



OpenShift Container Platform 4.10

Migrazione dalla versione 3 alla 4

Migrazione a OpenShift Container Platform 4

OpenShift Container Platform 4.10 Migrazione dalla versione 3 alla 4

Migrazione a OpenShift Container Platform 4

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

Nota Legale

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Migrating_from_version_3_to_4.ent file |.

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

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

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

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

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

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Sommario

Questo documento fornisce le istruzioni per eseguire la migrazione del cluster di OpenShift Container Platform dalla versione 3 alla versione 4.

Indice	
CAPITOLO 1. PANORAMICA SULLA MIGRAZIONE DA OPENSIFT CONTAINER PLATFORM 3 A 4	6
1.1. DIFFERENZE TRA OPENSIFT CONTAINER PLATFORM 3 E 4	6
1.2. CONSIDERAZIONI SULLA PIANIFICAZIONE DELLA RETE	6
1.3. INSTALLAZIONE DI MTC	6
1.4. AGGIORNAMENTO DI MTC	7
1.5. REVISIONE DELLE LISTE DI CONTROLLO PRE-MIGRAZIONE	7
1.6. MIGRAZIONE DI APPLICAZIONI	7
1.7. OPZIONI DI MIGRAZIONE AVANZATE	7
1.8. RISOLUZIONE DEI PROBLEMI DELLE MIGRAZIONI	7
1.9. ROLLBACK DI UNA MIGRAZIONE	7
1.10. DISINSTALLAZIONE DI MTC ED ELIMINAZIONE DI RISORSE	8
CAPITOLO 2. INFORMAZIONI SULLA MIGRAZIONE DA OPENSIFT CONTAINER PLATFORM 3 A 4	9
CAPITOLO 3. DIFFERENZE TRA OPENSIFT CONTAINER PLATFORM 3 E 4	10
3.1. ARCHITETTURA	10
Infrastruttura immutabile	10
Operatori	10
3.2. INSTALLAZIONE E AGGIORNAMENTO	11
Processo di installazione	11
Opzioni di infrastruttura	11
Aggiornamento del cluster	11
3.3. CONSIDERAZIONI SULLA MIGRAZIONE	11
3.3.1. Considerazioni sullo storage	11
Storage permanente dei volumi locali	11
Storage permanente FlexVolume	12
Storage permanente Container Storage Interface (CSI)	12
Red Hat OpenShift Container Storage	12
Opzioni di storage permanente non supportate	12
3.3.2. Considerazioni sulla rete	12
Modalità di isolamento della rete	12
3.3.3. Considerazioni sulla registrazione	13
Deployment di OpenShift Logging	13
Dati di registrazione aggregati	13
Configurazioni di registrazione non supportate	13
3.3.4. Considerazioni sulla sicurezza	13
Accesso non autenticato agli endpoint di rilevamento	13
Provider di identità	13
Formato di storage dei token OAuth	14
3.3.5. Considerazioni sul monitoraggio	14
Avviso per il monitoraggio della disponibilità dell'infrastruttura	14
CAPITOLO 4. CONSIDERAZIONI SULLA RETE	15
4.1. CONSIDERAZIONI SUL DNS	15
4.1.1. Isolare il dominio DNS del cluster di destinazione dai client	15
4.1.2. Impostare il cluster di destinazione per accettare il dominio DNS di origine	15
4.2. STRATEGIE DI REINDIRIZZAMENTO DEL TRAFFICO DI RETE	16
CAPITOLO 5. INFORMAZIONI SUL MIGRATION TOOLKIT FOR CONTAINERS	18
5.1. TERMINOLOGIA	18
5.2. WORKFLOW DI MTC	19
5.3. INFORMAZIONI SUI METODI DI COPIA DEI DATI	22

5.3.1. Metodo di copia del file system	22
5.3.2. Metodo di copia degli snapshot	23
5.4. MIGRAZIONE DIRETTA DEI VOLUMI E MIGRAZIONE DIRETTA DELLE IMMAGINI	23
CAPITOLO 6. INSTALLARE IL MIGRATION TOOLKIT FOR CONTAINERS	24
6.1. LINEE GUIDA PER LA COMPATIBILITÀ	24
6.2. INSTALLARE IL MIGRATION TOOLKIT FOR CONTAINERS OPERATOR ESISTENTE SU OPENSIFT CONTAINER PLATFORM 3	24
6.3. INSTALLARE IL MIGRATION TOOLKIT FOR CONTAINERS OPERATOR SU OPENSIFT CONTAINER PLATFORM 4.10	26
6.4. CONFIGURARE I PROXY	26
6.5. CONFIGURARE UN REPOSITORY DI REPLICA	28
6.5.1. Requisiti	28
6.5.2. Recuperare le credenziali di Multicloud Object Gateway	28
6.5.3. Configurazione di Amazon Web Services	29
6.5.4. Configurazione di Google Cloud Platform	31
6.5.5. Configurazione di Microsoft Azure	33
6.5.6. Risorse aggiuntive	34
6.6. DISINSTALLAZIONE DI MTC ED ELIMINAZIONE DI RISORSE	34
CAPITOLO 7. INSTALLARE MIGRATION TOOLKIT FOR CONTAINERS IN UN AMBIENTE DI RETE CON RESTRIZIONI	36
7.1. LINEE GUIDA PER LA COMPATIBILITÀ	36
7.2. INSTALLARE IL MIGRATION TOOLKIT FOR CONTAINERS OPERATOR SU OPENSIFT CONTAINER PLATFORM 4.10	36
7.3. INSTALLARE IL MIGRATION TOOLKIT FOR CONTAINERS OPERATOR ESISTENTE SU OPENSIFT CONTAINER PLATFORM 3	37
7.4. CONFIGURARE I PROXY	39
7.5. CONFIGURARE UN REPOSITORY DI REPLICA	41
7.5.1. Requisiti	41
7.5.2. Recuperare le credenziali di Multicloud Object Gateway	41
7.5.3. Risorse aggiuntive	41
7.6. DISINSTALLAZIONE DI MTC ED ELIMINAZIONE DI RISORSE	41
CAPITOLO 8. AGGIORNARE IL MIGRATION TOOLKIT FOR CONTAINERS	43
8.1. AGGIORNARE IL MIGRATION TOOLKIT FOR CONTAINERS SU OPENSIFT CONTAINER PLATFORM 4.10	43
8.2. AGGIORNARE IL MIGRATION TOOLKIT FOR CONTAINERS SU OPENSIFT CONTAINER PLATFORM 3	43
8.3. AGGIORNAMENTO DI MTC 1.3 A 1.6	45
CAPITOLO 9. LISTE DI CONTROLLO DI PRE-MIGRAZIONE	46
9.1. RISORSE	46
9.2. CLUSTER DI ORIGINE	46
9.3. CLUSTER DI DESTINAZIONE	47
9.4. PRESTAZIONI	48
CAPITOLO 10. MIGRAZIONE DELLE APPLICAZIONI	49
10.1. REQUISITI PER LA MIGRAZIONE	49
Risorse aggiuntive per i requisiti della migrazione	50
10.2. MIGRAZIONE DI APPLICAZIONI UTILIZZANDO LA CONSOLE WEB DI MTC	51
10.2.1. Lanciare la console web di MTC	51
10.2.2. Aggiungere un cluster alla console web di MTC	51
10.2.3. Aggiungere un repository di replica alla console web di MTC	53
10.2.4. Creare un piano di migrazione nella console web di MTC	54

Risorse aggiuntive	56
10.2.5. Eseguire un piano di migrazione nella console web di MTC	57
CAPITOLO 11. OPZIONI DI MIGRAZIONE AVANZATE	59
11.1. TERMINOLOGIA	59
11.2. ESEGUIRE LA MIGRAZIONE DI APPLICAZIONI USANDO LA RIGA DI COMANDO	60
11.2.1. Requisiti per la migrazione	60
11.2.2. Creare un percorso di registro per la migrazione diretta delle immagini	61
11.2.3. Configurare i proxy	62
11.2.4. Eseguire la migrazione di un'applicazione tramite l'API MTC	63
11.2.5. Migrazione dello stato	68
Risorse aggiuntive	69
11.3. HOOK DI MIGRAZIONE	69
11.3.1. Scrivere un Ansible Playbook per un hook di migrazione	70
11.3.1.1. Moduli Ansible	70
11.3.1.2. Variabili d'ambiente	71
11.4. OPZIONI DEL PIANO DI MIGRAZIONE	71
11.4.1. Escludere risorse	71
11.4.2. Mapping degli spazi dei nomi	73
11.4.3. Escludere le richieste di volumi permanenti	73
11.4.4. Mapping delle richieste di volumi permanenti	74
11.4.5. Modifica degli attributi dei volumi permanenti	75
Risorse aggiuntive	76
11.4.6. Eseguire la migrazione di oggetti Kubernetes	76
11.5. OPZIONI DEL CONTROLLER DI MIGRAZIONE	77
11.5.1. Aumentare i limiti per le migrazioni di grandi dimensioni	77
11.5.2. Abilitare il ridimensionamento dei volumi permanenti per la migrazione diretta dei volumi	78
11.5.3. Abilitare i client Kubernetes nella cache	79
CAPITOLO 12. RISOLUZIONE DEI PROBLEMI	81
12.1. WORKFLOW DI MTC	81
Informazioni sulle risorse personalizzate di MTC	83
12.2. MANIFEST DELLE RISORSE PERSONALIZZATE DI MTC	84
12.2.1. DirectImageMigration	84
12.2.2. DirectImageStreamMigration	85
12.2.3. DirectVolumeMigration	85
12.2.4. DirectVolumeMigrationProgress	86
12.2.5. MigAnalytic	86
12.2.6. MigCluster	87
12.2.7. MigHook	88
12.2.8. MigMigration	89
12.2.9. MigPlan	90
12.2.10. MigStorage	91
12.3. REGISTRI E STRUMENTI DI DEBUGGING	92
12.3.1. Visualizzare le risorse del piano di migrazione	92
12.3.2. Visualizzare il registro di un piano di migrazione	93
12.3.3. Usare il lettore dei registri di migrazione	94
12.3.4. Accedere alle metriche delle prestazioni	94
12.3.4.1. Metriche fornite	94
12.3.4.1.1. cam_app_workload_migrations	94
12.3.4.1.2. mtc_client_request_count	95
12.3.4.1.3. mtc_client_request_elapsed	95
12.3.4.1.4. Domande utili	96

12.3.5. Usare lo strumento must-gather	96
Visualizzare i dati delle metriche con la console Prometheus	97
12.3.6. Debugging delle risorse di Velero con lo strumento CLI di Velero	97
Syntax	98
Opzione aiuto	98
Comando descrizione	98
Comando dei registri	98
12.3.7. Debugging di un errore parziale della migrazione	98
12.3.8. Utilizzare le risorse personalizzate di MTC per la risoluzione dei problemi	100
12.4. PROBLEMI E PREOCCUPAZIONI COMUNI	103
12.4.1. Aggiornamento delle immagini interne deprecate	103
12.4.2. La migrazione diretta dei volumi non si completa	105
12.4.3. Messaggi di errore e risoluzioni	106
12.4.3.1. Errore del certificato CA visualizzato quando si accede alla console MTC per la prima volta	106
12.4.3.2. Errore di timeout OAuth nella console MTC	106
12.4.3.3. Errore del certificato firmato da un'autorità sconosciuta	106
12.4.3.4. Errori di posizione dello storage di backup nel log del pod Velero	107
12.4.3.5. Errore di timeout del backup del volume del pod nel log del pod Velero	107
12.4.3.6. Errori di verifica Restic nella risorsa personalizzata MigMigration	107
12.4.3.7. Errore di autorizzazione di Restic durante la migrazione dallo storage NFS con root_squash abilitato	109
12.4.4. Problemi noti	109
12.5. ROLLBACK DI UNA MIGRAZIONE	110
12.5.1. Rollback di una migrazione utilizzando la console web di MTC	110
12.5.2. Rollback di una migrazione dall'interfaccia della riga di comando	111
12.5.3. Rollback manuale di una migrazione	113
Risorse aggiuntive	113

CAPITOLO 1. PANORAMICA SULLA MIGRAZIONE DA OPENSIFT CONTAINER PLATFORM 3 A 4

I cluster di OpenShift Container Platform 4 sono diversi dai cluster di OpenShift Container Platform 3. I cluster di OpenShift Container Platform 4 contengono nuove tecnologie e funzionalità che consentono di ottenere un cluster autogestito, flessibile e automatizzato. Per saperne di più sulla migrazione da OpenShift Container Platform 3 a 4 consulta [Informazioni sulla migrazione da OpenShift Container Platform 3 a 4](#).

1.1. DIFFERENZE TRA OPENSIFT CONTAINER PLATFORM 3 E 4

Prima di eseguire la migrazione da OpenShift Container Platform 3 a 4, ti invitiamo a rivedere le [differenze tra OpenShift Container Platform 3 e 4](#). Rivedi le seguenti informazioni:

- [Architettura](#)
- [Installazione e aggiornamento](#)
- [Considerazioni su storage, rete, registrazione, sicurezza e monitoraggio](#)

1.2. CONSIDERAZIONI SULLA PIANIFICAZIONE DELLA RETE

Prima di eseguire la migrazione da OpenShift Container Platform 3 a 4, ti invitiamo a rivedere le [differenze tra OpenShift Container Platform 3 e 4](#) per informazioni sui seguenti argomenti:

- [Considerazioni sul DNS](#)
 - [Isolare il dominio DNS del cluster di destinazione dai client](#) .
 - [Impostare il cluster di destinazione per accettare il dominio DNS di origine](#) .

È possibile eseguire la migrazione dei carichi di lavoro delle applicazioni stateful da OpenShift Container Platform 3 a 4 al livello di granularità di uno spazio dei nomi. Per saperne di più su MTC consulta la sezione [I vantaggi di MTC](#) .



NOTA

Se esegui la migrazione da OpenShift Container Platform 3, consulta la sezione [Informazioni sulla migrazione da OpenShift Container Platform 3 a 4](#) e [Installazione del Migration Toolkit for Containers Operator](#) esistente su OpenShift Container Platform 3.

1.3. INSTALLAZIONE DI MTC

Esamina le seguenti attività per installare MTC:

1. [Installare il Migration Toolkit for Containers Operator sul cluster di destinazione utilizzando Operator Lifecycle Manager \(OLM\)](#).
2. [Installare manualmente il Migration Toolkit for Containers Operator esistente sul cluster di origine](#).
3. [Configurare lo storage a oggetti da usare come repository di replica](#) .

1.4. AGGIORNAMENTO DI MTC

Si aggiorna il [Migration Toolkit for Containers \(MTC\)](#) su OpenShift Container Platform 4.10 utilizzando OLM. Si aggiorna MTC su OpenShift Container Platform 3 reinstallando il Migration Toolkit for Containers Operator esistente.

1.5. REVISIONE DELLE LISTE DI CONTROLLO PRE-MIGRAZIONE

Prima di eseguire la migrazione dei carichi di lavoro delle applicazioni con Migration Toolkit for Containers (MTC), esamina le [liste di controllo pre-migrazione](#).

1.6. MIGRAZIONE DI APPLICAZIONI

È possibile eseguire la migrazione delle applicazioni utilizzando la [console web di MTC](#) o la [riga di comando](#).

1.7. OPZIONI DI MIGRAZIONE AVANZATE

È possibile automatizzare le migrazioni e modificare le risorse personalizzate di MTC per migliorare le prestazioni delle migrazioni di grandi dimensioni utilizzando le seguenti opzioni:

- [Eseguire una migrazione di stato](#)
- [Creare hook di migrazione](#)
- [Modificare, escludere e mappare le risorse migrate](#)
- [Configurare il controller di migrazione per grandi migrazioni](#)

1.8. RISOLUZIONE DEI PROBLEMI DELLE MIGRAZIONI

È possibile eseguire le seguenti attività di risoluzione dei problemi:

- [Visualizzare le risorse del piano di migrazione utilizzando la console web di MTC](#)
- [Visualizzare il file di registro aggregato del piano di migrazione](#)
- [Usare il lettore dei registri di migrazione](#)
- [Accedere alle metriche delle prestazioni](#)
- [Usare lo strumento **must-gather**](#)
- [Usare la CLI di Velero per eseguire il debugging delle risorse personalizzate](#) **Backup e Restore**
- [Utilizzare le risorse personalizzate di MTC per la risoluzione dei problemi](#)
- [Controllare problemi e dubbi comuni](#)

1.9. ROLLBACK DI UNA MIGRAZIONE

È possibile eseguire il [rollback di una migrazione](#) utilizzando la web console web di MTC, tramite la CLI o manualmente.

1.10. DISINSTALLAZIONE DI MTC ED ELIMINAZIONE DI RISORSE

È possibile [disinstallare l'MTC ed eliminare le relative risorse](#) per pulire il cluster.

CAPITOLO 2. INFORMAZIONI SULLA MIGRAZIONE DA OPENSIFT CONTAINER PLATFORM 3 A 4

OpenShift Container Platform 4 contiene nuove tecnologie e funzionalità che consentono di ottenere un cluster autogestito, flessibile e automatizzato. I cluster di OpenShift Container Platform 4 sono distribuiti e gestiti in modo molto diverso rispetto a OpenShift Container Platform 3.

Il modo più efficace per eseguire la migrazione da OpenShift Container Platform 3 a 4 è utilizzare una pipeline CI/CD per automatizzare i deployment in un framework di [gestione del ciclo di vita delle applicazioni](#).

Se non si dispone di una pipeline CI/CD o se si esegue la migrazione di applicazioni stateful, è possibile usare il Migration Toolkit for Containers (MTC) per eseguire la migrazione dei carichi di lavoro delle applicazioni.

Per una corretta migrazione a OpenShift Container Platform 4, ti invitiamo a rivedere le seguenti informazioni:

Differenze tra OpenShift Container Platform 3 e 4

- Architettura
- Installazione e aggiornamento
- Considerazioni su storage, rete, registrazione, sicurezza e monitoraggio

Informazioni sul Migration Toolkit for Containers

- Workflow
- Metodi di copia del file system e degli snapshot per volumi permanenti (PV)
- Migrazione diretta di volumi
- Migrazione diretta di immagini

Opzioni di migrazione avanzate

- Automatizzare la migrazione con gli hook di migrazione
- Usare l'API MTC
- Escludere risorse da un piano di migrazione
- Configurare la risorsa personalizzata **MigrationController** per le migrazioni di grandi dimensioni
- Abilitare il ridimensionamento automatico di PV per la migrazione diretta dei volumi
- Abilitare i client Kubernetes nella cache per migliorare le prestazioni

Per scoprire nuove funzionalità e miglioramenti, modifiche tecniche e problemi noti, consulta le [note di rilascio di MTC](#).

CAPITOLO 3. DIFFERENZE TRA OPENSIFT CONTAINER PLATFORM 3 E 4

OpenShift Container Platform 4.10 introduce modifiche architetturali e miglioramenti. Le procedure utilizzate per gestire il cluster di OpenShift Container Platform 3 potrebbero non essere applicabili a OpenShift Container Platform 4.

Per informazioni sulla configurazione del cluster di OpenShift Container Platform 4, consulta le sezioni appropriate della documentazione di OpenShift Container Platform. Per informazioni sulle nuove funzionalità e altre modifiche tecniche degne di nota, consulta le [note di rilascio di OpenShift Container Platform 4.10](#).

Non è possibile aggiornare il cluster esistente di OpenShift Container Platform 3 a OpenShift Container Platform 4. È necessario iniziare con una nuova installazione di OpenShift Container Platform 4. Sono disponibili strumenti per assistere con la migrazione delle impostazioni del piano di controllo e dei carichi di lavoro delle applicazioni.

3.1. ARCHITETTURA

Con OpenShift Container Platform 3, gli amministratori erano soliti distribuire individualmente gli host Red Hat Enterprise Linux (RHEL) per poi installare OpenShift Container Platform sopra questi host in modo da formare un cluster. Gli amministratori erano responsabili della corretta configurazione di questi host e dell'esecuzione degli aggiornamenti.

OpenShift Container Platform 4 rappresenta un cambiamento significativo nel modo in cui i cluster di OpenShift Container Platform sono distribuiti e gestiti. OpenShift Container Platform 4 include nuove tecnologie e funzionalità, come Operatori, set di macchine e Red Hat Enterprise Linux CoreOS (RHCOS), che sono fondamentali per il funzionamento del cluster. Questo cambiamento tecnologico permette ai cluster di gestire in automatico alcune funzioni precedentemente svolte dagli amministratori. Ciò assicura anche la stabilità e la coerenza della piattaforma, e semplifica l'installazione e la scalabilità.

Per ulteriori informazioni, consulta la sezione [Architettura di OpenShift Container Platform](#).

Infrastruttura immutabile

OpenShift Container Platform 4 utilizza Red Hat Enterprise Linux CoreOS (RHCOS), che è progettato per l'esecuzione di applicazioni containerizzate e fornisce un'installazione efficiente, una gestione basata su operatori e aggiornamenti semplificati. RHCOS è un host container immutabile, piuttosto che un sistema operativo personalizzabile come RHEL. RHCOS permette a OpenShift Container Platform 4 di gestire e automatizzare il deployment dell'host container sottostante. RHCOS fa parte di OpenShift Container Platform, il che significa che tutto viene eseguito all'interno di un container e viene distribuito utilizzando OpenShift Container Platform.

In OpenShift Container Platform 4, i nodi del piano di controllo devono eseguire RHCOS, assicurando che l'automazione full-stack sia mantenuta per il piano di controllo. Questo rende il rollout di aggiornamenti e upgrade un processo molto più facile che in OpenShift Container Platform 3.

Per ulteriori informazioni, consulta la pagina [Red Hat Enterprise Linux CoreOS \(RHCOS\)](#).

Operatori

Gli operatori sono un metodo per la creazione di pacchetti, il deployment e la gestione di un'applicazione Kubernetes. Gli operatori alleggeriscono la complessità operativa che deriva dall'esecuzione di un ulteriore componente software. Sorvegliano l'ambiente e utilizzano lo stato attuale per prendere decisioni in tempo reale. Gli operatori avanzati sono progettati per aggiornare e reagire ai guasti automaticamente.

Per maggiori informazioni, consulta [Capire gli operatori](#).

3.2. INSTALLAZIONE E AGGIORNAMENTO

Processo di installazione

Per installare OpenShift Container Platform 3.11, il processo di installazione prevedeva la preparazione degli host Red Hat Enterprise Linux (RHEL), l'impostazione di tutti i valori di configurazione necessari al cluster e infine l'esecuzione di un Ansible Playbook per installare e impostare il cluster.

In OpenShift Container Platform 4.10, si utilizza il programma di installazione di OpenShift per creare un set minimo di risorse necessarie per un cluster. Una volta che il cluster è in funzione, si usano gli operatori per configurare ulteriormente il cluster e per installare nuovi servizi. Dopo il primo avvio, i sistemi Red Hat Enterprise Linux CoreOS (RHCOS) sono gestiti dal Machine Config Operator (MCO) che viene eseguito nel cluster di OpenShift Container Platform.

Per maggiori informazioni, consulta la sezione [Processo di installazione](#).

Per aggiungere macchine di lavoro Red Hat Enterprise Linux (RHEL) al cluster OpenShift Container Platform 4.10, usa un Ansible Playbook per unire le macchine di lavoro RHEL una volta che il cluster è in esecuzione. Per ulteriori informazioni, consulta la sezione [Aggiungere macchine di calcolo RHEL a un cluster OpenShift Container Platform](#).

Opzioni di infrastruttura

In OpenShift Container Platform 3.11, il cluster veniva installato su un'infrastruttura preparata e mantenuta dal personale. Oltre a fornire la propria infrastruttura, OpenShift Container Platform 4 offre un'opzione per distribuire un cluster su un'infrastruttura fornita dal programma di installazione di OpenShift Container Platform e mantenuta dal cluster.

Per ulteriori informazioni, consulta la sezione [Panoramica sull'installazione di OpenShift Container Platform](#).

Aggiornamento del cluster

In OpenShift Container Platform 3.11, il cluster veniva aggiornato eseguendo gli Ansible Playbook. In OpenShift Container Platform 4.10, il cluster gestisce i propri aggiornamenti, compresi quelli a Red Hat Enterprise Linux CoreOS (RHCOS) sui nodi del cluster. È possibile aggiornare il cluster con facilità utilizzando la console web o il comando **oc adm upgrade** dalla CLI di OpenShift; gli operatori si aggiorneranno automaticamente. Se il cluster OpenShift Container Platform 4.10 dispone di macchine di lavoro RHEL, sarà comunque necessario eseguire un Ansible Playbook per aggiornare quelle macchine di lavoro.

Per maggiori informazioni, consulta la sezione [Aggiornamento dei cluster](#).

3.3. CONSIDERAZIONI SULLA MIGRAZIONE

Esamina i cambiamenti e altre considerazioni che potrebbero influenzare la transizione da OpenShift Container Platform 3.11 a OpenShift Container Platform 4.

3.3.1. Considerazioni sullo storage

Rivedi le seguenti modifiche allo storage da tenere in considerazione durante la transizione da OpenShift Container Platform 3.11 a OpenShift Container Platform 4.10.

Storage permanente dei volumi locali

Lo storage locale è supportato solo utilizzando la funzione Local Storage Operator in OpenShift Container Platform 4.10. Non è supportato l'uso del metodo del provisioner locale di OpenShift Container Platform 3.11.

Per maggiori informazioni, consulta la sezione [Storage permanente usando volumi locali](#).

Storage permanente FlexVolume

La posizione del plug-in FlexVolume è cambiata rispetto a OpenShift Container Platform 3.11. La nuova posizione in OpenShift Container Platform 4.10 è `/etc/kubernetes/kubelet-plugins/volume/exec`. I plug-in FlexVolume collegabili non sono più supportati.

Per maggiori informazioni, consulta la sezione [Storage permanente usando FlexVolume](#).

Storage permanente Container Storage Interface (CSI)

La funzione di storage permanente utilizzando la Container Storage Interface (CSI) è stata un'[anteprima tecnica](#) in OpenShift Container Platform 3.11. OpenShift Container Platform 4.10 viene fornita con [diversi driver CSI](#). È possibile anche installare il proprio driver.

Per maggiori informazioni, consulta la sezione [Storage permanente usando la Container Storage Interface \(CSI\)](#).

Red Hat OpenShift Container Storage

Red Hat OpenShift Container Storage 3, che è disponibile per l'uso con OpenShift Container Platform 3.11, utilizza Red Hat Gluster Storage come storage ausiliario.

Red Hat OpenShift Container Storage 4, che è disponibile per l'uso con OpenShift Container Platform 4, utilizza Red Hat Ceph Storage come storage ausiliario.

Per ulteriori informazioni, consulta la sezione [Storage permanente usando Red Hat OpenShift Container Storage](#) e l'articolo sulla [matrice di interoperabilità](#).

Opzioni di storage permanente non supportate

Il supporto per le seguenti opzioni di storage permanente da OpenShift Container Platform 3.11 è cambiato in OpenShift Container Platform 4.10:

- GlusterFS non è più supportato.
- CephFS come prodotto standalone non è più supportato.
- Ceph RBD come prodotto standalone non è più supportato.

Se hai utilizzato una di queste opzioni di storage in OpenShift Container Platform 3.11, devi scegliere una diversa opzione di storage permanente per garantire un supporto completo in OpenShift Container Platform 4.10.

Per maggiori informazioni, consulta la sezione [I vantaggi dello storage permanente](#).

3.3.2. Considerazioni sulla rete

Esamina le seguenti modifiche di rete da tenere in considerazione durante la transizione da OpenShift Container Platform 3.11 a OpenShift Container Platform 4.10.

Modalità di isolamento della rete

La modalità di isolamento di rete predefinita per OpenShift Container Platform 3.11 era **ovs-subnet**, anche se gli utenti spesso passavano all'utilizzo di **ovn-multitenant**. La modalità predefinita di isolamento della rete per OpenShift Container Platform 4.10 è controllata da un criterio di rete.

Se il cluster OpenShift Container Platform 3.11 usava la modalità **ovs-subnet** o **ovs-multitenant**, si raccomanda di passare a un criterio di rete per il cluster OpenShift Container Platform 4.10. I criteri di rete sono supportati upstream, sono più flessibili e forniscono la stessa funzionalità di **ovs-multitenant**. Per mantenere il comportamento **ovs-multitenant** mentre si usa un criterio di rete in OpenShift Container Platform 4.10, si consiglia di seguire i passaggi per [configurare l'isolamento multitenant usando un criterio di rete](#).

Per ulteriori informazioni, consulta la sezione [Informazioni sui criteri di rete](#).

3.3.3. Considerazioni sulla registrazione

Esamina le seguenti modifiche alla registrazione da tenere in considerazione durante la transizione da OpenShift Container Platform 3.11 a OpenShift Container Platform 4.10.

Deployment di OpenShift Logging

OpenShift Container Platform 4 fornisce un semplice meccanismo di deployment per OpenShift Logging, utilizzando una risorsa personalizzata Cluster Logging.

Per maggiori informazioni, consulta la sezione [Installazione di OpenShift Logging](#).

Dati di registrazione aggregati

Non è possibile trasferire i dati di registrazione aggregati da OpenShift Container Platform 3.11 al nuovo cluster di OpenShift Container Platform 4.

Per maggiori informazioni, consulta la sezione [Informazioni su OpenShift Logging](#).

Configurazioni di registrazione non supportate

Alcune configurazioni di registrazione che erano disponibili in OpenShift Container Platform 3.11 non sono più supportate in OpenShift Container Platform 4.10.

