



OpenShift Container Platform 4.5

클러스터 업데이트

OpenShift Container Platform 클러스터 업데이트

OpenShift Container Platform 4.5 클러스터 업데이트

OpenShift Container Platform 클러스터 업데이트

법적 공지

Copyright © 2020 Red Hat, Inc.

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.

초록

이 문서는 OpenShift Container Platform 클러스터 업데이트 또는 업그레이드에 대한 정보를 제공합니다. 클러스터 업데이트는 클러스터를 오프라인으로 전환할 필요없이 간단한 프로세스로 실행할 수 있습니다.

차례

1장. 마이너 버전 간 클러스터 업데이트	3
1.1. OPENSIFT CONTAINER PLATFORM 업데이트 서비스 정보	3
1.2. OPENSIFT CONTAINER PLATFORM 업그레이드 채널 및 릴리스 버전	4
1.3. 웹 콘솔을 사용하여 클러스터 업데이트	6
2장. 웹 콘솔에서 마이너 버전으로 클러스터 업데이트	8
2.1. OPENSIFT CONTAINER PLATFORM 업데이트 서비스 정보	8
2.2. OPENSIFT CONTAINER PLATFORM 업그레이드 채널 및 릴리스 버전	9
2.3. 웹 콘솔을 사용하여 클러스터 업데이트	11
3장. CLI를 사용하여 마이너 버전에서 클러스터 업데이트	12
3.1. OPENSIFT CONTAINER PLATFORM 업데이트 서비스 정보	12
3.2. OPENSIFT CONTAINER PLATFORM 업그레이드 채널 및 릴리스 버전	13
3.3. CLI를 사용하여 클러스터 업데이트	15
4장. RHEL 컴퓨팅 시스템을 포함하는 클러스터 업데이트	18
4.1. OPENSIFT CONTAINER PLATFORM 업데이트 서비스 정보	18
4.2. OPENSIFT CONTAINER PLATFORM 업그레이드 채널 및 릴리스 버전	19
4.3. 웹 콘솔을 사용하여 클러스터 업데이트	21
4.4. (선택 사항) RHEL 시스템에서 ANSIBLE 작업을 실행하기 위한 후크 추가	22
4.5. 클러스터에서 RHEL 컴퓨팅 시스템 업데이트	24
5장. 네트워크가 제한된 환경에서 클러스터 업데이트	27
5.1. 미러 호스트 준비	27
5.2. 이미지를 미러링할 수 있는 인증 정보 설정	29
5.3. OPENSIFT CONTAINER PLATFORM 이미지 저장소 미러링	30
5.4. 이미지 서명 CONFIGMAP 생성	31
5.5. 네트워크가 제한된 환경의 클러스터 업데이트	33
5.6. 이미지 레지스트리 저장소 미러링 설정	34

1장. 마이너 버전 간 클러스터 업데이트

마이너 버전 간에 OpenShift Container Platform 클러스터를 업데이트하거나 업그레이드할 수 있습니다.



참고

oc를 사용하여 업데이트 채널을 변경하기가 어렵기 때문에 웹 콘솔을 사용하여 업데이트 채널을 변경합니다. 웹 콘솔 내에서 업데이트 프로세스를 완료하는 것이 좋습니다. 4.5 채널을 변경한 후 업데이트를 완료하기 위해 **CLI를 사용하여 마이너 버전에서 클러스터 업데이트** 단계를 실행할 수 있습니다.

전제 조건

- **admin** 권한이 있는 사용자로 클러스터에 액세스합니다. [Using RBAC to define and apply permissions](#)을 참조하십시오.
- 업그레이드에 실패할 경우 **etcd backup**이 있어야 하고 **클러스터를 이전 상태로 복원** 해야 합니다.



중요

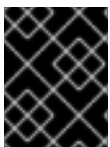
unsupportedConfigOverrides 섹션을 사용하여 Operator 설정을 변경하는 것은 지원되지 않으며 클러스터 업그레이드를 차단할 수 있습니다. 클러스터를 업그레이드하기 전에 이 설정을 삭제해야 합니다.

1.1. OPENSIFT CONTAINER PLATFORM 업데이트 서비스 정보

OpenShift Container Platform 업데이트 서비스는 OpenShift Container Platform 및 RHCOS (Red Hat Enterprise Linux CoreOS) 모두에 무선(OTA, Over-the-air) 업데이트를 제공하는 호스팅 서비스입니다. 구성 요소 Operator의 **정점**과 이를 연결하는 **에지**를 포함하는 그래프 또는 다이어그램을 제공합니다. 그래프의 에지에는 안전하게 업데이트할 수 있는 버전이 표시되고 정점은 관리되는 클러스터 구성 요소의 상태를 확인하는 업데이트 페이로드입니다.

클러스터의 CVO (Cluster Version Operator)는 OpenShift Container Platform 업데이트 서비스를 확인하여 현재 구성 요소 버전 및 그래프의 정보를 기반으로 유효한 업데이트 및 업데이트 경로를 확인합니다. 업데이트를 요청하면 OpenShift Container Platform CVO는 해당 업데이트의 릴리스 이미지를 사용하여 클러스터를 업그레이드합니다. 릴리스 아티팩트는 Quay에서 컨테이너 이미지로 호스팅됩니다.

OpenShift Container Platform 업데이트 서비스가 호환 가능한 업데이트 만 제공할 수 있도록 자동화를 지원하는 버전 확인 파이프 라인이 제공됩니다. 각 릴리스 아티팩트는 지원되는 클라우드 플랫폼 및 시스템 아키텍처 및 기타 구성 요소 패키지와의 호환성 여부를 확인합니다. 파이프 라인에서 적용 가능한 버전이 있음을 확인한 후 OpenShift Container Platform 업데이트 서비스는 해당 버전 업데이트를 사용할 수 있음을 알려줍니다.



중요

업데이트 서비스는 모든 유효한 업데이트를 표시하므로 업데이트 서비스가 표시하지 않는 버전으로 강제 업데이트할 수 없습니다.

연속 업데이트 모드에서는 두 개의 컨트롤러가 실행됩니다. 하나의 컨트롤러는 페이로드 매니페스트를 지속적으로 업데이트하여 클러스터에 적용한 다음 사용 가능한지, 업그레이드했는지 또는 실패했는지에 따라 Operator의 제어된 롤아웃 상태를 출력합니다. 두 번째 컨트롤러는 OpenShift Container Platform 업데이트 서비스를 폴링하여 업데이트를 사용할 수 있는지 확인합니다.



중요

클러스터를 이전 버전으로 되돌리거나 롤백을 수행하는 것은 지원되지 않습니다. 최신 버전으로의 업그레이드 만 지원됩니다.

업그레이드 프로세스 중에 MCO (Machine Config Operator)는 새 설정을 클러스터 시스템에 적용합니다. 이는 시스템 설정 폴의 **maxUnavailable** 필드에 지정된 노드 수를 제한하고 이를 사용할 수 없는 것으로 표시합니다. 기본적으로 이 값은 **1**로 설정됩니다. 그 다음 새 설정을 적용하여 컴퓨터를 다시 시작합니다. RHEL (Red Hat Enterprise Linux) 시스템을 작업자로 사용하는 경우 먼저 OpenShift API를 업데이트해야 하기 때문에 MCO는 이 시스템에서 kubelet을 업데이트하지 않습니다. 새 버전의 사양이 이전 kubelet에 적용되므로 RHEL 시스템을 **Ready** 상태로 되돌릴 수 없습니다. 컴퓨터를 사용할 수 있을 때까지 업데이트를 완료할 수 없습니다. 그러나 사용 불가능한 최대 노드 수를 설정하면 사용할 수 없는 시스템의 수가 이 값을 초과하지 않는 경우에도 정상적인 클러스터 작업을 계속할 수 있습니다.

추가 리소스

- [관리되지 않는 Operator에 대한 지원 정책](#)

1.2. OPENSIFT CONTAINER PLATFORM 업그레이드 채널 및 릴리스 버전

OpenShift Container Platform 4.1에서 Red Hat은 클러스터 업그레이드에 적합한 버전을 권장하기 위해 업그레이드 채널 개념을 도입했습니다. 업그레이드 속도를 제어함으로써 이러한 업그레이드 채널을 통해 업그레이드 전략을 선택할 수 있습니다. 업그레이드 채널은 OpenShift Container Platform의 마이너 버전과 연결되어 있습니다. 예를 들어 OpenShift Container Platform 4.5 업그레이드 채널에는 4.5 릴리스 버전으로의 업그레이드가 포함되어 있지 않습니다. 이를 통해 관리자는 OpenShift Container Platform의 다음 마이너 버전으로 업그레이드하기 위한 명확한 결정을 내릴 수 있습니다. 업그레이드 채널은 릴리스 선택만 제어하며 설치한 클러스터 버전에는 영향을 미치지 않습니다. OpenShift Container Platform 특정 버전의 **openshift-install** 바이너리 파일은 항상 해당 버전을 설치합니다.

OpenShift Container Platform 4.5는 다음과 같은 업그레이드 채널을 제공합니다.

- **candidate-4.5**
- **fast-4.5**
- **stable-4.5**

candidate-4.5 채널

candidate-4.5 채널에는 z-stream (4.5.z) 릴리스에 대한 후보 빌드가 포함되어 있습니다. 릴리스 후보 버전에는 제품의 모든 기능이 포함되어 있지만 지원되지는 않습니다. 릴리스 후보 버전을 사용하여 새 버전의 기능을 테스트하고 다음 OpenShift Container Platform 버전이 시스템에 적합한지 확인할 수 있습니다. 릴리스 후보는 이름에 **-rc**가 포함되어 있지 않은 후보 채널에서 사용 가능한 빌드를 말합니다. 후보 채널에서 버전을 사용할 수 있게 되면 더 많은 품질 테스트가 수행됩니다. 품질 기준을 충족하는 경우 **fast-4.5** 또는 **stable-4.5** 채널로 확장됩니다. 이로 인해 특정 릴리스가 **candidate-4.5** 채널과 **fast-4.5** 또는 **stable-4.5** 채널 모두에서 사용 가능한 경우 이는 Red Hat에서 지원되는 버전임을 의미합니다. **candidate-4.5** 채널에는 채널에 권장되는 업데이트가 없는 릴리스 버전이 포함되어 있을 수 있습니다.

candidate-4.5 채널을 사용하여 OpenShift Container Platform의 이전 마이너 버전에서 업그레이드할 수 있습니다.



참고

릴리스 후보는 <https://www.openshift.com/try> 사이트에 있는 Nightly build와 다릅니다. Nightly build는 기능에 조기 액세스하기 위해 사용할 수 있지만 Nightly build로 또는 Nightly build에서 업데이트하는 것은 권장되거나 지원되지 않습니다. 업그레이드 채널에서는 Nightly build를 사용할 수 없습니다.

fast-4.5 채널

Red Hat이 특정 버전이 공식 릴리스가 된다고 선언하면 **fast-4.5** 채널이 새로운 4.5 버전으로 업데이트됩니다. 이와 같이 이 릴리스는 완전히 지원되고, 프로덕션 환경에 적합한 품질이며, **candidate-4.5** 채널에서 릴리스 후보로 사용 가능한 동안 문제 없이 제대로 작동하는 것으로 표시됩니다. **fast-4.5** 채널에 릴리스가 표시된 후 **stable-4.5** 채널에 추가됩니다. 릴리스는 **fast-4.5** 채널에 표시되기 전에 **stable-4.5** 채널에 표시되지 않습니다.

fast-4.5 채널을 사용하여 OpenShift Container Platform의 이전 마이너스 버전에서 업그레이드할 수 있습니다.

stable-4.5 채널

에라타가 출시되면 곧 **fast-4.5** 채널에 표시되지만 릴리스는 지연 후 **stable-4.5** 채널에 추가됩니다. 이 지연 시간 동안 Connected Customer Program에 참여하는 Red Hat SRE 팀, Red Hat 지원 서비스, 사전 프로덕션 및 프로덕션 환경에서 릴리스의 안정성에 대한 데이터가 수집됩니다.

stable-4.5 채널을 사용하여 OpenShift Container Platform의 이전 마이너스 버전에서 업그레이드할 수 있습니다.

업그레이드 버전 경로

OpenShift Container Platform은 설치된 OpenShift Container Platform 버전과 다음 릴리스로 액세스하기 위해 선택한 채널의 경로를 확인할 수 있는 업그레이드 권장 서비스를 제공합니다. **fast-4.5** 채널에서는 다음을 확인할 수 있습니다.

- 4.5.0
- 4.5.1
- 4.5.3
- 4.5.4

이 서비스는 테스트를 통해 심각한 문제가 없는 업그레이드만 권장합니다. 클러스터가 4.5.1에 있고 OpenShift Container Platform에서 4.5.4를 권장하는 경우 .4.5.1에서 .4.5.4로 안전하게 업데이트할 수 있습니다. 연속적인 패치 번호에 의존하지 않도록 하십시오. 이 예에서 4.5.2는 채널에서 사용 가능하지 않습니다. 업데이트 서비스는 알려진 취약점이 포함된 OpenShift Container Platform 버전으로 업데이트할 것을 권장하지 않습니다.

