	Opis
image string	Obraz kontenera, który będzie używany.
imagePullPolicy string	Ustawienie wpływające na to, kiedy kubelet próbuje pobrać określony obraz. Wartością domyślną jest <code>IfNotPresent</code> .
livenessProbe TracingProbe	Ustawienia sterujące sondą działania.
readinessProbe TracingProbe	Ustawienia sterujące sondą gotowości.

.spec.tracing.collector.livenessProbe

Ustawienia sterujące sondą działania.

Występuje w:

- [“.spec.tracing.collector”](#) na stronie 204

Pole	Opis
failureThreshold integer	Minimalna liczba następujących po sobie niepowodzeń, poprzedzonych powodzeniem, po której przyjmuje się, że doszło do niepowodzenia sondy. Wartością domyślną jest 1.
initialDelaySeconds integer	Liczba sekund od uruchomienia kontenera, po której inicjowane są sondy działania. Wartością domyślną jest 10 sekund. Więcej informacji: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds integer	Co ile sekund ma być uruchamiana sonda. Wartością domyślną jest 10 sekund.
successThreshold integer	Minimalna liczba następujących po sobie powodzeń, poprzedzonych niepowodzeniem, po której przyjmuje się, że sonda zadziałała pomyślnie. Wartością domyślną jest 1.
timeoutSeconds integer	Limit czasu sondy w sekundach. Wartością domyślną są 2 sekundy. Więcej informacji: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.tracing.collector.readinessProbe

Ustawienia sterujące sondą gotowości.

Występuje w:

- [“.spec.tracing.collector”](#) na stronie 204

Pole	Opis
failureThreshold integer	Minimalna liczba następujących po sobie niepowodzeń, poprzedzonych powodzeniem, po której przyjmuje się, że doszło do niepowodzenia sondy. Wartością domyślną jest 1.
initialDelaySeconds integer	Liczba sekund od uruchomienia kontenera, po której inicjowane są sondy działania. Wartością domyślną jest 10 sekund. Więcej informacji: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds integer	Co ile sekund ma być uruchamiana sonda. Wartością domyślną jest 10 sekund.

Pole	Opis
successThreshold integer	Minimalna liczba następujących po sobie powodzeń, poprzedzonych niepowodzeniem, po której przyjmuje się, że sonda zadziałała pomyślnie. Wartością domyślną jest 1.
timeoutSeconds integer	Limit czasu sondy w sekundach. Wartością domyślną są 2 sekundy. Więcej informacji: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.web

Ustawienia serwera WWW produktu MQ.

Występuje w:

- [“.spec” na stronie 189](#)

Pole	Opis
console Konsola	Ustawienia konsoli WWW produktu MQ . Wymaga produktu MQ Operator 3.0.0 lub nowszego.
enabled boolean	Określa, czy serwer WWW ma być włączony. Wartością domyślną jest false.
manualConfig ManualConfig	Ustawienia dotyczące dostarczania konfiguracji XML serwera WWW. Wymaga produktu MQ Operator 3.0.0 lub nowszego.

.spec.web.console

Ustawienia konsoli WWW produktu MQ . Wymaga produktu MQ Operator 3.0.0 lub nowszego.

Występuje w:

- [“.spec.web” na stronie 206](#)

Pole	Opis
authentication Uwierzytelnianie	Ustawienia uwierzytelniania dla konsoli WWW produktu MQ . Wymaga produktu MQ Operator 3.0.0 lub nowszego.
authorization Autoryzacja	Ustawienia autoryzacji dla konsoli WWW produktu MQ . Wymaga produktu MQ Operator 3.0.0 lub nowszego.

.spec.web.console.authentication

Ustawienia uwierzytelniania dla konsoli WWW produktu MQ . Wymaga produktu MQ Operator 3.0.0 lub nowszego.

Występuje w:

- [“.spec.web.console” na stronie 206](#)

Pole	Opis
provider string	Dostawca uwierzytelniania, który ma być używany dla konsoli WWW produktu MQ . Użyj <code>integration-keycloak</code> , aby użyć pojedynczego logowania w interfejsie użytkownika platformy Cloud Pak for Integration (Keycloak). Wartością domyślną jest <code>integration-keycloak</code> , jeśli używana jest licencja na produkt Cloud Pak for Integration lub <code>manual</code> , jeśli używana jest licencja na produkt MQ . Aby udostępnić własną konfigurację, należy użyć opcji <code>manual</code> .

.spec.web.console.authorization

Ustawienia autoryzacji dla konsoli WWW produktu MQ . Wymaga produktu MQ Operator 3.0.0 lub nowszego.

Występuje w:

- [“.spec.web.console” na stronie 206](#)

Pole	Opis
provider string	Dostawca autoryzacji, który ma być używany dla konsoli WWW produktu MQ . Użyj <code>integration-keycloak</code> , aby użyć ról udostępnianych przez Cloud Pak for Integration Keycloak. Aby udostępnić własną konfigurację, należy użyć opcji <code>manual</code> . Wartością domyślną jest <code>integration-keycloak</code> , jeśli używana jest licencja na produkt Cloud Pak for Integration lub <code>manual</code> , jeśli używana jest licencja na produkt MQ .

.spec.web.manualConfig

Ustawienia dotyczące dostarczania konfiguracji XML serwera WWW. Wymaga produktu MQ Operator 3.0.0 lub nowszego.

Występuje w:

- [“.spec.web” na stronie 206](#)

Pole	Opis
configMap ConfigMap	ConfigMap reprezentuje obiekt Kubernetes ConfigMap , który zawiera konfigurację XML serwera WWW.
secret Secret	Klucz tajny reprezentuje klucz tajny Kubernetes , który zawiera konfigurację XML serwera WWW. Użycie klucza tajnego chroni wszystkie informacje autoryzacyjne w warstwie Kubernetes , ale istnieje możliwość, że narzędzia do monitorowania i rozwiązywania problemów mogą w sposób niezabezpieczony ujawnić bazowy plik. Aby zwiększyć bezpieczeństwo, należy zakodować referencje za pomocą programu narzędziowego "securityUtility".

.spec.web.manualConfig.configMap

ConfigMap reprezentuje obiekt Kubernetes ConfigMap , który zawiera konfigurację XML serwera WWW.