Per maggiori informazioni sui casi di registrazione esplicitamente non supportati, consulta la sezione [Manutenzione e supporto](#).

3.3.4. Considerazioni sulla sicurezza

Rivedi le seguenti modifiche alla sicurezza da tenere in considerazione durante la transizione da OpenShift Container Platform 3.11 a OpenShift Container Platform 4.10.

Accesso non autenticato agli endpoint di rilevamento

In OpenShift Container Platform 3.11, un utente non autenticato poteva accedere agli endpoint di rilevamento (per esempio, `/api/*` e `/apis/*`). Per ragioni di sicurezza, l'accesso non autenticato agli endpoint di rilevamento non è più consentito in OpenShift Container Platform 4.10. Se hai bisogno di consentire l'accesso non autenticato, puoi configurare le impostazioni RBAC come necessario; tuttavia, assicurati di considerare le implicazioni per la sicurezza poiché questo può esporre i componenti interni del cluster alla rete esterna.

Provider di identità

La configurazione per i provider di identità è cambiata in OpenShift Container Platform 4, comprese le seguenti modifiche degne di nota:

- Il provider di identità dell'intestazione della richiesta in OpenShift Container Platform 4.10 richiede TLS reciproco, mentre questo non era richiesto in OpenShift Container Platform 3.11.
- La configurazione del provider di identità OpenID Connect è stata semplificata in OpenShift Container Platform 4.10. Ora ottiene i dati, che in precedenza dovevano essere specificati in OpenShift Container Platform 3.11, dall'endpoint `/.well-known/openid-configuration` del

provider.

Per maggiori informazioni, consulta la sezione [Informazioni sulla configurazione del provider di identità](#) .

Formato di storage dei token OAuth

I token di connessione HTTP OAuth appena creati non corrispondono più ai nomi dei relativi oggetti token di accesso OAuth. I nomi degli oggetti sono ora un hash del token di connessione e non sono più sensibili. Questo riduce il rischio di fuga di informazioni sensibili.

3.3.5. Considerazioni sul monitoraggio

Esamina le seguenti modifiche sul monitoraggio da tenere in considerazione durante la transizione da OpenShift Container Platform 3.11 a OpenShift Container Platform 4.10.

Avviso per il monitoraggio della disponibilità dell'infrastruttura

L'avviso predefinito che si attiva per garantire la disponibilità della struttura di monitoraggio era chiamato **DeadMansSwitch** in OpenShift Container Platform 3.11. L'avviso è stato rinominato **Watchdog** in OpenShift Container Platform 4. Se in OpenShift Container Platform 3.11 era impostata l'integrazione di PagerDuty con questo avviso, è necessario impostare l'integrazione di PagerDuty per l'avviso **Watchdog** in OpenShift Container Platform 4.

Per maggiori informazioni, consulta la sezione [Applicare la configurazione personalizzata di Alertmanager](#).

CAPITOLO 4. CONSIDERAZIONI SULLA RETE

Esamina le strategie per reindirizzare il traffico di rete delle applicazioni dopo la migrazione.

4.1. CONSIDERAZIONI SUL DNS

Il dominio DNS del cluster di destinazione è diverso da quello del cluster di origine. Per impostazione predefinita, le applicazioni ottengono i FQDN del cluster di destinazione dopo la migrazione.

Per preservare il dominio DNS di origine delle applicazioni migrate, seleziona una delle due opzioni descritte di seguito.

4.1.1. Isolare il dominio DNS del cluster di destinazione dai client

È possibile consentire alle richieste dei client inviate al dominio DNS del cluster di origine di raggiungere il dominio DNS del cluster di destinazione senza esporre il cluster di destinazione ai client.

Procedura

1. Posizionare un componente di rete esterno, come uno strumento di bilanciamento del carico o un proxy inverso, tra i client e il cluster di destinazione.
2. Aggiornare il FQDN dell'applicazione sul cluster di origine nel server DNS per restituire l'indirizzo IP del componente di rete esterno.
3. Configurare il componente di rete per inviare le richieste ricevute per l'applicazione nel dominio di origine allo strumento di bilanciamento del carico nel dominio del cluster di destinazione.
4. Creare un record DNS jolly per il dominio ***.apps.source.example.com** che punti all'indirizzo IP dello strumento di bilanciamento del carico del cluster di origine.
5. Creare un record DNS per ogni applicazione che punti all'indirizzo IP del componente di rete esterno di fronte al cluster di destinazione. Un record DNS specifico ha una priorità più alta di un record jolly, quindi non sorge alcun conflitto quando l'FQDN dell'applicazione viene risolto.



NOTA

- Il componente di rete esterno deve terminare tutte le connessioni TLS sicure. Se le connessioni passano attraverso lo strumento di bilanciamento del carico del cluster di destinazione, l'FQDN dell'applicazione di destinazione è esposto al client e si verificano errori di certificato.
- Le applicazioni non devono restituire ai client i link che fanno riferimento al dominio del cluster di destinazione. Altrimenti, alcune parti dell'applicazione potrebbero non caricarsi o funzionare correttamente.

4.1.2. Impostare il cluster di destinazione per accettare il dominio DNS di origine

È possibile impostare il cluster di destinazione per accettare richieste per un'applicazione migrata nel dominio DNS del cluster di origine.

Procedura

Sia per l'accesso HTTP non sicuro che per l'accesso HTTPS sicuro, eseguire i seguenti passaggi:

1. Creare un percorso nel progetto del cluster di destinazione che sia configurato per accettare richieste indirizzate all'FQDN dell'applicazione nel cluster di origine:

```
$ oc expose svc <app1-svc> --hostname <app1.apps.source.example.com> \  
-n <app1-namespace>
```

Con questo nuovo percorso in funzione, il server accetta qualsiasi richiesta per quel FQDN e la invia ai pod dell'applicazione corrispondenti. Inoltre, quando si esegue la migrazione dell'applicazione, viene creato un altro percorso nel dominio del cluster di destinazione. Le richieste raggiungono l'applicazione migrata usando uno di questi nomi host.

2. Creare un record DNS con il provider DNS che punti l'FQDN dell'applicazione nel cluster di origine all'indirizzo IP dello strumento di bilanciamento del carico predefinito del cluster di destinazione. In questo modo si reindirizzerà il traffico dal cluster di origine al cluster di destinazione.

L'FQDN dell'applicazione si risolve nello strumento di bilanciamento del carico del cluster di destinazione. Il router del controller di ingresso predefinito accetta le richieste per quel FQDN perché è esposto un percorso per quel nome host.

Per un accesso HTTPS sicuro, eseguire il seguente passaggio aggiuntivo:

1. Sostituire il certificato x509 del controller di ingresso predefinito creato durante il processo di installazione con un certificato personalizzato.
2. Configurare questo certificato in modo da includere i domini DNS jolly per entrambi i cluster di origine e di destinazione nel campo **subjectAltName**.
Il nuovo certificato è valido per proteggere le connessioni effettuate utilizzando entrambi i domini DNS.

Risorse aggiuntive

- Consulta la sezione [Sostituzione del certificato di ingresso predefinito](#) per maggiori informazioni.

4.2. STRATEGIE DI REINDIRIZZAMENTO DEL TRAFFICO DI RETE

Dopo una migrazione riuscita, è necessario reindirizzare il traffico di rete delle applicazioni stateless dal cluster di origine al cluster di destinazione.

Le strategie di reindirizzamento del traffico di rete si basano sui seguenti presupposti:

- I pod dell'applicazione sono in esecuzione su entrambi i cluster di origine e di destinazione.
- Ogni applicazione ha un percorso che contiene il nome host del cluster di origine.
- Il percorso con il nome host del cluster di origine contiene un certificato CA.
- Per HTTPS, il certificato CA del router di destinazione contiene un nome alternativo del soggetto per il record DNS jolly del cluster di origine.

Considera le seguenti strategie e scegli quella che soddisfa al meglio gli obiettivi.

- Reindirizzamento di tutto il traffico di rete per tutte le applicazioni allo stesso tempo
Cambia il record DNS jolly del cluster di origine per puntare all'indirizzo IP virtuale (VIP) del router del cluster di destinazione.

Questa strategia è adatta per applicazioni semplici o piccole migrazioni.

- Reindirizzamento del traffico di rete per applicazioni individuali
Crea un record DNS per ogni applicazione con il nome host del cluster di origine che punta al VIP del router del cluster di destinazione. Questo record DNS ha la precedenza sul record DNS jolly del cluster di origine.
- Reindirizzamento graduale del traffico di rete per le singole applicazioni
 1. Crea un proxy che possa dirigere il traffico sia al VIP del router del cluster di origine che al VIP del router del cluster di destinazione, per ogni applicazione.
 2. Crea un record DNS per ogni applicazione con il nome host del cluster di origine che punta al proxy.
 3. Configura la voce proxy per l'applicazione per instradare una percentuale del traffico al VIP del router del cluster di destinazione e il resto del traffico al VIP del router del cluster di origine.
 4. Aumenta gradualmente la percentuale di traffico che instrada verso il VIP del router del cluster di destinazione fino a quando tutto il traffico di rete viene reindirizzato.
- Reindirizzamento del traffico in base all'utente per applicazioni individuali
Usando questa strategia, è possibile filtrare le intestazioni TCP/IP delle richieste degli utenti per reindirizzare il traffico di rete per gruppi predefiniti di utenti. Questo permette di testare il processo di reindirizzamento su segmenti specifici di utenti prima di reindirizzare l'intero traffico di rete.
 1. Crea un proxy che possa dirigere il traffico sia al VIP del router del cluster di origine che al VIP del router del cluster di destinazione, per ogni applicazione.
 2. Crea un record DNS per ogni applicazione con il nome host del cluster di origine che punta al proxy.
 3. Configura la voce proxy per l'applicazione per instradare il traffico che corrisponde a un dato schema di intestazione, come i **clienti di prova**, al VIP del router del cluster di destinazione e il resto del traffico al VIP del router del cluster di origine.
 4. Reindirizza il traffico al VIP del router del cluster di destinazione per gradi fino a quando tutto il traffico è sul VIP di questo router.

CAPITOLO 5. INFORMAZIONI SUL MIGRATION TOOLKIT FOR CONTAINERS

Il Migration Toolkit for Containers (MTC) permette di eseguire la migrazione dei carichi di lavoro delle applicazioni stateful da OpenShift Container Platform 3 a 4.10 al livello di granularità di uno spazio dei nomi.



IMPORTANTE

Prima di iniziare la migrazione, assicurati di rivedere le [differenze tra OpenShift Container Platform 3 e 4](#).

MTC fornisce una console web e un'API, basata sulle risorse personalizzate di Kubernetes, per consentire di controllare la migrazione e ridurre al minimo l'indisponibilità delle applicazioni.

La console MTC è installata sul cluster di destinazione per impostazione predefinita. È possibile configurare il Migration Toolkit for Containers Operator per installare la console su un [cluster di origine di OpenShift Container Platform 3](#) o su un [cluster remoto](#).

MTC supporta i metodi di copia del file system e dei dati degli snapshot per la migrazione dei dati dal cluster di origine al cluster di destinazione. È possibile selezionare un metodo adatto al proprio ambiente e supportato dal provider di storage.

Il catalogo dei servizi è deprecato in OpenShift Container Platform 4. È possibile eseguire la migrazione delle risorse dei carichi di lavoro fornite con il catalogo dei servizi da OpenShift Container Platform 3 a 4, ma non è possibile eseguire azioni del catalogo dei servizi come il **provisioning**, il **deprovisioning** o l'**aggiornamento** su questi carichi di lavoro dopo la migrazione. La console MTC mostra un messaggio se non è possibile eseguire la migrazione delle risorse del catalogo dei servizi.

5.1. TERMINOLOGIA

Tabella 5.1. Terminologia MTC

Termine	Definizione
Cluster di origine	Cluster da cui viene eseguita la migrazione delle applicazioni.
Cluster di destinazione ^[1]	Cluster verso cui viene eseguita la migrazione delle applicazioni.
Repository di replica	Storage a oggetti utilizzato per copiare immagini, volumi e oggetti Kubernetes durante la migrazione indiretta o per gli oggetti Kubernetes durante la migrazione diretta di volumi o la migrazione diretta di immagini. Il repository di replica deve essere accessibile a tutti i cluster.
Cluster host	Cluster su cui sono in esecuzione il pod migration-controller e la console web. Il cluster host è di solito il cluster di destinazione ma non deve esserlo necessariamente. Il cluster host non richiede un percorso di registro esposto per la migrazione diretta delle immagini.

Termine	Definizione
Cluster remoto	<p>Un cluster remoto è di solito il cluster di origine ma non deve esserlo necessariamente.</p> <p>Un cluster remoto richiede una risorsa personalizzata Secret che contiene il token dell'account di servizio del migration-controller.</p> <p>Un cluster remoto richiede un percorso di registro sicuro esposto per la migrazione diretta delle immagini.</p>
Migrazione indiretta	Immagini, volumi e oggetti Kubernetes vengono copiati dal cluster di origine al repository di replica e poi dal repository di replica al cluster di destinazione.
Migrazione diretta di volumi	I volumi permanenti sono copiati direttamente dal cluster di origine al cluster di destinazione.
Migrazione diretta di immagini	Le immagini sono copiate direttamente dal cluster di origine al cluster di destinazione.
Migrazione di fase	<p>I dati vengono copiati nel cluster di destinazione senza fermare l'applicazione.</p> <p>L'esecuzione di una migrazione di fase più volte riduce la durata della migrazione completa.</p>
Migrazione completa	L'applicazione viene fermata sul cluster di origine e viene eseguita la migrazione delle sue risorse sul cluster di destinazione.
Migrazione dello stato	Viene eseguita la migrazione dello stato dell'applicazione copiando specifiche richieste di volumi permanenti e oggetti Kubernetes nel cluster di destinazione.
Migrazione rollback	Con una migrazione rollback viene eseguito il rollback di una migrazione completata.

¹ Chiamate il cluster di *destinazione* nella console web di MTC.

5.2. WORKFLOW DI MTC

È possibile eseguire la migrazione delle risorse di Kubernetes, i dati dei volumi permanenti e le immagini container interne a OpenShift Container Platform 4.10 utilizzando la console web di Migration Toolkit for Containers (MTC) o l'API di Kubernetes.

MTC esegue la migrazione delle seguenti risorse:

- Uno spazio dei nomi specificato in un piano di migrazione.
- Risorse con spazio dei nomi: quando MTC esegue la migrazione di uno spazio dei nomi, vengono migrati tutti gli oggetti e le risorse associate a quello spazio dei nomi, come i servizi o i pod. Inoltre, se una risorsa che esiste nello spazio dei nomi ma non a livello di cluster dipende da una risorsa che esiste a livello di cluster, MTC esegue la migrazione di entrambe le risorse.

Per esempio, un security context constraint (SCC) è una risorsa che esiste a livello di cluster e un service account (SA) è una risorsa che esiste a livello di spazio dei nomi. Se un SA esiste in uno spazio dei nomi di cui MTC esegue la migrazione, MTC individua automaticamente tutti gli SCC che sono collegati al SA ed esegue la migrazione anche di questi. Allo stesso modo, MTC esegue la migrazione delle richieste di volumi permanenti che sono legate ai volumi permanenti dello spazio dei nomi.



NOTA

Potrebbe essere necessario eseguire la migrazione delle risorse con cluster manualmente, a seconda della risorsa.

- Risorse personalizzate (CR) e definizioni di risorse personalizzate (CRD): MTC esegue automaticamente la migrazione delle CR e delle CRD a livello di spazio dei nomi.

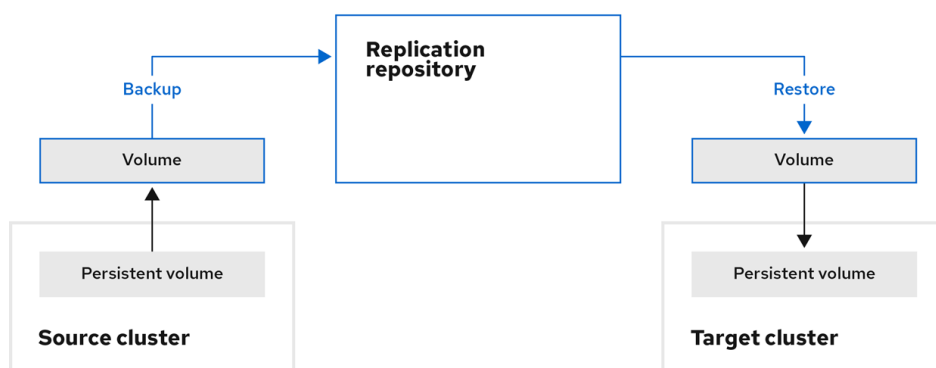
La migrazione di un'applicazione con la console web di MTC comporta i seguenti passaggi:

1. Installare il Migration Toolkit for Containers Operator su tutti i cluster.
È possibile installare il Migration Toolkit for Containers Operator in un ambiente ristretto con accesso Internet limitato o assente. I cluster di origine e di destinazione devono avere accesso di rete tra loro e a un registro speculare.
2. Configurare il repository di replica, uno storage a oggetti intermedio che MTC usa per eseguire la migrazione dei dati.
I cluster di origine e di destinazione devono avere accesso di rete al repository di replica durante la migrazione. Se si utilizza un server proxy, è necessario configurarlo per permettere il traffico di rete tra il repository di replica e i cluster.
3. Aggiungere il cluster di origine alla console web di MTC.
4. Aggiungere il repository di replica alla console web di MTC.
5. Creare un piano di migrazione, con una delle seguenti opzioni di migrazione dei dati:
 - **Copia:** MTC copia i dati dal cluster di origine al repository di replica e dal repository di replica al cluster di destinazione.



NOTA

Se si utilizza la migrazione diretta delle immagini o la migrazione diretta dei volumi, le immagini o i volumi sono copiati direttamente dal cluster di origine al cluster di destinazione.



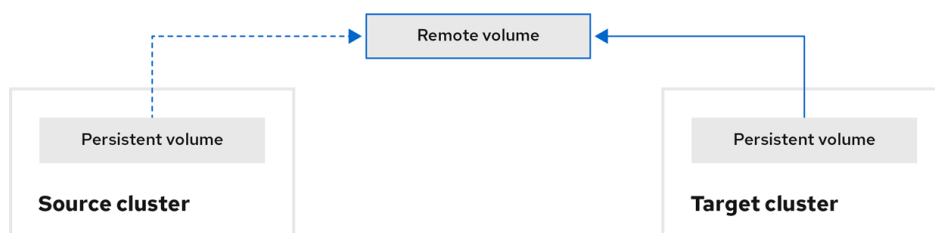
OpenShift_45_1019

- **Spostamento:** MTC smonta un volume remoto, per esempio NFS, dal cluster di origine, crea una risorsa PV sul cluster di destinazione che punta al volume remoto, e poi monta il volume remoto sul cluster di destinazione. Le applicazioni in esecuzione sul cluster di destinazione utilizzano lo stesso volume remoto utilizzato dal cluster di origine. Il volume remoto deve essere accessibile ai cluster di origine e di destinazione.



NOTA

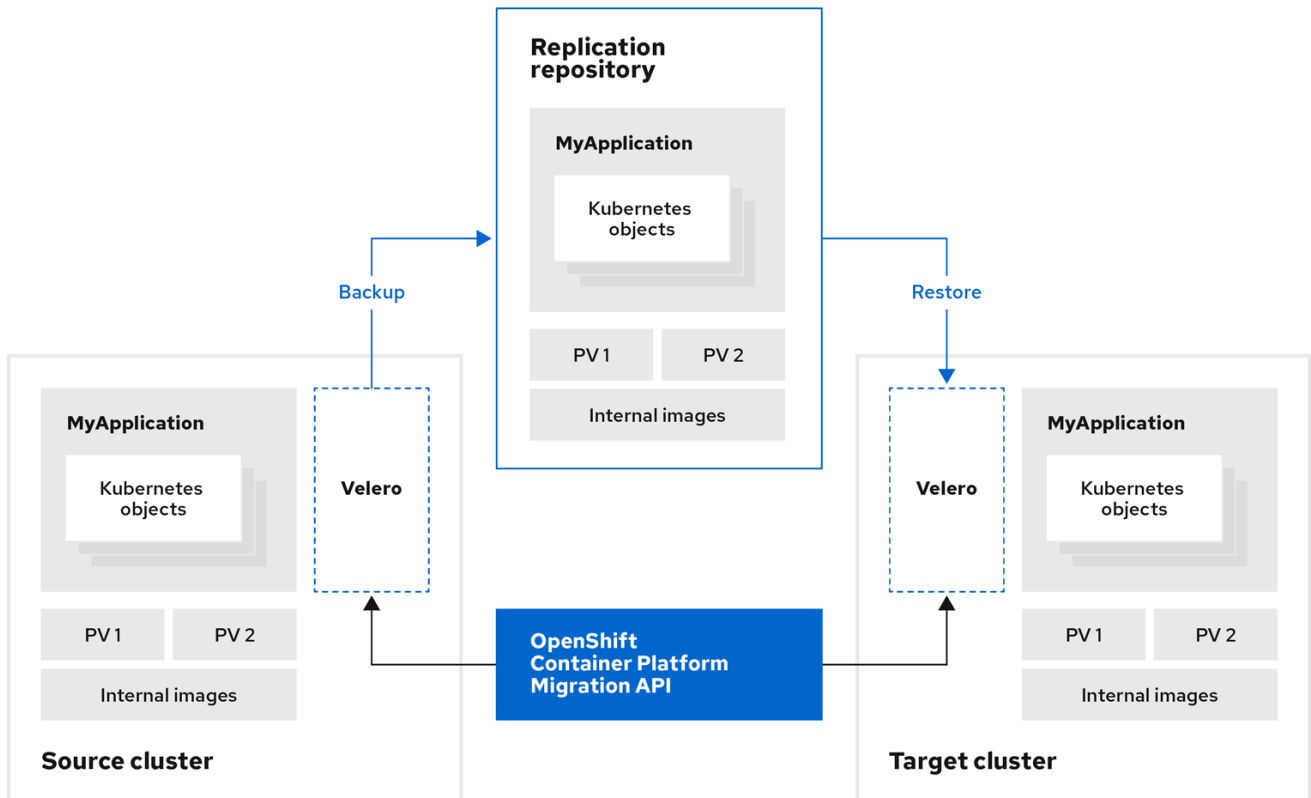
Anche se il repository di replica non appare in questo diagramma, è necessario per la migrazione.



OpenShift_45_1019

6. Eseguire il piano di migrazione, con una delle seguenti opzioni:

- Con la migrazione a **fasi** viene eseguita la migrazione dei dati nel cluster di destinazione senza fermare l'applicazione.
Una migrazione di fase può essere eseguita più volte in modo che la maggior parte dei dati sia copiata nella destinazione prima della migrazione. L'esecuzione di una o più migrazioni di fase riduce la durata della migrazione completa.
- La migrazione **completa** ferma l'applicazione sul cluster di origine e sposta le risorse sul cluster di destinazione.
Facoltativo: è possibile deselezionare la casella di controllo **Halt transactions (Arresta transazioni)** sul cluster di origine durante la migrazione.



OpenShift_45_1019

5.3. INFORMAZIONI SUI METODI DI COPIA DEI DATI

Il Migration Toolkit for Containers (MTC) supporta i metodi di copia del file system e dei dati degli snapshot per la migrazione dei dati dal cluster di origine al cluster di destinazione. È possibile selezionare un metodo adatto al proprio ambiente e supportato dal provider di storage.

5.3.1. Metodo di copia del file system

MTC copia i file di dati dal cluster di origine al repository di replica, e da qui al cluster di destinazione.

Il metodo di copia del file system usa Restic per la migrazione indiretta o Rsync per la migrazione diretta dei volumi.

Tabella 5.2. Riepilogo del metodo di copia del file system

Vantaggi	Limitazioni
<ul style="list-style-type: none"> ● I cluster possono avere diverse classi di storage. ● Supportato per tutti i provider di storage S3. ● Verifica opzionale dei dati con checksum. ● Supporta la migrazione diretta dei volumi, che aumenta significativamente le prestazioni. 	<ul style="list-style-type: none"> ● Più lento del metodo della copia degli snapshot. ● La verifica opzionale dei dati riduce significativamente le prestazioni.

5.3.2. Metodo di copia degli snapshot

MTC copia uno snapshot dei dati del cluster di origine nel repository di replica di un provider cloud. I dati vengono ripristinati sul cluster di destinazione.

Il metodo di copia degli snapshot può essere usato con Amazon Web Services, Google Cloud Provider e Microsoft Azure.

Tabella 5.3. Riepilogo del metodo di copia degli snapshot

Vantaggi	Limitazioni
<ul style="list-style-type: none"> ● Più veloce del metodo di copia del file system. 	<ul style="list-style-type: none"> ● Il provider cloud deve supportare gli snapshot. ● I cluster devono essere sullo stesso provider cloud. ● I cluster devono essere nella stessa località o paese. ● I cluster devono avere la stessa classe di storage. ● La classe di storage deve essere compatibile con gli snapshot. ● Non supporta la migrazione diretta dei volumi.

5.4. MIGRAZIONE DIRETTA DEI VOLUMI E MIGRAZIONE DIRETTA DELLE IMMAGINI

È possibile utilizzare la migrazione diretta delle immagini (DIM) e la migrazione diretta dei volumi (DVM) per eseguire la migrazione di immagini e dati direttamente dal cluster di origine al cluster di destinazione.

Se si esegue DVM con nodi che si trovano in diverse aree di disponibilità, la migrazione potrebbe fallire perché i pod migrati non possono accedere alla richiesta di volumi permanenti.

DIM e DVM hanno vantaggi significativi in termini di prestazioni perché i passaggi intermedi del backup dei file dal cluster di origine al repository di replica e il ripristino dei file dal repository di replica al cluster di destinazione vengono saltati. I dati sono trasferiti con [Rsync](#).

DIM e DVM hanno ulteriori requisiti.

CAPITOLO 6. INSTALLARE IL MIGRATION TOOLKIT FOR CONTAINERS

È possibile installare il Migration Toolkit for Containers (MTC) su OpenShift Container Platform 3 e 4.

Dopo aver installato il Migration Toolkit for Containers Operator su OpenShift Container Platform 4.10 usando Operator Lifecycle Manager, si installa manualmente il Migration Toolkit for Containers Operator esistente su OpenShift Container Platform 3.

Per impostazione predefinita, la console web di MTC e il pod **Migration Controller** vengono eseguiti sul cluster di destinazione. È possibile configurare il manifest delle risorse personalizzate di **Migration Controller** per eseguire la console web di MTC e il pod **Migration Controller** su un [cluster di origine o su un cluster remoto](#).

Dopo aver installato MTC, è necessario configurare uno storage a oggetti da utilizzare come repository di replica.

Per disinstallare MTC, consulta la sezione [Disinstallazione di MTC ed eliminazione delle risorse](#).

6.1. LINEE GUIDA PER LA COMPATIBILITÀ

È necessario installare il Migration Toolkit for Containers (MTC) Operator compatibile con la versione di OpenShift Container Platform.

Non è possibile installare MTC 1.6 su OpenShift Container Platform 4.5, o versioni precedenti, perché le versioni API delle definizioni delle risorse personalizzate sono incompatibili.

È possibile eseguire la migrazione dei carichi di lavoro da un cluster di origine con MTC 1.5.3 a un cluster di destinazione con MTC 1.6 purché la risorsa personalizzata **MigrationController** e la console web di MTC siano in esecuzione sul cluster di destinazione.

Tabella 6.1. OpenShift Container Platform e compatibilità MTC

Versione OpenShift Container Platform	Versione MTC	Migration Toolkit for Containers Operator
4.5 e precedenti	1.5.3	Migration Toolkit for Containers Operator esistente, installato manualmente con il file operator.yml .
4.6 e successivi	Ultima versione 1.6.x z-stream	Migration Toolkit for Containers Operator, installato con Operator Lifecycle Manager.

6.2. INSTALLARE IL MIGRATION TOOLKIT FOR CONTAINERS OPERATOR ESISTENTE SU OPENSIFT CONTAINER PLATFORM 3

È possibile installare manualmente il Migration Toolkit for Containers Operator esistente su OpenShift Container Platform 3.

Requisiti

- È necessario aver eseguito l'accesso come utente con privilegi di **cluster-admin** su tutti i cluster.

- È necessario disporre di accesso a **registry.redhat.io**.
- È necessario avere installato **podman**.
- È necessario creare un [segreto del flusso di immagini](#) e copiarlo su ogni nodo del cluster.

Procedura

1. Accedere a **registry.redhat.io** con le proprie credenziali del Red Hat Customer Portal:

```
$ sudo podman login registry.redhat.io
```

2. Scaricare il file **operator.yml**:

```
$ sudo podman cp $(sudo podman create \
registry.redhat.io/rhmtc/openshift-migration-legacy-rhel8-operator:v1.5.3):/operator.yml ./
```

3. Scaricare il file **controller.yml**:

```
$ sudo podman cp $(sudo podman create \
registry.redhat.io/rhmtc/openshift-migration-legacy-rhel8-operator:v1.5.3):/controller.yml ./
```

4. Accedere al cluster di OpenShift Container Platform 3.

5. Verificare che il cluster possa autenticarsi con **registry.redhat.io**:

```
$ oc run test --image registry.redhat.io/ubi8 --command sleep infinity
```

6. Creare l'oggetto Migration Toolkit for Containers Operator:

```
$ oc create -f operator.yml
```

Esempio di output

```
namespace/openshift-migration created
rolebinding.rbac.authorization.k8s.io/system:deployers created
serviceaccount/migration-operator created
customresourcedefinition.apiextensions.k8s.io/migrationcontrollers.migration.openshift.io
created
role.rbac.authorization.k8s.io/migration-operator created
rolebinding.rbac.authorization.k8s.io/migration-operator created
clusterrolebinding.rbac.authorization.k8s.io/migration-operator created
deployment.apps/migration-operator created
Error from server (AlreadyExists): error when creating "./operator.yml":
rolebindings.rbac.authorization.k8s.io "system:image-builders" already exists 1
Error from server (AlreadyExists): error when creating "./operator.yml":
rolebindings.rbac.authorization.k8s.io "system:image-pullers" already exists
```

- 1** Puoi ignorare i messaggi **Error from server (AlreadyExists)**. Sono causati dal Migration Toolkit for Containers Operator che crea risorse per le versioni precedenti di OpenShift Container Platform 3 fornite nelle versioni successive.