업데이트의 안정성은 채널에 따라 다릅니다. **candidate-4.5** 채널에 업데이트 권장 사항이 있다고 해서 해당 업데이트가 지원되는 것은 아닙니다. 업데이트와 관련하여 아직 심각한 문제가 발견되지 않았지만 업데이트를 통한 트래픽이 많지 않아 안정성이 확인되지 않을 수 있습니다. **fast-4.5** 또는 **secure-4.5** 채널에 업데이트 권장 사항이 있음은 업데이트가 채널에 있는 동안 완전히 지원됨을 나타냅니다. 릴리스는 채널에서 제거되지 않지만 심각한 문제가 있는 업데이트 권장 사항은 모든 채널에서 제거됩니다. 업데이트 권장 사항이 제거된 후 시작된 업데이트는 지원되지 않을 수 있습니다.

Red Hat은 **fast-4.5** 또는 **stable-4.5** 채널에서 지원되는 모든 릴리스에서 4.5.z의 최신 릴리스로 지원되는 업데이트 경로를 제공합니다. 그러나 문제가 발생한 릴리스로부터 안전한 경로를 구축하고 확인하는 동안 지연이 발생할 수 있습니다.

빠르고 안정적인 채널 사용 및 전략

fast-4.5 및 **stable-4.5** 채널에서는 공식 버전이 릴리스되는 대로 이를 즉시 수신하거나 Red Hat이 해당 업데이트의 롤아웃을 제어하는 것을 허용할지 여부를 선택할 수 있습니다. 롤아웃 중 또는 이후에 문제가 발견되면 해당 버전으로의 업그레이드가 **fast-4.5** 및 **stable-4.5** 채널에서 차단될 수 있으며 새로 권장되는 업그레이드 대상의 새 버전이 도입될 수 있습니다.

고객은 **fast-4.5** 채널에서 사전 프로덕션 시스템을 설정하고, **stable-4.5** 채널에서 프로덕션 시스템을 설정한 뒤, Red Hat의 Connected Customer Program에 참여하여 고객의 프로세스를 개선할 수 있습니다. Red Hat은 이 프로그램을 사용하여 사용자의 특정 하드웨어 및 소프트웨어 설정에 대한 업데이트의 영향을 관찰합니다. 향후 릴리스에서는 업데이트가 **fast-4.5**에서 **stable-4.5** 채널로 이동하는 속도가 향상되거나 변경될 수 있습니다.

네트워크가 제한된 환경의 클러스터

OpenShift Container Platform 클러스터의 컨테이너 이미지를 직접 관리하는 경우 제품 릴리스와 관련된 Red Hat 에라타를 참조하고 업그레이드에 영향을 미치는 영향을 고려해야 합니다. 업그레이드하는 동안 사용자 인터페이스에서 이러한 버전 간 전환에 대해 경고가 표시될 수 있으므로 이러한 경고를 무시하기 전에 적절한 버전을 선택했는지 확인해야 합니다.

채널 간 전환

stable-4.5 채널에서 **fast-4.5** 채널로 변경해도 클러스터는 계속 지원됩니다. 언제든지 **candidate-4.5** 채널로 전환할 수 있지만 해당 채널의 일부 릴리스는 지원되지 않을 수 있습니다. 현재 릴리스가 정식 사용 버전 릴리스인 경우 **candidate-4.5** 채널에서 **fast-4.5** 채널로 전환할 수 있습니다. 현재 릴리스가 최근에 **fast-4.5**로 승격된 경우 릴리스가 **stable-4.5**로 승격되기까지 최대 하루가 지연될 수 있지만 **fast-4.5** 채널에서 **stable-4.5** 채널로 전환할 수 있습니다. 현재 릴리스가 포함되지 않은 채널로 변경하면 경고가 표시되고 업데이트는 권장되지 않지만 언제든지 원래 채널로 안전하게 전환할 수 있습니다.

1.3. 웹 콘솔을 사용하여 클러스터 업데이트

사용 가능한 업데이트가 있으면 웹 콘솔에서 클러스터를 업데이트할 수 있습니다.

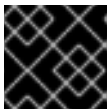
사용 가능한 OpenShift Container Platform 권고 및 업데이트는 고객 포털의 [에라타 섹션](#)을 참조하십시오.

전제 조건

- **admin** 권한이있는 사용자로 웹 콘솔에 액세스합니다.

프로세스

1. 웹 콘솔에서 **Administration > Cluster Settings**을 클릭하고 **Overview** 탭의 내용을 확인합니다.
2. 프로덕션 클러스터의 경우 **CHANNEL**이 **stable-4.5** 같은 현재 마이너 버전에 대한 올바른 채널로 설정되어 있는지 확인합니다.



중요

프로덕션 클러스터의 경우 **stable-*** 또는 **fast-*** 채널에 가입해야 합니다.

- **UPDATE STATUS**가 **Updates Available**가 아닌 경우 클러스터를 업그레이드할 수 없습니다.
- **DESIRED VERSION**은 클러스터가 실행 중이거나 업데이트 중인 클러스터 버전을 나타냅니다.

3. **Updates Available**을 클릭하고 사용 가능한 최고 버전을 선택한 다음 **Update**를 클릭합니다. **UPDATE STATUS**는 **Updating**으로 변경하고 **Cluster Operators** 탭에서 Operator 업그레이드 진행 상황을 확인할 수 있습니다.
 4. 업데이트가 완료되고 Cluster Version Operator가 사용 가능한 업데이트를 새로 고침한 후 현재 채널에서 사용 가능한 추가 업데이트가 있는지 확인합니다.
 - 업데이트가 있는 경우 더 이상 업데이트할 수 없을 때까지 현재 채널에서 업데이트를 계속 수행합니다.
 - 사용할 수 있는 업데이트가 없는 경우, **CHANNEL**을 다음 마이너 버전의 **stable-*** 또는 **fast-*** 채널로 변경하고 해당 채널에서 원하는 버전으로 업데이트합니다.
- 필요한 버전에 도달할 때까지 여러 중간 업데이트를 수행해야 할 수도 있습니다.

2장. 웹 콘솔에서 마이너 버전으로 클러스터 업데이트

웹 콘솔을 사용하여 OpenShift Container Platform 클러스터를 업데이트하거나 업그레이드할 수 있습니다.

전제 조건

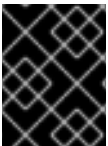
- **admin** 권한이 있는 사용자로 클러스터에 액세스합니다. [Using RBAC to define and apply permissions](#)을 참조하십시오.
- 업그레이드에 실패할 경우 **etcd backup**이 있어야 하고 **클러스터를 이전 상태로 복원**해야 합니다.

2.1. OPENSIFT CONTAINER PLATFORM 업데이트 서비스 정보

OpenShift Container Platform 업데이트 서비스는 OpenShift Container Platform 및 RHCOS (Red Hat Enterprise Linux CoreOS) 모두에 무선(OTA, Over-the-air) 업데이트를 제공하는 호스팅 서비스입니다. 구성 요소 Operator의 **정점**과 이를 연결하는 **에지**를 포함하는 그래프 또는 다이어그램을 제공합니다. 그래프의 에지에는 안전하게 업데이트할 수 있는 버전이 표시되고 정점은 관리되는 클러스터 구성 요소의 상태를 확인하는 업데이트 페이로드입니다.

클러스터의 CVO (Cluster Version Operator)는 OpenShift Container Platform 업데이트 서비스를 확인하여 현재 구성 요소 버전 및 그래프의 정보를 기반으로 유효한 업데이트 및 업데이트 경로를 확인합니다. 업데이트를 요청하면 OpenShift Container Platform CVO는 해당 업데이트의 릴리스 이미지를 사용하여 클러스터를 업그레이드합니다. 릴리스 아티팩트는 Quay에서 컨테이너 이미지로 호스팅됩니다.

OpenShift Container Platform 업데이트 서비스가 호환 가능한 업데이트 만 제공할 수 있도록 자동화를 지원하는 버전 확인 파이프 라인이 제공됩니다. 각 릴리스 아티팩트는 지원되는 클라우드 플랫폼 및 시스템 아키텍처 및 기타 구성 요소 패키지와의 호환성 여부를 확인합니다. 파이프 라인에서 적용 가능한 버전이 있음을 확인한 후 OpenShift Container Platform 업데이트 서비스는 해당 버전 업데이트를 사용할 수 있음을 알려줍니다.



중요

업데이트 서비스는 모든 유효한 업데이트를 표시하므로 업데이트 서비스가 표시하지 않는 버전으로 강제 업데이트할 수 없습니다.

연속 업데이트 모드에서는 두 개의 컨트롤러가 실행됩니다. 하나의 컨트롤러는 페이로드 매니페스트를 지속적으로 업데이트하여 클러스터에 적용한 다음 사용 가능한지, 업그레이드했는지 또는 실패했는지에 따라 Operator의 제어된 롤아웃 상태를 출력합니다. 두 번째 컨트롤러는 OpenShift Container Platform 업데이트 서비스를 폴링하여 업데이트를 사용할 수 있는지 확인합니다.



중요

클러스터를 이전 버전으로 되돌리거나 롤백을 수행하는 것은 지원되지 않습니다. 최신 버전으로의 업그레이드 만 지원됩니다.

업그레이드 프로세스 중에 MCO (Machine Config Operator)는 새 설정을 클러스터 시스템에 적용합니다. 이는 시스템 설정 폴의 **maxUnavailable** 필드에 지정된 노드 수를 제한하고 이를 사용할 수 없는 것으로 표시합니다. 기본적으로 이 값은 **1**로 설정됩니다. 그 다음 새 설정을 적용하여 컴퓨터를 다시 시작합니다. RHEL (Red Hat Enterprise Linux) 시스템을 작업자로 사용하는 경우 먼저 OpenShift API를 업데이트해야 하기 때문에 MCO는 이 시스템에서 kubelet을 업데이트하지 않습니다. 새 버전의 사양이 이전 kubelet에 적용되므로 RHEL 시스템을 **Ready** 상태로 되돌릴 수 없습니다. 컴퓨터를 사용할 수 있을 때까지 업데이트를 완료할 수 없습니다. 그러나 사용 불가능한 최대 노드 수를 설정하면 사용할 수 없는 시스템의 수가 이 값을 초과하지 않는 경우에도 정상적인 클러스터 작업을 계속할 수 있습니다.

추가 리소스

- [관리되지 않는 Operator에 대한 지원 정책](#)

2.2. OPENSIFT CONTAINER PLATFORM 업그레이드 채널 및 릴리스 버전

OpenShift Container Platform 4.1에서 Red Hat은 클러스터 업그레이드에 적합한 버전을 권장하기 위해 업그레이드 채널 개념을 도입했습니다. 업그레이드 속도를 제어함으로써 이러한 업그레이드 채널을 통해 업그레이드 전략을 선택할 수 있습니다. 업그레이드 채널은 OpenShift Container Platform의 마이너 버전과 연결되어 있습니다. 예를 들어 OpenShift Container Platform 4.5 업그레이드 채널에는 4.5 릴리스 버전으로의 업그레이드가 포함되어 있지 않습니다. 이를 통해 관리자는 OpenShift Container Platform의 다음 마이너 버전으로 업그레이드하기 위한 명확한 결정을 내릴 수 있습니다. 업그레이드 채널은 릴리스 선택만 제어하며 설치한 클러스터 버전에는 영향을 미치지 않습니다. OpenShift Container Platform 특정 버전의 **openshift-install** 바이너리 파일은 항상 해당 버전을 설치합니다.

OpenShift Container Platform 4.5는 다음과 같은 업그레이드 채널을 제공합니다.

- **candidate-4.5**
- **fast-4.5**
- **stable-4.5**

candidate-4.5 채널

candidate-4.5 채널에는 z-stream (4.5.z) 릴리스에 대한 후보 빌드가 포함되어 있습니다. 릴리스 후보 버전에는 제품의 모든 기능이 포함되어 있지만 지원되지는 않습니다. 릴리스 후보 버전을 사용하여 새 버전의 기능을 테스트하고 다음 OpenShift Container Platform 버전이 시스템에 적합한지 확인할 수 있습니다. 릴리스 후보는 이름에 **-rc**가 포함되어 있지 않은 후보 채널에서 사용 가능한 빌드를 말합니다. 후보 채널에서 버전을 사용할 수 있게 되면 더 많은 품질 테스트가 수행됩니다. 품질 기준을 충족하는 경우 **fast-4.5** 또는 **stable-4.5** 채널로 확장됩니다. 이로 인해 특정 릴리스가 **candidate-4.5** 채널과 **fast-4.5** 또는 **stable-4.5** 채널 모두에서 사용 가능한 경우 이는 Red Hat에서 지원되는 버전임을 의미합니다.

candidate-4.5 채널에는 채널에 권장되는 업데이트가 없는 릴리스 버전이 포함되어 있을 수 있습니다.

candidate-4.5 채널을 사용하여 OpenShift Container Platform의 이전 마이너 버전에서 업그레이드할 수 있습니다.