Występuje w:

- [“.spec.web.manualConfig” na stronie 207](#)

Pole	Opis
name string	Nazwa źródła Kubernetes.

.spec.web.manualConfig.secret

Klucz tajny reprezentuje klucz tajny Kubernetes , który zawiera konfigurację XML serwera WWW. Użycie klucza tajnego chroni wszystkie informacje autoryzacyjne w warstwie Kubernetes , ale istnieje możliwość, że narzędzia do monitorowania i rozwiązywania problemów mogą w sposób niezabezpieczony ujawnić bazowy plik. Aby zwiększyć bezpieczeństwo, należy zakodować referencje za pomocą programu narzędziowego "securityUtility".

Występuje w:

- [“.spec.web.manualConfig” na stronie 207](#)

Pole	Opis
name string	Nazwa źródła Kubernetes.

.status

Obserwowany stan menedżera kolejek.

Występuje w:

- [“QueueManager” na stronie 188](#)

Pole	Opis
adminUiUrl string	Adres URL administracyjnego interfejsu użytkownika.
availability Availability	Status dostępności menedżera kolejek.
conditions QueueManagerStatusCondition array	Warunki reprezentują najnowsze dostępne obserwacje stanu menedżera kolejek.
endpoints QueueManagerStatusEndpoint array	Informacje dotyczące punktów końcowych, które eksponuje ten menedżer kolejek, takich jak punkty końcowe interfejsu API lub interfejsu użytkownika.
metadata Metadane	Metadane reprezentują dodatkowe informacje dla menedżera kolejek, w tym statusKeycloak integracji.
name string	Nazwa menedżera kolejek.
phase string	Faza stanu menedżera kolejek.
versions QueueManagerStatusVersion	Używana wersja produktu MQ oraz inne wersje dostępne w rejestrze IBM Entitled Registry.

.status.availability

Status dostępności menedżera kolejek.

Występuje w:

- [“.status” na stronie 208](#)

Pole	Opis
initialQuorumEstablished boolean	Określa, czy dla NativeHA ustanowiono początkowe kworum.

.status.conditions

[QueueManagerStatusCondition](#) definiuje stany menedżera kolejek.

Występuje w:

- [“.status” na stronie 208](#)

Pole	Opis
lastTransitionTime string	Czas ostatniego przejścia stanu z jednego statusu do innego.
message string	Czytelny dla człowieka komunikat opisujący szczegóły ostatniej zmiany statusu.
reason string	Przyczyna ostatniej zmiany tego statusu.

Pole	Opis
status string	Status stanu.
type string	Typ stanu.

.status.endpoints

QueueManagerStatusEndpoint definiuje punkty końcowe menedżera kolejek.

Występuje w:

- [“.status” na stronie 208](#)

Pole	Opis
name string	Nazwa punktu końcowego.
type string	Typ punktu końcowego, na przykład 'UI' dla punktu końcowego interfejsu użytkownika, 'API' dla punktu końcowego interfejsu API, 'OpenAPI' dla dokumentacji API.
uri string	Identyfikator URI punktu końcowego.

.status.metadata

Metadane reprezentują dodatkowe informacje dla menedżera kolejek, w tym statusKeycloak integracji.

Występuje w:

- [“.status” na stronie 208](#)

Pole	Opis
integrationKeycloak IntegrationKeycloak	QueueManagerStatusIntegrationKeycloak definiuje statusKeycloak integracji dla menedżera kolejek QueueManager.

.status.metadata.integrationKeycloak

QueueManagerStatusIntegrationKeycloak definiuje statusKeycloak integracji dla menedżera kolejek QueueManager.

Występuje w:

- [“.status.metadata” na stronie 209](#)

Pole	Opis
clientName string	

.status.versions

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

Występuje w:

- [“.status” na stronie 208](#)

Pole	Opis
available QueueManagerStatusVersionA available	Inne wersje produktu MQ dostępne w rejestrze IBM Entitled Registry.

Pole	Opis
reconciled string	Konkretna wersja produktu IBM MQ, która jest używana. Jeśli zostanie określony obraz niestandardowy, może wystąpić niezgodność z używaną obecnie wersją produktu MQ.

.status.versions.available

Inne wersje produktu MQ dostępne w rejestrze IBM Entitled Registry.

Występuje w:

- [“.status.versions”](#) na stronie 209

Pole	Opis
channels array	Kanały dostępne do automatycznego aktualizowania wersji produktu MQ.
versions Versions array	Konkretne wersje produktu MQ, które są dostępne.

.status.versions.available.versions

QueueManagerStatusVersion definiuje wersję produktu MQ.

Występuje w:

- [“.status.versions.available”](#) na stronie 210

Pole	Opis
Tablica licenses licencji	Licencje odpowiednie dla tej wersji menedżera kolejek QueueManager.
name string	Nazwa (name) tej wersji menedżera kolejek. Są to poprawne wartości pola <code>spec.version</code> .

.status.versions.available.versions.licenses

QueueManagerStatusLicense definiuje licencję.

Występuje w:

- [“.status.versions.available.versions”](#) na stronie 210

Pole	Opis
displayName string	Nazwa wyświetlana licencji.
link string	Odsyłacz do treści licencji.
matchesCurrentType boolean	Określa, czy licencja jest zgodna z typem aktualnie używanej licencji.
name string	Nazwa licencji.

Warunki stanu menedżera kolejek (mq.ibm.com/v1beta1)

Pola **status.conditions** są aktualizowane w celu odzwierciedlenia stanu zasobu QueueManager. Co do zasady stany opisują sytuacje nieprawidłowe. Menedżer kolejek bez nieprawidłowości i będący w stanie gotowości nie ma stanów **Error** ani **Pending**. Może mieć stany informacyjne **Warning**.

Obsługa stanów została wprowadzona w wersji IBM MQ Operator 1.2.