7. Creare l'oggetto **MigrationController**:

```
$ oc create -f controller.yml
```

8. Verificare che i pod MTC siano in esecuzione:

```
$ oc get pods -n openshift-migration
```

6.3. INSTALLARE IL MIGRATION TOOLKIT FOR CONTAINERS OPERATOR SU OPENSIFT CONTAINER PLATFORM 4.10

Si installa il Migration Toolkit for Containers Operator su OpenShift Container Platform 4.10 utilizzando Operator Lifecycle Manager.

Requisiti

- È necessario aver eseguito l'accesso come utente con privilegi di **cluster-admin** su tutti i cluster.

Procedura

1. Nella console web di OpenShift Container Platform, cliccare su **Operators** → **OperatorHub**.
2. Usare il campo **Filter by keyword** per trovare il **Migration Toolkit for Containers Operator**.
3. Selezionare il **Migration Toolkit for Containers Operator** e cliccare su **Install**.
4. Cliccare su **Install**.
Nella pagina degli **operatori installati**, il **Migration Toolkit for Containers Operator** appare nel progetto **openshift-migration** con lo stato **Succeeded**.
5. Cliccare su **Migration Toolkit for Containers Operator**.
6. In **Provided APIs**, individuare il file **Migration Controller** e cliccare su **Create Instance**.
7. Cliccare su **Create**.
8. Cliccare su **Workloads** → **Pods** per verificare che i pod MTC siano in esecuzione.

6.4. CONFIGURARE I PROXY

Per OpenShift Container Platform 4.1 e versioni precedenti, è necessario configurare i proxy nel manifest della risorsa personalizzata **MigrationController** dopo aver installato il Migration Toolkit for Containers Operator perché queste versioni non supportano un oggetto **proxy** a livello di cluster.

Per OpenShift Container Platform da 4.2 a 4.10, il Migration Toolkit for Containers (MTC) eredita le impostazioni proxy a livello di cluster. Puoi cambiare i parametri del proxy se desideri sovrascrivere le impostazioni del proxy a livello di cluster.

È necessario configurare i proxy per consentire il protocollo SPDY e per inoltrare l'intestazione **Upgrade HTTP** al server API. Altrimenti, viene visualizzato l'errore **Upgrade request required**. La risorsa personalizzata **MigrationController** usa SPDY per eseguire comandi all'interno di pod remoti. L'intestazione **Upgrade HTTP** è necessaria per aprire una connessione websocket con il server API.

Migrazione diretta di volumi

Se si esegue una migrazione diretta di volumi (DVM) da un cluster di origine dietro un proxy, è necessario configurare un proxy Stunnel. Stunnel crea un tunnel trasparente tra i cluster di origine e destinazione per la connessione TCP senza cambiare i certificati.

DVM supporta solo un proxy. Il cluster di origine non può accedere al percorso del cluster di destinazione se anche il cluster di destinazione è dietro un proxy.

Requisiti

- È necessario aver eseguito l'accesso come utente con privilegi di **cluster-admin** su tutti i cluster.

Procedura

1. Ottenere il manifest della risorsa personalizzata **MigrationController**:

```
$ oc get migrationcontroller <migration_controller> -n openshift-migration
```

2. Aggiornare i parametri del proxy:

```
apiVersion: migration.openshift.io/v1 alpha1
kind: MigrationController
metadata:
  name: <migration_controller>
  namespace: openshift-migration
...
spec:
  stunnel_tcp_proxy: http://<username>:<password>@<ip>:<port> 1
  httpProxy: http://<username>:<password>@<ip>:<port> 2
  httpsProxy: https://<username>:<password>@<ip>:<port> 3
  noProxy: example.com 4
```

- 1 URL proxy Stunnel per la migrazione diretta dei volumi.
- 2 URL proxy per creare connessioni HTTP all'esterno del cluster. Lo schema dell'URL deve essere **http**.
- 3 URL proxy per creare connessioni HTTPS all'esterno del cluster. Se questo non è specificato, allora viene utilizzato **httpProxy** per entrambe le connessioni HTTP e HTTPS.
- 4 Elenco separato da virgole di nomi di dominio di destinazione, domini, indirizzi IP o altri CIDR di rete per escludere il proxy.

Precedere un dominio con **.** per far corrispondere solo i sottodomini. Ad esempio, **.y.com** corrisponde a **x.y.com**, ma non a **y.com**. Usare ***** per bypassare il proxy per tutte le destinazioni. Se si aumenta il numero di lavoratori che non sono inclusi nella rete definita dal campo **networking.machineNetwork[].cidr** della configurazione di installazione, è necessario aggiungerli a questa lista per prevenire problemi di connessione.

Questo campo viene ignorato se non è impostato né il campo **httpProxy** né il campo **httpsProxy**.

3. Salvare il manifest come **migration-controller.yaml**.

4. Applicare il manifest aggiornato:

```
$ oc replace -f migration-controller.yaml -n openshift-migration
```

Per maggiori informazioni, consulta la sezione [Configurare il proxy a livello di cluster](#).

6.5. CONFIGURARE UN REPOSITORY DI REPLICA

È necessario configurare uno storage a oggetti da utilizzare come repository di replica. Il Migration Toolkit for Containers (MTC) copia i dati dal cluster di origine al repository di replica, e poi dal repository di replica al cluster di destinazione.

MTC supporta i [metodi di copia del file system e dei dati degli snapshot](#) per la migrazione dei dati dal cluster di origine al cluster di destinazione. È possibile selezionare un metodo adatto al proprio ambiente e supportato dal provider di storage.

Sono supportati i seguenti provider di storage:

- [Multicloud Object Gateway](#)
- [Amazon Web Services S3](#)
- [Google Cloud Platform](#)
- [Microsoft Azure Blob](#)
- Storage a oggetti S3 generico, ad esempio, Minio o Ceph S3

6.5.1. Requisiti

- Tutti i cluster devono avere un accesso di rete ininterrotto al repository di replica.
- Se si utilizza un server proxy con un repository di replica ospitato internamente, è necessario assicurarsi che il proxy permetta l'accesso al repository di replica.

6.5.2. Recuperare le credenziali di Multicloud Object Gateway

È necessario recuperare le credenziali Multicloud Object Gateway (MCG) e l'endpoint S3 per configurare MCG come repository di replica per il Migration Toolkit for Containers (MTC). È necessario recuperare le credenziali Multicloud Object Gateway (MCG) per creare una risorsa personalizzata (CR) **Secret** per OpenShift API for Data Protection (OADP).

MCG è un componente di OpenShift Container Storage.

Requisiti

- È necessario distribuire OpenShift Container Storage utilizzando la [guida per il deployment di OpenShift Container Storage](#) appropriata.

Procedura

1. Ottenere l'endpoint S3, **AWS_ACCESS_KEY_ID** e **AWS_SECRET_ACCESS_KEY** eseguendo il comando [describe](#) sulla risorsa personalizzata **NooBaa**.

Usare queste credenziali per aggiungere MCG come repository di replica.

6.5.3. Configurazione di Amazon Web Services

Si configura lo storage a oggetti Amazon Web Services (AWS) S3 come repository di replica per il Migration Toolkit for Containers (MTC).

Requisiti

- È necessario che sia installata l'[interfaccia della riga di comando di AWS](#).
- Il bucket di storage AWS S3 deve essere accessibile ai cluster di origine e di destinazione.
- Se si utilizza il metodo di copia degli snapshot:
 - È necessario avere accesso a EC2 Elastic Block Storage (EBS).
 - I cluster di origine e di destinazione devono trovarsi nella stessa area geografica.
 - I cluster di origine e di destinazione devono avere la stessa classe di storage.
 - La classe di storage deve essere compatibile con gli snapshot.

Procedura

1. Impostare la variabile **BUCKET**:

```
$ BUCKET=<your_bucket>
```

2. Impostare la variabile **REGION**:

```
$ REGION=<your_region>
```

3. Creare un bucket AWS S3:

```
$ aws s3api create-bucket \
  --bucket $BUCKET \
  --region $REGION \
  --create-bucket-configuration LocationConstraint=$REGION 1
```

- 1** **us-east-1** non supporta un **LocationConstraint**. Se l'area geografica è **us-east-1**, omettere **--create-bucket-configuration LocationConstraint=\$REGION**.

4. Creare un utente IAM:

```
$ aws iam create-user --user-name velero 1
```

- 1** Se si desidera utilizzare Velero per eseguire il backup di più cluster con più bucket S3, creare un nome utente unico per ogni cluster.

5. Creare un file **velero-policy.json**:

```

$ cat > velero-policy.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVolumes",
        "ec2:DescribeSnapshots",
        "ec2:CreateTags",
        "ec2:CreateVolume",
        "ec2:CreateSnapshot",
        "ec2>DeleteSnapshot"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3>DeleteObject",
        "s3:PutObject",
        "s3:AbortMultipartUpload",
        "s3:ListMultipartUploadParts"
      ],
      "Resource": [
        "arn:aws:s3:::${BUCKET}/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::${BUCKET}"
      ]
    }
  ]
}
EOF

```

6. Allegare i criteri per dare all'utente **velero** le autorizzazioni necessarie:

```

$ aws iam put-user-policy \
  --user-name velero \
  --policy-name velero \
  --policy-document file://velero-policy.json

```

7. Creare una chiave di accesso per l'utente **velero**:

```

$ aws iam create-access-key --user-name velero

```

Esempio di output

■

```
{
  "AccessKey": {
    "UserName": "velero",
    "Status": "Active",
    "CreateDate": "2017-07-31T22:24:41.576Z",
    "SecretAccessKey": <AWS_SECRET_ACCESS_KEY>,
    "AccessKeyId": <AWS_ACCESS_KEY_ID>
  }
}
```

Registrare **AWS_SECRET_ACCESS_KEY** e **AWS_ACCESS_KEY_ID**. Si utilizzano le credenziali per aggiungere AWS come repository di replica.

6.5.4. Configurazione di Google Cloud Platform

Si configura un bucket di storage di Google Cloud Platform (GCP) come repository di replica per il Migration Toolkit for Containers (MTC).

Requisiti

- È necessario che gli strumenti dell'Interfaccia della riga di comando **gcloud** e **gsutil** siano installati. Consulta la [documentazione di Google Cloud](#) per maggiori dettagli.
- Il bucket di storage di GCP deve essere accessibile ai cluster di origine e di destinazione.
- Se si utilizza il metodo di copia degli snapshot:
 - I cluster di origine e di destinazione devono trovarsi nella stessa area geografica.
 - I cluster di origine e di destinazione devono avere la stessa classe di storage.
 - La classe di storage deve essere compatibile con gli snapshot.

Procedura

1. Accedere a GCP:

```
$ gcloud auth login
```

2. Impostare la variabile **BUCKET**:

```
$ BUCKET=<bucket> 1
```

- 1** Specificare il nome del bucket.

3. Creare il bucket di storage:

```
$ gsutil mb gs://$BUCKET/
```

4. Impostare la variabile **PROJECT_ID** nel progetto attivo:

```
$ PROJECT_ID=$(gcloud config get-value project)
```

5. Creare un account di servizio:

```
$ gcloud iam service-accounts create velero \
  --display-name "Velero service account"
```

6. Elencare gli account di servizio:

```
$ gcloud iam service-accounts list
```

7. Impostare la variabile **SERVICE_ACCOUNT_EMAIL** in modo che corrisponda al suo valore di **email**:

```
$ SERVICE_ACCOUNT_EMAIL=$(gcloud iam service-accounts list \
  --filter="displayName:Velero service account" \
  --format 'value(email)')
```

8. Allegare i criteri per dare all'utente **velero** le autorizzazioni necessarie:

```
$ ROLE_PERMISSIONS=(
  compute.disks.get
  compute.disks.create
  compute.disks.createSnapshot
  compute.snapshots.get
  compute.snapshots.create
  compute.snapshots.useReadOnly
  compute.snapshots.delete
  compute.zones.get
)
```

9. Creare il ruolo personalizzato **velero.server**:

```
$ gcloud iam roles create velero.server \
  --project $PROJECT_ID \
  --title "Velero Server" \
  --permissions "$(IFS=","; echo "${ROLE_PERMISSIONS[*]}")"
```

10. Aggiungere il binding del criterio IAM al progetto:

```
$ gcloud projects add-iam-policy-binding $PROJECT_ID \
  --member serviceAccount:$SERVICE_ACCOUNT_EMAIL \
  --role projects/$PROJECT_ID/roles/velero.server
```

11. Aggiornare l'account di servizio IAM:

```
$ gsutil iam ch serviceAccount:$SERVICE_ACCOUNT_EMAIL:objectAdmin gs://{BUCKET}
```

12. Salvare le chiavi dell'account di servizio IAM nel file **credentials-velero** nella directory corrente:

```
$ gcloud iam service-accounts keys create credentials-velero \
  --iam-account $SERVICE_ACCOUNT_EMAIL
```

Si utilizza il file **credentials-velero** per aggiungere GCP come repository di replica.

6.5.5. Configurazione di Microsoft Azure

Si configura un container di storage Microsoft Azure Blob come repository di replica per il Migration Toolkit for Containers (MTC).

Requisiti

- È necessario avere installato l'[interfaccia della riga di comando di Azure](#).
- Il container di storage Azure Blob deve essere accessibile ai cluster di origine e di destinazione.
- Se si utilizza il metodo di copia degli snapshot:
 - I cluster di origine e di destinazione devono trovarsi nella stessa area geografica.
 - I cluster di origine e di destinazione devono avere la stessa classe di storage.
 - La classe di storage deve essere compatibile con gli snapshot.

Procedura

1. Accedere ad Azure:

```
$ az login
```

2. Impostare la variabile **AZURE_RESOURCE_GROUP**:

```
$ AZURE_RESOURCE_GROUP=Velero_Backups
```

3. Creare un gruppo di risorse Azure:

```
$ az group create -n $AZURE_RESOURCE_GROUP --location CentralUS 1
```

- 1** Specificare la propria posizione.

4. Impostare la variabile **AZURE_STORAGE_ACCOUNT_ID**:

```
$ AZURE_STORAGE_ACCOUNT_ID="velero$(uuidgen | cut -d '-' -f5 | tr 'A-Z' '[a-z])"
```

5. Creare un account di storage Azure:

```
$ az storage account create \  
  --name $AZURE_STORAGE_ACCOUNT_ID \  
  --resource-group $AZURE_BACKUP_RESOURCE_GROUP \  
  --sku Standard_GRS \  
  --encryption-services blob \  
  --https-only true \  
  --kind BlobStorage \  
  --access-tier Hot
```

6. Impostare la variabile **BLOB_CONTAINER**:

```
$ BLOB_CONTAINER=velero
```

7. Creare un container di storage Azure Blob:

```
$ az storage container create \
  -n $BLOB_CONTAINER \
  --public-access off \
  --account-name $AZURE_STORAGE_ACCOUNT_ID
```

8. Creare un'entità servizio e le credenziali per **velero**:

```
$ AZURE_SUBSCRIPTION_ID=`az account list --query '[?isDefault].id' -o tsv` \
  AZURE_TENANT_ID=`az account list --query '[?isDefault].tenantId' -o tsv` \
  AZURE_CLIENT_SECRET=`az ad sp create-for-rbac --name "velero" \
  --role "Contributor" --query 'password' -o tsv` \
  AZURE_CLIENT_ID=`az ad sp list --display-name "velero" \
  --query '[0].appId' -o tsv`
```

9. Salvare le credenziali dell'entità servizio nel file **credentials-velero**:

```
$ cat << EOF > ./credentials-velero
  AZURE_SUBSCRIPTION_ID=${AZURE_SUBSCRIPTION_ID}
  AZURE_TENANT_ID=${AZURE_TENANT_ID}
  AZURE_CLIENT_ID=${AZURE_CLIENT_ID}
  AZURE_CLIENT_SECRET=${AZURE_CLIENT_SECRET}
  AZURE_RESOURCE_GROUP=${AZURE_RESOURCE_GROUP}
  AZURE_CLOUD_NAME=AzurePublicCloud
  EOF
```

Si utilizza il file **credentials-velero** per aggiungere Azure come repository di replica.

6.5.6. Risorse aggiuntive

- [Workflow di MTC](#)
- [Informazioni sui metodi di copia dei dati](#)
- [Aggiungere un repository di replica alla console web di MTC](#)

6.6. DISINSTALLAZIONE DI MTC ED ELIMINAZIONE DI RISORSE

È possibile disinstallare il Migration Toolkit for Containers (MTC) ed eliminare le relative risorse per pulire il cluster.



NOTA

L'eliminazione dei CRD **velero** rimuove Velero dal cluster.

Requisiti

- È necessario aver eseguito l'accesso come utente con privilegi di **cluster-admin**.

Procedura

1. Eliminare la risorsa personalizzata **MigrationController** su tutti i cluster:

-

```
$ oc delete migrationcontroller <migration_controller>
```

2. Disinstallare il Migration Toolkit for Containers Operator su OpenShift Container Platform 4 utilizzando Operator Lifecycle Manager.
3. Eliminare le risorse in ambito cluster su tutti i cluster eseguendo i seguenti comandi:

- **migration** delle definizioni di risorse personalizzate (CRD):

```
$ oc delete $(oc get crds -o name | grep 'migration.openshift.io')
```

- **velero** CRD:

```
$ oc delete $(oc get crds -o name | grep 'velero')
```

- **migration** ruoli del cluster:

```
$ oc delete $(oc get clusterroles -o name | grep 'migration.openshift.io')
```

- **migration-operator** ruolo del cluster:

```
$ oc delete clusterrole migration-operator
```

- **velero** ruoli del cluster:

```
$ oc delete $(oc get clusterroles -o name | grep 'velero')
```

- **migration** binding dei ruoli del cluster:

```
$ oc delete $(oc get clusterrolebindings -o name | grep 'migration.openshift.io')
```

- **migration-operator** binding dei ruoli del cluster:

```
$ oc delete clusterrolebindings migration-operator
```

- **velero** binding dei ruoli del cluster:

```
$ oc delete $(oc get clusterrolebindings -o name | grep 'velero')
```

CAPITOLO 7. INSTALLARE MIGRATION TOOLKIT FOR CONTAINERS IN UN AMBIENTE DI RETE CON RESTRIZIONI

È possibile installare il Migration Toolkit for Containers (MTC) su OpenShift Container Platform 3 e 4 in un ambiente di rete con restrizioni eseguendo le seguenti procedure:

1. Creare un [catalogo Operator speculare](#).
Questo processo crea il file **mapping.txt**, che contiene il mapping tra l'immagine di **registry.redhat.io** e l'immagine di registro speculare. Il file **mapping.txt** è necessario per installare Operator sul cluster di origine.
2. Installare il Migration Toolkit for Containers Operator sul cluster di destinazione di OpenShift Container Platform 4.10 utilizzando Operator Lifecycle Manager.
Per impostazione predefinita, la console web di MTC e il pod **Migration Controller** vengono eseguiti sul cluster di destinazione. È possibile configurare il manifest delle risorse personalizzate di **Migration Controller** per eseguire la console web di MTC e il pod **Migration Controller** su un [cluster di origine o su un cluster remoto](#).
3. Installare il Migration Toolkit for Containers Operator *esistente* sul cluster di origine di OpenShift Container Platform 3 dall'interfaccia della riga di comando.
4. Configurare lo storage a oggetti da usare come repository di replica.

Per disinstallare MTC, consulta la sezione [Disinstallazione di MTC ed eliminazione delle risorse](#).

7.1. LINEE GUIDA PER LA COMPATIBILITÀ

È necessario installare il Migration Toolkit for Containers (MTC) Operator compatibile con la versione di OpenShift Container Platform.

Non è possibile installare MTC 1.6 su OpenShift Container Platform 4.5, o versioni precedenti, perché le versioni API delle definizioni delle risorse personalizzate sono incompatibili.

È possibile eseguire la migrazione dei carichi di lavoro da un cluster di origine con MTC 1.5.3 a un cluster di destinazione con MTC 1.6 purché la risorsa personalizzata **MigrationController** e la console web di MTC siano in esecuzione sul cluster di destinazione.

Tabella 7.1. OpenShift Container Platform e compatibilità MTC

Versione OpenShift Container Platform	Versione MTC	Migration Toolkit for Containers Operator
4.5 e precedenti	1.5.3	Migration Toolkit for Containers Operator esistente, installato manualmente con il file operator.yml .
4.6 e successivi	Ultima versione 1.6.x z-stream	Migration Toolkit for Containers Operator, installato con Operator Lifecycle Manager.

7.2. INSTALLARE IL MIGRATION TOOLKIT FOR CONTAINERS OPERATOR SU OPENSIFT CONTAINER PLATFORM 4.10

Si installa il Migration Toolkit for Containers Operator su OpenShift Container Platform 4.10 utilizzando Operator Lifecycle Manager.

Requisiti

- È necessario aver eseguito l'accesso come utente con privilegi di **cluster-admin** su tutti i cluster.
- È necessario creare un catalogo Operator da un'immagine speculare in un registro locale.

Procedura

1. Nella console web di OpenShift Container Platform, cliccare su **Operators** → **OperatorHub**.
2. Usare il campo **Filter by keyword** per trovare il **Migration Toolkit for Containers Operator**.
3. Selezionare il **Migration Toolkit for Containers Operator** e cliccare su **Install**.
4. Cliccare su **Install**.
Nella pagina degli **operatori installati**, il **Migration Toolkit for Containers Operator** appare nel progetto **openshift-migration** con lo stato **Succeeded**.
5. Cliccare su **Migration Toolkit for Containers Operator**.
6. In **Provided APIs**, individuare il file **Migration Controller** e cliccare su **Create Instance**.
7. Cliccare su **Create**.
8. Cliccare su **Workloads** → **Pods** per verificare che i pod MTC siano in esecuzione.

7.3. INSTALLARE IL MIGRATION TOOLKIT FOR CONTAINERS OPERATOR ESISTENTE SU OPENSIFT CONTAINER PLATFORM 3

È possibile installare manualmente il Migration Toolkit for Containers Operator esistente su OpenShift Container Platform 3.

Requisiti

- È necessario aver eseguito l'accesso come utente con privilegi di **cluster-admin** su tutti i cluster.
- È necessario disporre di accesso a **registry.redhat.io**.
- È necessario avere installato **podman**.
- È necessario creare un [segreto del flusso di immagini](#) e copiarlo su ogni nodo del cluster.
- È necessario avere una workstation Linux con accesso alla rete per poter scaricare i file da **registry.redhat.io**.
- È necessario creare un'immagine speculare del catalogo Operator.
- È necessario installare Migration Toolkit for Containers Operator dal catalogo Operator speculare su OpenShift Container Platform 4.10.

Procedura

1. Accedere a **registry.redhat.io** con le proprie credenziali del Red Hat Customer Portal:

```
$ sudo podman login registry.redhat.io
```

2. Scaricare il file **operator.yml**:

```
$ sudo podman cp $(sudo podman create \
registry.redhat.io/rhmtc/openshift-migration-legacy-rhel8-operator:v1.5.3):/operator.yml ./
```

3. Scaricare il file **controller.yml**:

```
$ sudo podman cp $(sudo podman create \
registry.redhat.io/rhmtc/openshift-migration-legacy-rhel8-operator:v1.5.3):/controller.yml ./
```

4. Ottenere il mapping delle immagini di Operator eseguendo il seguente comando:

```
$ grep openshift-migration-legacy-rhel8-operator ./mapping.txt | grep rhmtc
```

Il file **mapping.txt** è stato creato insieme al catalogo Operator speculare. L'output mostra il mapping tra l'immagine di **registry.redhat.io** e l'immagine di registro speculare.

Esempio di output

```
registry.redhat.io/rhmtc/openshift-migration-legacy-rhel8-
operator@sha256:468a6126f73b1ee12085ca53a312d1f96ef5a2ca03442bcb63724af5e2614e8
a=<registry.apps.example.com>/rhmtc/openshift-migration-legacy-rhel8-operator
```

5. Aggiornare i valori **image** per i container **ansible** e **operator** e il valore **REGISTRY** nel file **operator.yml**:

```
containers:
  - name: ansible
    image: <registry.apps.example.com>/rhmtc/openshift-migration-legacy-rhel8-
operator@sha256:
<468a6126f73b1ee12085ca53a312d1f96ef5a2ca03442bcb63724af5e2614e8a> 1
  ...
  - name: operator
    image: <registry.apps.example.com>/rhmtc/openshift-migration-legacy-rhel8-
operator@sha256:
<468a6126f73b1ee12085ca53a312d1f96ef5a2ca03442bcb63724af5e2614e8a> 2
  ...
  env:
    - name: REGISTRY
      value: <registry.apps.example.com> 3
```

1 2 Specificare il registro speculare e il valore **sha256** dell'immagine di Operator.

3 Specificare il registro speculare.

6. Accedere al cluster di OpenShift Container Platform 3.

7. Creare l'oggetto Migration Toolkit for Containers Operator:

```
$ oc create -f operator.yml
```

Esempio di output

```
namespace/openshift-migration created
rolebinding.rbac.authorization.k8s.io/system:deployers created
serviceaccount/migration-operator created
customresourcedefinition.apiextensions.k8s.io/migrationcontrollers.migration.openshift.io
created
role.rbac.authorization.k8s.io/migration-operator created
rolebinding.rbac.authorization.k8s.io/migration-operator created
clusterrolebinding.rbac.authorization.k8s.io/migration-operator created
deployment.apps/migration-operator created
Error from server (AlreadyExists): error when creating "./operator.yml":
rolebindings.rbac.authorization.k8s.io "system:image-builders" already exists 1
Error from server (AlreadyExists): error when creating "./operator.yml":
rolebindings.rbac.authorization.k8s.io "system:image-pullers" already exists
```

- 1** Puoi ignorare i messaggi **Error from server (AlreadyExists)**. Sono causati dal Migration Toolkit for Containers Operator che crea risorse per le versioni precedenti di OpenShift Container Platform 3 fornite nelle versioni successive.

8. Creare l'oggetto **MigrationController**:

```
$ oc create -f controller.yml
```

9. Verificare che i pod MTC siano in esecuzione:

```
$ oc get pods -n openshift-migration
```

7.4. CONFIGURARE I PROXY

Per OpenShift Container Platform 4.1 e versioni precedenti, è necessario configurare i proxy nel manifest della risorsa personalizzata **MigrationController** dopo aver installato il Migration Toolkit for Containers Operator perché queste versioni non supportano un oggetto **proxy** a livello di cluster.

Per OpenShift Container Platform da 4.2 a 4.10, il Migration Toolkit for Containers (MTC) eredita le impostazioni proxy a livello di cluster. Puoi cambiare i parametri del proxy se desideri sovrascrivere le impostazioni del proxy a livello di cluster.

È necessario configurare i proxy per consentire il protocollo SPDY e per inoltrare l'intestazione **Upgrade HTTP** al server API. Altrimenti, viene visualizzato l'errore **Upgrade request required**. La risorsa personalizzata **MigrationController** usa SPDY per eseguire comandi all'interno di pod remoti. L'intestazione **Upgrade HTTP** è necessaria per aprire una connessione websocket con il server API.

Migrazione diretta di volumi

Se si esegue una migrazione diretta di volumi (DVM) da un cluster di origine dietro un proxy, è necessario configurare un proxy Stunnel. Stunnel crea un tunnel trasparente tra i cluster di origine e destinazione per la connessione TCP senza cambiare i certificati.

DVM supporta solo un proxy. Il cluster di origine non può accedere al percorso del cluster di destinazione se anche il cluster di destinazione è dietro un proxy.

Requisiti

- È necessario aver eseguito l'accesso come utente con privilegi di **cluster-admin** su tutti i cluster.

Procedura

1. Ottenere il manifest della risorsa personalizzata **MigrationController**:

```
$ oc get migrationcontroller <migration_controller> -n openshift-migration
```

2. Aggiornare i parametri del proxy:

```
apiVersion: migration.openshift.io/v1alpha1
kind: MigrationController
metadata:
  name: <migration_controller>
  namespace: openshift-migration
...
spec:
  stunnel_tcp_proxy: http://<username>:<password>@<ip>:<port> 1
  httpProxy: http://<username>:<password>@<ip>:<port> 2
  httpsProxy: http://<username>:<password>@<ip>:<port> 3
  noProxy: example.com 4
```

- 1 URL proxy Stunnel per la migrazione diretta dei volumi.
- 2 URL proxy per creare connessioni HTTP all'esterno del cluster. Lo schema dell'URL deve essere **http**.
- 3 URL proxy per creare connessioni HTTPS all'esterno del cluster. Se questo non è specificato, allora viene utilizzato **httpProxy** per entrambe le connessioni HTTP e HTTPS.
- 4 Elenco separato da virgole di nomi di dominio di destinazione, domini, indirizzi IP o altri CIDR di rete per escludere il proxy.