참고

릴리스 후보는 <https://www.openshift.com/try> 사이트에 있는 Nightly build와 다릅니다. Nightly build는 기능에 조기 액세스하기 위해 사용할 수 있지만 Nightly build로 또는 Nightly build에서 업데이트하는 것은 권장되거나 지원되지 않습니다. 업그레이드 채널에서는 Nightly build를 사용할 수 없습니다.

fast-4.5 채널

Red Hat이 특정 버전이 공식 릴리스가 된다고 선언하면 **fast-4.5** 채널이 새로운 4.5 버전으로 업데이트됩니다. 이와 같이 이 릴리스는 완전히 지원되고, 프로덕션 환경에 적합한 품질이며, **candidate-4.5** 채널에서 릴리스 후보로 사용 가능한 동안 문제 없이 제대로 작동하는 것으로 표시됩니다. **fast-4.5** 채널에 릴리스가 표시된 후 **stable-4.5** 채널에 추가됩니다. 릴리스는 **fast-4.5** 채널에 표시되기 전에 **stable-4.5** 채널에 표시되지 않습니다.

fast-4.5 채널을 사용하여 OpenShift Container Platform의 이전 마이너스 버전에서 업그레이드할 수 있습니다.

stable-4.5 채널

에라타가 출시되면 곧 **fast-4.5** 채널에 표시되지만 릴리스는 지연 후 **stable-4.5** 채널에 추가됩니다. 이 지연 시간 동안 Connected Customer Program에 참여하는 Red Hat SRE 팀, Red Hat 지원 서비스, 사전 프로덕션 및 프로덕션 환경에서 릴리스의 안정성에 대한 데이터가 수집됩니다.

stable-4.5 채널을 사용하여 OpenShift Container Platform의 이전 바이너스 버전에서 업그레이드할 수 있습니다.

업그레이드 버전 경로

OpenShift Container Platform은 설치된 OpenShift Container Platform 버전과 다음 릴리스로 액세스하기 위해 선택한 채널의 경로를 확인할 수 있는 업그레이드 권장 서비스를 제공합니다. **fast-4.5** 채널에서는 다음을 확인할 수 있습니다.

- 4.5.0
- 4.5.1
- 4.5.3
- 4.5.4

이 서비스는 테스트를 통해 심각한 문제가 없는 업그레이드만 권장합니다. 클러스터가 4.5.1에 있고 OpenShift Container Platform에서 4.5.4를 권장하는 경우 .4.5.1에서 .4.5.4로 안전하게 업데이트할 수 있습니다. 연속적인 패치 번호에 의존하지 않도록 하십시오. 이 예에서 4.5.2는 채널에서 사용 가능하지 않습니다. 업데이트 서비스는 알려진 취약점이 포함된 OpenShift Container Platform 버전으로 업데이트할 것을 권장하지 않습니다.

업데이트의 안정성은 채널에 따라 다릅니다. **candidate-4.5** 채널에 업데이트 권장 사항이 있다고 해서 해당 업데이트가 지원되는 것은 아닙니다. 업데이트와 관련하여 아직 심각한 문제가 발견되지 않았지만 업데이트를 통한 트래픽이 많지 않아 안정성이 확인되지 않을 수 있습니다. **fast-4.5** 또는 **secure-4.5** 채널에 업데이트 권장 사항이 있음은 업데이트가 채널에 있는 동안 완전히 지원됨을 나타냅니다. 릴리스는 채널에서 제거되지 않지만 심각한 문제가 있는 업데이트 권장 사항은 모든 채널에서 제거됩니다. 업데이트 권장 사항이 제거된 후 시작된 업데이트는 지원되지 않을 수 있습니다.

Red Hat은 **fast-4.5** 또는 **stable-4.5** 채널에서 지원되는 모든 릴리스에서 4.5.z의 최신 릴리스로 지원되는 업데이트 경로를 제공합니다. 그러나 문제가 발생한 릴리스로부터 안전한 경로를 구축하고 확인하는 동안 지연이 발생할 수 있습니다.

빠르고 안정적인 채널 사용 및 전략

fast-4.5 및 **stable-4.5** 채널에서는 공식 버전이 릴리스되는 대로 이를 즉시 수신하거나 Red Hat이 해당 업데이트의 롤아웃을 제어하는 것을 허용할지 여부를 선택할 수 있습니다. 롤아웃 중 또는 이후에 문제가 발견되면 해당 버전으로의 업그레이드가 **fast-4.5** 및 **stable-4.5** 채널에서 차단될 수 있으며 새로 권장되는 업그레이드 대상의 새 버전이 도입될 수 있습니다.

고객은 **fast-4.5** 채널에서 사전 프로덕션 시스템을 설정하고, **stable-4.5** 채널에서 프로덕션 시스템을 설정한 뒤, Red Hat의 Connected Customer Program에 참여하여 고객의 프로세스를 개선할 수 있습니다. Red Hat은 이 프로그램을 사용하여 사용자의 특정 하드웨어 및 소프트웨어 설정에 대한 업데이트의 영향을 관찰합니다. 향후 릴리스에서는 업데이트가 **fast-4.5**에서 **stable-4.5** 채널로 이동하는 속도가 향상되거나 변경될 수 있습니다.

네트워크가 제한된 환경의 클러스터

OpenShift Container Platform 클러스터의 컨테이너 이미지를 직접 관리하는 경우 제품 릴리스와 관련된 Red Hat 에라타를 참조하고 업그레이드에 영향을 미치는 영향을 고려해야 합니다. 업그레이드하는 동안 사용자 인터페이스에서 이러한 버전 간 전환에 대해 경고가 표시될 수 있으므로 이러한 경고를 무시하기 전에 적절한 버전을 선택했는지 확인해야 합니다.

채널 간 전환

stable-4.5 채널에서 **fast-4.5** 채널로 변경해도 클러스터는 계속 지원됩니다. 언제든지 **candidate-4.5** 채널

로 전환할 수 있지만 해당 채널의 일부 릴리스는 지원되지 않을 수 있습니다. 현재 릴리스가 정식 사용 버전 릴리스인 경우 **candidate-4.5** 채널에서 **fast-4.5** 채널로 전환할 수 있습니다. 현재 릴리스가 최근에 **fast-4.5**로 승격된 경우 릴리스가 **stable-4.5**로 승격되기까지 최대 하루가 지연될 수 있지만 **fast-4.5** 채널에서 **stable-4.5** 채널로 전환할 수 있습니다. 현재 릴리스가 포함되지 않은 채널로 변경하면 경고가 표시되고 업데이트는 권장되지 않지만 언제든지 원래 채널로 안전하게 전환할 수 있습니다.

2.3. 웹 콘솔을 사용하여 클러스터 업데이트

사용 가능한 업데이트가 있으면 웹 콘솔에서 클러스터를 업데이트할 수 있습니다.

사용 가능한 OpenShift Container Platform 권고 및 업데이트는 고객 포털의 [에라타 섹션](#)을 참조하십시오.

전제 조건

- **admin** 권한이 있는 사용자로 웹 콘솔에 액세스합니다.

프로세스

1. 웹 콘솔에서 **Administration > Cluster Settings**을 클릭하고 **Overview** 탭의 내용을 확인합니다.
2. 프로덕션 클러스터의 경우, **CHANNEL**이 업데이트하려는 버전에 대해 현재 마이너 버전 (예: **steady-4.5**)의 올바른 채널로 설정되어 있는지 확인합니다.



중요

프로덕션 클러스터의 경우 **stable-*** 또는 **fast-*** 채널에 가입해야 합니다.

- **UPDATE STATUS**가 **Updates Available**가 아닌 경우 클러스터를 업그레이드할 수 없습니다.
 - **DESIRED VERSION**은 클러스터가 실행 중이거나 업데이트 중인 클러스터 버전을 나타냅니다.
3. **Updates Available**을 클릭하여 사용 가능한 최신 버전을 선택하고 **Update**를 클릭합니다. **UPDATE STATUS**는 **Updating**으로 변경하고 **Cluster Operators** 탭에서 Operator 업그레이드 진행 상황을 확인할 수 있습니다.
 4. 업데이트가 완료되고 Cluster Version Operator가 사용 가능한 업데이트를 새로 고침한 후 현재 채널에서 사용 가능한 추가 업데이트가 있는지 확인합니다.
 - 업데이트가 있는 경우 더 이상 업데이트할 수 없을 때까지 현재 채널에서 업데이트를 계속 수행합니다.
 - 사용할 수 있는 업데이트가 없는 경우, **CHANNEL**을 다음 마이너 버전의 **stable-*** 또는 **fast-*** 채널로 변경하고 해당 채널에서 원하는 버전으로 업데이트합니다.

필요한 버전에 도달할 때까지 여러 중간 업데이트를 수행해야 할 수도 있습니다.

3장. CLI를 사용하여 마이너 버전에서 클러스터 업데이트

OpenShift CLI (**oc**)를 사용하여 마이너 버전에서 OpenShift Container Platform 클러스터를 업데이트하거나 업그레이드할 수 있습니다.

전제 조건

- **admin** 권한이 있는 사용자로 클러스터에 액세스합니다. [Using RBAC to define and apply permissions](#)을 참조하십시오.
- 업그레이드에 실패할 경우 **etcd backup**이 있어야 하고 **클러스터를 이전 상태로 복원**해야 합니다.

3.1. OPENSIFT CONTAINER PLATFORM 업데이트 서비스 정보

OpenShift Container Platform 업데이트 서비스는 OpenShift Container Platform 및 RHCOS (Red Hat Enterprise Linux CoreOS) 모두에 무선(OTA, Over-the-air) 업데이트를 제공하는 호스팅 서비스입니다. 구성 요소 Operator의 **정점**과 이를 연결하는 **에지**를 포함하는 그래프 또는 다이어그램을 제공합니다. 그래프의 에지에는 안전하게 업데이트할 수 있는 버전이 표시되고 정점은 관리되는 클러스터 구성 요소의 상태를 확인하는 업데이트 페이로드입니다.

클러스터의 CVO (Cluster Version Operator)는 OpenShift Container Platform 업데이트 서비스를 확인하여 현재 구성 요소 버전 및 그래프의 정보를 기반으로 유효한 업데이트 및 업데이트 경로를 확인합니다. 업데이트를 요청하면 OpenShift Container Platform CVO는 해당 업데이트의 릴리스 이미지를 사용하여 클러스터를 업그레이드합니다. 릴리스 아티팩트는 Quay에서 컨테이너 이미지로 호스팅됩니다.

OpenShift Container Platform 업데이트 서비스가 호환 가능한 업데이트 만 제공할 수 있도록 자동화를 지원하는 버전 확인 파이프 라인이 제공됩니다. 각 릴리스 아티팩트는 지원되는 클라우드 플랫폼 및 시스템 아키텍처 및 기타 구성 요소 패키지와의 호환성 여부를 확인합니다. 파이프 라인에서 적용 가능한 버전이 있음을 확인한 후 OpenShift Container Platform 업데이트 서비스는 해당 버전 업데이트를 사용할 수 있음을 알려줍니다.



중요

업데이트 서비스는 모든 유효한 업데이트를 표시하므로 업데이트 서비스가 표시하지 않는 버전으로 강제 업데이트할 수 없습니다.

연속 업데이트 모드에서는 두 개의 컨트롤러가 실행됩니다. 하나의 컨트롤러는 페이로드 매니페스트를 지속적으로 업데이트하여 클러스터에 적용한 다음 사용 가능한지, 업그레이드했는지 또는 실패했는지에 따라 Operator의 제어된 롤아웃 상태를 출력합니다. 두 번째 컨트롤러는 OpenShift Container Platform 업데이트 서비스를 폴링하여 업데이트를 사용할 수 있는지 확인합니다.



중요

클러스터를 이전 버전으로 되돌리거나 롤백을 수행하는 것은 지원되지 않습니다. 최신 버전으로의 업그레이드 만 지원됩니다.

업그레이드 프로세스 중에 MCO (Machine Config Operator)는 새 설정을 클러스터 시스템에 적용합니다. 이는 시스템 설정 폴의 **maxUnavailable** 필드에 지정된 노드 수를 제한하고 이를 사용할 수 없는 것으로 표시합니다. 기본적으로 이 값은 **1**로 설정됩니다. 그 다음 새 설정을 적용하여 컴퓨터를 다시 시작합니다. RHEL (Red Hat Enterprise Linux) 시스템을 작업자로 사용하는 경우 먼저 OpenShift API를 업데이트해야 하기 때문에 MCO는 이 시스템에서 kubelet을 업데이트하지 않습니다. 새 버전의 사양이 이전 kubelet에 적용되므로 RHEL 시스템을 **Ready** 상태로 되돌릴 수 없습니다. 컴퓨터를 사용할 수 있을 때까지 업데이트를 완료할 수 없습니다. 그러나 사용 불가능한 최대 노드 수를 설정하면 사용할 수 없는 시스템의 수가 이 값을 초과하지 않는 경우에도 정상적인 클러스터 작업을 계속할 수 있습니다.

추가 리소스

- [관리되지 않는 Operator에 대한 지원 정책](#)

3.2. OPENSIFT CONTAINER PLATFORM 업그레이드 채널 및 릴리스 버전

OpenShift Container Platform 4.1에서 Red Hat은 클러스터 업그레이드에 적합한 버전을 권장하기 위해 업그레이드 채널 개념을 도입했습니다. 업그레이드 속도를 제어함으로써 이러한 업그레이드 채널을 통해 업그레이드 전략을 선택할 수 있습니다. 업그레이드 채널은 OpenShift Container Platform의 마이너 버전과 연결되어 있습니다. 예를 들어 OpenShift Container Platform 4.5 업그레이드 채널에는 4.5 릴리스 버전으로의 업그레이드가 포함되어 있지 않습니다. 이를 통해 관리자는 OpenShift Container Platform의 다음 마이너 버전으로 업그레이드하기 위한 명확한 결정을 내릴 수 있습니다. 업그레이드 채널은 릴리스 선택만 제어하며 설치한 클러스터 버전에는 영향을 미치지 않습니다. OpenShift Container Platform 특정 버전의 **openshift-install** 바이너리 파일은 항상 해당 버전을 설치합니다.