Dla zasobu QueueManager zdefiniowane są następujące stany:

Tabela 1. Stany menedżera kolejek

Komponent	Typ stanu	Kod przyczyny	Komunikat ostrzegawczy
QueueManager ⁹	Zablokowane	OperatorDependency	Aby można było zainstalować tę instancję, produkt Keycloak musi zostać skonfigurowany przez produkt [IBM Cloud Pak for Integration]. Ta instancja pozostanie w stanie [Oczekująca] do czasu, aż status Keycloak zostanie zgłoszony jako [KeycloakReady] w zasobie Cp4iServicesBinding dla tego QueueManager. Do zainstalowania tej instancji wymagany jest operator [IBM IAM]. Ta instancja pozostanie w stanie [Zablokowana] do czasu zainstalowania operatora przez usługi [IBM Cloud Pak foundational services].
	Oczekujący	Tworzenie	Menedżer kolejek MQ jest wdrażany
	Oczekujący	OidcPending	Menedżer kolejek MQ oczekuje na rejestrację klienta OIDC
	Błąd	Niepowodzenie	Wdrożenie menedżera kolejek MQ nie powiodło się
	Ostrzeżenie	UnsupportedVersion	¹⁰ Operand został zainstalowany przez operator, który nie jest obsługiwany w produkcie OCP w wersji <ocp_version>. Ten operand nie jest obsługiwany.
	Ostrzeżenie	Obsługa CP4I-LTS	¹¹ Operand CP4I-LTS <mq_version> został zainstalowany, ale jest zarządzany przez operator, który nie kwalifikuje się do rozszerzonego przedziału czasu obsługi. Ten operand nie kwalifikuje się do wydłużonego czasu obsługi.
	Ostrzeżenie	Obsługa CP4I-LTS	¹² Operand CP4I-LTS <mq_version> został zainstalowany, ale panel OCP w wersji 4<ocp_version> nie kwalifikuje się do rozszerzonego czasu obsługi. Ten operand nie kwalifikuje się do wydłużonego czasu obsługi.
	Ostrzeżenie	Obsługa CP4I-LTS	¹³ Operand CP4I-LTS <mq_version> został zainstalowany, ale wersja OCP <ocp_version> nie kwalifikuje się do rozszerzonego czasu obsługi.

⁹ Warunki Creating i Failed służą do monitorowania ogólnego postępu wdrażania menedżera kolejek. Jeśli używana jest licencja na produkt IBM Cloud Pak for Integration, a konsola WDW jest włączona, warunek OidcPending rejestruje status menedżera kolejek podczas oczekiwania na zakończenie rejestracji klienta OIDC w trybie IAM.

Tabela 1. Stany menedżera kolejek (kontynuacja)

Komponent	Typ stanu	Kod przyczyny	Komunikat ostrzegawczy
Zasobnik ¹⁴	Oczekujący	PodPending	Trwa wdrażanie zasobnika menedżera kolejek MQ
	Błąd	PodFailed	Trwa wdrażanie zasobnika menedżera kolejek MQ
Pamięć masowa ¹⁵	Oczekujący	StoragePending	Trwa udostępnianie pamięci masowej dla menedżera kolejek MQ
	Ostrzeżenie	StorageEphemeral	Dla produkcyjnego menedżera kolejek MQ używana jest efemeryczna pamięć masowa.
	Błąd	StorageFailed	Nie powiodła się próba udostępnienia pamięci masowej dla menedżera kolejek MQ

Multi Budowanie własnego kontenera IBM MQ i kodu wdrażania

Samodzielne budowanie kontenera. Jest to najbardziej elastyczne rozwiązanie kontenerowe, ale wymaga zaawansowanych kompetencji w dziedzinie konfigurowania kontenerów i "posiadania na własność" kontenera wynikowego.

Zanim rozpocznie

Przed utworzeniem własnego kontenera należy rozważyć, czy zamiast niego można użyć pliku IBM MQ Operator. Patrz: ["Wybór sposobu używania IBM MQ w kontenerach"](#) na stronie 5

O tym zadaniu

Procedura

- ["Planowanie wdrożenia własnego obrazu menedżera kolejek IBM MQ przy użyciu kontenera"](#) na stronie 213
- ["Budowanie przykładowego obrazu kontenera menedżera kolejek IBM MQ"](#) na stronie 213
- ["Uruchamianie aplikacji z powiązaniem lokalnymi w oddzielnych kontenerach"](#) na stronie 216
- [Przejrzyj IBM MQ przykładowy wykres Helm.](#)

¹⁰ Operator 1.4.0 i nowszy

¹¹ Operator 1.4.0 i nowszy

¹² Operator 1.4.0 i nowszy

¹³ Tylko operator 1.3.0

¹⁴ Warunki zasobnika służą do monitorowania statusu zasobników podczas wdrażania menedżera kolejek. Jeśli zostanie wyświetlony dowolny warunek PodFailed, ogólny warunek menedżera kolejek zostanie również ustawiony na wartość Failed.

¹⁵ Warunki pamięci masowej służą do monitorowania postępu (warunek StoragePending) żądań utworzenia woluminów dla składnicy trwałej i raportowania błędów powiązania i innych awarii. Jeśli podczas udostępniania pamięci masowej wystąpi jakikolwiek błąd, warunek StorageFailed zostanie dodany do listy warunków, a ogólny warunek menedżera kolejek również zostanie ustawiony na wartość Failed.

Planowanie wdrożenia własnego obrazu menedżera kolejek IBM MQ przy użyciu kontenera

Istnieje kilka wymagań, które należy wziąć pod uwagę, uruchamiając menedżer kolejek IBM MQ w kontenerze. Przykładowy obraz kontenera umożliwia spełnienie tych wymagań, ale używając własnego obrazu, trzeba zadbać o ich spełnienie samodzielnie.

Nadzór nad procesami

W momencie uruchamiania kontenera zasadniczo uruchamiany jest jeden proces (PID 1 wewnątrz kontenera), który może później tworzyć procesy potomne.

Gdy proces główny kończy działanie, środowisko wykonawcze kontenera zatrzymuje kontener. Menedżer kolejek IBM MQ wymaga działania wielu procesów w tle.

Z tego powodu należy dopilnować, aby proces główny pozostawał aktywny zawsze, gdy menedżer kolejek ma być uruchomiony. Dobrą praktyką jest sprawdzanie, czy menedżer kolejek jest aktywny, z tego procesu, na przykład poprzez kierowanie zapytań administracyjnych.

Wypełnianie /var/mqm

Kontenery muszą zostać skonfigurowane z /var/mqm jako woluminem.