Precedere un dominio con **.** per far corrispondere solo i sottodomini. Ad esempio, **.y.com** corrisponde a **x.y.com**, ma non a **y.com**. Usare ***** per bypassare il proxy per tutte le destinazioni. Se si aumenta il numero di lavoratori che non sono inclusi nella rete definita dal campo **networking.machineNetwork[].cidr** della configurazione di installazione, è necessario aggiungerli a questa lista per prevenire problemi di connessione.

Questo campo viene ignorato se non è impostato né il campo **httpProxy** né il campo **httpsProxy**.

3. Salvare il manifest come **migration-controller.yaml**.
4. Applicare il manifest aggiornato:

```
$ oc replace -f migration-controller.yaml -n openshift-migration
```

Per maggiori informazioni, consulta la sezione [Configurare il proxy a livello di cluster](#).

7.5. CONFIGURARE UN REPOSITORY DI REPLICA

Il Multicloud Object Gateway è l'unica opzione supportata in un ambiente di rete con restrizioni.

MTC supporta i [metodi di copia del file system e dei dati degli snapshot](#) per la migrazione dei dati dal cluster di origine al cluster di destinazione. È possibile selezionare un metodo adatto al proprio ambiente e supportato dal provider di storage.

7.5.1. Requisiti

- Tutti i cluster devono avere un accesso di rete ininterrotto al repository di replica.
- Se si utilizza un server proxy con un repository di replica ospitato internamente, è necessario assicurarsi che il proxy permetta l'accesso al repository di replica.

7.5.2. Recuperare le credenziali di Multicloud Object Gateway

È necessario recuperare le credenziali Multicloud Object Gateway (MCG) per creare una risorsa personalizzata (CR) **Secret** per OpenShift API for Data Protection (OADP).

MCG è un componente di OpenShift Container Storage.

Requisiti

- È necessario distribuire OpenShift Container Storage utilizzando la [guida per il deployment di OpenShift Container Storage](#) appropriata.

Procedura

1. Ottenere l'endpoint S3, **AWS_ACCESS_KEY_ID** e **AWS_SECRET_ACCESS_KEY** eseguendo il comando **describe** sulla risorsa personalizzata **NooBaa**.

7.5.3. Risorse aggiuntive

- [Ambiente scollegato](#) nella documentazione di Red Hat OpenShift Container Storage.
- [Workflow di MTC](#)
- [Informazioni sui metodi di copia dei dati](#)
- [Aggiungere un repository di replica alla console web di MTC](#)

7.6. DISINSTALLAZIONE DI MTC ED ELIMINAZIONE DI RISORSE

È possibile disinstallare il Migration Toolkit for Containers (MTC) ed eliminare le relative risorse per pulire il cluster.



NOTA

L'eliminazione dei CRD **velero** rimuove Velero dal cluster.

Requisiti

- È necessario aver eseguito l'accesso come utente con privilegi di **cluster-admin**.

Procedura

1. Eliminare la risorsa personalizzata **MigrationController** su tutti i cluster:

```
$ oc delete migrationcontroller <migration_controller>
```

2. Disinstallare il Migration Toolkit for Containers Operator su OpenShift Container Platform 4 utilizzando Operator Lifecycle Manager.

3. Eliminare le risorse in ambito cluster su tutti i cluster eseguendo i seguenti comandi:

- **migration** delle definizioni di risorse personalizzate (CRD):

```
$ oc delete $(oc get crds -o name | grep 'migration.openshift.io')
```

- **velero** CRD:

```
$ oc delete $(oc get crds -o name | grep 'velero')
```

- **migration** ruoli del cluster:

```
$ oc delete $(oc get clusterroles -o name | grep 'migration.openshift.io')
```

- **migration-operator** ruolo del cluster:

```
$ oc delete clusterrole migration-operator
```

- **velero** ruoli del cluster:

```
$ oc delete $(oc get clusterroles -o name | grep 'velero')
```

- **migration** binding dei ruoli del cluster:

```
$ oc delete $(oc get clusterrolebindings -o name | grep 'migration.openshift.io')
```

- **migration-operator** binding dei ruoli del cluster:

```
$ oc delete clusterrolebindings migration-operator
```

- **velero** binding dei ruoli del cluster:

```
$ oc delete $(oc get clusterrolebindings -o name | grep 'velero')
```

CAPITOLO 8. AGGIORNARE IL MIGRATION TOOLKIT FOR CONTAINERS

Puoi aggiornare il Migration Toolkit for Containers (MTC) su OpenShift Container Platform 4.10 utilizzando Operator Lifecycle Manager.

È possibile aggiornare MTC su OpenShift Container Platform 3 reinstallando il Migration Toolkit for Containers Operator esistente.



IMPORTANTE

Se si aggiorna da MTC versione 1.3, è necessario eseguire una procedura aggiuntiva per aggiornare la risorsa personalizzata **MigPlan**.

8.1. AGGIORNARE IL MIGRATION TOOLKIT FOR CONTAINERS SU OPENSIFT CONTAINER PLATFORM 4.10

È possibile aggiornare il Migration Toolkit for Containers (MTC) su OpenShift Container Platform 4.10 utilizzando Operator Lifecycle Manager.

Requisiti

- È necessario aver eseguito l'accesso come utente con privilegi di **cluster-admin**.

Procedura

1. Nella console di OpenShift Container Platform, andare su **Operators → Installed Operators**. Gli operatori che hanno un aggiornamento in sospeso mostrano lo stato **Upgrade available**.
2. Cliccare su **Migration Toolkit for Containers Operator**.
3. Cliccare sulla scheda **Subscription**. Tutti gli aggiornamenti che richiedono l'approvazione sono visualizzati accanto alla voce **Upgrade Status**. Per esempio, potrebbe essere visualizzato il messaggio **1 requires approval**.
4. Cliccare su **1 requires approval**, quindi cliccare su **Preview Install Plan**.
5. Esaminare le risorse elencate come disponibili per l'aggiornamento e cliccare su **Approve**.
6. Tornare alla pagina **Operators → Installed Operators** per monitorare il progresso dell'aggiornamento. Una volta completato, lo stato cambia in **Success** e **Up to date**.
7. Cliccare su **Migration Toolkit for Containers Operator**.
8. In **Provided APIs**, individuare il file **Migration Controller** e cliccare su **Create Instance**.
9. Cliccare su **Workloads → Pods** per verificare che i pod MTC siano in esecuzione.

8.2. AGGIORNARE IL MIGRATION TOOLKIT FOR CONTAINERS SU OPENSIFT CONTAINER PLATFORM 3

È possibile aggiornare il Migration Toolkit for Containers (MTC) su OpenShift Container Platform 3 installando manualmente il Migration Toolkit for Containers Operator esistente.

Requisiti

- È necessario aver eseguito l'accesso come utente con privilegi di **cluster-admin**.
- È necessario disporre di accesso a **registry.redhat.io**.
- È necessario avere installato **podman**.

Procedura

1. Accedere a **registry.redhat.io** con le proprie credenziali del Red Hat Customer Portal:

```
$ sudo podman login registry.redhat.io
```

2. Scaricare il file **operator.yml**:

```
$ sudo podman cp $(sudo podman create \
registry.redhat.io/rhmtc/openshift-migration-legacy-rhel8-operator:v1.5.3)/operator.yml ./
```

3. Sostituire il Migration Toolkit for Containers Operator:

```
$ oc replace --force -f operator.yml
```

4. Ridimensionare il deployment del **migration-operator** su **0** per fermare il deployment:

```
$ oc scale -n openshift-migration --replicas=0 deployment/migration-operator
```

5. Ridimensionare il deployment del **migration-operator** su **1** per avviare il deployment e applicare le modifiche:

```
$ oc scale -n openshift-migration --replicas=1 deployment/migration-operator
```

6. Verificare che il **migration-operator** sia stato aggiornato:

```
$ oc -o yaml -n openshift-migration get deployment/migration-operator | grep image: | awk -F
":" '{ print $NF }'
```

7. Scaricare il file **controller.yml**:

```
$ sudo podman cp $(sudo podman create \
registry.redhat.io/rhmtc/openshift-migration-legacy-rhel8-operator:v1.5.3)/controller.yml ./
```

8. Creare l'oggetto **migration-controller**:

```
$ oc create -f controller.yml
```

9. Se precedentemente è stato aggiunto il cluster di OpenShift Container Platform 3 alla console web di MTC, occorre aggiornare il token dell'account di servizio nella console web perché il processo di aggiornamento elimina e ripristina lo spazio dei nomi **openshift-migration**:

- a. Ottenere il token dell'account di servizio:

```
$ oc sa get-token migration-controller -n openshift-migration
```


b. Nella console web di MTC, cliccare su **Cluster**.

c. Cliccare sul menu Opzioni  accanto al cluster e selezionare **Modifica**.

d. Inserire il nuovo token dell'account di servizio nel campo **Service account token**

e. Cliccare su **Update cluster** e poi su **Close**.

10. Verificare che i pod MTC siano in esecuzione:

```
$ oc get pods -n openshift-migration
```

8.3. AGGIORNAMENTO DI MTC 1.3 A 1.6

Se si aggiorna il Migration Toolkit for Containers (MTC) dalla versione 1.3.x alla 1.6, occorre aggiornare il manifest delle risorse personalizzate **MigPlan** sul cluster su cui è in esecuzione il pod **MigrationController**.

Poiché i parametri **indirectImageMigration** e **indirectVolumeMigration** non esistono in MTC 1.3, il loro valore predefinito nella versione 1.4 è **false**, il che significa che la migrazione diretta delle immagini e la migrazione diretta dei volumi sono abilitate. Poiché i requisiti per la migrazione diretta non sono soddisfatti, il piano di migrazione non può raggiungere lo stato **Ready** a meno che questi valori dei parametri non siano cambiati in **true**.

Requisiti

- È necessario aver eseguito l'accesso come utente con privilegi di **cluster-admin**.

Procedura

1. Accedere al cluster su cui è in esecuzione il pod **MigrationController**.

2. Ottenere il manifest della risorsa personalizzata **MigPlan**:

```
$ oc get migplan <migplan> -o yaml -n openshift-migration
```

3. Aggiornare i seguenti valori dei parametri e salvare il file come **migplan.yaml**:

```
...
spec:
  indirectImageMigration: true
  indirectVolumeMigration: true
```

4. Sostituire il manifest della risorsa personalizzata **MigPlan** per applicare le modifiche:

```
$ oc replace -f migplan.yaml -n openshift-migration
```

5. Ottenere il manifest della risorsa personalizzata **MigPlan** aggiornato per verificare le modifiche:

```
$ oc get migplan <migplan> -o yaml -n openshift-migration
```

CAPITOLO 9. LISTE DI CONTROLLO DI PRE-MIGRAZIONE

Prima di eseguire la migrazione dei carichi di lavoro delle applicazioni con il Migration Toolkit for Containers (MTC), rivedere le seguenti liste di controllo.

9.1. RISORSE

- Se l'applicazione usa una rete di servizi interna o un percorso esterno per comunicare con i servizi, il percorso pertinente esiste.
- Se l'applicazione usa risorse a livello di cluster, queste sono state ricreate sul cluster di destinazione.
- Sono stati [esclusi](#) volumi permanenti (PV), flussi di immagini e altre risorse di cui non si desidera eseguire la migrazione.
- I dati PV sono stati sottoposti a backup nel caso in cui un'applicazione mostri un comportamento inaspettato dopo la migrazione e corrompa i dati.

9.2. CLUSTER DI ORIGINE

- Il cluster soddisfa i [requisiti hardware minimi](#).
- È installata la versione corretta del Migration Toolkit for Containers Operator:
 - **operator-3.7.yml** su OpenShift Container Platform versione 3.7.
 - **operator.yml** su OpenShift Container Platform versioni da 3.9 a 4.5.
- La versione dell'MTC è la stessa su tutti i cluster.
- Tutti i nodi hanno una sottoscrizione attiva a OpenShift Container Platform.
- Hai eseguito tutte le [attività da eseguire una volta sola](#).
- Hai eseguito tutti i [controlli di integrità dell'ambiente](#).
- Hai controllato i PV con configurazioni anormali bloccati nello stato **Terminating** eseguendo il seguente comando:

```
$ oc get pv
```

- Hai controllato i pod il cui stato è diverso da **Running** o **Completed** eseguendo il seguente comando:

```
$ oc get pods --all-namespaces | egrep -v 'Running | Completed'
```

- Hai controllato i pod con un alto numero di riavvii eseguendo il seguente comando:

```
$ oc get pods --all-namespaces --field-selector=status.phase=Running \
-o json | jq '.items[]|select(any( .status.containerStatuses[]; \
.restartCount > 3))|.metadata.name'
```

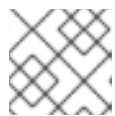
Anche se i pod sono nello stato **Running**, un alto numero di riavvii potrebbe indicare alcuni problemi.

- Hai rimosso vecchie build, deployment e immagini da ogni spazio dei nomi di cui deve essere eseguita la migrazione tramite [pruning](#).
- Il registro interno usa un [tipo di storage supportato](#).
- Solo migrazione diretta delle immagini: il registro interno è [esposto](#) al traffico esterno.
- È possibile leggere e scrivere immagini nel registro.
- Il [cluster etcd](#) è integro.
- Il [tempo medio di risposta del server API](#) sul cluster di origine è inferiore a 50 ms.
- I certificati del cluster sono [validi](#) per la durata del processo di migrazione.
- Hai controllato le richieste di firma dei certificati in sospeso eseguendo il seguente comando:


```
$ oc get csr -A | grep pending -i
```
- Il [provider di identità](#) sta funzionando.

9.3. CLUSTER DI DESTINAZIONE

- Hai installato il Migration Toolkit for Containers Operator versione 1.5.1.
- Tutti i [requisiti di MTC](#) sono soddisfatti.
- Il cluster soddisfa i requisiti hardware minimi per la piattaforma specifica e il metodo di installazione, per esempio, su [bare metal](#).
- Il cluster ha [classi di storage](#) definite per i tipi di storage utilizzati dal cluster di origine, per esempio, volume a blocchi, file system o storage a oggetti.



NOTA

NFS non richiede una classe di storage definita.

- Il cluster ha la configurazione di rete e le autorizzazioni corrette per accedere a servizi esterni, per esempio, database, repository di codice sorgente, registri di immagini container e strumenti CI/CD.
- Le applicazioni e i servizi esterni che utilizzano i servizi forniti dal cluster hanno la configurazione di rete e le autorizzazioni corrette per accedere al cluster.
- Le dipendenze interne delle immagini container sono soddisfatte.
Se un'applicazione utilizza un'immagine interna nello spazio dei nomi **openshift** che non è supportata da OpenShift Container Platform 4.10, è possibile aggiornare manualmente il [tag del flusso di immagini di OpenShift Container Platform 3](#) con **podman**.
- Il cluster di destinazione e il repository di replica hanno spazio di storage sufficiente.
- Il [provider di identità](#) sta funzionando.

- I record DNS dell'applicazione esistono sul cluster di destinazione.
- Impostare il valore del parametro **annotation.openshift.io/host.generated** su **true** per ogni percorso di OpenShift Container Platform per aggiornare il nome host per il cluster di destinazione. Altrimenti, i percorsi migrati mantengono il nome dell'host del cluster di origine.
- I certificati che l'applicazione utilizza esistono sul cluster di destinazione.
- Hai configurato regole firewall appropriate sul cluster di destinazione.
- Hai configurato correttamente il bilanciamento del carico sul cluster di destinazione.
- Se si esegue la migrazione di oggetti in uno spazio dei nomi esistente sul cluster di destinazione che ha lo stesso nome dello spazio dei nomi migrato dall'origine, lo spazio dei nomi di destinazione non contiene oggetti dello stesso nome e tipo degli oggetti di cui viene eseguita la migrazione.

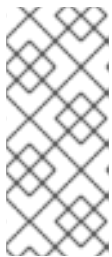


NOTA

Non creare spazi dei nomi per l'applicazione sul cluster di destinazione prima della migrazione perché questo potrebbe causare il cambiamento delle quote.

9.4. PRESTAZIONI

- La rete di migrazione ha una produttività minima di 10 Gbps.
- I cluster hanno risorse sufficienti per la migrazione.



NOTA

I cluster richiedono memoria, CPU e storage aggiuntivi per eseguire una migrazione oltre ai normali carichi di lavoro. I requisiti effettivi delle risorse dipendono dal numero di risorse Kubernetes che vengono migrate in un singolo piano di migrazione. È necessario testare le migrazioni in un ambiente non di produzione per stimare i requisiti delle risorse.

- L'[utilizzo di memoria e CPU](#) dei nodi è efficiente.
- Le [prestazioni del disco etcd](#) dei cluster sono state controllate con **fiio**.

CAPITOLO 10. MIGRAZIONE DELLE APPLICAZIONI

È possibile eseguire la migrazione delle applicazioni utilizzando la console web di Migration Toolkit for Containers (MTC) o dalla [riga di comando](#).

È possibile usare la migrazione di fase e la migrazione completa per eseguire la migrazione di un'applicazione tra cluster:

- La migrazione di fase copia i dati dal cluster di origine al cluster di destinazione senza fermare l'applicazione. È possibile eseguire una migrazione di fase più volte per ridurre la durata della migrazione completa.
- Una migrazione completa ferma le transazioni sul cluster di origine e sposta le risorse sul cluster di destinazione.

È possibile usare la migrazione dello stato per eseguire la migrazione dello stato di un'applicazione:

- La migrazione dello stato copia le richieste di volumi permanenti (PVC) e le risorse Kubernetes selezionate.
- È possibile usare la migrazione dello stato per eseguire la migrazione di uno spazio dei nomi all'interno dello stesso cluster.

La maggior parte delle risorse in ambito cluster non è ancora gestita da MTC. Se le applicazioni necessitano di risorse in ambito cluster, potrebbe essere necessario crearle manualmente sul cluster di destinazione.

Durante la migrazione, MTC conserva le seguenti annotazioni dello spazio dei nomi:

- **openshift.io/sa.scc.mcs**
- **openshift.io/sa.scc.supplemental-groups**
- **openshift.io/sa.scc.uid-range**

Queste annotazioni preservano l'intervallo UID, assicurando che i container mantengano le autorizzazioni del file system sul cluster di destinazione. Esiste il rischio che gli UID migrati possano duplicare gli UID all'interno di uno spazio dei nomi esistente o futuro sul cluster di destinazione.

10.1. REQUISITI PER LA MIGRAZIONE

- È necessario aver eseguito l'accesso come utente con privilegi di **cluster-admin** su tutti i cluster.

Migrazione diretta di immagini

- È necessario assicurarsi che il registro interno sicuro del cluster di origine sia esposto.
- È necessario creare un percorso verso il registro esposto.

Migrazione diretta di volumi

- Se i cluster usano dei proxy, è necessario configurare un proxy Stunnel TCP.

Immagini interne

- Se l'applicazione utilizza immagini interne dello spazio dei nomi **openshift**, è necessario assicurarsi che le versioni richieste delle immagini siano presenti sul cluster di destinazione. È possibile aggiornare manualmente un tag di flusso di immagini per utilizzare un'immagine deprecata di OpenShift Container Platform 3 su un cluster di OpenShift Container Platform 4.10.

Cluster

- Il cluster di origine deve essere aggiornato all'ultima versione z-stream di MTC.
- La versione MTC deve essere la stessa su tutti i cluster.

Rete

- I cluster hanno accesso illimitato alla rete tra loro e al repository di replica.
- Se si copiano i volumi permanenti con **move**, i cluster devono avere un accesso di rete illimitato ai volumi remoti.
- È necessario abilitare le seguenti porte su un cluster OpenShift Container Platform 3:
 - **8443** (server API)
 - **443** (percorsi)
 - **53** (DNS)
- È necessario abilitare le seguenti porte su un cluster OpenShift Container Platform 4:
 - **6443** (server API)
 - **443** (percorsi)
 - **53** (DNS)
- È necessario abilitare la porta **443** sul repository di replica se si utilizza TLS.

Volumi permanenti (PV)

- I PV devono essere validi.
- I PV devono essere legati a richieste di volumi permanenti.
- Se si usano gli snapshot per copiare i PV, si applicano i seguenti requisiti aggiuntivi:
 - Il provider cloud deve supportare gli snapshot.
 - I PV devono avere lo stesso provider cloud.
 - I PV devono essere situati nella stessa area geografica.
 - I PV devono avere la stessa classe di storage.

Risorse aggiuntive per i requisiti della migrazione

- [Esporre manualmente un registro sicuro per OpenShift Container Platform 3](#)

- [Aggiornamento delle immagini interne deprecate](#)

10.2. MIGRAZIONE DI APPLICAZIONI UTILIZZANDO LA CONSOLE WEB DI MTC

È possibile configurare i cluster e un repository di replica utilizzando la console web di MTC. Quindi, è possibile creare ed eseguire un piano di migrazione.

10.2.1. Lanciare la console web di MTC

È possibile lanciare la console web di Migration Toolkit for Containers (MTC) in un browser.

Requisiti

- La console web di MTC deve disporre di un accesso di rete alla console web di OpenShift Container Platform.
- La console web di MTC deve avere un accesso di rete al server di autorizzazione OAuth.

Procedura

1. Accedere al cluster di OpenShift Container Platform su cui è installato MTC.
2. Ottenere l'URL della console web di MTC inserendo il seguente comando:

```
$ oc get -n openshift-migration route/migration -o go-template='https://{ .spec.host }'
```

L'output assomiglia al seguente: **https://migration-openshift-migration.apps.cluster.openshift.com**.

3. Lanciare un browser e andare alla console web di MTC.



NOTA

Se provi ad accedere alla console web di MTC subito dopo aver installato Migration Toolkit for Containers Operator, la console potrebbe non caricarsi perché l'Operator sta ancora configurando il cluster. Aspettare qualche minuto e riprovare.

4. Se si utilizzano certificati CA autofirmati, verrà richiesto di accettare il certificato CA del server API del cluster di origine. La pagina web ti guida attraverso il processo di accettazione dei certificati rimanenti.
5. Accedere con il **nome utente** e **password** di OpenShift Container Platform.

10.2.2. Aggiungere un cluster alla console web di MTC

È possibile aggiungere un cluster alla console web di Migration Toolkit for Containers (MTC).

Requisiti

- Se si utilizzano gli snapshot di Azure per copiare i dati:


```
$ oc create route passthrough --service=image-registry --port=5000 -n openshift-  
image-registry
```

- **Cluster Azure:** è necessario selezionare questa opzione se si utilizzano gli snapshot di Azure per copiare i dati.
 - **Gruppo di risorse Azure:** questo campo viene visualizzato se è selezionata l'opzione **cluster Azure**. Specificare il gruppo di risorse Azure.
 - **Require SSL verification:** Facoltativo: selezionare questa opzione per verificare le connessioni SSL al cluster.
 - **File bundle CA:** questo campo viene visualizzato se è selezionata l'opzione **Require SSL verification**. Se si è creato un file bundle del certificato CA personalizzato per i certificati autofirmati, cliccare su **Browse**, selezionare il file bundle CA e caricarlo.
6. Cliccare su **Add cluster**.
Il cluster appare nell'elenco **Clusters**.

10.2.3. Aggiungere un repository di replica alla console web di MTC

È possibile aggiungere uno storage a oggetti come repository di replica alla console web di Migration Toolkit for Containers (MTC).

L'MTC supporta i seguenti provider di storage:

- Amazon Web Services (AWS) S3
- Multi-Cloud Object Gateway (MCG)
- Storage a oggetti S3 generico, ad esempio, Minio o Ceph S3
- Google Cloud Provider (GCP)
- Microsoft Azure Blob

Requisiti

- È necessario configurare lo storage a oggetti come repository di replica.

Procedura

1. Nella console web di MTC, cliccare su **Replication repositories**.
2. Cliccare su **Add repository**.
3. Selezionare un **tipo di provider di storage** e compilare i seguenti campi:
 - **AWS** per i provider S3, compresi AWS e MCG:
 - **Nome del repository di replica** specificare il nome del repository di replica nella console web di MTC.
 - **Nome bucket S3:** specificare il nome del bucket S3.

- **Area geografica del bucket S3:** specificare l'area geografica del bucket S3. **Richiesto** per AWS S3. **Facoltativo** per alcuni provider S3. Controllare la documentazione del prodotto del provider S3 per i valori attesi.
 - **Endpoint S3:** specificare l'URL del servizio S3, non il bucket, per esempio, **https://<s3-storage.apps.cluster.com>**. **Required** per un provider S3 generico. Occorre usare il prefisso **https://**.
 - **Chiave di accesso del provider S3** specificare la chiave **<AWS_SECRET_ACCESS_KEY>** per AWS o la chiave di accesso del provider S3 per MCG e altri provider S3.
 - **Chiave di accesso segreta** del provider S3: specificare l'ID **<AWS_ACCESS_KEY_ID>** per AWS o la chiave di accesso segreta del provider S3 per MCG e altri provider S3.
 - **Require SSL verification:** deselezionare questa casella di controllo se si utilizza un provider S3 generico.
 - Se si è creato un bundle di certificati CA personalizzati per i certificati autofirmati, cliccare su **Browse** e cercare il file con codifica Base64.
- **GCP:**
 - **Nome del repository di replica** specificare il nome del repository di replica nella console web di MTC.
 - **Nome bucket GCP:** specificare il nome del bucket GCP.
 - **Blob JSON delle credenziali GCP:** specificare la stringa nel file **credentials-velero**.
 - **Azure:**
 - **Nome del repository di replica** specificare il nome del repository di replica nella console web di MTC.
 - **Gruppo di risorse Azure:** specificare il gruppo di risorse dello storage Azure Blob.
 - **Nome account di storage Azure** specificare il nome account dello storage Azure Blob.
 - **Credenziali Azure - contenuto del file INI** specificare la stringa nel file **credentials-velero**.
4. Cliccare su **Add repository** e aspettare la convalida della connessione.
 5. Cliccare su **Close**.
Il nuovo repository appare nella lista **Replication repositories**.

10.2.4. Creare un piano di migrazione nella console web di MTC

È possibile creare un piano di migrazione nella console web di Migration Toolkit for Containers (MTC).

Requisiti

- È necessario aver eseguito l'accesso come utente con privilegi di **cluster-admin** su tutti i cluster.
- È necessario assicurarsi che la stessa versione di MTC sia installata su tutti i cluster.

- È necessario aggiungere i cluster e il repository di replica alla console web di MTC.
- Se si desidera usare il metodo di copia dei dati di *spostamento* per eseguire la migrazione di un volume permanente (PV), i cluster di origine e di destinazione devono avere un accesso di rete ininterrotto al volume remoto.
- Se si desidera usare la migrazione diretta delle immagini, è necessario specificare il percorso esposto al registro delle immagini del cluster di origine. Questo può essere fatto tramite la console web di MTC o aggiornando il manifest delle risorse personalizzate **MigCluster**.

Procedura

1. Nella console web di MTC, cliccare su **Migration plans**.
2. Cliccare su **Add migration plan**.
3. Inserire il **nome del piano**.
Il nome del piano di migrazione non deve superare i 253 caratteri alfanumerici minuscoli (**a-z, 0-9**) e non deve contenere spazi o trattino basso (`_`).
4. Selezionare un **cluster di origine**, un **cluster di destinazione** e un **repository**.
5. Cliccare su **Next**.
6. Selezionare i progetti per la migrazione.
7. Facoltativo: cliccare sull'icona di modifica accanto a un progetto per cambiare lo spazio dei nomi di destinazione.
8. Cliccare su **Next**.
9. Selezionare un **tipo di migrazione** per ogni PV:
 - L'opzione **Copy** copia i dati dal PV di un cluster di origine al repository di replica e poi ripristina i dati su un PV appena creato, con caratteristiche simili, nel cluster di destinazione.
 - L'opzione **Move** smonta un volume remoto, per esempio NFS, dal cluster di origine, crea una risorsa PV sul cluster di destinazione che punta al volume remoto, e poi monta il volume remoto sul cluster di destinazione. Le applicazioni in esecuzione sul cluster di destinazione utilizzano lo stesso volume remoto utilizzato dal cluster di origine.
10. Cliccare su **Next**.
11. Selezionare un **metodo di copia** per ogni PV:
 - **Lacopia degli snapshot** esegue il backup e il ripristino dei dati utilizzando la funzionalità di snapshot del provider cloud. È significativamente più veloce della **copia del file system**.
 - **Lacopia del file system** esegue il backup dei file sul cluster di origine e li ripristina sul cluster di destinazione.
Il metodo di copia del file system è richiesto per la migrazione diretta dei volumi.
12. È possibile selezionare **Verify copy** per verificare i dati migrati con la **copia del file system**. I dati sono verificati generando un checksum per ogni file di origine e controllando il checksum dopo il ripristino. La verifica dei dati riduce significativamente le prestazioni.
13. Selezionare una **classe di storage di destinazione**

Se è selezionata la **copia del file system**, è possibile cambiare la classe di storage di destinazione.

14. Cliccare su **Next**.

15. Nella pagina **Migration options**, è selezionata l'opzione **Direct image migration** se si è specificato un percorso esposto del registro delle immagini per il cluster di origine. L'opzione **Direct PV migration** è selezionata se si esegue la migrazione dei dati con la **copia del file system**.

Le opzioni di migrazione diretta copiano immagini e file direttamente dal cluster di origine al cluster di destinazione. Questa opzione è molto più veloce che copiare immagini e file dal cluster di origine al repository di replica e poi dal repository di replica al cluster di destinazione.