OpenShift Container Platform 4.5는 다음과 같은 업그레이드 채널을 제공합니다.

- **candidate-4.5**
- **fast-4.5**
- **stable-4.5**

candidate-4.5 채널

candidate-4.5 채널에는 z-stream (4.5.z) 릴리스에 대한 후보 빌드가 포함되어 있습니다. 릴리스 후보 버전에는 제품의 모든 기능이 포함되어 있지만 지원되지는 않습니다. 릴리스 후보 버전을 사용하여 새 버전의 기능을 테스트하고 다음 OpenShift Container Platform 버전이 시스템에 적합한지 확인할 수 있습니다. 릴리스 후보는 이름에 **-rc**가 포함되어 있지 않은 후보 채널에서 사용 가능한 빌드를 말합니다. 후보 채널에서 버전을 사용할 수 있게 되면 더 많은 품질 테스트가 수행됩니다. 품질 기준을 충족하는 경우 **fast-4.5** 또는 **stable-4.5** 채널로 확장됩니다. 이로 인해 특정 릴리스가 **candidate-4.5** 채널과 **fast-4.5** 또는 **stable-4.5** 채널 모두에서 사용 가능한 경우 이는 Red Hat에서 지원되는 버전임을 의미합니다. **candidate-4.5** 채널에는 채널에 권장되는 업데이트가 없는 릴리스 버전이 포함되어 있을 수 있습니다.

candidate-4.5 채널을 사용하여 OpenShift Container Platform의 이전 마이너 버전에서 업그레이드할 수 있습니다.



참고

릴리스 후보는 <https://www.openshift.com/try> 사이트에 있는 Nightly build와 다릅니다. Nightly build는 기능에 조기 액세스하기 위해 사용할 수 있지만 Nightly build로 또는 Nightly build에서 업데이트하는 것은 권장되거나 지원되지 않습니다. 업그레이드 채널에서는 Nightly build를 사용할 수 없습니다.

fast-4.5 채널

Red Hat이 특정 버전이 공식 릴리스가 된다고 선언하면 **fast-4.5** 채널이 새로운 4.5 버전으로 업데이트됩니다. 이와 같이 이 릴리스는 완전히 지원되고, 프로덕션 환경에 적합한 품질이며, **candidate-4.5** 채널에서 릴리스 후보로 사용 가능한 동안 문제 없이 제대로 작동하는 것으로 표시됩니다. **fast-4.5** 채널에 릴리스가 표시된 후 **stable-4.5** 채널에 추가됩니다. 릴리스는 **fast-4.5** 채널에 표시되기 전에 **stable-4.5** 채널에 표시되지 않습니다.

fast-4.5 채널을 사용하여 OpenShift Container Platform의 이전 마이너스 버전에서 업그레이드할 수 있습니다.

stable-4.5 채널

에라타가 출시되면 곧 **fast-4.5** 채널에 표시되지만 릴리스는 지연 후 **stable-4.5** 채널에 추가됩니다. 이 지연 시간 동안 Connected Customer Program에 참여하는 Red Hat SRE 팀, Red Hat 지원 서비스, 사전 프로덕션 및 프로덕션 환경에서 릴리스의 안정성에 대한 데이터가 수집됩니다.

stable-4.5 채널을 사용하여 OpenShift Container Platform의 이전 바이너스 버전에서 업그레이드할 수 있습니다.

업그레이드 버전 경로

OpenShift Container Platform은 설치된 OpenShift Container Platform 버전과 다음 릴리스로 액세스하기 위해 선택한 채널의 경로를 확인할 수 있는 업그레이드 권장 서비스를 제공합니다. **fast-4.5** 채널에서는 다음을 확인할 수 있습니다.

- 4.5.0
- 4.5.1
- 4.5.3
- 4.5.4

이 서비스는 테스트를 통해 심각한 문제가 없는 업그레이드만 권장합니다. 클러스터가 4.5.1에 있고 OpenShift Container Platform에서 4.5.4를 권장하는 경우 .4.5.1에서 .4.5.4로 안전하게 업데이트할 수 있습니다. 연속적인 패치 번호에 의존하지 않도록 하십시오. 이 예에서 4.5.2는 채널에서 사용 가능하지 않습니다. 업데이트 서비스는 알려진 취약점이 포함된 OpenShift Container Platform 버전으로 업데이트할 것을 권장하지 않습니다.

업데이트의 안정성은 채널에 따라 다릅니다. **candidate-4.5** 채널에 업데이트 권장 사항이 있다고 해서 해당 업데이트가 지원되는 것은 아닙니다. 업데이트와 관련하여 아직 심각한 문제가 발견되지 않았지만 업데이트를 통한 트래픽이 많지 않아 안정성이 확인되지 않을 수 있습니다. **fast-4.5** 또는 **secure-4.5** 채널에 업데이트 권장 사항이 있음은 업데이트가 채널에 있는 동안 완전히 지원됨을 나타냅니다. 릴리스는 채널에서 제거되지 않지만 심각한 문제가 있는 업데이트 권장 사항은 모든 채널에서 제거됩니다. 업데이트 권장 사항이 제거된 후 시작된 업데이트는 지원되지 않을 수 있습니다.

Red Hat은 **fast-4.5** 또는 **stable-4.5** 채널에서 지원되는 모든 릴리스에서 4.5.z의 최신 릴리스로 지원되는 업데이트 경로를 제공합니다. 그러나 문제가 발생한 릴리스로부터 안전한 경로를 구축하고 확인하는 동안 지연이 발생할 수 있습니다.

빠르고 안정적인 채널 사용 및 전략

fast-4.5 및 **stable-4.5** 채널에서는 공식 버전이 릴리스되는 대로 이를 즉시 수신하거나 Red Hat이 해당 업데이트의 롤아웃을 제어하는 것을 허용할지 여부를 선택할 수 있습니다. 롤아웃 중 또는 이후에 문제가 발견되면 해당 버전으로의 업그레이드가 **fast-4.5** 및 **stable-4.5** 채널에서 차단될 수 있으며 새로 권장되는 업그레이드 대상의 새 버전이 도입될 수 있습니다.

고객은 **fast-4.5** 채널에서 사전 프로덕션 시스템을 설정하고, **stable-4.5** 채널에서 프로덕션 시스템을 설정한 뒤, Red Hat의 Connected Customer Program에 참여하여 고객의 프로세스를 개선할 수 있습니다. Red Hat은 이 프로그램을 사용하여 사용자의 특정 하드웨어 및 소프트웨어 설정에 대한 업데이트의 영향을 관찰합니다. 향후 릴리스에서는 업데이트가 **fast-4.5**에서 **stable-4.5** 채널로 이동하는 속도가 향상되거나 변경될 수 있습니다.

네트워크가 제한된 환경의 클러스터

OpenShift Container Platform 클러스터의 컨테이너 이미지를 직접 관리하는 경우 제품 릴리스와 관련된 Red Hat 에라타를 참조하고 업그레이드에 영향을 미치는 영향을 고려해야 합니다. 업그레이드하는 동안 사용자 인터페이스에서 이러한 버전 간 전환에 대해 경고가 표시될 수 있으므로 이러한 경고를 무시하기 전에 적절한 버전을 선택했는지 확인해야 합니다.

채널 간 전환

stable-4.5 채널에서 **fast-4.5** 채널로 변경해도 클러스터는 계속 지원됩니다. 언제든지 **candidate-4.5** 채널

로 전환할 수 있지만 해당 채널의 일부 릴리스는 지원되지 않을 수 있습니다. 현재 릴리스가 정식 사용 버전 릴리스인 경우 **candidate-4.5** 채널에서 **fast-4.5** 채널로 전환할 수 있습니다. 현재 릴리스가 최근에 **fast-4.5**로 승격된 경우 릴리스가 **stable-4.5**로 승격되기까지 최대 하루가 지연될 수 있지만 **fast-4.5** 채널에서 **stable-4.5** 채널로 전환할 수 있습니다. 현재 릴리스가 포함되지 않은 채널로 변경하면 경고가 표시되고 업데이트는 권장되지 않지만 언제든지 원래 채널로 안전하게 전환할 수 있습니다.

3.3. CLI를 사용하여 클러스터 업데이트

업데이트가 있는 경우 OpenShift CLI (**oc**)를 사용하여 클러스터를 업데이트할 수 있습니다.

사용 가능한 OpenShift Container Platform 권고 및 업데이트는 고객 포털의 [에라타 섹션](#)을 참조하십시오.

전제 조건

- 업데이트된 버전의 버전과 일치하는 일반적으로 **oc**로 알려진 OpenShift CLI (명령 줄 인터페이스) 버전을 설치합니다.
- **cluster-admin** 권한이 있는 사용자로 클러스터에 로그인합니다.
- **jq** 패키지를 설치합니다.

프로세스

1. 클러스터를 사용할 수 있는지 확인합니다.

```
$ oc get clusterversion
```

```
NAME      VERSION AVAILABLE PROGRESSING SINCE STATUS
version  4.5.0   True      False      158m Cluster version is 4.5.0
```

2. 현재 업데이트 채널 정보를 확인하고 채널이 **stable-4.5**로 설정되어 있는지 확인합니다.

```
$ oc get clusterversion -o json|jq ".items[0].spec"
```

```
{
  "channel": "stable-4.5",
  "clusterID": "990f7ab8-109b-4c95-8480-2bd1deec55ff",
  "upstream": "https://api.openshift.com/api/upgrades_info/v1/graph"
}
```



중요

프로덕션 클러스터의 경우 **stable-*** 채널에 가입해야 합니다.

3. 사용 가능한 업데이트를 확인하고 적용하려는 업데이트의 버전 번호를 기록해 둡니다.

```
$ oc adm upgrade
```

```
Cluster version is 4.1.0
```

```
Updates:
```

VERSION IMAGE

```
4.1.2 quay.io/openshift-release-dev/ocp-
release@sha256:9c5f0df8b192a0d7b46cd5f6a4da2289c155fd5302dec7954f8f06c878160b8b
```

4. 업데이트를 적용합니다.

- 최신 버전으로 업데이트하려면 다음을 수행합니다.

```
$ oc adm upgrade --to-latest=true 1
```

- 특정 버전으로 업데이트하려면 다음을 수행합니다.

```
$ oc adm upgrade --to=<version> 1
```

1 1 <version>은 이전 명령의 출력에서 얻을 수 있는 업데이트 버전입니다.

5. 클러스터 버전 Operator의 상태를 확인합니다.

```
$ oc get clusterversion -o json|jq ".items[0].spec"
{
  "channel": "stable-4.5",
  "clusterID": "990f7ab8-109b-4c95-8480-2bd1deec55ff",
  "desiredUpdate": {
    "force": false,
    "image": "quay.io/openshift-release-dev/ocp-
release@sha256:9c5f0df8b192a0d7b46cd5f6a4da2289c155fd5302dec7954f8f06c878160b8b",
    "version": "4.5.0" 1
  },
  "upstream": "https://api.openshift.com/api/upgrades_info/v1/graph"
}
```

- 1 **desiredUpdate** 부분의 **version** 번호가 지정한 값과 일치하는 경우 업데이트가 진행 중입니다.

6. 클러스터 버전 상태 기록에서 업데이트 상태를 모니터링합니다. 모든 개체가 업데이트를 완료하는 데 시간이 걸릴 수 있습니다.

```
$ oc get clusterversion -o json|jq ".items[0].status.history"
[
  {
    "completionTime": null,
    "image": "quay.io/openshift-release-dev/ocp-
release@sha256:9c5f0df8b192a0d7b46cd5f6a4da2289c155fd5302dec7954f8f06c878160b8b",
    "startedTime": "2019-06-19T20:30:50Z",
    "state": "Partial",
    "verified": true,
    "version": "4.1.2"
  },
  {
```

```

"completionTime": "2019-06-19T20:30:50Z",
"image": "quay.io/openshift-release-dev/ocp-
release@sha256:b8307ac0f3ec4ac86c3f3b52846425205022da52c16f56ec31cbe428501001d6
",
"startedTime": "2019-06-19T17:38:10Z",
"state": "Completed",
"verified": false,
"version": "4.1.0"
}
]

```

기록에는 클러스터에 적용된 최신 버전 목록이 포함되어 있습니다. 이 값은 CVO가 업데이트를 적용할 때 업데이트됩니다. 목록은 날짜순으로 정렬되며 최신 업데이트가 목록의 맨 처음에 표시됩니다. 롤아웃이 완료되면 기록의 업데이트 상태는 **Completed**로 표시되고 업데이트가 실패하거나 완료되지 않은 경우 **Partial**로 표시됩니다.



중요

업그레이드가 실패하면 Operator가 중지하고 실패한 구성 요소의 상태를 보고합니다. 클러스터를 이전 버전으로 롤백하는 것은 지원되지 않습니다. 업그레이드에 실패할 경우 Red Hat 지원에 문의하십시오.