W takim przypadku przy pierwszym uruchomieniu kontenera katalog woluminu jest pusty. Ten katalog jest zwykle wypełniany podczas instalacji, ale w przypadku korzystania z kontenera środowisko instalacji i środowisko wykonawcze są oddzielnymi środowiskami.

Aby rozwiązać ten problem, podczas uruchamiania kontenera można użyć komendy `crtmqdir`, aby wstępnie wypełnić katalog /var/mqm.

Zabezpieczenia kontenera

W celu ograniczenia wymagań co do zabezpieczeń środowiska wykonawczego przykładowe obrazy kontenera są instalowane przy użyciu instalacji IBM MQ niepodlegającej kompresji. Zapewnia to ustawienie bitów `setuid` i powoduje, że kontener nie musi korzystać z eskalacji uprawnień. Niektóre systemy kontenerowe definiują identyfikatory użytkowników, które mogą być używane, a w instalacji bez kompresji nie czyni się żadnych założeń co do dostępnych użytkowników systemu operacyjnego.

Budowanie przykładowego obrazu kontenera menedżera kolejek IBM MQ

Jest to instrukcja tworzenia przykładowego obrazu kontenera do uruchamiania menedżera kolejek IBM MQ w kontenerze.

O tym zadaniu

Najpierw budujemy podstawowy obraz zawierający system plików Red Hat Universal Base Image i czystą instalację produktu IBM MQ.

Następnie tworzymy kolejną warstwę obrazu kontenera, dodając w niej konfigurację IBM MQ, aby umożliwić podstawową identyfikację użytkowników i zabezpieczenie hasłem.

Na koniec 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 214](#)
- [“Budowanie przykładowego skonfigurowanego obrazu menedżera kolejek IBM MQ” na stronie 214](#)

Multi Budowanie przykładowego podstawowego obrazu menedżera kolejek IBM MQ

Aby używać produktu IBM MQ we własnym obrazie kontenera, należy najpierw zbudować obraz podstawowy z czystą instalacją produktu IBM MQ. W poniższych krokach przedstawiono sposób budowania przykładowego obrazu podstawowego przy użyciu przykładowego kodu udostępnianego na serwerze GitHub.

Procedura

- Użyj plików make dostarczonych w [repozytorium GitHub mq-container](#), aby zbudować obraz kontenera produkcyjnego.

Postępuj zgodnie z instrukcjami w sekcji [Budowanie obrazu kontenera](#) w serwisie GitHub.

- Opcjonalne: Jeśli planowane jest skonfigurowanie bezpiecznego dostępu przy użyciu ograniczenia kontekstu zabezpieczeń (SCC) Red Hat OpenShift Container Platform "z ograniczeniem", należy użyć jednego z obrazów nieinstalacyjnych systemu IBM MQ.

Odsyłacze umożliwiające pobranie tych obrazów są dostępne w sekcji Kontenery w sekcji [Pobieranie produktu IBM MQ](#).

Wyniki

Mamy teraz podstawowy obraz kontenera z zainstalowanym produktem IBM MQ.

Teraz można przystąpić do [budowania przykładowego skonfigurowanego obrazu menedżera kolejek IBM MQ](#).

Multi Budowanie przykładowego skonfigurowanego obrazu menedżera kolejek IBM MQ

Po utworzeniu ogólnego podstawowego obrazu kontenera IBM MQ należy zastosować w nim własną konfigurację, aby zabezpieczyć dostęp do niego. W tym celu należy utworzyć własną warstwę obrazu kontenera, korzystając z obrazu ogólnego jako obrazu nadrzędnego.

Zanim rozpoczniesz

W tym zadaniu przyjęto założenie, że do zbudowania przykładowego podstawowego obrazu menedżera kolejek IBM MQ użyto pakietu "No-Install" IBM MQ. W przeciwnym razie nie można będzie skonfigurować bezpiecznego dostępu przy użyciu ograniczenia kontekstu zabezpieczeń (SCC) typu Red Hat OpenShift Container Platform "restricted". Ograniczenie SCC typu "restricted", które jest stosowane domyślnie, korzysta z losowych identyfikatorów użytkowników i uniemożliwia eskalację uprawnień poprzez zmianę konta użytkownika na inne. Tradycyjny instalator IBM MQ oparty na RPM korzysta z użytkownika i grupy mqm, a także korzysta z bitów setuid w programach wykonywalnych. W bieżącej wersji systemu IBM MQ, gdy używany jest pakiet "No-Install" IBM MQ, nie ma już użytkownika mqm ani grupy mqm.

Procedura

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

```
DEFINE QLOCAL(EXAMPLE.QUEUE.1) REPLACE
```

Należy zauważyć, że w powyższym przykładzie zastosowano proces uwierzytelniania na podstawie identyfikatora i hasła użytkownika. Istnieje jednak możliwość zastosowania dowolnej konfiguracji zabezpieczeń, której wymaga przedsiębiorstwo.

2. Utwórz plik o nazwie Dockerfile o następującej zawartości:

```
FROM mq
COPY config.mqsc /etc/mqm/
```

3. Zbuduj niestandardowy obraz kontenera, używając następującej komendy:

```
docker build -t mymq .
```

gdzie "." to katalog, w którym znajdują się właśnie utworzone dwa pliki.

Docker utworzy wówczas tymczasowy kontener, używając tego obrazu, a następnie uruchomi pozostałe komendy.

Uwaga: W systemie Red Hat Enterprise Linux (RHEL) stosuje się komendę **docker** (RHEL V7) lub **podman** (RHEL V7 lub RHEL V8). W systemie Linux komendy **docker** muszą być poprzedzane komendą **sudo** zapewniającą dodatkowe uprawnienia.

4. Uruchom nowy dostosowany obraz, aby utworzyć nowy kontener z obrazem dysku, który został właśnie utworzony.

Nowa warstwa obrazu nie określa żadnej konkretnej komendy do uruchomienia, dlatego komenda jest dziedziczona z obrazu nadrzędnego. Punkt wejścia obrazu nadrzędnego (kod jest dostępny w serwisie GitHub):

- Tworzy menedżer kolejek
- Uruchamia menedżer kolejek
- Tworzy domyślny proces nasłuchujący
- Następnie uruchamia wszystkie komendy MQSC z produktu /etc/mqm/config.mqsc .