16. Cliccare su **Next**.

17. Facoltativo: cliccare su **Add Hook** per aggiungere un hook al piano di migrazione.

Un hook esegue del codice personalizzato. È possibile aggiungere fino a quattro hook a un singolo piano di migrazione. Ogni hook viene eseguito durante un diverso stadio della migrazione.

a. Inserire il nome dell'hook da visualizzare nella console web.

b. Se l'hook è un Ansible Playbook, selezionare **Ansible Playbook** e cliccare su **Browse** per caricare il playbook o incollare il contenuto del playbook nel campo.

c. Facoltativo: specificare un'immagine di runtime di Ansible se non si utilizza l'immagine dell'hook predefinita.

d. Se l'hook non è un Ansible Playbook, selezionare **Custom container image** e specificare il nome e il percorso dell'immagine.

Un'immagine container personalizzata può includere degli Ansible Playbook.

e. Selezionare il **cluster di origine** o il **cluster di destinazione**

f. Inserire il **nome dell'account di servizio** e lo **spazio dei nomi dell'account di servizio**

g. Selezionare lo stadio della migrazione per l'hook:

- **preBackup**: prima che il carico di lavoro dell'applicazione sia sottoposto a backup sul cluster di origine
- **postBackup**: dopo il backup del carico di lavoro dell'applicazione sul cluster di origine
- **preRestore**: prima che il carico di lavoro dell'applicazione venga ripristinato sul cluster di destinazione
- **postRestore**: dopo che il carico di lavoro dell'applicazione è stato ripristinato sul cluster di destinazione

h. Cliccare su **Add**.

18. Cliccare su **Finish**.

Il piano di migrazione viene visualizzato nell'elenco **Migration plans**.

Risorse aggiuntive

- [Metodo di copia del file system MTC](#)

- [Metodo di copia degli snapshot MTC](#)

10.2.5. Eseguire un piano di migrazione nella console web di MTC

È possibile eseguire la migrazione di applicazioni e dati con il piano di migrazione creato nella console web di Migration Toolkit for Containers (MTC).



NOTA

Durante la migrazione, MTC imposta il criterio di recupero dei volumi permanenti (PV) migrati su **Retain** sul cluster di destinazione.

La risorsa personalizzata **Backup** contiene l'annotazione **PVOriginalReclaimPolicy** che indica il criterio di recupero originale. È possibile ripristinare manualmente il criterio di recupero dei PV migrati.


Requisiti

La console web di MTC deve contenere quanto segue:

- Cluster di origine nello stato **Ready**
- Cluster di destinazione nello stato **Ready**
- Repository di replica
- Piano di migrazione valido

Procedura

1. Accedere alla console web di MTC e cliccare su **Migration plans**.

2. Cliccare sul menu Opzioni  accanto a un piano di migrazione e selezionare una delle seguenti opzioni sotto **Migration**:

- La migrazione **di fase** copia i dati dal cluster di origine al cluster di destinazione senza fermare l'applicazione.
- La migrazione **completa** ferma le transazioni sul cluster di origine e sposta le risorse sul cluster di destinazione.
Facoltativo: nella finestra di dialogo della **migrazione completa**, è possibile deselezionare la casella di controllo **Halt transactions on the source cluster during migration**
- La migrazione dello **stato** copia le richieste di volumi permanenti (PVC) selezionate e le risorse Kubernetes che costituiscono lo stato di un'applicazione. È possibile usare la migrazione dello stato per eseguire la migrazione di uno spazio dei nomi all'interno dello stesso cluster.



IMPORTANTE

Non utilizzare la migrazione dello stato per eseguire la migrazione di uno spazio dei nomi tra cluster. Usare invece la migrazione di fase o completa.

- Selezionare una o più PVC nella finestra di dialogo **State migration** e cliccare su **Migrate**.
3. Quando la migrazione è stata completata, verificare che l'applicazione sia migrata con successo nella console web di OpenShift Container Platform:
- a. Cliccare su **Home → Projects**.
 - b. Cliccare sul progetto migrato per visualizzarne lo stato.
 - c. Nella sezione **Routes**, cliccare su **Location** per verificare che l'applicazione sia funzionante, se applicabile.
 - d. Cliccare su **Workloads → Pods** per verificare che i pod siano in esecuzione nello spazio dei nomi migrato.
 - e. Cliccare su **Storage → Persistent volumes** per verificare che il provisioning dei volumi permanenti migrati sia avvenuto correttamente.

CAPITOLO 11. OPZIONI DI MIGRAZIONE AVANZATE

È possibile automatizzare le migrazioni e modificare le risorse personalizzate **MigPlan** e **MigrationController** per eseguire migrazioni di grandi dimensioni e migliorare le prestazioni.

11.1. TERMINOLOGIA

Tabella 11.1. Terminologia MTC

Termine	Definizione
Cluster di origine	Cluster da cui viene eseguita la migrazione delle applicazioni.
Cluster di destinazione ^[1]	Cluster verso cui viene eseguita la migrazione delle applicazioni.
Repository di replica	Storage a oggetti utilizzato per copiare immagini, volumi e oggetti Kubernetes durante la migrazione indiretta o per gli oggetti Kubernetes durante la migrazione diretta di volumi o la migrazione diretta di immagini. Il repository di replica deve essere accessibile a tutti i cluster.
Cluster host	Cluster su cui sono in esecuzione il pod migration-controller e la console web. Il cluster host è di solito il cluster di destinazione ma non deve esserlo necessariamente. Il cluster host non richiede un percorso di registro esposto per la migrazione diretta delle immagini.
Cluster remoto	Un cluster remoto è di solito il cluster di origine ma non deve esserlo necessariamente. Un cluster remoto richiede una risorsa personalizzata Secret che contiene il token dell'account di servizio del migration-controller . Un cluster remoto richiede un percorso di registro sicuro esposto per la migrazione diretta delle immagini.
Migrazione indiretta	Immagini, volumi e oggetti Kubernetes vengono copiati dal cluster di origine al repository di replica e poi dal repository di replica al cluster di destinazione.
Migrazione diretta di volumi	I volumi permanenti sono copiati direttamente dal cluster di origine al cluster di destinazione.
Migrazione diretta di immagini	Le immagini sono copiate direttamente dal cluster di origine al cluster di destinazione.
Migrazione di fase	I dati vengono copiati nel cluster di destinazione senza fermare l'applicazione. L'esecuzione di una migrazione di fase più volte riduce la durata della migrazione completa.

Termine	Definizione
Migrazione completa	L'applicazione viene fermata sul cluster di origine e viene eseguita la migrazione delle sue risorse sul cluster di destinazione.
Migrazione dello stato	Viene eseguita la migrazione dello stato dell'applicazione copiando specifiche richieste di volumi permanenti e oggetti Kubernetes nel cluster di destinazione.
Migrazione rollback	Con una migrazione rollback viene eseguito il rollback di una migrazione completata.

¹ Chiamate il cluster di *destinazione* nella console web di MTC.

11.2. ESEGUIRE LA MIGRAZIONE DI APPLICAZIONI USANDO LA RIGA DI COMANDO

È possibile eseguire la migrazione delle applicazioni con l'API MTC utilizzando l'interfaccia a riga di comando (CLI) per automatizzare la migrazione.

11.2.1. Requisiti per la migrazione

- È necessario aver eseguito l'accesso come utente con privilegi di **cluster-admin** su tutti i cluster.

Migrazione diretta di immagini

- È necessario assicurarsi che il registro interno sicuro del cluster di origine sia esposto.
- È necessario creare un percorso verso il registro esposto.

Migrazione diretta di volumi

- Se i cluster usano dei proxy, è necessario configurare un proxy Stunnel TCP.

Immagini interne

- Se l'applicazione utilizza immagini interne dello spazio dei nomi **openshift**, è necessario assicurarsi che le versioni richieste delle immagini siano presenti sul cluster di destinazione. È possibile aggiornare manualmente un tag di flusso di immagini per utilizzare un'immagine deprecata di OpenShift Container Platform 3 su un cluster di OpenShift Container Platform 4.10.

Cluster

- Il cluster di origine deve essere aggiornato all'ultima versione z-stream di MTC.
- La versione MTC deve essere la stessa su tutti i cluster.

Rete

- I cluster hanno accesso illimitato alla rete tra loro e al repository di replica.
- Se si copiano i volumi permanenti con **move**, i cluster devono avere un accesso di rete illimitato ai volumi remoti.
- È necessario abilitare le seguenti porte su un cluster OpenShift Container Platform 3:
 - **8443** (server API)
 - **443** (percorsi)
 - **53** (DNS)
- È necessario abilitare le seguenti porte su un cluster OpenShift Container Platform 4:
 - **6443** (server API)
 - **443** (percorsi)
 - **53** (DNS)
- È necessario abilitare la porta **443** sul repository di replica se si utilizza TLS.

Volumi permanenti (PV)

- I PV devono essere validi.
- I PV devono essere legati a richieste di volumi permanenti.
- Se si usano gli snapshot per copiare i PV, si applicano i seguenti requisiti aggiuntivi:
 - Il provider cloud deve supportare gli snapshot.
 - I PV devono avere lo stesso provider cloud.
 - I PV devono essere situati nella stessa area geografica.
 - I PV devono avere la stessa classe di storage.

11.2.2. Creare un percorso di registro per la migrazione diretta delle immagini

Per la migrazione diretta delle immagini, è necessario creare un percorso verso il registro interno esposto su tutti i cluster remoti.

Requisiti

- Il registro interno deve essere esposto al traffico esterno su tutti i cluster remoti. Il registro di OpenShift Container Platform 4 è esposto per impostazione predefinita. Il registro di OpenShift Container Platform 3 deve essere [esposto manualmente](#).

Procedura

- Per creare un percorso verso un registro di OpenShift Container Platform 3, eseguire il seguente comando:

```
$ oc create route passthrough --service=docker-registry -n default
```

- - Per creare un percorso verso un registro di OpenShift Container Platform 4, eseguire il seguente comando:

```
$ oc create route passthrough --service=image-registry -n openshift-image-registry
```

11.2.3. Configurare i proxy

Per OpenShift Container Platform 4.1 e versioni precedenti, è necessario configurare i proxy nel manifest della risorsa personalizzata **MigrationController** dopo aver installato il Migration Toolkit for Containers Operator perché queste versioni non supportano un oggetto **proxy** a livello di cluster.

Per OpenShift Container Platform da 4.2 a 4.10, il Migration Toolkit for Containers (MTC) eredita le impostazioni proxy a livello di cluster. Puoi cambiare i parametri del proxy se desideri sovrascrivere le impostazioni del proxy a livello di cluster.

È necessario configurare i proxy per consentire il protocollo SPDY e per inoltrare l'intestazione **Upgrade HTTP** al server API. Altrimenti, viene visualizzato l'errore **Upgrade request required**. La risorsa personalizzata **MigrationController** usa SPDY per eseguire comandi all'interno di pod remoti. L'intestazione **Upgrade HTTP** è necessaria per aprire una connessione websocket con il server API.

Migrazione diretta di volumi

Se si esegue una migrazione diretta di volumi (DVM) da un cluster di origine dietro un proxy, è necessario configurare un proxy Stunnel. Stunnel crea un tunnel trasparente tra i cluster di origine e destinazione per la connessione TCP senza cambiare i certificati.

DVM supporta solo un proxy. Il cluster di origine non può accedere al percorso del cluster di destinazione se anche il cluster di destinazione è dietro un proxy.

Requisiti

- È necessario aver eseguito l'accesso come utente con privilegi di **cluster-admin** su tutti i cluster.

Procedura

1. Ottenere il manifest della risorsa personalizzata **MigrationController**:

```
$ oc get migrationcontroller <migration_controller> -n openshift-migration
```

2. Aggiornare i parametri del proxy:

```
apiVersion: migration.openshift.io/v1alpha1
kind: MigrationController
metadata:
  name: <migration_controller>
  namespace: openshift-migration
...
spec:
  stunnel_tcp_proxy: http://<username>:<password>@<ip>:<port> 1
  httpProxy: http://<username>:<password>@<ip>:<port> 2
  httpsProxy: http://<username>:<password>@<ip>:<port> 3
  noProxy: example.com 4
```

- 1 URL proxy Stunnel per la migrazione diretta dei volumi.
- 2 URL proxy per creare connessioni HTTP all'esterno del cluster. Lo schema dell'URL deve essere **http**.
- 3 URL proxy per creare connessioni HTTPS all'esterno del cluster. Se questo non è specificato, allora viene utilizzato **httpProxy** per entrambe le connessioni HTTP e HTTPS.
- 4 Elenco separato da virgole di nomi di dominio di destinazione, domini, indirizzi IP o altri CIDR di rete per escludere il proxy.

Precedere un dominio con `.` per far corrispondere solo i sottodomini. Ad esempio, **.y.com** corrisponde a **x.y.com**, ma non a **y.com**. Usare `*` per bypassare il proxy per tutte le destinazioni. Se si aumenta il numero di lavoratori che non sono inclusi nella rete definita dal campo **networking.machineNetwork[].cidr** della configurazione di installazione, è necessario aggiungerli a questa lista per prevenire problemi di connessione.

Questo campo viene ignorato se non è impostato né il campo **httpProxy** né il campo **httpsProxy**.

3. Salvare il manifest come **migration-controller.yaml**.
4. Applicare il manifest aggiornato:

```
$ oc replace -f migration-controller.yaml -n openshift-migration
```

11.2.4. Eseguire la migrazione di un'applicazione tramite l'API MTC

È possibile eseguire la migrazione di un'applicazione dalla riga di comando tramite l'API Migration Toolkit for Containers (MTC).

Procedura

1. Creare il manifest della risorsa personalizzata **MigCluster** per il cluster host:

```
$ cat << EOF | oc apply -f -
apiVersion: migration.openshift.io/v1alpha1
kind: MigCluster
metadata:
  name: <host_cluster>
  namespace: openshift-migration
spec:
  isHostCluster: true
EOF
```

2. Creare il manifest dell'oggetto **Secret** per ogni cluster remoto:

```
$ cat << EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: <cluster_secret>
  namespace: openshift-config
type: Opaque
```

```
data:
  saToken: <sa_token> 1
EOF
```

- 1 Specificare il token dell'account di servizio (SA) **migration-controller** con codifica base64 del cluster remoto. È possibile ottenere il token eseguendo il seguente comando:

```
$ oc sa get-token migration-controller -n openshift-migration | base64 -w 0
```

3. Creare il manifest della risorsa personalizzata **MigCluster** per ogni cluster remoto:

```
$ cat << EOF | oc apply -f -
apiVersion: migration.openshift.io/v1alpha1
kind: MigCluster
metadata:
  name: <remote_cluster> 1
  namespace: openshift-migration
spec:
  exposedRegistryPath: <exposed_registry_route> 2
  insecure: false 3
  isHostCluster: false
  serviceAccountSecretRef:
    name: <remote_cluster_secret> 4
    namespace: openshift-config
  url: <remote_cluster_url> 5
EOF
```

- 1 Specificare la risorsa personalizzata **Cluster** del cluster remoto.
- 2 Facoltativo: per la migrazione diretta delle immagini, specificare il percorso del registro esposto.
- 3 La verifica SSL è abilitata se **false**. I certificati CA non sono richiesti o controllati se **true**.
- 4 Specificare l'oggetto **Secret** del cluster remoto.
- 5 Specificare l'URL del cluster remoto.

4. Verificare che tutti i cluster siano nello stato **Ready**:

```
$ oc describe cluster <cluster>
```

5. Creare il manifest dell'oggetto **Secret** per il repository di replica:

```
$ cat << EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  namespace: openshift-config
  name: <migstorage_creds>
type: Opaque
data:
```

```
aws-access-key-id: <key_id_base64> 1
aws-secret-access-key: <secret_key_base64> 2
EOF
```

- 1 Specificare l'ID della chiave in formato base64.
- 2 Specificare la chiave segreta in formato base64.

Le credenziali di AWS sono con codifica base64 per impostazione predefinita. Per gli altri provider di storage, è necessario codificare le proprie credenziali eseguendo il seguente comando con ogni chiave:

```
$ echo -n "<key>" | base64 -w 0 1
```

- 1 Specificare l'ID della chiave o la chiave segreta. Entrambe le chiavi devono essere con codifica base64.

6. Creare il manifest della risorsa personalizzata **MigStorage** per il repository di replica:

```
$ cat << EOF | oc apply -f -
apiVersion: migration.openshift.io/v1alpha1
kind: MigStorage
metadata:
  name: <migstorage>
  namespace: openshift-migration
spec:
  backupStorageConfig:
    awsBucketName: <bucket> 1
    credsSecretRef:
      name: <storage_secret> 2
      namespace: openshift-config
  backupStorageProvider: <storage_provider> 3
  volumeSnapshotConfig:
    credsSecretRef:
      name: <storage_secret> 4
      namespace: openshift-config
  volumeSnapshotProvider: <storage_provider> 5
EOF
```

- 1 Specificare il nome del bucket.
- 2 Specificare la risorsa personalizzata **Secrets** dello storage a oggetti. È necessario assicurarsi che le credenziali memorizzate nella risorsa personalizzata **Secrets** dello storage a oggetti siano corrette.
- 3 Specificare il provider di storage.
- 4 Facoltativo: se si copiano i dati usando gli snapshot, specificare la risorsa personalizzata **Secrets** dello storage a oggetti. È necessario assicurarsi che le credenziali memorizzate nella risorsa personalizzata **Secrets** dello storage a oggetti siano corrette.
- 5 Facoltativo: se si copiano i dati usando gli snapshot, specificare il provider di storage.

7. Verificare che la risorsa personalizzata **MigStorage** sia nello stato **Ready**:

```
$ oc describe migstorage <migstorage>
```

8. Creare il manifest della risorsa personalizzata **MigPlan**:

```
$ cat << EOF | oc apply -f -
apiVersion: migration.openshift.io/v1alpha1
kind: MigPlan
metadata:
  name: <migplan>
  namespace: openshift-migration
spec:
  destMigClusterRef:
    name: <host_cluster>
    namespace: openshift-migration
  indirectImageMigration: true 1
  indirectVolumeMigration: true 2
  migStorageRef:
    name: <migstorage> 3
    namespace: openshift-migration
  namespaces:
    - <source_namespace_1> 4
    - <source_namespace_2>
    - <source_namespace_3>:<destination_namespace> 5
  srcMigClusterRef:
    name: <remote_cluster> 6
    namespace: openshift-migration
EOF
```

- 1** La migrazione diretta delle immagini è abilitata se **false**.
- 2** La migrazione diretta dei volumi è abilitata se **false**.
- 3** Specificare il nome dell'istanza della risorsa personalizzata **MigStorage**.
- 4** Specificare uno o più spazi dei nomi di origine. Per impostazione predefinita, lo spazio dei nomi di destinazione ha lo stesso nome.
- 5** Specificare uno spazio dei nomi di destinazione se è diverso dallo spazio dei nomi di origine.
- 6** Specificare il nome dell'istanza **MigCluster** del cluster di origine.

9. Verificare che l'istanza **MigPlan** sia nello stato **Ready**:

```
$ oc describe migplan <migplan> -n openshift-migration
```

10. Creare il manifest della risorsa personalizzata **MigMigration** per avviare la migrazione definita nell'istanza **MigPlan**:

```
$ cat << EOF | oc apply -f -
apiVersion: migration.openshift.io/v1alpha1
```

```

kind: MigMigration
metadata:
  name: <migmigration>
  namespace: openshift-migration
spec:
  migPlanRef:
    name: <migplan> 1
    namespace: openshift-migration
  quiescePods: true 2
  stage: false 3
  rollback: false 4
EOF

```

- 1 Specificare il nome della risorsa personalizzata **MigPlan**.
- 2 I pod sul cluster di origine vengono fermati prima della migrazione, se **true**.
- 3 Una migrazione di fase, che copia la maggior parte dei dati senza fermare l'applicazione, viene eseguita se **true**.
- 4 Se **true**, una migrazione completata viene annullata.

11. Verificare la migrazione guardando il progresso della risorsa personalizzata **MigMigration**:

```
$ oc watch migmigration <migmigration> -n openshift-migration
```

L'output assomiglia al seguente:

Esempio di output

```

Name:      c8b034c0-6567-11eb-9a4f-0bc004db0fbc
Namespace: openshift-migration
Labels:    migration.openshift.io/migplan-name=django
Annotations: openshift.io/touch: e99f9083-6567-11eb-8420-0a580a81020c
API Version: migration.openshift.io/v1alpha1
Kind:      MigMigration
...
Spec:
  Mig Plan Ref:
    Name:      migplan
    Namespace: openshift-migration
  Stage:      false
Status:
  Conditions:
    Category:      Advisory
    Last Transition Time: 2021-02-02T15:04:09Z
    Message:        Step: 19/47
    Reason:         InitialBackupCreated
    Status:         True
    Type:           Running
    Category:      Required
    Last Transition Time: 2021-02-02T15:03:19Z
    Message:        The migration is ready.
    Status:        True

```

```

Type:          Ready
Category:      Required
Durable:       true
Last Transition Time: 2021-02-02T15:04:05Z
Message:       The migration registries are healthy.
Status:        True
Type:          RegistriesHealthy
Itinerary:     Final
Observed Digest:
7fae9d21f15979c71ddc7dd075cb97061895caac5b936d92fae967019ab616d5
Phase:         InitialBackupCreated
Pipeline:
  Completed: 2021-02-02T15:04:07Z
  Message:    Completed
  Name:       Prepare
  Started:    2021-02-02T15:03:18Z
  Message:    Waiting for initial Velero backup to complete.
  Name:       Backup
  Phase:      InitialBackupCreated
  Progress:
    Backup openshift-migration/c8b034c0-6567-11eb-9a4f-0bc004db0fbc-wpc44: 0 out of
    estimated total of 0 objects backed up (5s)
  Started:    2021-02-02T15:04:07Z
  Message:    Not started
  Name:       StageBackup
  Message:    Not started
  Name:       StageRestore
  Message:    Not started
  Name:       DirectImage
  Message:    Not started
  Name:       DirectVolume
  Message:    Not started
  Name:       Restore
  Message:    Not started
  Name:       Cleanup
Start Timestamp: 2021-02-02T15:03:18Z
Events:
Type Reason Age          From          Message
---- -
Normal Running 57s          migmigration_controller Step: 2/47
Normal Running 57s          migmigration_controller Step: 3/47
Normal Running 57s (x3 over 57s) migmigration_controller Step: 4/47
Normal Running 54s          migmigration_controller Step: 5/47
Normal Running 54s          migmigration_controller Step: 6/47
Normal Running 52s (x2 over 53s) migmigration_controller Step: 7/47
Normal Running 51s (x2 over 51s) migmigration_controller Step: 8/47
Normal Ready 50s (x12 over 57s) migmigration_controller The migration is ready.
Normal Running 50s          migmigration_controller Step: 9/47
Normal Running 50s          migmigration_controller Step: 10/47

```

11.2.5. Migrazione dello stato

È possibile eseguire migrazioni ripetibili, solo dello stato, utilizzando il Migration Toolkit for Containers (MTC) per eseguire la migrazione delle richieste di volumi permanenti (PVC) che costituiscono lo stato di un'applicazione. Si esegue la migrazione delle PVC specificate escludendo altre PVC dal piano di

migrazione. I dati dei volumi permanenti (PV) vengono copiati nel cluster di destinazione. I riferimenti dei PV non vengono spostati. I pod dell'applicazione continuano a funzionare sul cluster di origine. È possibile mappare le PVC per assicurarsi che le PVC di origine e di destinazione siano sincronizzate.

È possibile eseguire una migrazione un tantum degli oggetti Kubernetes che costituiscono lo stato di un'applicazione.

Se è presente una pipeline CI/CD, è possibile eseguire la migrazione dei componenti stateless distribuendoli sul cluster di destinazione. Quindi è possibile eseguire la migrazione dei componenti stateful tramite MTC.

È possibile eseguire una migrazione dello stato tra cluster o all'interno dello stesso cluster.



IMPORTANTE

La migrazione dello stato trasferisce solo i componenti che costituiscono lo stato di un'applicazione. Per eseguire la migrazione di un intero spazio dei nomi, utilizzare la migrazione di fase o completa.

Risorse aggiuntive

- Vedere [Escludere le PVC dalla migrazione](#) per selezionare le PVC per la migrazione dello stato.
- Vedere [Mapping delle PVC](#) per migrare i dati PV di origine alle PVC con provisioning sul cluster di destinazione.
- Vedere [Migrazione degli oggetti Kubernetes](#) per eseguire la migrazione degli oggetti Kubernetes che costituiscono lo stato di un'applicazione.

11.3. HOOK DI MIGRAZIONE

È possibile aggiungere fino a quattro hook di migrazione a un singolo piano di migrazione, con ogni hook in esecuzione in una diversa fase della migrazione. Gli hook di migrazione eseguono compiti come la personalizzazione della inattività delle applicazioni, la migrazione manuale dei tipi di dati non supportati e l'aggiornamento delle applicazioni dopo la migrazione.

Un hook di migrazione viene eseguito su un cluster di origine o di destinazione in uno dei seguenti stadi della migrazione:

- **PreBackup:** prima di eseguire il backup delle risorse sul cluster di origine.
- **PostBackup:** dopo il backup delle risorse sul cluster di origine.
- **PreRestore:** prima che le risorse siano ripristinate sul cluster di destinazione.
- **PostRestore:** dopo il ripristino delle risorse sul cluster di destinazione.

È possibile creare un hook creando un Ansible Playbook che viene eseguito con l'immagine predefinita di Ansible o con un container degli hook personalizzato.

Ansible Playbook

L'Ansible Playbook è montato su un container degli hook come mappa di configurazione. Il container degli hook viene eseguito come un processo, utilizzando il cluster, l'account di servizio e lo spazio dei nomi specificati nella risorsa personalizzata **MigPlan**. Il processo continua a essere eseguito fino a

quando non raggiunge il limite predefinito di 6 tentativi o viene completato correttamente. Questo continua anche se il pod iniziale viene eliminato o terminato.

L'immagine di runtime predefinita di Ansible è **registry.redhat.io/rhmtc/openshift-migration-hook-runner-rhel7:1.6**. Questa immagine è basata sull'immagine di Ansible Runner e include **python-openshift** per le risorse Kubernetes Ansible e un file binario **oc** aggiornato.

Container degli hook personalizzato

È possibile usare un container degli hook personalizzato invece dell'immagine predefinita di Ansible.

11.3.1. Scrivere un Ansible Playbook per un hook di migrazione

È possibile scrivere un Ansible Playbook da usare come hook di migrazione. L'hook viene aggiunto a un piano di migrazione usando la console web di MTC o specificando i valori per i parametri **spec.hooks** nel manifest della risorsa personalizzata **MigPlan**.

L'Ansible Playbook è montato su un container degli hook come mappa di configurazione. Il container degli hook viene eseguito come un processo, utilizzando il cluster, l'account di servizio e lo spazio dei nomi specificati nella risorsa personalizzata **MigPlan**. Il container degli hook utilizza un token di account di servizio specificato in modo che i compiti non richiedano l'autenticazione prima di essere eseguiti nel cluster.

11.3.1.1. Moduli Ansible

È possibile usare il modulo Ansible **shell** per eseguire i comandi **oc**.

Modulo shell di esempio

```
- hosts: localhost
gather_facts: false
tasks:
- name: get pod name
  shell: oc get po --all-namespaces
```

È possibile usare i moduli **kubernetes.core**, come **k8s_info**, per interagire con le risorse di Kubernetes.

Modulo k8s_facts di esempio

```
- hosts: localhost
gather_facts: false
tasks:
- name: Get pod
  k8s_info:
    kind: pods
    api: v1
    namespace: openshift-migration
    name: "{{ lookup('env', 'HOSTNAME') }}"
    register: pods

- name: Print pod name
  debug:
    msg: "{{ pods.resources[0].metadata.name }}"
```

È possibile usare il modulo **fail** per generare uno stato di uscita non zero in casi in cui uno stato di uscita

non zero non sarebbe normalmente generato, assicurando così di rilevare la riuscita o l'esito negativo di un hook. Gli hook vengono eseguiti come processi e lo stato di successo o esito negativo di un hook è basato sullo stato di uscita del container dei processi.

Modulo fail di esempio

```
- hosts: localhost
gather_facts: false
tasks:
- name: Set a boolean
  set_fact:
    do_fail: true

- name: "fail"
  fail:
    msg: "Cause a failure"
  when: do_fail
```

11.3.1.2. Variabili d'ambiente

Il nome della risorsa personalizzata **MigPlan** e gli spazi dei nomi di migrazione vengono passati come variabili d'ambiente al container degli hook. Si accede a queste variabili usando il plug-in **lookup**.

Esempio di variabili d'ambiente

```
- hosts: localhost
gather_facts: false
tasks:
- set_fact:
  namespaces: "{{ (lookup('env', 'migration_namespaces')).split(',') }}"

- debug:
  msg: "{{ item }}"
  with_items: "{{ namespaces }}"

- debug:
  msg: "{{ lookup('env', 'migplan_name') }}"
```

11.4. OPZIONI DEL PIANO DI MIGRAZIONE

È possibile escludere, modificare e mappare i componenti nella risorsa personalizzata (CR) **MigPlan**.

11.4.1. Escludere risorse

È possibile escludere risorse, per esempio, flussi di immagini, volumi permanenti (PV) o sottoscrizioni, da un piano di migrazione di Migration Toolkit for Containers (MTC) per ridurre il carico di risorse per la migrazione o per eseguire la migrazione di immagini o PV con uno strumento diverso.