- 업데이트가 완료되면 클러스터 버전이 새 버전으로 업데이트되었는지 확인합니다.

```
$ oc get clusterversion
```

```

NAME      VERSION  AVAILABLE  PROGRESSING  SINCE   STATUS
version  4.5.0    True       False        2m     Cluster version is 4.5.0

```

4장. RHEL 컴퓨팅 시스템을 포함하는 클러스터 업데이트

OpenShift Container Platform 클러스터를 업데이트하거나 업그레이드할 수 있습니다. 클러스터에 RHEL (Red Hat Enterprise Linux) 시스템이 포함된 경우 해당 시스템을 업데이트하기 위해 추가 단계를 수행해야 합니다.

전제 조건

- **admin** 권한이 있는 사용자로 클러스터에 액세스합니다. [Using RBAC to define and apply permissions](#)을 참조하십시오.
- 업그레이드에 실패할 경우 **etcd backup**이 있어야 하고 **클러스터를 이전 상태로 복원**해야 합니다.

4.1. OPENSIFT CONTAINER PLATFORM 업데이트 서비스 정보

OpenShift Container Platform 업데이트 서비스는 OpenShift Container Platform 및 RHCOS (Red Hat Enterprise Linux CoreOS) 모두에 무선(OTA, Over-the-air) 업데이트를 제공하는 호스팅 서비스입니다. 구성 요소 Operator의 **경로**와 이를 연결하는 **에지**를 포함하는 그래프 또는 다이어그램을 제공합니다. 그래프의 에지에는 안전하게 업데이트할 수 있는 버전이 표시되고 정점은 관리되는 클러스터 구성 요소의 상태를 확인하는 업데이트 페이로드입니다.

클러스터의 CVO (Cluster Version Operator)는 OpenShift Container Platform 업데이트 서비스를 확인하여 현재 구성 요소 버전 및 그래프의 정보를 기반으로 유효한 업데이트 및 업데이트 경로를 확인합니다. 업데이트를 요청하면 OpenShift Container Platform CVO는 해당 업데이트의 릴리스 이미지를 사용하여 클러스터를 업그레이드합니다. 릴리스 아티팩트는 Quay에서 컨테이너 이미지로 호스팅됩니다.

OpenShift Container Platform 업데이트 서비스가 호환 가능한 업데이트 만 제공할 수 있도록 자동화를 지원하는 버전 확인 파이프 라인이 제공됩니다. 각 릴리스 아티팩트는 지원되는 클라우드 플랫폼 및 시스템 아키텍처 및 기타 구성 요소 패키지와의 호환성 여부를 확인합니다. 파이프 라인에서 적용 가능한 버전이 있음을 확인한 후 OpenShift Container Platform 업데이트 서비스는 해당 버전 업데이트를 사용할 수 있음을 알려줍니다.



중요

업데이트 서비스는 모든 유효한 업데이트를 표시하므로 업데이트 서비스가 표시하지 않는 버전으로 강제 업데이트할 수 없습니다.

연속 업데이트 모드에서는 두 개의 컨트롤러가 실행됩니다. 하나의 컨트롤러는 페이로드 매니페스트를 지속적으로 업데이트하여 클러스터에 적용한 다음 사용 가능한지, 업그레이드했는지 또는 실패했는지에 따라 Operator의 제어된 롤아웃 상태를 출력합니다. 두 번째 컨트롤러는 OpenShift Container Platform 업데이트 서비스를 폴링하여 업데이트를 사용할 수 있는지 확인합니다.



중요

클러스터를 이전 버전으로 되돌리거나 롤백을 수행하는 것은 지원되지 않습니다. 최신 버전으로의 업그레이드 만 지원됩니다.

업그레이드 프로세스 중에 MCO (Machine Config Operator)는 새 설정을 클러스터 시스템에 적용합니다. 이는 시스템 설정 폴의 **maxUnavailable** 필드에 지정된 노드 수를 제한하고 이를 사용할 수 없는 것으로 표시합니다. 기본적으로 이 값은 **1**로 설정됩니다. 그 다음 새 설정을 적용하여 컴퓨터를 다시 시작합니다. RHEL (Red Hat Enterprise Linux) 시스템을 작업자로 사용하는 경우 먼저 OpenShift API를 업데이트해야 하기 때문에 MCO는 이 시스템에서 kubelet을 업데이트하지 않습니다. 새 버전의 사양이 이전 kubelet에

적용되므로 RHEL 시스템을 **Ready** 상태로 되돌릴 수 없습니다. 컴퓨터를 사용할 수 있을 때까지 업데이트를 완료할 수 없습니다. 그러나 사용 불가능한 최대 노드 수를 설정하면 사용할 수 없는 시스템의 수가 이 값을 초과하지 않는 경우에도 정상적인 클러스터 작업을 계속할 수 있습니다.

추가 리소스

- [관리되지 않는 Operator에 대한 지원 정책](#)

4.2. OPENSIFT CONTAINER PLATFORM 업그레이드 채널 및 릴리스 버전

OpenShift Container Platform 4.1에서 Red Hat은 클러스터 업그레이드에 적합한 버전을 권장하기 위해 업그레이드 채널 개념을 도입했습니다. 업그레이드 속도를 제어함으로써 이러한 업그레이드 채널을 통해 업그레이드 전략을 선택할 수 있습니다. 업그레이드 채널은 OpenShift Container Platform의 마이너 버전과 연결되어 있습니다. 예를 들어 OpenShift Container Platform 4.5 업그레이드 채널에는 4.5 릴리스 버전으로의 업그레이드가 포함되어 있지 않습니다. 이를 통해 관리자는 OpenShift Container Platform의 다음 마이너 버전으로 업그레이드하기 위한 명확한 결정을 내릴 수 있습니다. 업그레이드 채널은 릴리스 선택만 제어하며 설치한 클러스터 버전에는 영향을 미치지 않습니다. OpenShift Container Platform 특정 버전의 **openshift-install** 바이너리 파일은 항상 해당 버전을 설치합니다.

OpenShift Container Platform 4.5는 다음과 같은 업그레이드 채널을 제공합니다.

- **candidate-4.5**
- **fast-4.5**
- **stable-4.5**

candidate-4.5 채널

candidate-4.5 채널에는 z-stream (4.5.z) 릴리스에 대한 후보 빌드가 포함되어 있습니다. 릴리스 후보 버전에는 제품의 모든 기능이 포함되어 있지만 지원되지는 않습니다. 릴리스 후보 버전을 사용하여 새 버전의 기능을 테스트하고 다음 OpenShift Container Platform 버전이 시스템에 적합한지 확인할 수 있습니다. 릴리스 후보는 이름에 **-rc**가 포함되어 있지 않은 후보 채널에서 사용 가능한 빌드를 말합니다. 후보 채널에서 버전을 사용할 수 있게 되면 더 많은 품질 테스트가 수행됩니다. 품질 기준을 충족하는 경우 **fast-4.5** 또는 **stable-4.5** 채널로 확장됩니다. 이로 인해 특정 릴리스가 **candidate-4.5** 채널과 **fast-4.5** 또는 **stable-4.5** 채널 모두에서 사용 가능한 경우 이는 Red Hat에서 지원되는 버전임을 의미합니다.

candidate-4.5 채널에는 채널에 권장되는 업데이트가 없는 릴리스 버전이 포함되어 있을 수 있습니다.

candidate-4.5 채널을 사용하여 OpenShift Container Platform의 이전 마이너 버전에서 업그레이드할 수 있습니다.



참고

릴리스 후보는 <https://www.openshift.com/try> 사이트에 있는 Nightly build와 다릅니다. Nightly build는 기능에 조기 액세스하기 위해 사용할 수 있지만 Nightly build로 또는 Nightly build에서 업데이트하는 것은 권장되거나 지원되지 않습니다. 업그레이드 채널에서는 Nightly build를 사용할 수 없습니다.

fast-4.5 채널

Red Hat이 특정 버전이 공식 릴리스가 된다고 선언하면 **fast-4.5** 채널이 새로운 4.5 버전으로 업데이트됩니다. 이와 같이 이 릴리스는 완전히 지원되고, 프로덕션 환경에 적합한 품질이며, **candidate-4.5** 채널에서 릴리스 후보로 사용 가능한 동안 문제 없이 제대로 작동하는 것으로 표시됩니다. **fast-4.5** 채널에 릴리스가 표시된 후 **stable-4.5** 채널에 추가됩니다. 릴리스는 **fast-4.5** 채널에 표시되기 전에 **stable-4.5** 채널에 표시되지 않습니다.

fast-4.5 채널을 사용하여 OpenShift Container Platform의 이전 마이너스 버전에서 업그레이드할 수 있습니다.

stable-4.5 채널

에라타가 출시되면 곧 **fast-4.5** 채널에 표시되지만 릴리스는 지연 후 **stable-4.5** 채널에 추가됩니다. 이 지연 시간 동안 Connected Customer Program에 참여하는 Red Hat SRE 팀, Red Hat 지원 서비스, 사전 프로덕션 및 프로덕션 환경에서 릴리스의 안정성에 대한 데이터가 수집됩니다.

stable-4.5 채널을 사용하여 OpenShift Container Platform의 이전 마이너스 버전에서 업그레이드할 수 있습니다.

업그레이드 버전 경로

OpenShift Container Platform은 설치된 OpenShift Container Platform 버전과 다음 릴리스로 액세스하기 위해 선택한 채널의 경로를 확인할 수 있는 업그레이드 권장 서비스를 제공합니다. **fast-4.5** 채널에서는 다음을 확인할 수 있습니다.

- 4.5.0
- 4.5.1
- 4.5.3
- 4.5.4

이 서비스는 테스트를 통해 심각한 문제가 없는 업그레이드만 권장합니다. 클러스터가 4.5.1에 있고 OpenShift Container Platform에서 4.5.4를 권장하는 경우 .4.5.1에서 .4.5.4로 안전하게 업데이트할 수 있습니다. 연속적인 패치 번호에 의존하지 않도록 하십시오. 이 예에서 4.5.2는 채널에서 사용 가능하지 않습니다. 업데이트 서비스는 알려진 취약점이 포함된 OpenShift Container Platform 버전으로 업데이트할 것을 권장하지 않습니다.

업데이트의 안정성은 채널에 따라 다릅니다. **candidate-4.5** 채널에 업데이트 권장 사항이 있다고 해서 해당 업데이트가 지원되는 것은 아닙니다. 업데이트와 관련하여 아직 심각한 문제가 발견되지 않았지만 업데이트를 통한 트래픽이 많지 않아 안정성이 확인되지 않을 수 있습니다. **fast-4.5** 또는 **secure-4.5** 채널에 업데이트 권장 사항이 있음은 업데이트가 채널에 있는 동안 완전히 지원됨을 나타냅니다. 릴리스는 채널에서 제거되지 않지만 심각한 문제가 있는 업데이트 권장 사항은 모든 채널에서 제거됩니다. 업데이트 권장 사항이 제거된 후 시작된 업데이트는 지원되지 않을 수 있습니다.

Red Hat은 **fast-4.5** 또는 **stable-4.5** 채널에서 지원되는 모든 릴리스에서 4.5.z의 최신 릴리스로 지원되는 업데이트 경로를 제공합니다. 그러나 문제가 발생한 릴리스로부터 안전한 경로를 구축하고 확인하는 동안 지연이 발생할 수 있습니다.

빠르고 안정적인 채널 사용 및 전략

fast-4.5 및 **stable-4.5** 채널에서는 공식 버전이 릴리스되는 대로 이를 즉시 수신하거나 Red Hat이 해당 업데이트의 롤아웃을 제어하는 것을 허용할지 여부를 선택할 수 있습니다. 롤아웃 중 또는 이후에 문제가 발견되면 해당 버전으로의 업그레이드가 **fast-4.5** 및 **stable-4.5** 채널에서 차단될 수 있으며 새로 권장되는 업그레이드 대상의 새 버전이 도입될 수 있습니다.

고객은 **fast-4.5** 채널에서 사전 프로덕션 시스템을 설정하고, **stable-4.5** 채널에서 프로덕션 시스템을 설정한 뒤, Red Hat의 Connected Customer Program에 참여하여 고객의 프로세스를 개선할 수 있습니다. Red Hat은 이 프로그램을 사용하여 사용자의 특정 하드웨어 및 소프트웨어 설정에 대한 업데이트의 영향을 관찰합니다. 향후 릴리스에서는 업데이트가 **fast-4.5**에서 **stable-4.5** 채널로 이동하는 속도가 향상되거나 변경될 수 있습니다.

네트워크가 제한된 환경의 클러스터

OpenShift Container Platform 클러스터의 컨테이너 이미지를 직접 관리하는 경우 제품 릴리스와 관련된 Red Hat 에라타를 참조하고 업그레이드에 영향을 미치는 영향을 고려해야 합니다. 업그레이드하는 동안 사용자 인터페이스에서 이러한 버전 간 전환에 대해 경고가 표시될 수 있으므로 이러한 경고를 무시하기

전에 적절한 버전을 선택했는지 확인해야 합니다.