Aby uruchomić nowy dostosowany obraz, wydaj następujące komendy:

```
docker run \
  --env LICENSE=accept \
  --env MQ_QMGR_NAME=QM1 \
  --volume /var/example:/var/mqm \
  --publish 1414:1414 \
  --detach \
  mymq
```

gdzie:

Pierwszy parametr env

Przekazuje do kontenera zmienną środowiskową, która potwierdza akceptację licencji na produkt IBM IBM WebSphere MQ. Można także ustawić zmienną LICENSE, aby wyświetlić licencję.

Więcej informacji na temat licencji na produkt [IBM MQ](#) można znaleźć w sekcji [Informacje o licencji produktu IBM MQ](#).

Drugi parametr env

Określa nazwę używanego menedżera kolejek.

Parametr Volume

Przekazuje do kontenera informacje o tym, że wszystkie operacje zapisu MQ w produkcie /var/mqm powinny zostać zapisane w /var/example na hoście.

Ta opcja powoduje, że można później usunąć kontener, zachowując utrwalone dane. Ta opcja ułatwia również wyświetlanie plików dziennika.

Parametr Publish

Odwzorowuje porty w systemie hosta na porty w kontenerze. Domyślnie kontener działa z własnym wewnętrznym adresem IP, co oznacza, że konieczne jest jawne odwzorowanie wszelkich portów, które mają być wyeksponowane.

W tym przykładzie oznacza to odwzorowanie portu 1414 na hoście na port 1414 w kontenerze.

Parametr Detach

Uruchamia kontener w tle.

Wyniki

Zbudowaliśmy skonfigurowany obraz kontenera i możemy przeglądać uruchomione kontenery za pomocą komendy **docker ps**. Procesy programu IBM MQ działające w kontenerze można wyświetlić za pomocą komendy **docker top**.



Ostrzeżenie:

Dzienniki kontenera można wyświetlić przy użyciu komendy **docker logs \${CONTAINER_ID}**.

Co dalej

- Brak kontenera w wynikach komendy **docker ps** oznacza, że prawdopodobnie wystąpiło niepowodzenie w działaniu kontenera. Uszkodzone kontenery można wyświetlić za pomocą komendy **docker ps -a**.
- Użycie komendy **docker ps -a** powoduje wyświetlenie identyfikatora kontenera. Ten identyfikator jest również drukowany po wydaniu komendy **docker run**.
- Dzienniki kontenera można wyświetlić przy użyciu komendy **docker logs \${CONTAINER_ID}**.

Multi Uruchamianie aplikacji z powiązaniem lokalnymi w oddzielnych kontenerach

W przypadku współużytkowania przestrzeni nazw procesów między kontenerami można uruchamiać aplikacje, które wymagają lokalnego połączenia powiązania z produktem IBM MQ w oddzielnych kontenerach niż menedżer kolejek produktu IBM MQ .

O tym zadaniu

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`.

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 217.

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 216.

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 zwracana pustą odpowiedź:

```
docker exec container dspmq
```

- Jeśli są współużytkowane wszystkie wymagane przestrzenie nazw, ale wolumin współużytkowany nie został podłączony do katalogu `/var/mqm` i nie jest dostępna poprawna ścieżka danych IBM MQ (lub

nie ma dostępnej ścieżki danych IBM MQ), wyświetlane są różne błędy, ponieważ ścieżka danych jest kluczowym komponentem instalacji produktu IBM MQ. Bez ścieżki do danych program IBM MQ nie może działać.

Jeśli zostanie uruchomiona dowolna z następujących komend, a odpowiedzi są podobne do pokazanych przykładach, należy sprawdzić, czy katalog został podłączony i czy utworzono katalog danych IBM MQ:

```
docker exec container dspmq
'No such file or directory' from /var/mqm/mqs.ini
AMQ6090: IBM MQ was unable to display an error message FFFFFFFF.
AMQffff

docker exec container dspmqver
AMQ7047: An unexpected error was encountered by a command. Reason code is 0.

docker exec container mqrc
<file path>/mqrc.c[1152]
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
```

MQ Adv. Tworzenie rodzimej grupy wysokiej dostępności w przypadku tworzenia własnych kontenerów

Aby utworzyć grupę rodzimej wysokiej dostępności, należy utworzyć, skonfigurować i uruchomić trzy menedżery kolejek.

O tym zadaniu

Zalecaną metodą tworzenia rodzimego rozwiązania wysokiej dostępności jest użycie operatora IBM MQ (patrz [Native HA](#)). Alternatywnie, jeśli tworzone są własne kontenery, można postępować zgodnie z poniższymi instrukcjami.

Aby utworzyć grupę rodzimej wysokiej dostępności, należy utworzyć trzy menedżery kolejek w trzech węzłach z typem dziennika ustawionym na `log replication`. Następnie należy przeprowadzić edycję pliku `qm.ini` dla każdego menedżera kolejek, aby dodać szczegóły połączenia dla każdego z trzech węzłów, tak aby mogły one replikować dane dziennika do siebie nawzajem.

Następnie należy uruchomić wszystkie trzy menedżery kolejek, aby mogły sprawdzić, czy wszystkie trzy instancje mogą komunikować się ze sobą i określić, która z nich będzie instancją aktywną, a która replikami.

Uwaga: W ten sposób można utworzyć grupę rodzimej wysokiej dostępności we własnych kontenerach tylko wtedy, gdy używany jest system Kubernetes lub Red Hat OpenShift.

Procedura

1. Na każdym z trzech węzłów utwórz menedżer kolejek, określając typ dziennika repliki dziennika i podając unikalną nazwę dla każdej instancji dziennika. Każdy menedżer kolejek ma taką samą nazwę:

```
crtmqm -lr instance_name qmname
```

Na przykład:

```
node 1> crtmqm -lr qm1_inst1 qm1
node 2> crtmqm -lr qm1_inst2 qm1
node 3> crtmqm -lr qm1_inst3 qm1
```

2. Po pomyślnym utworzeniu każdego menedżera kolejek do pliku konfiguracyjnego menedżera kolejek `qm.ini` dodawana jest dodatkowa sekcja o nazwie `NativeHALocalInstance`. Atrybut `Name` jest dodawany do sekcji określającej podaną nazwę instancji.