Per impostazione predefinita, MTC esclude dalla migrazione le risorse del catalogo dei servizi e le risorse dell'Operator Lifecycle Manager (OLM). Queste risorse fanno parte del gruppo API del catalogo dei servizi e del gruppo API OLM, nessuno dei quali è supportato per la migrazione in questo momento.

Procedura

1. Modificare il manifest delle risorse personalizzate di **MigrationController**:

```
$ oc edit migrationcontroller <migration_controller> -n openshift-migration
```

2. Aggiornare la sezione **spec** aggiungendo un parametro per escludere risorse specifiche o aggiungendo una risorsa al parametro **excluded_resources** se non ha il proprio parametro di esclusione:

```
apiVersion: migration.openshift.io/v1alpha1
kind: MigrationController
metadata:
  name: migration-controller
  namespace: openshift-migration
spec:
  disable_image_migration: true 1
  disable_pv_migration: true 2
  ...
  excluded_resources: 3
  - imagedtags
  - templateinstances
  - clusterserviceversions
  - packagemanifests
  - subscriptions
  - servicebrokers
  - servicebindings
  - serviceclasses
  - serviceinstances
  - serviceplans
  - operatorgroups
  - events
  - events.events.k8s.io
```

- 1 Aggiungere **disable_image_migration: true** per escludere i flussi di immagini dalla migrazione. Non modificare il parametro **excluded_resources.imagestreams** viene aggiunto a **excluded_resources** quando il pod **MigrationController** si riavvia.
- 2 Aggiungere **disable_pv_migration: true** per escludere i PV dal piano di migrazione. Non modificare il parametro **excluded_resources.persistentvolumes** e **persistentvolumeclaims** sono aggiunti a **excluded_resources** quando il pod **MigrationController** si riavvia. Disabilitando la migrazione dei PV si disabilita anche il rilevamento dei PV quando si crea il piano di migrazione.
- 3 È possibile aggiungere le risorse di OpenShift Container Platform all'elenco **excluded_resources**. Non eliminare le risorse escluse predefinite. Queste risorse sono problematiche da migrare e devono essere escluse.

3. Attendere due minuti che il pod **MigrationController** si riavvii in modo che le modifiche siano applicate.
4. Verificare che la risorsa sia esclusa:

```
$ oc get deployment -n openshift-migration migration-controller -o yaml | grep EXCLUDED_RESOURCES -A1
```

L'output contiene le risorse escluse:

Esempio di output

```
- name: EXCLUDED_RESOURCES
  value:
```

```
imagetags,templateinstances,clusterserviceversions,packagemanifests,subscriptions,servicebro
ers,servicebindings,serviceclasses,serviceinstances,serviceplans,imagestreams,persistentvolun
es,persistentvolumeclaims
```

11.4.2. Mapping degli spazi dei nomi

Se si mappano gli spazi dei nomi nella risorsa personalizzata **MigPlan**, è necessario assicurarsi che gli spazi dei nomi non siano duplicati sui cluster di origine o di destinazione perché gli intervalli UID e GID degli spazi dei nomi vengono copiati durante la migrazione.

Due spazi dei nomi di origine mappati allo stesso spazio dei nomi di destinazione

```
spec:
  namespaces:
    - namespace_2
    - namespace_1:namespace_2
```

Per mappare uno spazio dei nomi di origine su uno spazio dei nomi con lo stesso nome, non è necessario creare un mapping. Per impostazione predefinita, uno spazio dei nomi di origine e uno spazio dei nomi di destinazione hanno lo stesso nome.

Mapping dello spazio dei nomi non corretto

```
spec:
  namespaces:
    - namespace_1:namespace_1
```

Riferimento corretto allo spazio dei nomi

```
spec:
  namespaces:
    - namespace_1
```

11.4.3. Escludere le richieste di volumi permanenti

Si selezionano le richieste di volumi permanenti (PVC) per la migrazione dello stato escludendo le PVC di cui non si desidera eseguire la migrazione. Si escludono le PVC impostando il parametro **spec.persistentVolumes.pvc.selection.action** della risorsa personalizzata **MigPlan** (CR) dopo che i volumi permanenti (PV) sono stati rilevati.

Requisiti

- La risorsa personalizzata **MigPlan** è nello stato **Ready**.

Procedura

- Aggiungere il parametro **spec.persistentVolumes.pvc.selection.action** alla risorsa personalizzata **MigPlan** e impostarlo su **skip**:

```

apiVersion: migration.openshift.io/v1alpha1
kind: MigPlan
metadata:
  name: <migplan>
  namespace: openshift-migration
spec:
  ...
  persistentVolumes:
  - capacity: 10Gi
    name: <pv_name>
    pvc:
  ...
  selection:
    action: skip

```

11.4.4. Mapping delle richieste di volumi permanenti

È possibile eseguire la migrazione dei dati dei volumi permanenti (PV) dal cluster di origine alle richieste di volumi permanenti (PVC) che sono già con provisioning nel cluster di destinazione nella risorsa personalizzata **MigPlan** mappando le PVC. Questo mapping assicura che le PVC di destinazione delle applicazioni migrate siano sincronizzate con le PVC di origine.

Si mappano le PVC aggiornando il parametro **spec.persistentVolumes.pvc.name** nella risorsa personalizzata **MigPlan** dopo che i PV sono stati rilevati.

Requisiti

- La risorsa personalizzata **MigPlan** è nello stato **Ready**.

Procedura

- Aggiornare il parametro **spec.persistentVolumes.pvc.name** nella risorsa personalizzata **MigPlan**:

```

apiVersion: migration.openshift.io/v1alpha1
kind: MigPlan
metadata:
  name: <migplan>
  namespace: openshift-migration
spec:
  ...
  persistentVolumes:
  - capacity: 10Gi
    name: <pv_name>
    pvc:
      name: <source_pvc>:<destination_pvc> 1

```

- 1** Specificare la PVC sul cluster di origine e la PVC sul cluster di destinazione. Se la PVC di destinazione non esiste, verrà creata. È possibile usare questo mapping per cambiare il nome della PVC durante la migrazione.

11.4.5. Modifica degli attributi dei volumi permanenti

Dopo aver creato una risorsa personalizzata (CR) **MigPlan**, la risorsa personalizzata **MigrationController** rileva i volumi permanenti (PV). Il blocco **spec.persistentVolumes** e il blocco **status.destStorageClasses** sono aggiunti alla risorsa personalizzata **MigPlan**.

È possibile modificare i valori nel blocco **spec.persistentVolumes.selection**. Se si cambiano i valori al di fuori del blocco **spec.persistentVolumes.selection**, i valori vengono sovrascritti quando la risorsa personalizzata **MigPlan** viene riconciliata dalla risorsa personalizzata **MigrationController**.

NOTA

Il valore predefinito per il parametro **spec.persistentVolumes.selection.storageClass** è determinato dalla seguente logica:

1. Se il PV del cluster di origine è Gluster o NFS, il valore predefinito è **cephfs**, per **accessMode: ReadWriteMany**, o **cephrbd**, per **accessMode: ReadWriteOnce**.
2. Se il PV non è né Gluster né NFS o se **cephfs** o **cephrbd** non sono disponibili, il valore predefinito è una classe di storage per lo stesso provisioner.
3. Se una classe di storage per lo stesso provisioner non è disponibile, il valore predefinito è la classe di storage predefinita del cluster di destinazione.

È possibile cambiare il valore **storageClass** con il valore di qualsiasi parametro **name** nel blocco **status.destStorageClasses** della risorsa personalizzata **MigPlan**.

Se il valore **storageClass** è vuoto, il PV non avrà alcuna classe di storage dopo la migrazione. Questa opzione è appropriata se, per esempio, si desidera spostare il PV su un volume NFS sul cluster di destinazione.

Requisiti

- La risorsa personalizzata **MigPlan** è nello stato **Ready**.

Procedura

- Modificare i valori **spec.persistentVolumes.selection** nella risorsa personalizzata **MigPlan**:

```
apiVersion: migration.openshift.io/v1alpha1
kind: MigPlan
metadata:
  name: <migplan>
  namespace: openshift-migration
spec:
  persistentVolumes:
    - capacity: 10Gi
      name: pvc-095a6559-b27f-11eb-b27f-021bddcaf6e4
      proposedCapacity: 10Gi
  pvc:
    accessModes:
      - ReadWriteMany
    hasReference: true
    name: mysql
    namespace: mysql-persistent
  selection:
```

```

action: <copy> 1
copyMethod: <filesystem> 2
verify: true 3
storageClass: <gp2> 4
accessMode: <ReadWriteMany> 5
storageClass: cephfs

```

- 1 I valori consentiti sono **move**, **copy** e **skip**. Se è supportata una sola azione, il valore predefinito è l'azione supportata. Se sono supportate più azioni, il valore predefinito è **copy**.
- 2 I valori consentiti sono **snapshot** e **filesystem**. Il valore predefinito è **filesystem**.
- 3 Viene visualizzato il parametro **verify** se si seleziona l'opzione di verifica per la copia del file system nella console web di MTC. È possibile impostarlo su **false**.
- 4 È possibile cambiare il valore predefinito con il valore di qualsiasi parametro **name** nel blocco **status.destStorageClasses** della risorsa personalizzata **MigPlan**. Se non viene specificato alcun valore, il PV non avrà alcuna classe di storage dopo la migrazione.
- 5 I valori consentiti sono **ReadWriteOnce** e **ReadWriteMany**. Se questo valore non è specificato, il valore predefinito è la modalità di accesso della PVC del cluster di origine. È possibile modificare la modalità di accesso solo nella risorsa personalizzata **MigPlan**. Non è possibile modificarla usando la console web di MTC.

Risorse aggiuntive

- Per i dettagli sulle azioni **move** e **copy**, vedere [Workflow di MTC](#).
- Per i dettagli sull'azione **skip**, vedere [Escludere i PVC dalla migrazione](#).
- Per i dettagli sui metodi di copia del file system e degli snapshot, vedere [Informazioni sui metodi di copia dei dati](#).

11.4.6. Eseguire la migrazione di oggetti Kubernetes

È possibile eseguire una migrazione una tantum degli oggetti Kubernetes che costituiscono lo stato di un'applicazione.



NOTA

Dopo la migrazione, il parametro **closed** della risorsa personalizzata **MigPlan** è impostato su **true**. Non è possibile creare un'altra risorsa personalizzata **MigMigration** per questa risorsa personalizzata **MigPlan**.

Si aggiungono oggetti Kubernetes alla risorsa personalizzata **MigPlan** utilizzando una delle seguenti opzioni:

- Aggiungere gli oggetti Kubernetes alla sezione **includedResources**.
- Usare il parametro **labelSelector** per fare riferimento a oggetti Kubernetes etichettati.
- Aggiungere oggetti Kubernetes alla sezione **includedResources** e poi filtrarli con il parametro **labelSelector**, per esempio, le risorse **Secret** e **ConfigMap** con l'etichetta **app: frontend**.

Procedura

- Aggiornare la risorsa personalizzata **MigPlan**:

```

apiVersion: migration.openshift.io/v1alpha1
kind: MigPlan
metadata:
  name: <migplan>
  namespace: openshift-migration
spec:
  includedResources:
    - kind: <kind> 1
      group: ""
    - kind: <kind>
      group: ""
  ...
  labelSelector:
    matchLabels:
      <label> 2

```

- 1 Specificare l'oggetto Kubernetes, per esempio, **Secret** o **ConfigMap**.
- 2 Specificare l'etichetta delle risorse di cui eseguire la migrazione, per esempio, **app: frontend**.

11.5. OPZIONI DEL CONTROLLER DI MIGRAZIONE

È possibile modificare i limiti del piano di migrazione, abilitare il ridimensionamento dei volumi permanenti o abilitare i client Kubernetes nella cache nella risorsa personalizzata **MigrationController** per migrazioni di grandi dimensioni e migliori prestazioni.

11.5.1. Aumentare i limiti per le migrazioni di grandi dimensioni

È possibile aumentare i limiti degli oggetti di migrazione e delle risorse del container per migrazioni di grandi dimensioni con il Migration Toolkit for Containers (MTC).



IMPORTANTE

È necessario testare queste modifiche prima di eseguire una migrazione in un ambiente di produzione.

Procedura

1. Modificare il manifest della risorsa personalizzata **MigrationController**:

```
$ oc edit migrationcontroller -n openshift-migration
```

2. Aggiornare i seguenti parametri:

```

...
mig_controller_limits_cpu: "1" 1
mig_controller_limits_memory: "10Gi" 2
...

```

```

mig_controller_requests_cpu: "100m" 3
mig_controller_requests_memory: "350Mi" 4
...
mig_pv_limit: 100 5
mig_pod_limit: 100 6
mig_namespace_limit: 10 7
...

```

- 1 Specificare il numero di CPU disponibili per la risorsa personalizzata **MigrationController**.
- 2 Specificare la quantità di memoria disponibile nella risorsa personalizzata **MigrationController**.
- 3 Specificare il numero di unità CPU disponibili per le richieste della risorsa personalizzata **MigrationController**. **100m** rappresenta 0,1 unità CPU (100 * 1e-3).
- 4 Specificare la quantità di memoria disponibile per le richieste della risorsa personalizzata **MigrationController**.
- 5 Specificare il numero di volumi permanenti che possono essere migrati.
- 6 Specificare il numero di pod che possono essere migrati.
- 7 Specificare il numero di spazi dei nomi che possono essere migrati.

3. Creare un piano di migrazione che utilizza i parametri aggiornati per verificare le modifiche. Se il piano di migrazione supera i limiti della risorsa personalizzata **MigrationController**, la console MTC mostra un messaggio di avvertimento quando si salva il piano di migrazione.

11.5.2. Abilitare il ridimensionamento dei volumi permanenti per la migrazione diretta dei volumi

È possibile abilitare il ridimensionamento dei volumi permanenti (PV) per la migrazione diretta dei volumi per evitare di esaurire lo spazio su disco sul cluster di destinazione.

Quando l'utilizzo del disco di un PV raggiunge un livello configurato, la risorsa personalizzata **MigrationController** confronta la capacità di storage richiesta di una richiesta di volume permanente (PVC) con la sua effettiva capacità con provisioning. Quindi, calcolare lo spazio richiesto sul cluster di destinazione.

Un parametro **pv_resizing_threshold** determina quando viene usato il ridimensionamento dei PV. La soglia predefinita è il **3%**. Questo significa che il ridimensionamento dei PV avviene quando l'utilizzo del disco di un PV è superiore al **97%**. È possibile aumentare questa soglia in modo che il ridimensionamento dei PV avvenga a un livello di utilizzo del disco inferiore.

La capacità delle PVC è calcolata secondo i seguenti criteri:

- Se la capacità di storage richiesta (**spec.resources.requests.storage**) delle PVC non è uguale alla sua capacità effettiva con provisioning(**status.capacity.storage**), viene usato il valore maggiore.
- Se un PV è fornito attraverso una PVC e poi successivamente cambiato in modo che le sue capacità PV e PVC non corrispondano più, viene utilizzato il valore maggiore.

Requisiti

- Le PVC devono essere collegate a uno o più pod in esecuzione in modo che la risorsa personalizzata **MigrationController** possa eseguire i comandi.

Procedura

- Accedere al cluster dell'host.
- Abilitare il ridimensionamento dei PV applicando una patch alla risorsa personalizzata **MigrationController**:

```
$ oc patch migrationcontroller migration-controller -p '{"spec":
{"enable_dvm_pv_resizing":true}}' \ 1
--type='merge' -n openshift-migration
```

- Impostare il valore su **false** per disabilitare il ridimensionamento dei PV.

- Facoltativo: aggiornare il parametro **pv_resizing_threshold** per aumentare la soglia:

```
$ oc patch migrationcontroller migration-controller -p '{"spec":{"pv_resizing_threshold":41}}' \
1
--type='merge' -n openshift-migration
```

- Il valore predefinito è **3**.

Quando la soglia viene superata, il seguente messaggio di stato viene visualizzato nello stato della risorsa personalizzata **MigPlan**:

```
status:
  conditions:
  ...
  - category: Warn
    durable: true
    lastTransitionTime: "2021-06-17T08:57:01Z"
    message: 'Capacity of the following volumes will be automatically adjusted to avoid disk
capacity issues in the target cluster: [pvc-b800eb7b-cf3b-11eb-a3f7-0eae3e0555f3]'
    reason: Done
    status: "False"
    type: PvCapacityAdjustmentRequired
```



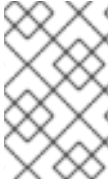
NOTA

Per lo storage AWS gp2, questo messaggio non appare a meno che il valore **pv_resizing_threshold** non sia pari o superiore al 42% a causa del modo in cui gp2 calcola l'utilizzo e la dimensione del volume. ([BZ#1973148](#))

11.5.3. Abilitare i client Kubernetes nella cache

È possibile abilitare i client Kubernetes nella cache nella risorsa personalizzata **MigrationController** per migliorare le prestazioni durante la migrazione. Il maggior beneficio in termini di prestazioni viene mostrato quando si esegue la migrazione tra cluster in aree geografiche diverse o con una latenza di rete

significativa.



NOTA

I compiti delegati, per esempio, il backup Rsync per la migrazione diretta dei volumi o il backup e il ripristino Velero, tuttavia, non mostrano prestazioni migliorate con i client nella cache.

I client in cache richiedono memoria extra perché la risorsa personalizzata **MigrationController** mette in cache tutte le risorse API che sono richieste per interagire con le risorse personalizzate **MigCluster**. Le richieste che sono normalmente inviate al server API sono invece dirette alla cache. La cache controlla il server API per gli aggiornamenti.

È possibile aumentare i limiti di memoria e le richieste della risorsa personalizzata **MigrationController** se si verificano errori **OOMKilled** dopo aver abilitato i client nella cache.

Procedura

1. Abilitare i client nella cache eseguendo il seguente comando:

```
$ oc -n openshift-migration patch migrationcontroller migration-controller --type=json --patch \
'[{ "op": "replace", "path": "/spec/mig_controller_enable_cache", "value": true}]'
```

2. Facoltativo: aumentare i limiti di memoria della risorsa personalizzata **MigrationController** eseguendo il seguente comando:

```
$ oc -n openshift-migration patch migrationcontroller migration-controller --type=json --patch \
'[{ "op": "replace", "path": "/spec/mig_controller_limits_memory", "value": <10Gi>}]'
```

3. Facoltativo: aumentare le richieste di memoria della risorsa personalizzata **MigrationController** eseguendo il seguente comando:

```
$ oc -n openshift-migration patch migrationcontroller migration-controller --type=json --patch \
'[{ "op": "replace", "path": "/spec/mig_controller_requests_memory", "value": <350Mi>}]'
```

CAPITOLO 12. RISOLUZIONE DEI PROBLEMI

Questa sezione descrive le risorse per la risoluzione dei problemi del Migration Toolkit for Containers (MTC).

Per i problemi noti, vedere le [note di rilascio di MTC](#).

12.1. WORKFLOW DI MTC

È possibile eseguire la migrazione delle risorse di Kubernetes, i dati dei volumi permanenti e le immagini container interne a OpenShift Container Platform 4.10 utilizzando la console web di Migration Toolkit for Containers (MTC) o l'API di Kubernetes.

MTC esegue la migrazione delle seguenti risorse:

- Uno spazio dei nomi specificato in un piano di migrazione.
- Risorse con spazio dei nomi: quando MTC esegue la migrazione di uno spazio dei nomi, vengono migrati tutti gli oggetti e le risorse associate a quello spazio dei nomi, come i servizi o i pod. Inoltre, se una risorsa che esiste nello spazio dei nomi ma non a livello di cluster dipende da una risorsa che esiste a livello di cluster, MTC esegue la migrazione di entrambe le risorse. Per esempio, un security context constraint (SCC) è una risorsa che esiste a livello di cluster e un service account (SA) è una risorsa che esiste a livello di spazio dei nomi. Se un SA esiste in uno spazio dei nomi di cui MTC esegue la migrazione, MTC individua automaticamente tutti gli SCC che sono collegati al SA ed esegue la migrazione anche di questi. Allo stesso modo, MTC esegue la migrazione delle richieste di volumi permanenti che sono legate ai volumi permanenti dello spazio dei nomi.



NOTA

Potrebbe essere necessario eseguire la migrazione delle risorse con cluster manualmente, a seconda della risorsa.

- Risorse personalizzate (CR) e definizioni di risorse personalizzate (CRD): MTC esegue automaticamente la migrazione delle CR e delle CRD a livello di spazio dei nomi.

La migrazione di un'applicazione con la console web di MTC comporta i seguenti passaggi:

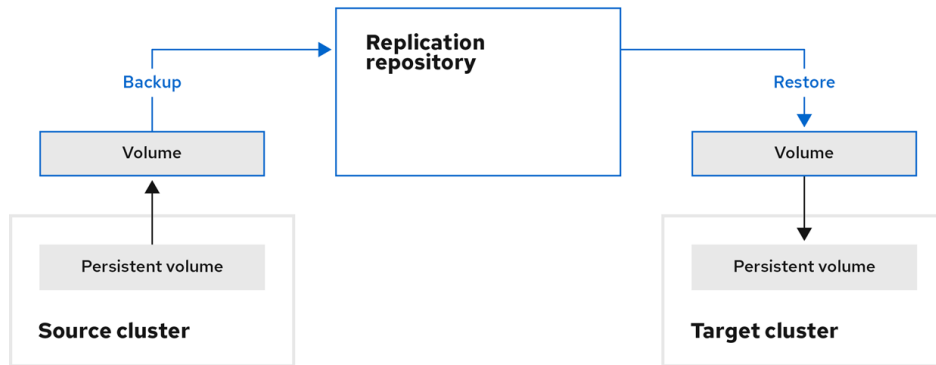
1. Installare il Migration Toolkit for Containers Operator su tutti i cluster.
È possibile installare il Migration Toolkit for Containers Operator in un ambiente ristretto con accesso Internet limitato o assente. I cluster di origine e di destinazione devono avere accesso di rete tra loro e a un registro speculare.
2. Configurare il repository di replica, uno storage a oggetti intermedio che MTC usa per eseguire la migrazione dei dati.
I cluster di origine e di destinazione devono avere accesso di rete al repository di replica durante la migrazione. Se si utilizza un server proxy, è necessario configurarlo per permettere il traffico di rete tra il repository di replica e i cluster.
3. Aggiungere il cluster di origine alla console web di MTC.
4. Aggiungere il repository di replica alla console web di MTC.
5. Creare un piano di migrazione, con una delle seguenti opzioni di migrazione dei dati:

- **Copia:** MTC copia i dati dal cluster di origine al repository di replica e dal repository di replica al cluster di destinazione.



NOTA

Se si utilizza la migrazione diretta delle immagini o la migrazione diretta dei volumi, le immagini o i volumi sono copiati direttamente dal cluster di origine al cluster di destinazione.



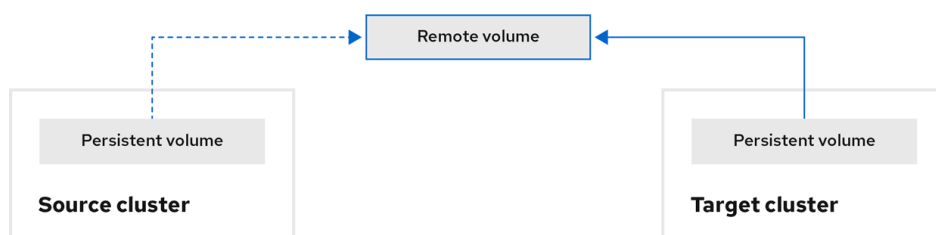
OpenShift_45_1019

- **Spostamento:** MTC smonta un volume remoto, per esempio NFS, dal cluster di origine, crea una risorsa PV sul cluster di destinazione che punta al volume remoto, e poi monta il volume remoto sul cluster di destinazione. Le applicazioni in esecuzione sul cluster di destinazione utilizzano lo stesso volume remoto utilizzato dal cluster di origine. Il volume remoto deve essere accessibile ai cluster di origine e di destinazione.



NOTA

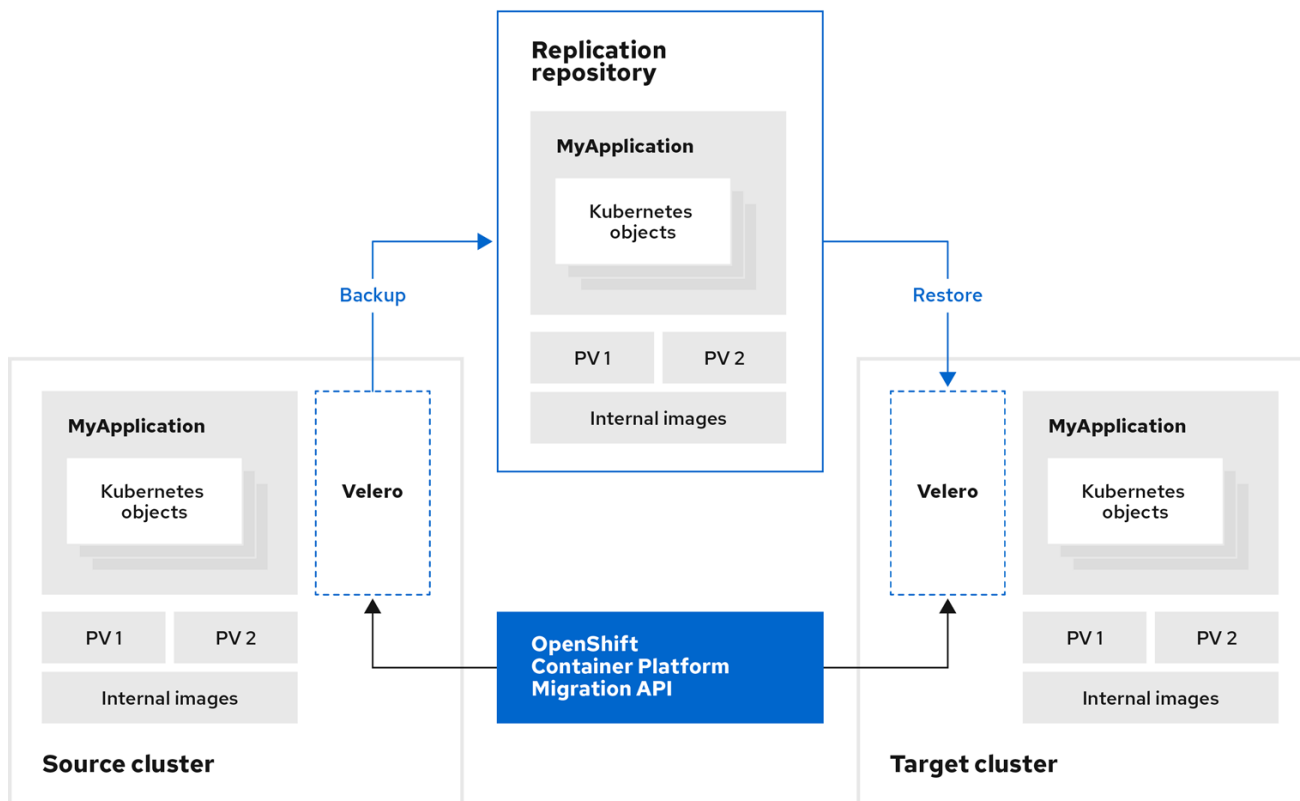
Anche se il repository di replica non appare in questo diagramma, è necessario per la migrazione.



OpenShift_45_1019

6. Eseguire il piano di migrazione, con una delle seguenti opzioni:

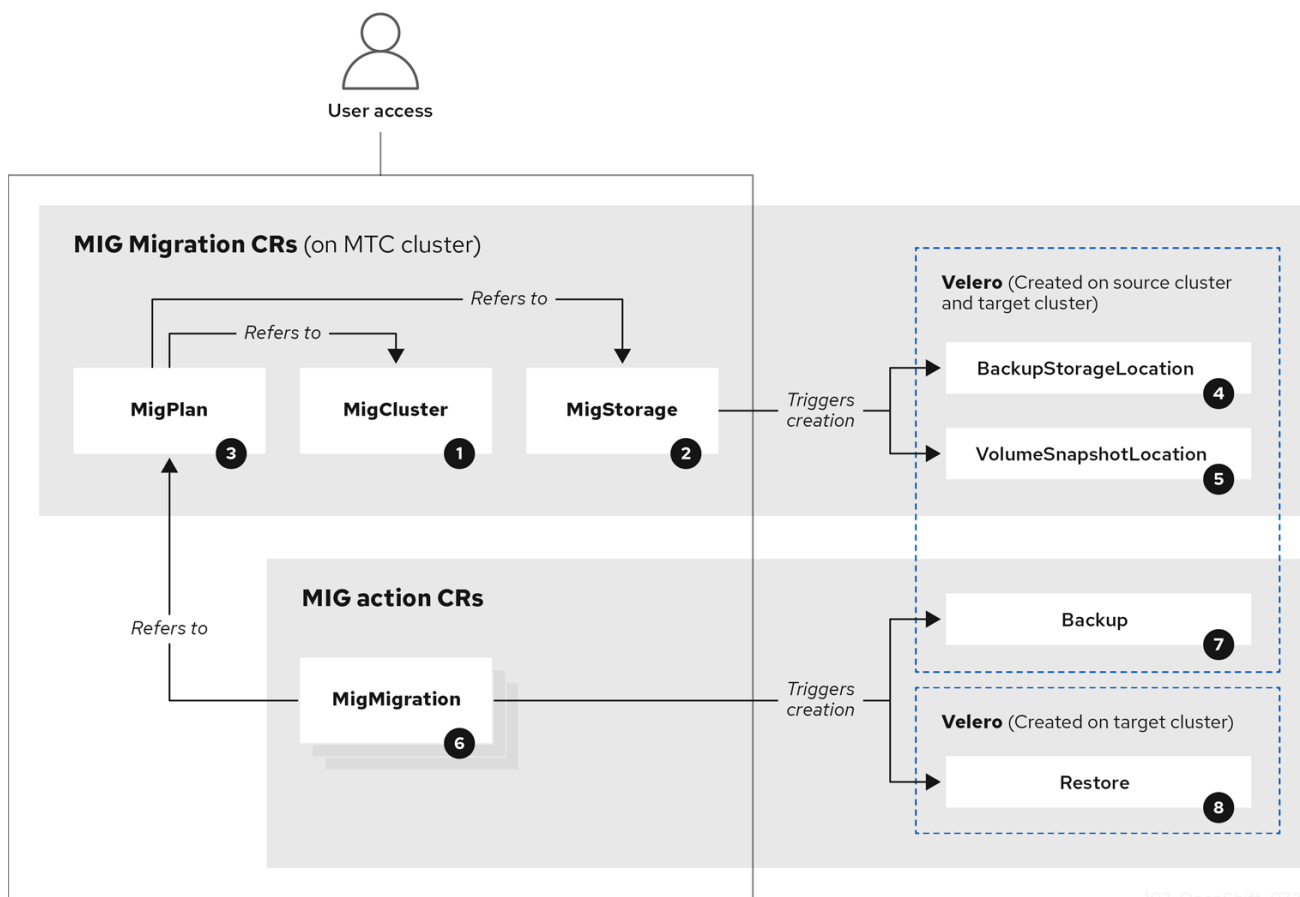
- Con la migrazione a **fasi** viene eseguita la migrazione dei dati nel cluster di destinazione senza fermare l'applicazione.
Una migrazione di fase può essere eseguita più volte in modo che la maggior parte dei dati sia copiata nella destinazione prima della migrazione. L'esecuzione di una o più migrazioni di fase riduce la durata della migrazione completa.
- La migrazione **completa** ferma l'applicazione sul cluster di origine e sposta le risorse sul cluster di destinazione.
Facoltativo: è possibile deselezionare la casella di controllo **Halt transactions (Arresta transazioni)** sul cluster di origine durante la migrazione.