채널 간 전환

stable-4.5 채널에서 **fast-4.5** 채널로 변경해도 클러스터는 계속 지원됩니다. 언제든지 **candidate-4.5** 채널로 전환할 수 있지만 해당 채널의 일부 릴리스는 지원되지 않을 수 있습니다. 현재 릴리스가 정식 사용 버전 릴리스인 경우 **candidate-4.5** 채널에서 **fast-4.5** 채널로 전환할 수 있습니다. 현재 릴리스가 최근에 **fast-4.5**로 승격된 경우 릴리스가 **stable-4.5**로 승격되기까지 최대 하루가 지연될 수 있지만 **fast-4.5** 채널에서 **stable-4.5** 채널로 전환할 수 있습니다. 현재 릴리스가 포함되지 않은 채널로 변경하면 경고가 표시되고 업데이트는 권장되지 않지만 언제든지 원래 채널로 안전하게 전환할 수 있습니다.

4.3. 웹 콘솔을 사용하여 클러스터 업데이트

사용 가능한 업데이트가 있으면 웹 콘솔에서 클러스터를 업데이트할 수 있습니다.

사용 가능한 OpenShift Container Platform 권고 및 업데이트는 고객 포털의 [에라타 섹션](#)을 참조하십시오.

전제 조건

- **admin** 권한이 있는 사용자로 웹 콘솔에 액세스합니다.

프로세스

1. 웹 콘솔에서 **Administration > Cluster Settings**을 클릭하고 **Overview** 탭의 내용을 확인합니다.
2. 프로덕션 클러스터의 경우, **CHANNEL**이 업데이트하려는 버전에 대해 현재 마이너 버전 (예: **steady-4.5**)의 올바른 채널로 설정되어 있는지 확인합니다.



중요

프로덕션 클러스터의 경우 **stable-*** 또는 **fast-*** 채널에 가입해야 합니다.

- **UPDATE STATUS**가 **Updates Available**가 아닌 경우 클러스터를 업그레이드할 수 없습니다.
 - **DESIRED VERSION**은 클러스터가 실행 중이거나 업데이트 중인 클러스터 버전을 나타냅니다.
3. **Updates Available**을 클릭하여 사용 가능한 최신 버전을 선택하고 **Update**를 클릭합니다. **UPDATE STATUS**는 **Updating**으로 변경하고 **Cluster Operators** 탭에서 Operator 업그레이드 진행 상황을 확인할 수 있습니다.
 4. 업데이트가 완료되고 Cluster Version Operator가 사용 가능한 업데이트를 새로 고침한 후 현재 채널에서 사용 가능한 추가 업데이트가 있는지 확인합니다.
 - 업데이트가 있는 경우 더 이상 업데이트할 수 없을 때까지 현재 채널에서 업데이트를 계속 수행합니다.
 - 사용할 수 있는 업데이트가 없는 경우, **CHANNEL**을 다음 마이너 버전의 **stable-*** 또는 **fast-*** 채널로 변경하고 해당 채널에서 원하는 버전으로 업데이트합니다.

필요한 버전에 도달할 때까지 여러 중간 업데이트를 수행해야 할 수도 있습니다.



참고

RHEL (Red Hat Enterprise Linux) 작업자 시스템이 포함된 클러스터를 업데이트 하면 업데이트 프로세스 중에 해당 작업자를 일시적으로 사용할 수 없게 됩니다. 클러스터가 **NotReady** 상태가 되면 각 RHEL 시스템에 대해 업그레이드 플레이 북을 실행하여 업데이트를 완료해야 합니다.

4.4. (선택 사항) RHEL 시스템에서 ANSIBLE 작업을 실행하기 위한 후크 추가

OpenShift Container Platform을 업데이트할 때 후크를 사용하여 RHEL 컴퓨팅 시스템에서 Ansible 작업을 실행할 수 있습니다.

4.4.1. 업그레이드를 위한 Ansible Hook 사용 정보

OpenShift Container Platform을 업데이트할 때 후크를 사용하여 특정 작업 중에 RHEL (Red Hat Enterprise Linux) 노드에서 사용자 정의 작업을 실행할 수 있습니다. 후크를 사용하면 특정 업데이트 작업 전후에 실행할 작업을 정의하는 파일을 지정할 수 있습니다. OpenShift Container Platform 클러스터에서 RHEL 컴퓨팅 노드를 업데이트할 때 후크를 사용하여 사용자 정의 인프라의 유효성을 검사하거나 변경할 수 있습니다.

후크가 실패하면 작업도 실패하므로 후크를 여러 번 실행하고 동일한 결과를 얻도록 설계해야 합니다.

후크에는 다음과 같은 제한이 있습니다.-후크에는 정의되거나 버전이 지정된 인터페이스가 없습니다. 후크는 내부 **openshift-ansible** 변수를 사용할 수 있지만 이러한 변수는 향후 OpenShift Container Platform 릴리스에서 수정되거나 제거될 수 있습니다.-후크에는 오류 처리 기능이 없으므로 후크에 오류가 발생하면 업데이트 프로세스가 중단됩니다. 오류가 발생하면 문제를 해결한 다음 업그레이드를 다시 시작해야 합니다.

4.4.2. 후크를 사용하도록 Ansible 인벤토리 파일 설정

all : vars 섹션 아래의 **hosts** 인벤토리 파일에서 작업자 시스템이라고도 하는 RHEL (Red Hat Enterprise Linux) 컴퓨팅 시스템을 업데이트할 때 사용할 후크를 정의합니다.

전제 조건

- RHEL 컴퓨팅 시스템 클러스터를 추가하는 데 사용되는 컴퓨터에 액세스할 수 있어야 합니다. RHEL 시스템을 정의하는 **hosts** Ansible 인벤토리 파일에 액세스할 수 있어야 합니다.

프로세스

1. 후크를 설계한 후 Ansible 작업을 정의하는 YAML 파일을 만듭니다. 이 파일은 다음 예와 같이 Playbook이 아닌 일련의 작업으로 구성되어 있어야 합니다.

```
---
# Trivial example forcing an operator to acknowledge the start of an upgrade
# file=/home/user/openshift-ansible/hooks/pre_compute.yml

- name: note the start of a compute machine update
  debug:
    msg: "Compute machine upgrade of {{ inventory_hostname }} is about to start"

- name: require the user agree to start an upgrade
  pause:
    prompt: "Press Enter to start the compute machine update"
```

2. **hosts** Ansible 인벤토리 파일을 수정하여 후크 파일을 지정합니다. 후크 파일은 다음과 같이 **[all : vars]** 섹션에서 매개 변수 값으로 지정됩니다.

인벤토리 파일에서 후크 정의의 예

```
[all:vars]
openshift_node_pre_upgrade_hook=/home/user/openshift-ansible/hooks/pre_node.yml
openshift_node_post_upgrade_hook=/home/user/openshift-ansible/hooks/post_node.yml
```

후크 경로의 모호성을 피하려면 정의에서 상대 경로 대신 절대 경로를 사용합니다.

4.4.3. RHEL 컴퓨팅 시스템에서 사용 가능한 후크

OpenShift Container Platform 클러스터에서 RHEL (Red Hat Enterprise Linux) 컴퓨팅 시스템을 업데이트 할 때 다음 후크를 사용할 수 있습니다.

후크 이름	설명
openshift_node_pre_cordon_hook	<ul style="list-style-type: none"> ● 각 노드가 차단 (cordon)되기 이전에 실행됩니다. ● 이 후크는 각 노드에 대해 연속적으로 실행됩니다. ● 작업이 다른 호스트에서 실행되어야 하는 경우 해당 작업은 delegate_to 또는 local_action을 사용해야 합니다.
openshift_node_pre_upgrade_hook	<ul style="list-style-type: none"> ● 각 노드가 차단(cordon)된 후 업데이트되기 전에 실행됩니다. ● 이 후크는 각 노드에 대해 연속적으로 실행됩니다. ● 작업이 다른 호스트에서 실행되어야 하는 경우 해당 작업은 delegate_to 또는 local_action을 사용해야 합니다.
openshift_node_pre_uncordon_hook	<ul style="list-style-type: none"> ● 각 노드가 업데이트 된 후 차단 해제 (uncordon)되기 전에 실행됩니다. ● 이 후크는 각 노드에 대해 연속적으로 실행됩니다. ● 작업이 다른 호스트에서 실행되어야 하는 경우, 해당 작업은 delegate_to 또는 local_action을 사용해야 합니다.

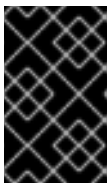
후크 이름	설명
openshift_node_post_upgrade_hook	<ul style="list-style-type: none"> ● 각 노드가 차단 해제 (uncordon) 후에 실행됩니다. 이는 마지막 노드 업데이트 작업입니다. ● 이 후크는 각 노드에 대해 연속적으로 실행됩니다. ● 작업이 다른 호스트에서 실행되어야 하는 경우 해당 작업은 delegate_to 또는 local_action 을 사용해야 합니다.

4.5. 클러스터에서 RHEL 컴퓨팅 시스템 업데이트

클러스터를 업데이트한 후 클러스터의 RHEL (Red Hat Enterprise Linux) 컴퓨팅 시스템을 업데이트해야 합니다.

전제 조건

- 클러스터가 업데이트되었습니다.



중요

RHEL 시스템에는 업데이트 프로세스를 완료하기 위해 클러스터에서 생성된 자산이 필요하므로 RHEL 컴퓨팅 시스템을 업데이트하기 전에 클러스터를 업데이트해야 합니다.

- RHEL 컴퓨팅 시스템 클러스터를 추가하는 데 사용되는 컴퓨터에 액세스할 수 있어야 합니다. RHEL 시스템 및 **upgrade** Playbook을 정의하는 **hosts** Ansible 인벤토리 파일에 액세스할 수 있어야 합니다.

프로세스

1. 호스트에서 firewalld를 중지하고 비활성화합니다.

```
# systemctl disable --now firewalld.service
```



참고

나중에 firewalld를 활성화할 수 없습니다. 활성화하면 작업자의 OpenShift Container Platform 로그에 액세스할 수 없습니다.

2. OpenShift Container Platform 4.5에 필요한 리포지토리를 활성화합니다.
 - a. Ansible Playbook을 실행하는 컴퓨터에서 필요한 리포지토리를 업데이트합니다.

```
# subscription-manager repos --disable=rhel-7-server-ose-4.4-rpms \
    --enable=rhel-7-server-ansible-2.9-rpms \
    --enable=rhel-7-server-ose-4.5-rpms
```

- b. Ansible Playbook을 실행하는 시스템에서 **openshift-ansible**을 포함하여 필요한 패키지를 업데이트합니다.

```
# yum update openshift-ansible openshift-clients
```

- c. 각 RHEL 컴퓨팅 노드에 필요한 리포지토리를 업데이트합니다.

```
# subscription-manager repos --disable=rhel-7-server-ose-4.4-rpms \
--enable=rhel-7-server-ose-4.5-rpms
```

3. RHEL 작업자 시스템을 업데이트합니다.

- a. 현재 노드 상태를 확인하고 업데이트할 RHEL 작업자를 결정합니다.

```
# oc get node
NAME                                STATUS                                ROLES  AGE  VERSION
mycluster-control-plane-0          Ready                                master 145m v1.18.3
mycluster-control-plane-1          Ready                                master 145m v1.18.3
mycluster-control-plane-2          Ready                                master 145m v1.18.3
mycluster-rhel7-0                   NotReady,SchedulingDisabled        worker 98m
v1.14.6+97c81d00e
mycluster-rhel7-1                   Ready                                worker 98m  v1.14.6+97c81d00e
mycluster-rhel7-2                   Ready                                worker 98m  v1.14.6+97c81d00e
mycluster-rhel7-3                   Ready                                worker 98m  v1.14.6+97c81d00e
```

NotReady, SchedulingDisabled 상태인 시스템을 확인합니다.

- b. 다음 예와 같이 `/<path>/inventory/hosts`에서 Ansible 인벤토리 파일을 확인하고 **NotReady, SchedulingDisabled** 상태인 시스템만 **[workers]** 섹션에 나열되도록 내용을 업데이트합니다.

```
[all:vars]
ansible_user=root
#ansible_become=True

openshift_kubeconfig_path=~/.kube/config"

[workers]
mycluster-rhel7-0.example.com
```

- c. **openshift-ansible** 디렉토리로 변경하고 **upgrade** Playbook을 실행합니다.

```
$ cd /usr/share/ansible/openshift-ansible
$ ansible-playbook -i /<path>/inventory/hosts playbooks/upgrade.yml 1
```

- 1 **<path>**에 대해 생성한 Ansible 인벤토리 파일의 경로를 지정합니다.

4. 이전 단계의 프로세스에 따라 클러스터의 각 RHEL 작업자 시스템을 업데이트합니다.

5. 모든 작업자를 업데이트한 후 모든 클러스터 노드가 새 버전으로 업데이트되었는지 확인합니다.

```
# oc get node
NAME                                STATUS                                ROLES  AGE  VERSION
mycluster-control-plane-0          Ready                                master 145m v1.18.3
```

mycluster-control-plane-1	Ready	master	145m	v1.18.3
mycluster-control-plane-2	Ready	master	145m	v1.18.3
mycluster-rhel7-0	NotReady,SchedulingDisabled	worker	98m	v1.18.3
mycluster-rhel7-1	Ready	worker	98m	v1.18.3
mycluster-rhel7-2	Ready	worker	98m	v1.18.3
mycluster-rhel7-3	Ready	worker	98m	v1.18.3

5장. 네트워크가 제한된 환경에서 클러스터 업데이트

oc 명령행 인터페이스 (CLI)를 사용하여 네트워크가 제한된 환경에서 OpenShift Container Platform 클러스터를 업그레이드할 수 있습니다.