Opcjonalnie można dodać następujące atrybuty do sekcji `NativeHALocalInstance` w pliku `qm.ini`:

KeyRepository

Położenie repozytorium kluczy przechowującego certyfikat cyfrowy, który ma być używany do ochrony ruchu związanego z replikacją dzienników. Położenie jest podane w formacie rdzenia, czyli zawiera pełną ścieżkę i nazwę pliku bez rozszerzenia. Jeśli atrybut sekcji `KeyRepository` zostanie pominięty, dane replikacji dziennika będą wymieniane między instancjami przy użyciu zwykłego tekstu.

CertificateLabel

Etykieta certyfikatu identyfikująca certyfikat cyfrowy, który ma być używany do ochrony ruchu replikacji dziennika. Jeśli podano parametr `KeyRepository`, ale pominięto parametr `CertificateLabel`, zostanie użyta wartość domyślna `ibmwebspherequeue_manager`.

CipherSpec

CipherSpec produktu MQ, która ma być używana do ochrony ruchu związanego z replikacją dziennika. Jeśli ten atrybut sekcji jest podany, należy także podać `KeyRepository`. Jeśli podano parametr `KeyRepository`, ale pominięto parametr `CipherSpec`, zostanie użyta wartość domyślna `ANY`.

LocalAddress

Adres lokalnego interfejsu sieciowego, który akceptuje ruch replikacji dziennika. Jeśli podano ten atrybut sekcji, identyfikuje on lokalny interfejs sieciowy i/lub port w formacie "[addr] [(port)]". Adres sieciowy może być podany w postaci nazwy hosta, IPv4 w postaci dziesiętnej z kropkami lub w formacie szesnastkowym IPv6. Jeśli ten atrybut zostanie pominięty, menedżer kolejek podejmuje próbę powiązania ze wszystkimi interfejsami sieciowymi i używa portu określonego w pliku `ReplicationAddress` w sekcji `NativeHAInstances`, który jest zgodny z nazwą lokalnej instancji.

HeartbeatInterval

Interwał pulsu definiuje, co ile milisekund aktywna instancja menedżera kolejek o rodzimej wysokiej dostępności wysyła puls sieciowy. Poprawny zakres wartości interwału pulsu wynosi od 500 (0,5 sekundy) do 60000 (1 minuta), a wartość spoza tego zakresu powoduje, że uruchomienie menedżera kolejek nie powiedzie się. Jeśli ten atrybut zostanie pominięty, zostanie użyta wartość domyślna 5000 (5 sekund). Każda instancja musi używać tego samego interwału pulsu.

HeartbeatTimeout

Limit czasu pulsu definiuje, jak długo instancja-replika menedżera kolejek o rodzimej wysokiej dostępności oczekuje przed podjęciem decyzji, że aktywna instancja nie odpowiada. Poprawny zakres limitu czasu interwału pulsu wynosi od 500 (0,5 sekundy) do 120000 (2 minuty). Wartość limitu czasu pulsu musi być większa niż lub równa wartości interwału pulsu.

Niepoprawna wartość powoduje, że uruchomienie menedżera kolejek nie powiedzie się. Jeśli ten atrybut zostanie pominięty, replika oczekuje na $2 \times \text{HeartbeatInterval}$ przed rozpoczęciem procesu w celu wybrania nowej aktywnej instancji. Każda instancja musi używać tego samego limitu czasu pulsu.

RetryInterval

Interwał ponowienia definiuje, co ile milisekund menedżer kolejek o rodzimej wysokiej dostępności powinien ponawiać próbę nawiązania połączenia przez niesprawne łącze replikacji. Poprawny zakres interwału ponowienia wynosi od 500 (0,5 sekundy) do 120000 (2 minuty). Jeśli ten

atrybut zostanie pominięty, replika oczekuje na $2 \times \text{HeartbeatInterval}$ przed ponowną próbą utworzenia łącza replikacji, które zakończyło się niepowodzeniem.

3. Zmodyfikuj plik `qm.ini` dla każdego menedżera kolejek i dodaj szczegóły połączenia. Należy dodać trzy sekcje `NativeHAInstance`, po jednej dla każdej instancji menedżera kolejek w rodzimej grupie wysokiej dostępności (w tym dla instancji lokalnej). Dodaj następujące atrybuty:

Nazwa

Podaj nazwę instancji, która była używana podczas tworzenia instancji menedżera kolejek.

ReplicationAddress

Podaj nazwę hosta IPv4 w postaci dziesiętnej z kropkami lub adres instancji w formacie szesnastkowym IPv6. Adres można podać w postaci nazwy hosta, IPv4 w postaci dziesiętnej z kropkami lub adresu w formacie szesnastkowym IPv6. Adres replikacji musi być możliwy do rozstrzygnięcia i musi być możliwy do routingu z każdej instancji w grupie. Numer portu, który ma być używany na potrzeby replikacji dziennika, musi być podany w nawiasach kwadratowych, na przykład:

```
ReplicationAddress=host1.example.com(4444)
```

Uwaga: Sekcje `NativeHAInstance` są identyczne dla każdej instancji i mogą być udostępniane za pomocą automatycznej konfiguracji (`crtmqm -ii`).

4. Uruchom każdą z trzech instancji:

```
startmq QMgrName
```

Po uruchomieniu instancji komunikują się one w celu sprawdzenia, czy wszystkie trzy instancje są uruchomione, a następnie decydują, która z tych trzech instancji jest aktywna, podczas gdy pozostałe dwie instancje nadal działają jako repliki.

Przykład

W poniższym przykładzie przedstawiono sekcję pliku `qm.ini` określającą wymagane szczegóły rodzimej wysokiej dostępności dla jednej z trzech instancji:

```
NativeHALocalInstance:
  LocalName=node-1

NativeHAInstance:
  Name=node-1
  ReplicationAddress=host1.example.com(4444)
NativeHAInstance:
  Name=node-2
  ReplicationAddress=host2.example.com(4444)
NativeHAInstance:
  Name=node-3
  ReplicationAddress=host3.example.com(4444)
```

MQ Adv. Uwagi dotyczące wykonywania własnej sekwencyjnej aktualizacji menedżera kolejek rodzimej wysokiej dostępności

Każda aktualizacja wersji IBM MQ lub specyfikacji zasobnika dla menedżera kolejek rodzimej wysokiej dostępności będzie wymagała przeprowadzenia sekwencyjnej aktualizacji instancji menedżera kolejek. Produkt IBM MQ Operator obsługuje tę opcję automatycznie, ale jeśli stworzysz własny kod wdrażania, istnieje kilka ważnych kwestii.