OpenShift_45_1019

Informazioni sulle risorse personalizzate di MTC

Il Migration Toolkit for Containers (MTC) crea le seguenti risorse personalizzate (CR):



102_OpenShift_0720

- 1 **MigCluster** (configurazione, cluster MTC): Definizione del cluster
- 2 **MigStorage** (configurazione, cluster MTC): Definizione dello storage
- 3 **MigPlan** (configurazione, cluster MTC): piano di migrazione

La risorsa personalizzata **MigPlan** descrive i cluster di origine e di destinazione, il repository di replica e gli spazi dei nomi da migrare. È associata a 0, 1, o molte risorse personalizzate **MigMigration**.



NOTA

L'eliminazione di una risorsa personalizzata **MigPlan** elimina le risorse personalizzate **MigMigration** associate.

- 4 **BackupStorageLocation** (configurazione, cluster MTC): posizione degli oggetti di backup **Velero**
- 5 **VolumeSnapshotLocation** (configurazione, cluster MTC): posizione degli snapshot dei volumi **Velero**
- 6 **MigMigration** (azione, cluster MTC): migrazione, creata ogni volta che si organizzano o si migrano i dati. Ogni risorsa personalizzata **MigMigration** è associata a una risorsa personalizzata **MigPlan**.
- 7 **Backup** (azione, cluster di origine): quando si esegue un piano di migrazione, la risorsa personalizzata **MigMigration** crea due risorse personalizzate di backup **Velero** su ogni cluster di origine:
 - Risorsa personalizzata di backup n.1 per gli oggetti Kubernetes
 - Risorsa personalizzata di backup n.2 per i dati dei PV
- 8 **Restore** (azione, cluster di destinazione): quando si esegue un piano di migrazione, la risorsa personalizzata **MigMigration** crea due risorse personalizzate di ripristino **Velero** sul cluster di destinazione:
 - Risorsa personalizzata di ripristino n.1 (usando la risorsa personalizzata di backup n.2) per i dati dei PV
 - Risorsa personalizzata di ripristino n.2 (usando la risorsa personalizzata di backup n.1) per gli oggetti Kubernetes

12.2. MANIFEST DELLE RISORSE PERSONALIZZATE DI MTC

Migration Toolkit for Containers (MTC) utilizza i seguenti file manifest di risorse personalizzate (CR) per la migrazione delle applicazioni.

12.2.1. DirectImageMigration

La risorsa personalizzata **DirectImageMigration** copia le immagini direttamente dal cluster di origine al cluster di destinazione.

apiVersion: migration.openshift.io/v1alpha1


```

kind: DirectImageMigration
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: <direct_image_migration>
spec:
  srcMigClusterRef:
    name: <source_cluster>
    namespace: openshift-migration
  destMigClusterRef:
    name: <destination_cluster>
    namespace: openshift-migration
  namespaces: ❶
    - <source_namespace_1>
    - <source_namespace_2>:<destination_namespace_3> ❷

```

- ❶ Uno o più spazi di nomi contenenti immagini da migrare. Per impostazione predefinita, lo spazio dei nomi di destinazione ha lo stesso nome dello spazio dei nomi di origine.
- ❷ Spazio dei nomi di origine mappato su uno spazio dei nomi di destinazione con un nome diverso.

12.2.2. DirectImageStreamMigration

La risorsa personalizzata **DirectImageStreamMigration** copia i riferimenti ai flussi di immagini direttamente dal cluster di origine al cluster destinazione.

```

apiVersion: migration.openshift.io/v1alpha1
kind: DirectImageStreamMigration
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: <direct_image_stream_migration>
spec:
  srcMigClusterRef:
    name: <source_cluster>
    namespace: openshift-migration
  destMigClusterRef:
    name: <destination_cluster>
    namespace: openshift-migration
  imageStreamRef:
    name: <image_stream>
    namespace: <source_image_stream_namespace>
  destNamespace: <destination_image_stream_namespace>

```

12.2.3. DirectVolumeMigration

La risorsa personalizzata **DirectVolumeMigration** copia i volumi permanenti (PV) direttamente dal cluster di origine al cluster di destinazione.

```

apiVersion: migration.openshift.io/v1alpha1
kind: DirectVolumeMigration
metadata:
  name: <direct_volume_migration>

```

```

namespace: openshift-migration
spec:
  createDestinationNamespaces: false 1
  deleteProgressReportingCRs: false 2
  destMigClusterRef:
    name: <host_cluster> 3
    namespace: openshift-migration
  persistentVolumeClaims:
  - name: <pvc> 4
    namespace: <pvc_namespace>
  srcMigClusterRef:
    name: <source_cluster>
    namespace: openshift-migration

```

- 1** Impostare su **true** per creare spazi dei nomi per i PV sul cluster di destinazione.
- 2** Impostare su **true** per eliminare le risorse personalizzate **DirectVolumeMigrationProgress** dopo la migrazione. Il valore predefinito è **false** in modo che le risorse personalizzate **DirectVolumeMigrationProgress** siano conservate per la risoluzione dei problemi.
- 3** Aggiornare il nome del cluster se il cluster di destinazione non è il cluster host.
- 4** Specificare uno o più PVC di cui eseguire la migrazione.

12.2.4. DirectVolumeMigrationProgress

La risorsa personalizzata **DirectVolumeMigrationProgress** mostra il progresso della risorsa personalizzata **DirectVolumeMigration**.

```

apiVersion: migration.openshift.io/v1alpha1
kind: DirectVolumeMigrationProgress
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: <direct_volume_migration_progress>
spec:
  clusterRef:
    name: <source_cluster>
    namespace: openshift-migration
  podRef:
    name: <rsync_pod>
    namespace: openshift-migration

```

12.2.5. MigAnalytic

La risorsa personalizzata **MigAnalytic** raccoglie il numero di immagini, le risorse Kubernetes e la capacità dei volumi permanenti (PV) da una risorsa personalizzata **MigPlan** associata.

È possibile configurare i dati che raccoglie.

```

apiVersion: migration.openshift.io/v1alpha1
kind: MigAnalytic
metadata:

```

```

annotations:
  migplan: <migplan>
name: <miganalytic>
namespace: openshift-migration
labels:
  migplan: <migplan>
spec:
  analyzeImageCount: true 1
  analyzeK8SResources: true 2
  analyzePVCapacity: true 3
  listImages: false 4
  listImagesLimit: 50 5
  migPlanRef:
    name: <migplan>
    namespace: openshift-migration

```

- 1 Facoltativo: restituisce il numero di immagini.
- 2 Facoltativo: restituisce il numero, il tipo e la versione API delle risorse Kubernetes.
- 3 Facoltativo: restituisce la capacità PV.
- 4 Restituisce un elenco di nomi di immagini. Il valore predefinito è **false** in modo che l'output non sia eccessivamente lungo.
- 5 Facoltativo: specifica il numero massimo di nomi di immagini da restituire se **listImages** è **true**.

12.2.6. MigCluster

La risorsa personalizzata **MigCluster** definisce un cluster host, locale o remoto.

```

apiVersion: migration.openshift.io/v1alpha1
kind: MigCluster
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: <host_cluster> 1
  namespace: openshift-migration
spec:
  isHostCluster: true 2
  # The 'azureResourceGroup' parameter is relevant only for Microsoft Azure.
  azureResourceGroup: <azure_resource_group> 3
  caBundle: <ca_bundle_base64> 4
  insecure: false 5
  refresh: false 6
  # The 'restartRestic' parameter is relevant for a source cluster.
  restartRestic: true 7
  # The following parameters are relevant for a remote cluster.
  exposedRegistryPath: <registry_route> 8
  url: <destination_cluster_url> 9
  serviceAccountSecretRef:
    name: <source_secret> 10
    namespace: openshift-config

```

- 1 Aggiorna il nome del cluster se il pod **migration-controller** non è in esecuzione su questo cluster.
- 2 Il pod **migration-controller** viene eseguito su questo cluster, se è **true**.
- 3 Solo Microsoft Azure: specificare il gruppo di risorse.
- 4 Facoltativo: se si è creato un bundle di certificati per certificati CA autofirmati e se il valore del parametro **insecure** è **false**, specificare il bundle di certificati con codifica base64.
- 5 Impostare su **true** per disabilitare la verifica SSL.
- 6 Impostare su **true** per convalidare il cluster.
- 7 Impostare su **true** per riavviare i pod di **Restic** sul cluster di origine dopo la creazione dei pod **Stage**.
- 8 Solo cluster remoto e migrazione diretta delle immagini: specificare il percorso esposto del registro sicuro.
- 9 Solo cluster remoto: specificare l'URL.
- 10 Solo cluster remoto: specificare il nome dell'oggetto **Secret**.

12.2.7. MigHook

La risorsa personalizzata **MigHook** definisce un hook di migrazione che esegue codice personalizzato in una fase specifica della migrazione. È possibile creare fino a hook ganci di migrazione. Ogni hook viene eseguito durante una diversa fase della migrazione.

È possibile configurare il nome dell'hook, la durata di esecuzione, un'immagine personalizzata e il cluster dove l'hook verrà eseguito.

Le fasi di migrazione e gli spazi dei nomi degli hook sono configurati nella risorsa personalizzata **MigPlan**.

```

apiVersion: migration.openshift.io/v1alpha1
kind: MigHook
metadata:
  generateName: <hook_name_prefix> 1
  name: <mighook> 2
  namespace: openshift-migration
spec:
  activeDeadlineSeconds: 1800 3
  custom: false 4
  image: <hook_image> 5
  playbook: <ansible_playbook_base64> 6
  targetCluster: source 7

```

- 1 Facoltativo: un hash unico viene aggiunto al valore di questo parametro in modo che ogni hook di migrazione abbia un nome unico. Non è necessario specificare il valore del parametro **name**.
- 2 Specificare il nome dell'hook di migrazione, a meno che non si specifichi il valore del parametro **generateName**.

- 3 Facoltativo: specificare il numero massimo di secondi in cui un hook può essere eseguito. Il valore predefinito è **1800**.
- 4 L'hook è un'immagine personalizzata, se è **true**. L'immagine personalizzata può includere Ansible o può essere scritta in un diverso linguaggio di programmazione.
- 5 Specificare l'immagine personalizzata, per esempio, **quay.io/konveyor/hook-runner:latest**. Richiesto se **custom** è **true**.
- 6 Ansible Playbook con codifica base64. Richiesto se **custom** è **false**.
- 7 Specificare il cluster su cui verrà eseguito l'hook. I valori validi sono **source** o **destination**.

12.2.8. MigMigration

La risorsa personalizzata **MigMigration** esegue una risorsa personalizzata **MigPlan**.

È possibile configurare una risorsa personalizzata **Migmigration** per eseguire una migrazione di fase o incrementale, per annullare una migrazione in corso, o per annullare una migrazione completata.

```

apiVersion: migration.openshift.io/v1alpha1
kind: MigMigration
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: <migmigration>
  namespace: openshift-migration
spec:
  canceled: false 1
  rollback: false 2
  stage: false 3
  quiescePods: true 4
  keepAnnotations: true 5
  verify: false 6
  migPlanRef:
    name: <migplan>
    namespace: openshift-migration

```

- 1 Impostare su **true** per annullare una migrazione in corso.
- 2 Impostare su **true** per annullare una migrazione completata.
- 3 Impostare su **true** per eseguire una migrazione di fase. I dati vengono copiati in modo incrementale e i pod sul cluster di origine non vengono fermati.
- 4 Impostare su **true** per fermare l'applicazione durante la migrazione. I pod sul cluster di origine sono ridimensionati a **0** dopo la fase di **backup**.
- 5 Impostare su **true** per mantenere le etichette e le annotazioni applicate durante la migrazione.
- 6 Impostare su **true** per controllare lo stato dei pod migrati sul cluster di destinazione e per restituire i nomi dei pod che non sono nello stato **Running**.

12.2.9. MigPlan

La risorsa personalizzata **MigPlan** definisce i parametri di un piano di migrazione.

È possibile configurare gli spazi dei nomi di destinazione, le fasi degli hook e la migrazione diretta o indiretta.



NOTA

Per impostazione predefinita, uno spazio dei nomi di destinazione ha lo stesso nome dello spazio dei nomi di origine. Se si configura un diverso spazio dei nomi di destinazione, è necessario assicurarsi che gli spazi dei nomi non siano duplicati sui cluster di origine o di destinazione perché gli intervalli UID e GID sono copiati durante la migrazione.

```

apiVersion: migration.openshift.io/v1alpha1
kind: MigPlan
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: <migplan>
  namespace: openshift-migration
spec:
  closed: false 1
  srcMigClusterRef:
    name: <source_cluster>
    namespace: openshift-migration
  destMigClusterRef:
    name: <destination_cluster>
    namespace: openshift-migration
  hooks: 2
  - executionNamespace: <namespace> 3
    phase: <migration_phase> 4
    reference:
      name: <hook> 5
      namespace: <hook_namespace> 6
      serviceAccount: <service_account> 7
  indirectImageMigration: true 8
  indirectVolumeMigration: false 9
  migStorageRef:
    name: <migstorage>
    namespace: openshift-migration
  namespaces:
    - <source_namespace_1> 10
    - <source_namespace_2>
    - <source_namespace_3>:<destination_namespace_4> 11
  refresh: false 12

```

- 1 Se il valore è **true**, la migrazione è stata completata. Non è possibile creare un'altra risorsa personalizzata **MigMigration** per questa risorsa personalizzata **MigPlan**.
- 2 Facoltativo: è possibile specificare fino a quattro hook di migrazione. Ogni hook deve essere eseguito durante una diversa fase di migrazione.

- 3 Facoltativo: specificare lo spazio dei nomi in cui l'hook verrà eseguito.
- 4 Facoltativo: specificare la fase di migrazione durante la quale un hook viene eseguito. Un hook può essere assegnato a una fase. I valori validi sono **PreBackup**, **PostBackup**, **PreRestore** e **PostRestore**.
- 5 Facoltativo: specificare il nome della risorsa personalizzata **MigHook**.
- 6 Facoltativo: specificare lo spazio dei nomi della risorsa personalizzata **MigHook**.
- 7 Facoltativo: specificare un account di servizio con privilegi di **cluster-admin**.
- 8 La migrazione diretta dell'immagine è disabilitata se **true**. Le immagini sono copiate dal cluster di origine al repository di replica e dal repository di replica al cluster di destinazione.
- 9 La migrazione diretta del volume è disabilitata se **true**. I PV sono copiati dal cluster di origine al repository di replica e dal repository di replica al cluster di destinazione.
- 10 Specificare uno o più spazi dei nomi di origine. Se si specifica solo lo spazio dei nomi di origine, lo spazio dei nomi di destinazione è lo stesso.
- 11 Specificare lo spazio dei nomi di destinazione se è diverso da quello di origine.
- 12 La risorsa personalizzata **MigPlan** è convalidata se è **true**.

12.2.10. MigStorage

La risorsa personalizzata **MigStorage** descrive lo storage a oggetti per il repository di replica.

Sono supportati Amazon Web Services (AWS), Microsoft Azure, Google Cloud Storage, Multi-Cloud Object Gateway, e il cloud storage generico compatibile con S3.

AWS e il metodo di copia degli snapshot hanno parametri aggiuntivi.

```

apiVersion: migration.openshift.io/v1alpha1
kind: MigStorage
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: <migstorage>
  namespace: openshift-migration
spec:
  backupStorageProvider: <backup_storage_provider> 1
  volumeSnapshotProvider: <snapshot_storage_provider> 2
  backupStorageConfig:
    awsBucketName: <bucket> 3
    awsRegion: <region> 4
    credsSecretRef:
      namespace: openshift-config
      name: <storage_secret> 5
    awsKmsKeyId: <key_id> 6
    awsPublicUrl: <public_url> 7
    awsSignatureVersion: <signature_version> 8
  volumeSnapshotConfig:

```

```
awsRegion: <region> 9
credsSecretRef:
  namespace: openshift-config
  name: <storage_secret> 10
refresh: false 11
```

- 1** Specificare il provider di storage.
- 2** Solo per il metodo di copia degli snapshot: specificare il provider di storage.
- 3** Solo AWS: specificare il nome del bucket.
- 4** Solo AWS: specificare l'area geografica del bucket, per esempio, **us-east-1**.
- 5** Specificare il nome dell'oggetto **Secret** creato per lo storage.
- 6** Solo AWS: se si utilizza il servizio di gestione delle chiavi di AWS, specificare l'identificatore unico della chiave.
- 7** Solo AWS: se si è concesso accesso pubblico al bucket AWS, specificare l'URL del bucket.
- 8** Solo AWS: specificare la versione della firma AWS per autenticare le richieste al bucket, per esempio, **4**.
- 9** Solo metodo di copia degli snapshot: specificare l'area geografica dei cluster.
- 10** Solo metodo di copia degli snapshot: specificare il nome dell'oggetto **Secret** creato per lo storage.
- 11** Impostare su **true** per convalidare il cluster.

12.3. REGISTRI E STRUMENTI DI DEBUGGING

Questa sezione descrive i registri e gli strumenti di debugging che è possibile utilizzare per la risoluzione dei problemi.

12.3.1. Visualizzare le risorse del piano di migrazione

È possibile visualizzare le risorse del piano di migrazione per monitorare una migrazione in corso o per risolvere i problemi di una migrazione non riuscita utilizzando la console web di MTC e l'interfaccia della riga di comando (CLI).


Procedura

1. Nella console web di MTC, cliccare su **Migration Plans**.
2. Cliccare sul numero **Migrations** accanto a un piano di migrazione per visualizzare la pagina **Migrations**.
3. Cliccare su una migrazione per visualizzare i **dettagli della migrazione**.
4. Espandere **Migration resources** per visualizzare le risorse di migrazione e il loro stato in una vista ad albero.



NOTA

Per risolvere una migrazione non riuscita, iniziare con una risorsa di alto livello non riuscita e poi procedere lungo l'albero delle risorse verso le risorse di livello inferiore.

5. Cliccare sul menu Opzioni  accanto a una risorsa e selezionare una delle seguenti opzioni:
- L'opzione **Copy oc describe command** copia il comando negli appunti.
 - Accedere al cluster interessato e poi eseguire il comando.
Le condizioni e gli eventi della risorsa sono visualizzati in formato YAML.
 - L'opzione **Copy oc logs command** copia il comando negli appunti.
 - Accedere al cluster interessato e poi eseguire il comando.
Se la risorsa supporta il filtraggio dei registri, viene visualizzato un registro filtrato.
 - **View JSON** visualizza i dati della risorsa in formato JSON in un browser web.
I dati sono gli stessi dell'output del comando **oc get <resource>**.

12.3.2. Visualizzare il registro di un piano di migrazione

È possibile visualizzare un registro aggregato per un piano di migrazione. Si usa la console web di MTC per copiare un comando negli appunti e poi eseguire il comando dall'interfaccia della riga di comando (CLI).

Il comando visualizza i registri filtrati dei seguenti pod:

- **Migration Controller**
- **Velero**
- **Restic**
- **Rsync**
- **Stunnel**
- **Registro**

Procedura

1. Nella console web di MTC, cliccare su **Migration Plans**.
2. Cliccare sul numero di **Migrations** accanto a un piano di migrazione.
3. Cliccare su **View logs**.
4. Cliccare sull'icona Copy per copiare il comando **oc logs** negli appunti.
5. Accedere al cluster pertinente e inserire il comando nella CLI.
Viene visualizzato il registro aggregato per il piano di migrazione.

12.3.3. Usare il lettore dei registri di migrazione

Puoi usare il lettore dei registri di migrazione per visualizzare una singola vista filtrata di tutti i registri di migrazione.

Procedura

1. Ottenere il pod **mig-log-reader**:

```
$ oc -n openshift-migration get pods | grep log
```

2. Inserire il seguente comando per visualizzare un singolo registro di migrazione:

```
$ oc -n openshift-migration logs -f <mig-log-reader-pod> -c color 1
```

1 L'opzione **-c plain** mostra il registro senza colori.

12.3.4. Accedere alle metriche delle prestazioni

La risorsa personalizzata **MigrationController** registra le metriche e le immette nello storage di monitoraggio on-cluster. È possibile interrogare le metriche usando Prometheus Query Language (PromQL) per diagnosticare i problemi di prestazioni della migrazione. Tutte le metriche vengono reimpostate quando il pod Migration Controller viene riavviato.

È possibile accedere alle metriche delle prestazioni ed eseguire query utilizzando la console web di OpenShift Container Platform.

Procedura

1. Nella console web di OpenShift Container Platform, cliccare su **Observe → Metrics**.
2. Inserire una query PromQL, selezionare una finestra temporale da visualizzare e cliccare su **Run Queries**.
Se il browser web non mostra tutti i risultati, usare la console Prometheus.

12.3.4.1. Metriche fornite

La risorsa personalizzata **MigrationController** fornisce metriche per il conteggio delle risorse personalizzate **MigMigration** e per le relative richieste API.

12.3.4.1.1. cam_app_workload_migrations

Questa metrica è un conteggio delle risorse personalizzate **MigMigration** nel tempo. È utile da visualizzare insieme alle metriche **mtc_client_request_count** e **mtc_client_request_elapsed** per confrontare le informazioni sulle richieste API con i cambiamenti di stato della migrazione. Questa metrica è inclusa in Telemetry.

Tabella 12.1. cam_app_workload_migrations metric

Nome dell'etichetta interrogabile	Valori dell'etichetta campione	Descrizione dell'etichetta
stato	in esecuzione, inattivo, non riuscito, completato	Stato della risorsa personalizzata MigMigration
tipo	fase, finale	Tipo della risorsa personalizzata MigMigration

12.3.4.1.2. mtc_client_request_count

Questa metrica è un conteggio cumulativo delle richieste API Kubernetes che **MigrationController** ha emesso. Non è inclusa in Telemetry.

Tabella 12.2. mtc_client_request_count metric

Nome dell'etichetta interrogabile	Valori dell'etichetta campione	Descrizione dell'etichetta
cluster	https://migcluster-url:443	Cluster per cui la richiesta è stata emessa
componente	MigPlan, MigCluster	API sub-controller che ha emesso la richiesta
funzione	(*ReconcileMigPlan).Reconcile	Funzione da cui la richiesta è stata emessa
tipo	SecretList, Deployment	Tipo di Kubernetes per cui la richiesta è stata emessa

12.3.4.1.3. mtc_client_request_elapsed

Questa metrica è una latenza cumulativa, in millisecondi, delle richieste API di Kubernetes che **MigrationController** ha emesso. Non è inclusa in Telemetry.

Tabella 12.3. mtc_client_request_elapsed metric

Nome dell'etichetta interrogabile	Valori dell'etichetta campione	Descrizione dell'etichetta
cluster	https://cluster-url.com:443	Cluster per cui la richiesta è stata emessa
componente	migplan, migcluster	API sub-controller che ha emesso la richiesta
funzione	(*ReconcileMigPlan).Reconcile	Funzione da cui la richiesta è stata emessa

Nome dell'etichetta interrogabile	Valori dell'etichetta campione	Descrizione dell'etichetta
tipo	SecretList, Deployment	Risorsa Kubernetes per cui è stata emessa la richiesta

12.3.4.1.4. Domande utili

La tabella elenca alcune query utili che possono essere utilizzate per monitorare le prestazioni.

Tabella 12.4. Domande utili

Query	Descrizione
mtc_client_request_count	Numero di richieste API emesse, ordinate per tipo di richiesta
sum(mtc_client_request_count)	Numero totale di richieste API emesse
mtc_client_request_elapsed	Latenza delle richieste API, ordinate per tipo di richiesta
sum(mtc_client_request_elapsed)	Latenza totale delle richieste API
sum(mtc_client_request_elapsed) / sum(mtc_client_request_count)	Latenza media delle richieste API
mtc_client_request_elapsed / mtc_client_request_count	Latenza media delle richieste API, ordinate per tipo di richiesta
cam_app_workload_migrations{status="running"} * 100	Conteggio delle migrazioni in esecuzione, moltiplicato per 100 per una più facile visualizzazione insieme al conteggio delle richieste

12.3.5. Usare lo strumento must-gather

È possibile raccogliere registri, metriche e informazioni sulle risorse personalizzate MTC utilizzando lo strumento **must-gather**.

I dati **must-gather** devono essere allegati a tutti i casi dei clienti.

È possibile raccogliere dati per un'ora o un periodo di 24 ore e visualizzare i dati con la console Prometheus.

Requisiti

- È necessario essere connessi al cluster di OpenShift Container Platform come utente con il ruolo **cluster-admin**.
- È necessario avere installato la CLI OpenShift (**oc**).

Procedura

1. Andare alla directory dove si desidera memorizzare i dati **must-gather**.
2. Eseguire il comando **oc adm must-gather** per una delle seguenti opzioni di raccolta dati:

- Per raccogliere i dati dell'ultima ora:

```
$ oc adm must-gather --image=registry.redhat.io/rhmtc/openshift-migration-must-gather-rhel8:v1.6
```

I dati sono salvati come **must-gather/must-gather.tar.gz**. È possibile caricare questo file in un caso di supporto sul [Red Hat Customer Portal](#).

- Per raccogliere i dati delle ultime 24 ore:

```
$ oc adm must-gather --image=registry.redhat.io/rhmtc/openshift-migration-must-gather-rhel8:v1.6 \
  -- /usr/bin/gather_metrics_dump
```

Questa operazione può richiedere molto tempo. I dati sono salvati come **must-gather/metrics/prom_data.tar.gz**.

Visualizzare i dati delle metriche con la console Prometheus

È possibile visualizzare i dati delle metriche con la console Prometheus.

Procedura

1. Decomprimere il file **prom_data.tar.gz**:

```
$ tar -xvzf must-gather/metrics/prom_data.tar.gz
```

2. Creare un'istanza locale di Prometheus:

```
$ make prometheus-run
```

Il comando emette l'URL di Prometheus.

Output

```
Started Prometheus on http://localhost:9090
```

3. Avviare un browser web e andare all'URL per visualizzare i dati utilizzando la console web di Prometheus.
4. Dopo aver visualizzato i dati, eliminare l'istanza Prometheus e i dati:

```
$ make prometheus-cleanup
```

12.3.6. Debugging delle risorse di Velero con lo strumento CLI di Velero

È possibile eseguire il debugging delle risorse personalizzate di **Backup** e **Restore** e recuperare i registri con lo strumento CLI di Velero.

Lo strumento CLI di Velero fornisce informazioni più dettagliate dello strumento CLI di OpenShift.

Syntax

Usare il comando **oc exec** per eseguire un comando CLI di Velero:

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) \
-- ./velero <backup_restore_cr> <command> <cr_name>
```

Esempio

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) \
-- ./velero backup describe 0e44ae00-5dc3-11eb-9ca8-df7e5254778b-2d8ql
```

È possibile specificare **velero-<pod> -n openshift-migration** al posto di **\$(oc get pods -n openshift-migration -o name | grep velero)**.