네트워크가 제한된 환경은 클러스터 노드가 인터넷에 액세스할 수 없는 환경입니다. 따라서 레지스트리에 설치 이미지를 입력해야 합니다. 레지스트리 호스트가 인터넷과 클러스터에 모두에 액세스할 수 없는 경우 이미지를 해당 환경에서 분리된 파일 시스템으로 미리링한 다음 호스트 또는 이동식 미디어를 가져올 수 있습니다. 로컬 컨테이너 레지스트리와 클러스터가 미리 레지스트리의 호스트에 연결된 경우 릴리스 이미지를 로컬 레지스트리로 직접 푸시할 수 있습니다.

네트워크가 제한된 환경에 여러 개의 클러스터가 있는 경우 필요한 릴리스 이미지를 단일 컨테이너 이미지 레지스트리에 미리링하고 해당 레지스트리를 사용하여 모든 클러스터를 업데이트합니다.

전제 조건

- 필요한 컨테이너 이미지를 얻으려면 인터넷에 액세스할 수 있어야 합니다.
- 네트워크가 제한된 환경에서 컨테이너 레지스트리에 대한 쓰기 권한이 있어야 이미지를 푸시하고 가져올 수 있습니다. 컨테이너 레지스트리는 Docker 레지스트리 API v2와 호환되어야 합니다.
- **oc** 명령 줄 인터페이스 (CLI) 툴이 설치되어 있어야 합니다.
- **admin** 권한이 있는 사용자로 클러스터에 액세스합니다. [Using RBAC to define and apply permissions](#)을 참조하십시오.
- 업그레이드에 실패할 경우 **etcd backup**이 있어야 하고 클러스터를 이전 상태로 복원해야 합니다.

5.1. 미리 호스트 준비

미리 단계를 수행하기 전에 호스트는 콘텐츠를 검색하고 원격 위치로 푸시할 준비가 되어 있어야 합니다.

5.1.1. 바이너리를 다운로드하여 CLI 설치

명령 줄 인터페이스를 사용하여 OpenShift Container Platform과 상호 작용하기 위해 OpenShift CLI (**oc**)를 설치할 수 있습니다. Linux, Windows 또는 macOS에 **oc**를 설치할 수 있습니다.



중요

이전 버전의 **oc**를 설치한 경우, OpenShift Container Platform 4.5의 모든 명령을 완료하는 데 해당 버전을 사용할 수 없습니다. 새 버전의 **oc**를 다운로드하여 설치합니다. 네트워크가 제한된 환경에서 클러스터를 업그레이드하는 경우 업그레이드하려는 **oc** 버전을 설치합니다.

5.1.1.1. Linux에서 CLI 설치

다음 단계를 사용하여 Linux에서 OpenShift CLI (**oc**) 바이너리를 설치할 수 있습니다.

프로세스

1. Red Hat OpenShift Cluster Manager 사이트의 [Infrastructure Provider](#) 페이지로 이동합니다.
2. 인프라 제공 업체 (해당되는 경우)를 선택하고 설치 유형을 선택합니다.

3. **Command-line interface** 섹션의 드롭다운 메뉴에서 **Linux** 를 선택하고 **Download command-line tools**를 클릭합니다.
4. 아카이브의 압축을 풉니다.

```
$ tar xvzf <file>
```

5. **oc** 바이너리를 **PATH**에 있는 디렉토리에 배치합니다.
PATH를 확인하려면 다음 명령을 실행합니다.

```
$ echo $PATH
```

CLI를 설치한 후 **oc** 명령을 사용할 수 있습니다.

```
$ oc <command>
```

5.1.1.2. Windows에서 CLI 설치

다음 프로세스를 사용하여 Windows에 OpenShift CLI (**oc**) 바이너리를 설치할 수 있습니다.

프로세스

1. Red Hat OpenShift Cluster Manager 사이트의 [Infrastructure Provider](#) 페이지로 이동합니다.
2. 인프라 제공 업체 (해당되는 경우)를 선택하고 설치 유형을 선택합니다.
3. **Command-line interface** 섹션의 드롭다운 메뉴에서 **Windows**를 선택하고 **Download command-line tools**를 클릭합니다.
4. ZIP 프로그램으로 아카이브의 압축을 풉니다.
5. **oc** 바이너리를 **PATH**에 있는 디렉토리로 이동합니다.
PATH를 확인하려면 명령 프롬프트를 열고 다음 명령을 실행합니다.

```
C:\> path
```

CLI를 설치한 후 **oc** 명령을 사용할 수 있습니다.

```
C:\> oc <command>
```

5.1.1.3. macOS에 CLI 설치

다음 프로세스에 따라 macOS에서 OpenShift CLI (**oc**) 바이너리를 설치할 수 있습니다.

프로세스

1. Red Hat OpenShift Cluster Manager 사이트의 [Infrastructure Provider](#) 페이지로 이동합니다.
2. 인프라 제공 업체 (해당되는 경우)를 선택하고 설치 유형을 선택합니다.
3. **Command-line interface** 섹션의 드롭다운 메뉴에서 **MacOS**를 선택하고 **Download command-line tools**를 클릭합니다.

4. 아카이브의 압축을 해제하고 압축을 풉니다.
5. **oc** 바이너리 PATH의 디렉토리로 이동합니다.
PATH를 확인하려면 터미널을 열고 다음 명령을 실행합니다.

```
$ echo $PATH
```

CLI를 설치한 후 **oc** 명령을 사용할 수 있습니다.

```
$ oc <command>
```

5.2. 이미지를 미러링할 수 있는 인증 정보 설정

Red Hat에서 미러로 이미지를 미러링할 수 있는 컨테이너 이미지 레지스트리 인증 정보 파일을 생성합니다.



주의

클러스터를 설치할 때 이 이미지 레지스트리 인증 정보 파일을 풀 시크릿(pull secret)으로 사용하지 마십시오. 클러스터를 설치할 때 이 파일을 지정하면 클러스터의 모든 시스템에 미러 레지스트리에 대한 쓰기 권한이 부여됩니다.



주의

이 프로세스에서는 미러 레지스트리의 컨테이너 이미지 레지스트리에 대한 쓰기 권한이 있어야 하며 인증 정보를 레지스트리 풀 시크릿에 추가해야 합니다.



중요

클러스터를 설치할 때 이 이미지 레지스트리 인증 정보 파일을 풀 시크릿(pull secret)으로 사용하지 마십시오. 클러스터를 설치할 때 이 파일을 지정하면 클러스터의 모든 시스템에 미러 레지스트리에 대한 쓰기 권한이 부여됩니다.

전제 조건

- 네트워크가 제한된 환경에서 사용하도록 미러 레지스트리가 설정되어 있습니다.
- 미러 레지스트리에서 이미지를 미러링할 이미지 저장소 위치를 확인했습니다.
- 이미지를 해당 이미지 저장소에 업로드할 수 있는 미러 레지스트리 계정을 제공하고 있습니다.

프로세스

설치 호스트에서 다음 단계를 수행합니다.

1. Red Hat OpenShift Cluster Manager 사이트의 [Pull Secret](#) 페이지에서 **registry.redhat.io** pull secret을 다운로드하여 **.json** 파일에 저장합니다.
2. 다음 명령을 사용하여 레지스트리에 로그인합니다.

```
$ oc registry login --to ./pull-secret.json --registry "<registry_host_and_port>"
```

레지스트리의 사용자 이름 및 비밀번호를 입력합니다.

5.3. OPENSIFT CONTAINER PLATFORM 이미지 저장소 미러링

네트워크가 제한된 환경에서 프로비저닝하는 인프라에서 클러스터를 업그레이드하기 전에 필요한 컨테이너 이미지를 해당 환경에 미러링해야 합니다. 이 프로세스를 무제한 네트워크에서 사용하여 클러스터가 외부 콘텐츠에 대해 조직의 제어 조건을 충족하는 컨테이너 이미지만 사용하도록 할 수 있습니다.

프로세스

1. [OpenShift Container Platform 업그레이드 경로](#)를 확인하고 현재 클러스터 버전과 예약된 클러스터 버전 간의 업그레이드 경로가 있는지 확인합니다.
2. 필요한 환경 변수를 설정합니다.

```
$ OCP_RELEASE=<release_version> 1
$ LOCAL_REGISTRY=<local_registry_host_name>:<local_registry_host_port>' 2
$ LOCAL_REPOSITORY=<local_repository_name>' 3
$ PRODUCT_REPO='openshift-release-dev' 4
$ LOCAL_SECRET_JSON=<path_to_pull_secret>' 5
$ RELEASE_NAME='ocp-release' 6
$ ARCHITECTURE=<server_architecture> 7
$ REMOVABLE_MEDIA_PATH=<path> 8
```

- 1 **<release_version>**에 대해 업그레이드할 OpenShift Container Platform 버전에 해당하는 태그를 지정합니다 (예: **4.5.0**).
- 2 **<local_registry_host_name>**의 경우 미러 저장소의 레지스트리 도메인 이름을 지정하고 **<local_registry_host_port>**의 경우 콘텐츠를 제공하는 데 사용되는 포트를 지정합니다.
- 3 **<repository_name>**의 경우 레지스트리에 작성할 저장소 이름 (예: **ocp4/openshift4**)을 지정합니다.
- 4 미러링할 저장소입니다. 프로덕션 환경의 릴리스의 경우 **openshift-release-dev**를 지정해야 합니다.
- 5 **<path_to_pull_secret>**에 대해 생성한 미러 레지스트리에 대한 풀 시크릿의 절대 경로 및 파일 이름을 지정합니다.
- 6 프로덕션 환경의 릴리스의 경우 **ocp-release**를 지정해야 합니다.
- 7 **<server_architecture>**에 대해 **x86_64**와 같은 서버 아키텍처를 지정합니다.
- 8 **<path>**에 대해 미러링된 이미지를 호스트할 디렉토리의 경로를 지정합니다.

3. 미러링할 이미지 및 설정 매니페스트를 확인합니다.

```
$ oc adm release mirror -a ${LOCAL_SECRET_JSON} --to-dir=${REMOVABLE_MEDIA_PATH}/mirror quay.io/${PRODUCT_REPO}/${RELEASE_NAME}:${OCP_RELEASE}-${ARCHITECTURE} --dry-run
```

4. 버전 이미지를 내부 컨테이너 레지스트리에 미러링합니다.

- 미리 호스트가 인터넷에 액세스할 수 없는 경우 다음 작업을 수행합니다.
 - i. 이동식 미디어를 인터넷에 연결된 시스템에 연결합니다.
 - ii. 이미지 및 설정 매니페스트를 이동식 미디어의 디렉토리에 미러링합니다.

```
$ oc adm release mirror -a ${LOCAL_SECRET_JSON} --to-dir=${REMOVABLE_MEDIA_PATH}/mirror quay.io/${PRODUCT_REPO}/${RELEASE_NAME}:${OCP_RELEASE}-${ARCHITECTURE}
```

- iii. 미디어를 네트워크가 제한된 환경으로 가져와서 이미지를 로컬 컨테이너 레지스트리에 업로드합니다.

```
$ oc image mirror -a ${LOCAL_SECRET_JSON} --from-dir=${REMOVABLE_MEDIA_PATH}/mirror 'file://openshift/release:${OCP_RELEASE}*' ${LOCAL_REGISTRY}/${LOCAL_REPOSITORY}
```

- 로컬 컨테이너 레지스트리와 클러스터가 미리 호스트에 연결된 경우 릴리스 이미지를 로컬 레지스트리로 직접 푸시하고 다음 명령을 사용하여 ConfigMap을 클러스터에 적용합니다.

```
$ oc adm release mirror -a ${LOCAL_SECRET_JSON} --from=quay.io/${PRODUCT_REPO}/${RELEASE_NAME}:${OCP_RELEASE}-${ARCHITECTURE} \ --to=${LOCAL_REGISTRY}/${LOCAL_REPOSITORY} --apply-release-image-signature
```



참고

--apply-release-image-signature 옵션이 포함된 경우 이미지 서명 확인을 위해 ConfigMap을 작성하지 않습니다.

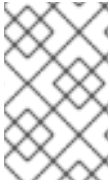
5.4. 이미지 서명 CONFIGMAP 생성

클러스터를 업데이트하기 전에 사용하는 릴리스 이미지의 서명이 포함된 ConfigMap을 수동으로 생성해야 합니다. 이 서명을 통해 CVO (Cluster Version Operator)는 예상되는 이미지와 실제 이미지 서명을 비교하여 릴리스 이미지가 변경되지 않았는지 확인할 수 있습니다.

버전 4.4.8 이상에서 업그레이드하는 경우 **oc** CLI를 사용하여 ConfigMap을 생성할 수 있습니다. 이전 버전에서 업그레이드하는 경우 수동 방법을 사용해야 합니다.

5.4.1. oc CLI를 사용하여 이미지 서명 확인을 위한 ConfigMap 생성

클러스터를 업데이트하기 전에 사용하는 릴리스 이미지의 서명이 포함된 ConfigMap을 수동으로 생성해야 합니다. 이 서명을 통해 CVO (Cluster Version Operator)는 예상되는 이미지와 실제 이미지 서명을 비교하여 릴리스 이미지가 변경되지 않았는지 확인할 수 있습니다.