Uwaga: Przykładowy wykres programu Helm zawiera skrypt powłoki do wykonywania sekwencyjnej aktualizacji, jednak skrypt ten **nie** nadaje się do użytku produkcyjnego, ponieważ nie odnosi się do zagadnień w tym temacie.

Kubernetes W produkcie Kubernetes zasoby `StatefulSet` służą do zarządzania uporządkowanymi uruchomieniami i sekwencyjnymi aktualizacjami. Częścią procedury uruchomienia jest indywidualne uruchomienie każdego zasobnika, oczekiwanie na jego gotowość, a następnie przejście do następnego zasobnika. To nie zadziała w przypadku Rdzennej HA, ponieważ wszystkie Pody muszą być uruchomione,

aby mogły przeprowadzić wybory lidera. Dlatego pole `.spec.podManagementPolicy` w `StatefulSet` musi zostać ustawione na wartość `Parallel`. Oznacza to również, że wszystkie zasobniki będą aktualizowane równoległe, co jest szczególnie niepożądane. Z tego powodu produkt `StatefulSet` powinien również używać strategii aktualizacji produktu `OnDelete`.

Niezdolność do wykorzystania kodu sekwencyjnej aktualizacji `StatefulSet` wymaga dostosowanego kodu sekwencyjnej aktualizacji, który powinien uwzględniać następujące kwestie:

- Ogólna procedura sekwencyjnej aktualizacji
- Minimalizowanie czasu przestoju poprzez aktualizację zasobników w najlepszej kolejności
- Obsługa zmian stanu klastra
- Obsługa błędów
- Rozwiązywanie problemów z synchronizacją

Ogólna procedura sekwencyjnej aktualizacji

Kod sekwencyjnej aktualizacji powinien odczekać, aż każda instancja wyświetli status `REPLICA` produktu `dspm`. Oznacza to, że instancja wykonała pewien poziom uruchamiania (na przykład kontener jest uruchomiony, a procesy produktu `MQ` działają), ale niekoniecznie udało się jeszcze skomunikować z innymi instancjami. Na przykład: zasobnik A zostanie zrestartowany, a gdy tylko będzie w stanie `REPLICA`, zasobnik B zostanie zrestartowany. Gdy zasobnik B rozpoczyna się od nowej konfiguracji, powinien być w stanie komunikować się z zasobnikiem A i może tworzyć kworum; wówczas albo A, albo B stanie się nową aktywną instancją.

W ramach tego przydatne jest opóźnienie po osiągnięciu stanu `REPLICA` przez każdy zasobnik, aby umożliwić mu połączenie się z innymi zasobnikami i ustanowienie kworum.

Minimalizowanie czasu przestoju poprzez aktualizację zasobników w najlepszej kolejności

Kod sekwencyjnej aktualizacji powinien usuwać zasobniki pojedynczo, zaczynając od zasobników, które są w znanym stanie błędu, a następnie wszystkich zasobników, które nie zostały pomyślnie uruchomione. Aktywny zasobnik menedżera kolejek powinien być na ogół aktualizowany jako ostatni.

Ważne jest również, aby wstrzymać usuwanie zasobników, jeśli ostatnia aktualizacja spowodowała przejście zasobnika do znanego stanu błędu. Zapobiega to wprowadzeniu przerwanej aktualizacji do wszystkich zasobników. Może się to zdarzyć na przykład wtedy, gdy zasobnik zostanie zaktualizowany, w celu użycia nowego obrazu kontenera, który nie jest dostępny (lub zawiera literówkę).

Obsługa zmian stanu klastra

Kod sekwencyjnej aktualizacji musi odpowiednio reagować na zmiany w czasie rzeczywistym w stanie klastra. Na przykład jeden z zasobników menedżera kolejek może zostać usunięty z powodu restartu węzła lub z powodu nacisku węzła. Możliwe, że usunięty zasobnik może nie zostać natychmiast ponownie zaplanowany, jeśli klaster jest zajęty. W takim przypadku kod sekwencyjnej aktualizacji powinien odpowiednio odczekać przed zrestartowaniem innych zasobników.

Obsługa błędów

Kod sekwencyjnej aktualizacji musi być odporny na awarie podczas wywoływania interfejsu API Kubernetes i innych nieoczekiwanych zachowań w klastrze.

Ponadto sam kod sekwencyjnej aktualizacji musi dopuszczać swoje restartowanie. Sekwencyjna aktualizacja może być długotrwała, a kod może wymagać zrestartowania.

Rozwiązywanie problemów z synchronizacją

Kod sekwencyjnej aktualizacji musi sprawdzić wersje aktualizacji zasobnika, aby mógł zapewnić restart zasobnika. Pozwala to uniknąć problemów z synchronizacją, w których zasobnik może wskazywać status "Uruchomiono", gdy w rzeczywistości jeszcze się nie zakończył.

Pojęcia pokrewne

["Wybór sposobu używania IBM MQ w kontenerach"](#) na stronie 5

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

MQ Adv.

Wyświetlanie statusu rodzimych menedżerów kolejek wysokiej dostępności dla kontenerów niestandardowych

W przypadku kontenerów niestandardowych można wyświetlić status instancji rodzimej wysokiej dostępności za pomocą komendy **dspmqr**.

O tym zadaniu

Za pomocą komendy **dspmqr** można wyświetlić status działania instancji menedżera kolejek w węźle. Zwrócone informacje zależą od tego, czy instancja jest aktywna, czy jest repliką. Informacje dostarczone przez aktywną instancję są definitywne, a informacje z węzłów replik mogą być nieaktualne.

Można wykonać następujące czynności:

- Sprawdź, czy instancja menedżera kolejek w bieżącym węźle jest aktywna, czy jest repliką.
- Wyświetl status działania rodzimej wysokiej dostępności dla instancji w bieżącym węźle.
- Wyświetl status działania wszystkich trzech instancji w konfiguracji rodzimej wysokiej dostępności.