Esempio

```
$ oc exec velero-<pod> -n openshift-migration -- ./velero backup describe 0e44ae00-5dc3-11eb-9ca8-df7e5254778b-2d8ql
```

Opzione aiuto

Usare l'opzione **velero --help** per elencare tutti i comandi CLI di Velero:

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) -- ./velero --help
```

Comando descrizione

Usare il comando **velero describe** per recuperare un riassunto degli avvertimenti e degli errori associati a una risorsa personalizzata **Backup** o **Restore**:

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) \
-- ./velero <backup_restore_cr> describe <cr_name>
```

Esempio

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) \
-- ./velero backup describe 0e44ae00-5dc3-11eb-9ca8-df7e5254778b-2d8ql
```

Comando dei registri

Usare il comando **velero logs** per recuperare i registri di una risorsa personalizzata **Backup** o **Restore**:

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) \
-- ./velero <backup_restore_cr> logs <cr_name>
```

Esempio

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) \
-- ./velero restore logs ccc7c2d0-6017-11eb-afab-85d0007f5a19-x4lbf
```

12.3.7. Debugging di un errore parziale della migrazione

È possibile eseguire il debugging di un messaggio di avviso di errore parziale della migrazione usando la CLI di Velero per esaminare i registri della risorsa personalizzata **Restore**.

Un errore parziale si verifica quando Velero incontra un problema che non causa l'esito negativo della migrazione. Per esempio, se manca la definizione di una risorsa personalizzata (CRD) o se è presente una discrepanza tra le versioni CRD sui cluster di origine e di destinazione, la migrazione viene completata ma la risorsa personalizzata non viene creata sul cluster di destinazione.

Velero registra il problema come un errore parziale e poi elabora il resto degli oggetti nella risorsa personalizzata **Backup**.

Procedura

1. Controllare lo stato di una risorsa personalizzata **MigMigration**:

```
$ oc get migmigration <migmigration> -o yaml
```

Esempio di output

```
status:
  conditions:
  - category: Warn
    durable: true
    lastTransitionTime: "2021-01-26T20:48:40Z"
    message: 'Final Restore openshift-migration/ccc7c2d0-6017-11eb-afab-85d0007f5a19-
x4lbf: partially failed on destination cluster'
    status: "True"
    type: VeleroFinalRestorePartiallyFailed
  - category: Advisory
    durable: true
    lastTransitionTime: "2021-01-26T20:48:42Z"
    message: The migration has completed with warnings, please look at `Warn` conditions.
    reason: Completed
    status: "True"
    type: SucceededWithWarnings
```

2. Controllare lo stato della risorsa personalizzata **Restore** usando il comando Velero **describe**:

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) -n openshift-migration -
- ./velero restore describe <restore>
```

Esempio di output

```
Phase: PartiallyFailed (run 'velero restore logs ccc7c2d0-6017-11eb-afab-85d0007f5a19-
x4lbf' for more information)

Errors:
  Velero: <none>
  Cluster: <none>
  Namespaces:
    migration-example: error restoring example.com/migration-example/migration-example:
the server could not find the requested resource
```

3. Controllare i registri della risorsa personalizzata **Restore** usando il comando Velero **logs**:

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) -n openshift-migration -
- ./velero restore logs <restore>
```

Esempio di output

```
time="2021-01-26T20:48:37Z" level=info msg="Attempting to restore migration-example:
migration-example" logSource="pkg/restore/restore.go:1107" restore=openshift-
migration/ccc7c2d0-6017-11eb-afab-85d0007f5a19-x4lbf
time="2021-01-26T20:48:37Z" level=info msg="error restoring migration-example: the server
could not find the requested resource" logSource="pkg/restore/restore.go:1170"
restore=openshift-migration/ccc7c2d0-6017-11eb-afab-85d0007f5a19-x4lbf
```

Il messaggio di errore del registro della risorsa personalizzata **Restore, the server could not find the requested resource**, indica la causa della migrazione parzialmente non riuscita.

12.3.8. Utilizzare le risorse personalizzate di MTC per la risoluzione dei problemi

È possibile controllare le seguenti risorse personalizzate (CR) di Migration Toolkit for Containers (MTC) per risolvere i problemi di una migrazione non riuscita:

- **MigCluster**

- **MigStorage**

- **MigPlan**

- **BackupStorageLocation**

La risorsa personalizzata **BackupStorageLocation** contiene l'etichetta **migrationcontroller** per identificare l'istanza MTC che l'ha creata:

```
labels:
  migrationcontroller: ebe13bee-c803-47d0-a9e9-83f380328b93
```

- **VolumeSnapshotLocation**

La risorsa personalizzata **VolumeSnapshotLocation** contiene l'etichetta **migrationcontroller** per identificare l'istanza MTC che l'ha creata:

```
labels:
  migrationcontroller: ebe13bee-c803-47d0-a9e9-83f380328b93
```

- **MigMigration**

- **Backup**

MTC cambia il criterio di recupero dei volumi permanenti (PV) migrati in **Retain** sul cluster di destinazione. La risorsa personalizzata **Backup** contiene l'annotazione **openshift.io/orig-reclaim-policy** che indica il criterio di recupero originale. È possibile ripristinare manualmente il criterio di recupero dei PV migrati.

- **Restore**

Procedura

1. Elencare le risorse personalizzate **MigMigration** nello spazio dei nomi **openshift-migration**:

-


```
$ oc get migmigration -n openshift-migration
```

Esempio di output

```
NAME                                     AGE
88435fe0-c9f8-11e9-85e6-5d593ce65e10  6m42s
```

2. Ispezionare la risorsa personalizzata **MigMigration**:

```
$ oc describe migmigration 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
```

L'output è simile ai seguenti esempi.

Esempio di output di MigMigration

```
name:      88435fe0-c9f8-11e9-85e6-5d593ce65e10
namespace: openshift-migration
labels:    <none>
annotations: touch: 3b48b543-b53e-4e44-9d34-33563f0f8147
apiVersion: migration.openshift.io/v1alpha1
kind:      MigMigration
metadata:
  creationTimestamp: 2019-08-29T01:01:29Z
  generation:       20
  resourceVersion:  88179
  selfLink:         /apis/migration.openshift.io/v1alpha1/namespaces/openshift-
migration/migmigrations/88435fe0-c9f8-11e9-85e6-5d593ce65e10
  uid:              8886de4c-c9f8-11e9-95ad-0205fe66cbb6
spec:
  migPlanRef:
    name:      socks-shop-mig-plan
    namespace: openshift-migration
  quiescePods: true
  stage:      false
status:
  conditions:
    category:      Advisory
    durable:       True
    lastTransitionTime: 2019-08-29T01:03:40Z
    message:       The migration has completed successfully.
    reason:        Completed
    status:        True
    type:          Succeeded
  phase:          Completed
  startTimestamp: 2019-08-29T01:01:29Z
  events:         <none>
```

Esempio di output di backup della risorsa personalizzata n.2 Velero che descrive i dati PV

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
```

```

openshift.io/migrate-quiesce-pods: "true"
openshift.io/migration-registry: 172.30.105.179:5000
openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-44dd3bd5-c9f8-11e9-95ad-0205fe66cbb6
openshift.io/orig-reclaim-policy: delete
creationTimestamp: "2019-08-29T01:03:15Z"
generateName: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-
generation: 1
labels:
  app.kubernetes.io/part-of: migration
  migmigration: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
  migration-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
  velero.io/storage-location: myrepo-vpzq9
name: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7
namespace: openshift-migration
resourceVersion: "87313"
selfLink: /apis/velero.io/v1/namespaces/openshift-migration/backups/88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7
uid: c80dbbc0-c9f8-11e9-95ad-0205fe66cbb6
spec:
  excludedNamespaces: []
  excludedResources: []
  hooks:
    resources: []
  includeClusterResources: null
  includedNamespaces:
  - sock-shop
  includedResources:
  - persistentvolumes
  - persistentvolumeclaims
  - namespaces
  - imagestreams
  - imagestreamtags
  - secrets
  - configmaps
  - pods
  labelSelector:
    matchLabels:
      migration-included-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
  storageLocation: myrepo-vpzq9
  ttl: 720h0m0s
  volumeSnapshotLocations:
  - myrepo-wv6fx
status:
  completionTimestamp: "2019-08-29T01:02:36Z"
  errors: 0
  expiration: "2019-09-28T01:02:35Z"
  phase: Completed
  startTimestamp: "2019-08-29T01:02:35Z"
  validationErrors: null
  version: 1
  volumeSnapshotsAttempted: 0
  volumeSnapshotsCompleted: 0
  warnings: 0

```

Esempio di output di ripristino della risorsa personalizzata n.2 Velero che descrive le risorse Kubernetes

```

apiVersion: velero.io/v1
kind: Restore
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.90.187:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-36f54ca7-c925-11e9-825a-06fa9fb68c88
  creationTimestamp: "2019-08-28T00:09:49Z"
  generateName: e13a1b60-c927-11e9-9555-d129df7f3b96-
  generation: 3
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: e18252c9-c927-11e9-825a-06fa9fb68c88
    migration-final-restore: e18252c9-c927-11e9-825a-06fa9fb68c88
  name: e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
  namespace: openshift-migration
  resourceVersion: "82329"
  selfLink: /apis/velero.io/v1/namespaces/openshift-migration/restores/e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
  uid: 26983ec0-c928-11e9-825a-06fa9fb68c88
spec:
  backupName: e13a1b60-c927-11e9-9555-d129df7f3b96-sz24f
  excludedNamespaces: null
  excludedResources:
    - nodes
    - events
    - events.events.k8s.io
    - backups.velero.io
    - restores.velero.io
    - resticrepositories.velero.io
  includedNamespaces: null
  includedResources: null
  namespaceMapping: null
  restorePVs: true
status:
  errors: 0
  failureReason: ""
  phase: Completed
  validationErrors: null
  warnings: 15

```

12.4. PROBLEMI E PREOCCUPAZIONI COMUNI

Questa sezione descrive i problemi e le preoccupazioni comuni che possono presentarsi durante una migrazione.

12.4.1. Aggiornamento delle immagini interne deprecate

Se l'applicazione usa immagini dello spazio dei nomi **openshift**, le versioni richieste delle immagini devono essere presenti sul cluster di destinazione.

Se un'immagine di OpenShift Container Platform 3 è deprecata in OpenShift Container Platform 4.10, è possibile aggiornare manualmente il tag del flusso di immagini utilizzando **podman**.

Requisiti

- È necessario avere installato **podman**.
- È necessario aver eseguito l'accesso come utente con privilegi di **cluster-admin**.
- Se si utilizzano registri insicuri, aggiungere i valori dell'host del registro alla sezione **[registries.insecure]** di **/etc/container/registries.conf** per assicurarsi che **podman** non incontri un errore di verifica TLS.
- I registri interni devono essere esposti sui cluster di origine e di destinazione.

Procedura

1. Assicurarsi che i registri interni siano esposti sui cluster OpenShift Container Platform 3 e 4. Il registro interno è esposto su OpenShift Container Platform 4 per impostazione predefinita.
2. Se si utilizzano registri insicuri, aggiungere i valori dell'host del registro alla sezione **[registries.insecure]** di **/etc/container/registries.conf** per assicurarsi che **podman** non incontri un errore di verifica TLS.

3. Accedere al registro di OpenShift Container Platform 3:

```
$ podman login -u $(oc whoami) -p $(oc whoami -t) --tls-verify=false <registry_url>:<port>
```

4. Accedere al registro di OpenShift Container Platform 4:

```
$ podman login -u $(oc whoami) -p $(oc whoami -t) --tls-verify=false <registry_url>:<port>
```

5. Eseguire pull dell'immagine di OpenShift Container Platform 3:

```
$ podman pull <registry_url>:<port>/openshift/<image>
```

6. Etichettare l'immagine di OpenShift Container Platform 3 per il registro di OpenShift Container Platform 4:

```
$ podman tag <registry_url>:<port>/openshift/<image> \ 1  
  <registry_url>:<port>/openshift/<image> 2
```

1 Specificare l'URL del registro e la porta per il cluster di OpenShift Container Platform 3.

2 Specificare l'URL del registro e la porta per il cluster di OpenShift Container Platform 4.

7. Eseguire push dell'immagine nel registro di OpenShift Container Platform 4:

```
$ podman push <registry_url>:<port>/openshift/<image> 1
```

1 Specificare il cluster di OpenShift Container Platform 4.

8. Verificare che l'immagine abbia un flusso di immagini valido:

```
$ oc get imagestream -n openshift | grep <image>
```

Esempio di output

```
NAME      IMAGE REPOSITORY                                TAGS  UPDATED
my_image  image-registry.openshift-image-registry.svc:5000/openshift/my_image  latest 32
seconds ago
```

12.4.2. La migrazione diretta dei volumi non si completa

Se la migrazione diretta dei volumi non venisse completata, il cluster di destinazione potrebbe non avere le stesse annotazioni **node-selector** del cluster di origine.

Migration Toolkit for Containers (MTC) esegue la migrazione degli spazi dei nomi con tutte le annotazioni per preservare i vincoli del contesto di sicurezza e i requisiti di scheduling. Durante la migrazione diretta dei volumi, MTC crea pod di trasferimento Rsync sul cluster di destinazione negli spazi dei nomi che sono stati migrati dal cluster di origine. Se uno spazio dei nomi del cluster di destinazione non ha le stesse annotazioni dello spazio dei nomi del cluster di origine, i pod di trasferimento Rsync non possono essere programmati. I pod Rsync rimangono nello stato **Pending**.

È possibile identificare e risolvere questo problema eseguendo la seguente procedura.

Procedura

1. Controllare lo stato della risorsa personalizzata **MigMigration**:

```
$ oc describe migmigration <pod> -n openshift-migration
```

L'output include il seguente messaggio di stato:

Esempio di output

```
Some or all transfer pods are not running for more than 10 mins on destination cluster
```

2. Sul cluster di origine, ottenere i dettagli di uno spazio dei nomi migrato:

```
$ oc get namespace <namespace> -o yaml 1
```

- 1** Specificare lo spazio dei nomi migrato.

3. Sul cluster di destinazione, modificare lo spazio dei nomi migrato:

```
$ oc edit namespace <namespace>
```

4. Aggiungere le annotazioni mancanti **openshift.io/node-selector** allo spazio dei nomi migrato come nell'esempio seguente:

```
apiVersion: v1
kind: Namespace
metadata:
```

```

annotations:
  openshift.io/node-selector: "region=east"
...

```

5. Eseguire di nuovo il piano di migrazione.

12.4.3. Messaggi di errore e risoluzioni

Questa sezione descrive i messaggi di errore comuni che si possono incontrare con il Migration Toolkit for Containers (MTC) e come risolvere le cause alla base.

12.4.3.1. Errore del certificato CA visualizzato quando si accede alla console MTC per la prima volta

Se viene visualizzato il messaggio **CA certificate error** la prima volta che si cerca di accedere alla console MTC, ciò è dovuto probabilmente all'uso di certificati CA autofirmati in uno dei cluster.

Per risolvere questo problema, andare all'URL **oauth-authorization-server** visualizzato nel messaggio di errore e accettare il certificato. Per risolvere questo problema in modo permanente, aggiungere il certificato all'archivio attendibilità del browser web.

Se viene visualizzato il messaggio **Unauthorized** dopo aver accettato il certificato, andare alla console MTC e aggiornare la pagina web.

12.4.3.2. Errore di timeout OAuth nella console MTC

Se viene visualizzato il messaggio **connection has timed out** nella console MTC dopo aver accettato un certificato autofirmato, è probabile che le cause siano le seguenti:

- Accesso di rete interrotto al server OAuth
- Accesso di rete interrotto alla console di OpenShift Container Platform
- Configurazione proxy che blocca l'accesso all'URL **oauth-authorization-server**. Vedere [Console MTC inaccessibile a causa di un errore di timeout OAuth](#) per i dettagli.

Per determinare la causa del timeout:

- Ispezionare la pagina web della console MTC con una finestra di ispezione del browser web.
- Controllare il log del pod **Migration UI** per la presenza di errori.

12.4.3.3. Errore del certificato firmato da un'autorità sconosciuta

Se si utilizza un certificato autofirmato per proteggere un cluster o un repository di replica per Migration Toolkit for Containers (MTC), la verifica del certificato potrebbe avere esito negativo con il seguente messaggio di errore: **Certificate signed by unknown authority**.

È possibile creare un file bundle di certificato CA personalizzato e caricarlo nella console web di MTC quando si aggiunge un cluster o un repository di replica.

Procedura

Scaricare un certificato CA da un endpoint remoto e salvarlo come file bundle CA:

```
$ echo -n | openssl s_client -connect <host_FQDN>:<port> \ 1
| sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > <ca_bundle.cert> 2
```

- 1 Specificare l'host FQDN e la porta dell'endpoint, per esempio, **api.my-cluster.example.com:6443**.
- 2 Specificare il nome del file del bundle CA.

12.4.3.4. Errori di posizione dello storage di backup nel log del pod Velero

Se una risorsa personalizzata **Backup** di **Velero** contiene un riferimento a una posizione di storage di backup (BSL) che non esiste, il log del pod **Velero** potrebbe mostrare i seguenti messaggi di errore:

```
$ oc logs <MigrationUI_Pod> -n openshift-migration
```

È possibile ignorare questi messaggi di errore. Un BSL mancante non può causare l'esito negativo di una migrazione.

12.4.3.5. Errore di timeout del backup del volume del pod nel log del pod Velero

Se una migrazione non riesce perché si verifica il timeout di Restic, il seguente errore viene visualizzato nel log del pod **Velero**.

```
level=error msg="Error backing up item" backup=velero/monitoring error="timed out waiting for all
PodVolumeBackups to complete"
error.file="/go/src/github.com/heptio/velero/pkg/restic/backupper.go:165"
error.function="github.com/heptio/velero/pkg/restic.(*backupper).BackupPodVolumes" group=v1
```

Il valore predefinito di **restic_timeout** è un'ora. È possibile aumentare questo parametro per migrazioni di grandi dimensioni, tenendo presente che un valore più alto può ritardare il ritorno dei messaggi di errore.

Procedura

1. Nella console web di OpenShift Container Platform, andare a **Operators** → **Installed Operators**.
2. Cliccare su **Migration Toolkit for Containers Operator**.
3. Nella scheda **MigrationController**, cliccare su **migration-controller**.
4. Nella scheda **YAML**, aggiornare il valore del seguente parametro:

```
spec:
  restic_timeout: 1h 1
```

- 1 Le unità valide sono **h** (ore), **m** (minuti) e **s** (secondi), per esempio, **3h30m15s**.

5. Cliccare su **Save**.

12.4.3.6. Errori di verifica Restic nella risorsa personalizzata MigMigration

Se la verifica dei dati non riesce durante la migrazione di un volume permanente con il metodo di copia dei dati del file system, viene visualizzato il seguente errore nella risorsa personalizzata **MigMigration**.

Esempio di output

```
status:
  conditions:
  - category: Warn
    durable: true
    lastTransitionTime: 2020-04-16T20:35:16Z
    message: There were verify errors found in 1 Restic volume restores. See restore `registry-
example-migration-rvwcm`
    for details 1
    status: "True"
    type: ResticVerifyErrors 2
```

1 Il messaggio di errore identifica il nome della risorsa personalizzata **Restore**.

2 **ResticVerifyErrors** è un tipo di avviso di errore generale che include gli errori di verifica.



NOTA

Un errore di verifica dei dati non causa l'esito negativo del processo di migrazione.

È possibile controllare la risorsa personalizzata **Restore** per identificare l'origine dell'errore di verifica dei dati.

Procedura

1. Accedere al cluster di destinazione.
2. Visualizzare la risorsa personalizzata **Restore**:

```
$ oc describe <registry-example-migration-rvwcm> -n openshift-migration
```

L'output identifica il volume permanente con errori di **PodVolumeRestore**.

Esempio di output

```
status:
  phase: Completed
  podVolumeRestoreErrors:
  - kind: PodVolumeRestore
    name: <registry-example-migration-rvwcm-98t49>
    namespace: openshift-migration
  podVolumeRestoreResticErrors:
  - kind: PodVolumeRestore
    name: <registry-example-migration-rvwcm-98t49>
    namespace: openshift-migration
```

3. Visualizzare la risorsa personalizzata **PodVolumeRestore**:

```
$ oc describe <migration-example-rvwcm-98t49>
```


■

L'output identifica il pod **Restic** che ha registrato gli errori.

Esempio di output

```
completionTimestamp: 2020-05-01T20:49:12Z
errors: 1
resticErrors: 1
...
resticPod: <restic-nr2v5>
```

4. Visualizzare il log del pod **Restic** per individuare gli errori:

```
$ oc logs -f <restic-nr2v5>
```

12.4.3.7. Errore di autorizzazione di Restic durante la migrazione dallo storage NFS con `root_squash` abilitato

Se si esegue la migrazione dei dati dallo storage NFS e `root_squash` è abilitato, **Restic** esegue il mapping a `nfsnobody` e non ha l'autorizzazione a eseguire la migrazione. Il seguente errore viene visualizzato nel log del pod **Restic**.

Esempio di output

```
backup=openshift-migration/<backup_id> controller=pod-volume-backup error="fork/exec
/usr/bin/restic: permission denied" error.file="/go/src/github.com/vmware-
tanzu/velero/pkg/controller/pod_volume_backup_controller.go:280"
error.function="github.com/vmware-tanzu/velero/pkg/controller.
(*podVolumeBackupController).processBackup"
logSource="pkg/controller/pod_volume_backup_controller.go:280" name=<backup_id>
namespace=openshift-migration
```

È possibile risolvere questo problema creando un gruppo supplementare per Restic e aggiungendo l'ID del gruppo al manifest della risorsa personalizzata **MigrationController**.

Procedura

1. Creare un gruppo supplementare per Restic sullo storage NFS.
2. Impostare il bit `setgid` sulle directory NFS in modo che la proprietà del gruppo sia ereditata.
3. Aggiungere il parametro `restic_supplemental_groups` al manifest della risorsa personalizzata **MigrationController** sui cluster di origine e di destinazione:

```
spec:
  restic_supplemental_groups: <group_id> ❶
```

- ❶ Specificare l'ID del gruppo supplementare.

4. Attendere il riavvio dei pod di **Restic** in modo che le modifiche siano applicate.

12.4.4. Problemi noti

Questa release ha i seguenti problemi noti:

- Durante la migrazione, il Migration Toolkit for Containers (MTC) conserva le seguenti annotazioni degli spazi dei nomi:
 - **openshift.io/sa.scc.mcs**
 - **openshift.io/sa.scc.supplemental-groups**
 - **openshift.io/sa.scc.uid-range**
 Queste annotazioni preservano l'intervallo UID, assicurando che i container mantengano le autorizzazioni del file system sul cluster di destinazione. Esiste il rischio che gli UID migrati possano duplicare gli UID all'interno di uno spazio dei nomi esistente o futuro sul cluster di destinazione. ([BZ#1748440](#))
- La maggior parte delle risorse in ambito cluster non è ancora gestita da MTC. Se le applicazioni necessitano di risorse in ambito cluster, potrebbe essere necessario crearle manualmente sul cluster di destinazione.
- Se una migrazione non riesce, il piano di migrazione non mantiene le impostazioni PV personalizzate per i pod in quiescenza. È necessario eseguire manualmente il rollback della migrazione, eliminare il piano di migrazione e creare un nuovo piano di migrazione con le proprie impostazioni PV. ([BZ#1784899](#))
- Se una migrazione di grandi dimensioni non riesce perché si verifica il timeout di Restic, è possibile aumentare il valore del parametro **restic_timeout** (predefinito: **1h**) nel manifest della risorsa personalizzata **MigrationController**.
- Se si seleziona l'opzione di verifica dei dati per i PV che sono migrati con il metodo di copia del file system, le prestazioni sono significativamente più lente.
- Se si esegue la migrazione dei dati dallo storage NFS e **root_squash** è abilitato, **Restic** esegue il mapping a **nfsnobody**. La migrazione non riesce e viene visualizzato un errore di autorizzazione nel log del pod **Restic**. ([BZ#1873641](#))
 È possibile risolvere questo problema aggiungendo gruppi supplementari per **Restic** al manifest della risorsa personalizzata **MigrationController**:


```
spec:
  ...
  restic_supplemental_groups:
  - 5555
  - 6666
```
- Se si esegue la migrazione diretta dei volumi con nodi che si trovano in diverse zone di disponibilità o set di disponibilità, la migrazione potrebbe non riuscire perché i pod migrati non possono accedere alle PVC. ([BZ#1947487](#))

12.5. ROLLBACK DI UNA MIGRAZIONE

È possibile eseguire il rollback di una migrazione utilizzando la console web di MTC o la CLI.

È anche possibile eseguire [il rollback di una migrazione manualmente](#).

12.5.1. Rollback di una migrazione utilizzando la console web di MTC

È possibile annullare una migrazione utilizzando la console web di Migration Toolkit for Containers (MTC).



NOTA

Le seguenti risorse rimangono negli spazi dei nomi migrati per il debugging dopo una migrazione di volumi diretta (DVM) non riuscita:

- Mappe di configurazione (cluster di origine e destinazione)
- Oggetti **Secret** (cluster di origine e destinazione)
- Risorse personalizzate **Rsync** (cluster di origine)

Queste risorse non influiscono sul rollback. È possibile eliminarle manualmente.


Se si esegue con successo lo stesso piano di migrazione in seguito, le risorse della migrazione non riuscita vengono eliminate automaticamente.

Se l'applicazione è stata fermata durante una migrazione non riuscita, è necessario eseguire il rollback della migrazione per prevenire la corruzione dei dati nel volume permanente.

Il rollback non è richiesto se l'applicazione non è stata fermata durante la migrazione perché l'applicazione originale è ancora in esecuzione sul cluster di origine.

Procedura

1. Nella console web di MTC, cliccare su **Migration plans**.

2. Cliccare sul menu Opzioni  accanto a un piano di migrazione e selezionare **Rollback** sotto **Migration**.

3. Cliccare su **Rollback** e attendere che il rollback sia completato.
Nei dettagli del piano di migrazione, viene visualizzato **Rollback succeeded**.

4. Verificare che il rollback sia andato a buon fine nella console web di OpenShift Container Platform del cluster di origine:

- a. Cliccare su **Home** → **Projects**.
- b. Cliccare sul progetto migrato per visualizzarne lo stato.
- c. Nella sezione **Routes**, cliccare su **Location** per verificare che l'applicazione sia funzionante, se applicabile.
- d. Cliccare su **Workloads** → **Pods** per verificare che i pod siano in esecuzione nello spazio dei nomi migrato.
- e. Cliccare su **Storage** → **Persistent volumes** per verificare che il provisioning del volume permanente migrato sia stato effettuato correttamente.

12.5.2. Rollback di una migrazione dall'interfaccia della riga di comando

È possibile annullare una migrazione creando una risorsa personalizzata **MigMigration** dall'interfaccia della riga di comando.



NOTA

Le seguenti risorse rimangono negli spazi dei nomi migrati per il debugging dopo una migrazione di volumi diretta (DVM) non riuscita:

- Mappe di configurazione (cluster di origine e destinazione)
- Oggetti **Secret** (cluster di origine e destinazione)
- Risorse personalizzate **Rsync** (cluster di origine)

Queste risorse non influiscono sul rollback. È possibile eliminarle manualmente.

Se si esegue con successo lo stesso piano di migrazione in seguito, le risorse della migrazione non riuscita vengono eliminate automaticamente.

Se l'applicazione è stata fermata durante una migrazione non riuscita, è necessario eseguire il rollback della migrazione per prevenire la corruzione dei dati nel volume permanente.

Il rollback non è richiesto se l'applicazione non è stata fermata durante la migrazione perché l'applicazione originale è ancora in esecuzione sul cluster di origine.

Procedura

1. Creare una risorsa personalizzata **MigMigration** basata sul seguente esempio:

```
$ cat << EOF | oc apply -f -
apiVersion: migration.openshift.io/v1alpha1
kind: MigMigration
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: <migmigration>
  namespace: openshift-migration
spec:
...
  rollback: true
...
  migPlanRef:
    name: <migplan> 1
    namespace: openshift-migration
EOF
```



1 Specificare il nome della risorsa personalizzata **MigPlan** associata.

2. Nella console web di MTC, verificare che le risorse del progetto migrato siano state rimosse dal cluster di destinazione.
3. Verificare che le risorse del progetto migrato siano presenti nel cluster di origine e che l'applicazione sia in esecuzione.

12.5.3. Rollback manuale di una migrazione

È possibile eseguire il rollback di una migrazione non riuscita manualmente eliminando i pod dello **stage** e riportando l'applicazione allo stato attivo.

Se si esegue con successo lo stesso piano di migrazione, le risorse della migrazione non riuscita vengono eliminate automaticamente.



NOTA

Le seguenti risorse rimangono negli spazi dei nomi migrati dopo una migrazione di volumi diretta (DVM) non riuscita:

- Mappe di configurazione (cluster di origine e destinazione)
- Oggetti **Secret** (cluster di origine e destinazione)
- Risorse personalizzate **Rsync** (cluster di origine)

Queste risorse non influiscono sul rollback. È possibile eliminarle manualmente.

Procedura

1. Eliminare i pod **stage** su tutti i cluster:

```
$ oc delete $(oc get pods -l migration.openshift.io/is-stage-pod -n <namespace>) 1
```

- 1** Spazi dei nomi specificati nella risorsa personalizzata **MigPlan**.

2. Riattivare l'applicazione sul cluster di origine ridimensionando le repliche sul numero di pre-migrazione:

```
$ oc scale deployment <deployment> --replicas=<premigration_replicas>
```

L'annotazione **migration.openshift.io/preQuiesceReplicas** nella risorsa personalizzata **Deployment** mostra il numero di repliche pre-migrazione:

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
    migration.openshift.io/preQuiesceReplicas: "1"
```

3. Verificare che i pod delle applicazioni siano in esecuzione sul cluster di origine:

```
$ oc get pod -n <namespace>
```

Risorse aggiuntive

- [Eliminare gli operatori da un cluster utilizzando la console web](#)