참고

버전 4.4.8 이전 릴리스에서 업그레이드하는 경우 이 절차 대신 ConfigMap을 생성하기 위한 수동 방법을 사용해야 합니다. 이 절차에 사용되는 명령은 이전 버전의 **oc** 명령행 인터페이스 (CLI)에서 제공되지 않습니다.

전제 조건

- **oc**로 알려진 OpenShift 명령행 인터페이스 (CLI)를 설치합니다 (버전 4.4.8 이상).

프로세스

1. mirror.openshift.com 또는 [Google Cloud Storage \(GCS\)](https://cloud.google.com/storage) 에서 업그레이드하려는 버전의 이미지 서명을 가져옵니다.
2. **oc** 명령행 인터페이스 (CLI)를 사용하여 업그레이드 중인 클러스터에 로그인합니다.
3. 미러링된 릴리스 이미지 서명 ConfigMap을 연결된 클러스터에 적용합니다.

```
$ oc apply -f <image_signature_file> ❶
```

- ❶ **<image_signature_file>** 의 경우 파일의 경로와 이름을 지정합니다 (예: **mirror/config/signature-sha256-81154f5c03294534.yaml**).

5.4.2. 이미지 서명 ConfigMap 수동으로 생성

이미지 서명 ConfigMap을 만들고 업데이트하려는 클러스터에 적용합니다.



참고

클러스터를 업데이트할 때 마다 다음 단계를 수행해야 합니다.

프로세스

1. [OpenShift Container Platform 업그레이드 경로](#) 지식 베이스 문서를 참조하여 클러스터의 유효한 업그레이드 경로를 결정합니다.
2. **OCP_RELEASE_NUMBER** 환경 변수에 버전을 추가합니다.

```
$ OCP_RELEASE_NUMBER=<release_version> ❶
```

- ❶ **<release_version>**에 대해 클러스터를 업데이트하려는 OpenShift Container Platform 버전에 해당하는 태그를 지정합니다 (예: **4.4.0**).

3. **ARCHITECTURE** 환경 변수에 클러스터의 시스템 아키텍처를 추가합니다.

```
$ ARCHITECTURE=<server_architecture> ❶
```

server_architecture에 대해 **x86_64**와 같은 서버 아키텍처를 지정합니다.

4. [Quay](#) 에서 릴리스 이미지 다이제스트를 가져옵니다.

```
$ DIGEST="$(oc adm release info quay.io/openshift-release-dev/ocp-
release:${OCP_RELEASE_NUMBER}-${ARCHITECTURE} | sed -n 's/Pull From: .*@/p')"
```

5. 다이제스트 알고리즘을 설정합니다.

```
$ DIGEST_ALGO="${DIGEST%%:*}"
```

6. 다이제스트 서명을 설정합니다.

```
$ DIGEST_ENCODED="${DIGEST#*:*}"
```

7. mirror.openshift.com 웹 사이트에서 이미지 서명을 가져옵니다.

```
$ SIGNATURE_BASE64=$(curl -s "https://mirror.openshift.com/pub/openshift-
v4/signatures/openshift/release/${DIGEST_ALGO}=${DIGEST_ENCODED}/signature-1" |
base64 -w0 && echo)
```

8. ConfigMap을 생성합니다.

```
$ cat >checksum-${OCP_RELEASE_NUMBER}.yaml <<EOF
apiVersion: v1
kind: ConfigMap
metadata:
  name: release-image-${OCP_RELEASE_NUMBER}
  namespace: openshift-config-managed
  labels:
    release.openshift.io/verification-signatures: ""
binaryData:
  ${DIGEST_ALGO}-${DIGEST_ENCODED}: ${SIGNATURE_BASE64}
EOF
```

9. 업데이트할 클러스터에 ConfigMap을 적용합니다.

```
$ oc apply -f checksum-${OCP_RELEASE_NUMBER}.yaml
```

5.5. 네트워크가 제한된 환경의 클러스터 업데이트

네트워크가 제한된 환경의 클러스터를 다운로드한 릴리스 이미지의 OpenShift Container Platform 버전으로 업데이트합니다.

전제 조건

- 새 릴리스의 이미지를 레지스트리에 미러링하고 있습니다.
- 새 릴리스의 릴리스 이미지 서명 ConfigMap을 클러스터에 적용하고 있습니다.
- 이미지 서명 ConfigMap에서 릴리스의 sha256 합계 값을 얻을 수 있습니다.
- **oc**로 알려진 OpenShift 명령행 인터페이스 (CLI)를 설치합니다 (버전 4.4.8 이상).

프로세스

- 클러스터를 업데이트합니다.

```
$ oc adm upgrade --allow-explicit-upgrade --to-image
${LOCAL_REGISTRY}/${LOCAL_REPOSITORY}<sha256_sum_value> 1
```

- 1 <sha256_sum_value> 값은 이미지 서명 ConfigMap에서 릴리스에 대한 sha256 합계입니다 (예: @sha256:81154f5c03294534e1eaf0319bef7a601134f891689ccede5d705ef659aa8c92).

미리 레지스트리에 **ImageContentSourcePolicy**를 사용하는 경우 **LOCAL_REGISTRY** 대신 표준 레지스트리 이름을 사용할 수 있습니다.

5.6. 이미지 레지스트리 저장소 미러링 설정

컨테이너 레지스트리 저장소 미러링을 설정하면 다음을 수행할 수 있습니다.

- 소스 이미지 레지스트리의 저장소에서 이미지를 가져오기 위해 요청을 리디렉션하고 미러링된 이미지 레지스트리의 저장소에서 해결할 수 있도록 OpenShift Container Platform 클러스터를 설정합니다.
- 하나의 미러가 다운된 경우 다른 미러를 사용할 수 있도록 각 대상 저장소에 대해 여러 미러링된 저장소를 확인합니다.

다음은 OpenShift Container Platform의 저장소 미러링의 몇 가지 속성입니다.

- 이미지 풀은 레지스트리 다운 타임에 대처할 수 있습니다.
- 네트워크가 제한된 환경의 클러스터는 중요한 위치(예 : quay.io)에서 이미지를 가져오도록 요청할 수 있으며 회사의 방화벽 뒤에 레지스트리가 요청된 이미지를 제공하도록 할 수 있습니다.
- 이미지의 풀 요청 시 특정 레지스트리 순서로 시도되고 일반적으로 영구 레지스트리는 마지막으로 시도됩니다.
- 입력한 미러링 정보는 OpenShift Container Platform 클러스터의 모든 노드에서 **/etc/containers/registries.conf** 파일에 추가됩니다.
- 노드가 소스 저장소에서 이미지를 요청하면 요청된 콘텐츠를 찾을 때 까지 미러링된 각 저장소를 차례로 시도합니다. 모든 미러가 실패하면 클러스터는 소스 저장소를 시도합니다. 성공하면 이미지가 노드로 풀됩니다.

저장소 미러링은 다음과 같은 방법으로 설정할 수 있습니다.

- OpenShift Container Platform 설치 시 : OpenShift Container Platform에 필요한 컨테이너 이미지를 가져온 다음 해당 이미지를 회사 방화벽 뒤에 배치하면 제한된 네트워크에있는 데이터 센터에 OpenShift Container Platform을 설치할 수 있습니다. 자세한 내용은 OpenShift Container Platform 이미지 저장소 미러링을 참조하십시오.
- OpenShift Container Platform 설치 후 : OpenShift Container Platform 설치 시 미러링을 설정하지 않고 **ImageContentSourcePolicy** 객체를 사용하여 나중에 설정할 수 있습니다.

다음 프로세스에서는 설치 후 미러를 설정하고 다음을 식별하는 **ImageContentSourcePolicy** 객체를 만듭니다.

- 미러링하려는 컨테이너 이미지 저장소의 소스
- 소스 저장소에서 요청된 콘텐츠를 제공하는 각 미러 저장소에 대한 개별 항목

전제 조건

- **cluster-admin** 역할을 가진 사용자로 클러스터에 액세스할 수 있어야 합니다.

프로세스

1. 미러링된 저장소를 설정합니다. 이를 위해 다음 중 하나를 수행합니다.

- [Red Hat Quay Repository Mirroring](#) 에 설명된대로 Red Hat Quay를 사용하여 미러링된 저장소를 설정합니다. Red Hat Quay를 사용하면 한 저장소에서 다른 저장소로 이미지를 복사하고 시간이 지남에 따라 해당 저장소를 반복해서 자동으로 동기화할 수 있습니다.
- **skopeo**와 같은 툴을 사용하여 소스 디렉토리에서 미러링된 저장소로 이미지를 수동으로 복사합니다.
예를 들어, Red Hat Enterprise Linux (RHEL 7 또는 RHEL 8) 시스템에 skopeo RPM 패키지를 설치한 후 다음 예와 같이 **skopeo** 명령을 사용합니다.

```
$ skopeo copy \
  docker://registry.access.redhat.com/ubi8/ubi-
  minimal@sha256:c505667389712dc337986e29ffcb65116879ef27629dc3ce6e1b17727c06
  e78f \
  docker://example.io/ubi8/ubi-minimal
```

이 예제에는 **example.io**라는 컨테이너 이미지 레지스트리가 있으며, **registry.access.redhat.com**에서 **ubi8 / ubi-minimal** 이미지를 복사하려는 **example**이라는 이미지 저장소가 있습니다. 레지스트리를 생성한 후 OpenShift Container Platform 클러스터를 설정하여 소스 저장소의 요청을 미러링된 저장소로 리디렉션할 수 있습니다.

2. OpenShift Container Platform 클러스터에 로그인합니다.

3. **ImageContentSourcePolicy** 파일(예: **registryrepositor.yaml**)을 만들고 소스 및 미러를 특정 레지스트리 및 저장소 쌍과 이미지로 대체합니다.

```
apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
  name: ubi8repo
spec:
  repositoryDigestMirrors:
  - mirrors:
    - example.io/example/ubi-minimal 1
    source: registry.access.redhat.com/ubi8/ubi-minimal 2
  - mirrors:
    - example.com/example/ubi-minimal
    source: registry.access.redhat.com/ubi8/ubi-minimal
```

- 1** 이미지 레지스트리 및 저장소의 이름을 가리킵니다.
- 2** 미러링된 콘텐츠를 포함하는 레지스트리 및 저장소를 가리킵니다.

4. 새 **ImageContentSourcePolicy**를 만듭니다.

```
$ oc create -f registryrepositor.yaml
```

ImageContentSourcePolicy가 생성된 후 새 설정이 각 노드에 배포되고 소스 저장소의 요청에 대해 미러링된 저장소를 사용하기 시작합니다.

5. 미러링된 설정이 작동했는지 확인하려면 노드 중 하나로 이동합니다. 예를 들면 다음과 같습니다.
 - a. 노드를 나열합니다.

```
$ oc get node
NAME                                STATUS    ROLES    AGE    VERSION
ip-10-0-137-44.ec2.internal        Ready    worker   7m    v1.18.3
ip-10-0-138-148.ec2.internal        Ready    master   11m   v1.18.3
ip-10-0-139-122.ec2.internal        Ready    master   11m   v1.18.3
ip-10-0-147-35.ec2.internal        Ready,SchedulingDisabled  worker   7m    v1.18.3
ip-10-0-153-12.ec2.internal        Ready    worker   7m    v1.18.3
ip-10-0-154-10.ec2.internal        Ready    master   11m   v1.18.3
```

변경 사항이 적용됨에 따라 각 작업자 노드의 스케줄링이 비활성화되어 있음을 알 수 있습니다.

- b. **/etc/containers/registries.conf** 파일을 체크하여 변경 사항이 적용되었는지 확인합니다.

```
$ oc debug node/ip-10-0-147-35.ec2.internal
Starting pod/ip-10-0-147-35ec2internal-debug ...
To use host binaries, run `chroot /host`

sh-4.2# chroot /host
sh-4.2# cat /etc/containers/registries
unqualified-search-registries = ["registry.access.redhat.com", "docker.io"]
[[registry]]
  location = "registry.access.redhat.com/ubi8/"
  insecure = false
  blocked = false
  mirror-by-digest-only = true
  prefix = ""

[[registry.mirror]]
  location = "example.io/example/ubi8-minimal"
  insecure = false

[[registry.mirror]]
  location = "example.com/example/ubi8-minimal"
  insecure = false
```

- c. 소스에서 이미지를 노드로 가져와 실제로 미러링에 의해 해결되는지 확인합니다.

```
sh-4.2# podman pull --log-level=debug registry.access.redhat.com/ubi8/ubi-minimal
```

저장소 미러링 문제 해결

설명대로 저장소 미러링 프로세스가 작동하지 않는 경우 저장소 미러링 작동 방법에 대한 다음 정보를 사용하여 문제점을 해결하십시오.

- 먼저 작업 중인 미러는 풀되는 이미지를 지정하는 데 사용됩니다.
- 주요 레지스트리는 다른 미러가 작동하지 않는 경우에만 사용됩니다.

- 시스템 컨텍스트에서 **Insecure** 플래그가 폴백으로 사용됩니다.
- **/etc/containers/registries** 파일 형식이 최근에 변경되었습니다. 현재 버전은 TOML 형식의 버전 2입니다.*