Następujące pola statusu są używane do raportowania statusu konfiguracji rodzimej wysokiej dostępności:

ROLE

Określa bieżącą rolę instancji i jest jedną z następujących wartości: Active, Replica lub Unknown.

INSTANCE

Nazwa podana dla tej instancji menedżera kolejek, gdy była ona tworzona przy użyciu opcji **-lr** komendy **crtmqm**.

INSYNC

Wskazuje, czy w razie potrzeby instancja jest w stanie przejąć rolę aktywnej instancji.

QUORUM

Raportuje status kworum w formacie *liczba zsynchronizowanych instancji/liczba skonfigurowanych instancji*.

REPLADDR

Adres replikacji instancji menedżera kolejek.

CONNECTV

Wskazuje, czy węzeł jest połączony z aktywną instancją.

BACKLOG

Określa zaległość instancji w kB.

CONNINST

Informuje, czy instancja o podanej nazwie jest połączona z tą instancją.

ALTDATA

Wskazuje datę ostatniej aktualizacji tych informacji (wartość pusta, jeśli nigdy nie zostały zaktualizowane).

ALTTIME

Wskazuje godzinę ostatniej aktualizacji tych informacji (wartość pusta, jeśli nigdy nie zostały zaktualizowane).

Procedura

- Aby określić, czy instancja menedżera kolejek działa jako aktywna instancja, czy jako replika:

```
dspmqr -o status -m QMgrName
```

Aktywna instancja menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          STATUS(Running)
```

Instancja-replika menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          STATUS(Replica)
```

Nieaktywna instancja zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          STATUS(Ended Immediately)
```

- Aby określić status działania rodzimej wysokiej dostępności instancji w bieżącym węźle:

```
dspmqr -o nativeha -m QMgrName
```

Aktywna instancja menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

Instancja-replika menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```

Nieaktywna menedżera kolejek o nazwie BOB zwróciłaby następujące informacje o statusie:

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- Aby określić status działania rodzimej wysokiej dostępności dla wszystkich instancji w konfiguracji o rodzimej wysokiej dostępności:

```
dspmqr -o nativeha -x -m QMgrName
```

Jeśli ta komenda zostanie wydana w węźle, w którym działa aktywna instancja menedżera kolejek BOB, może zostać wyświetlony następujący status:

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

Jeśli ta komenda zostanie wydana w węźle, w którym uruchomiona jest instancja-replika menedżera kolejek BOB, może zostać wyświetlony następujący status, który wskazuje, że jedna z replik ma zaległości:

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No) BACKLOG(435)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

Jeśli ta komenda zostanie wydana w węźle, w którym uruchomiona jest nieaktywna instancja menedżera kolejek BOB, może zostać wyświetlony następujący status:

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
INSTANCE(inst1)     ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown)   CONNINST(No) ALTDATE() ALTTIME()
INSTANCE(inst2)     ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown)   CONNINST(No) ALTDATE() ALTTIME()
INSTANCE(inst3)     ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown)   CONNINST(No) ALTDATE() ALTTIME()
```

W przypadku wydania komendy w momencie, gdy instancje nadal negocjują, która będzie aktywna, a które będą replikami, zwrócony zostanie następujący status:

```
QMNAME(BOB)          STATUS(Negotiating)
```

Odsyłacze pokrewne

[Komenda dspmq \(wyświetl menedżery kolejek\)](#)

MQ Adv.

Kończenie działania rodzimych menedżerów kolejek HA

Za pomocą komendy **endmqm** można zakończyć działanie aktywnego menedżera kolejek lub menedżera kolejek repliki, który jest częścią rodzimej grupy wysokiej dostępności.

Procedura

- Aby zakończyć aktywną instancję menedżera kolejek, należy zapoznać się z sekcją [Kończenie działania rodzimych menedżerów kolejek o wysokiej dostępności](#) w sekcji Konfigurowanie tej dokumentacji.

Uwagi

Niniejsza publikacja została opracowana z myślą o produktach i usługach oferowanych w Stanach Zjednoczonych.

IBM może nie oferować w innych krajach produktów, usług lub opcji omawianych w tej publikacji. Informacje o produktach i usługach dostępnych w danym kraju można uzyskać od lokalnego przedstawiciela IBM. Odwołanie do produktu, programu lub usługi IBM nie oznacza, że można użyć wyłącznie tego produktu, programu lub usługi IBM. Zamiast nich można zastosować ich odpowiednik funkcjonalny pod warunkiem, że nie narusza to praw własności intelektualnej firmy IBM. Jednakże cała odpowiedzialność za ocenę przydatności i sprawdzenie działania produktu, programu lub usługi pochodzących od producenta innego niż IBM spoczywa na użytkowniku.

IBM może posiadać patenty lub złożone wnioski patentowe na towary i usługi, o których mowa w niniejszej publikacji. Przedstawienie niniejszej publikacji nie daje żadnych uprawnień licencyjnych do tychże 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 oprogramowania, dział 49XA
3605 Autostrada 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 o interfejsie programistycznym, jeśli są dostępne, mają na celu pomóc w tworzeniu aplikacji do użycia z tym programem.

Podręcznik ten zawiera informacje na temat 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 o diagnostyce, modyfikacji i strojeniu nie należy używać jako interfejsu programistycznego, ponieważ mogą one 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 dostępna jest w serwisie WWW IBM, w sekcji "Copyright and trademark information" (Informacje o prawach autorskich i znakach towarowych), pod adresem 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 firmy Microsoft Corporation w Stanach Zjednoczonych i/lub innych krajach.

UNIX jest zastrzeżonym znakiem towarowym The Open Group w Stanach Zjednoczonych i/lub w innych krajach.

Linux jest zastrzeżonym znakiem towarowym Linusa Torvaldsa w Stanach Zjednoczonych i/lub w innych krajach.

Ten produkt zawiera oprogramowanie opracowane przez Eclipse Project (<https://www.eclipse.org/>).

Java oraz wszystkie znaki towarowe i logo dotyczące języka Java są znakami towarowymi lub zastrzeżonymi znakami towarowymi Oracle i/lub przedsiębiorstw afiliowanych Oracle.



Numer pozycji:

(1P) P/N: