



# Red Hat OpenStack Platform 16.1

## Red Hat OpenStack Platform 업데이트 유지

Red Hat OpenStack Platform의 마이너 업데이트 수행



# Red Hat OpenStack Platform 16.1 Red Hat OpenStack Platform 업데이트 유지

---

Red Hat OpenStack Platform의 마이너 업데이트 수행

OpenStack Team  
rhos-docs@redhat.com

## 법적 공지

Copyright © 2023 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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 문서에서는 Red Hat OpenStack Platform 환경의 마이너 업데이트를 수행하는 절차를 설명합니다.

<b>차례</b>	
보다 포괄적 수용을 위한 오픈 소스 용어 교체 .....	3
RED HAT 문서에 관한 피드백 제공 .....	4
<b>1장. 소개</b> .....	<b>5</b>
1.1. 상위 수준의 워크플로 .....	5
1.2. 업데이트를 차단할 수 있는 알려진 문제 .....	5
<b>2장. 마이너 업데이트 준비</b> .....	<b>7</b>
2.1. RED HAT ENTERPRISE LINUX 릴리스에 대한 환경 잠금 .....	7
2.2. EUS 리포지토리에서 TUS 리포지토리로 변경 .....	8
2.3. RED HAT OPENSTACK PLATFORM 및 ANSIBLE 리포지토리 업데이트 .....	10
2.4. CONTAINER-TOOLS 버전 설정 .....	12
2.5. 컨테이너 이미지 준비 파일 업데이트 .....	13
2.6. SSL/TLS 구성 업데이트 .....	14
2.7. 오버클라우드에서 펜싱 비활성화 .....	15
<b>3장. 언더클라우드 업데이트</b> .....	<b>17</b>
3.1. 컨테이너화된 언더클라우드의 마이너 업데이트 수행 .....	17
3.2. 오버클라우드 이미지 업데이트 .....	17
3.3. 언더클라우드 업그레이드 후 참고 사항 .....	19
<b>4장. 오버클라우드 업데이트</b> .....	<b>20</b>
4.1. 오버클라우드 업데이트 준비 실행 .....	20
4.2. 컨테이너 이미지 준비 실행 .....	21
4.3. 선택 사항: 모든 오버클라우드 서버에서 OVN-CONTROLLER 컨테이너 업데이트 .....	22
4.4. 모든 컨트롤러 노드 업데이트 .....	23
4.5. 모든 컴퓨팅 노드 업데이트 .....	24
4.6. 모든 HCI 컴퓨팅 노드 업데이트 .....	25
4.7. 모든 CEPH STORAGE 노드 업데이트 .....	26
4.8. 온라인 데이터베이스 업데이트 수행 .....	28
4.9. 업데이트 종료 .....	29
<b>5장. 오버클라우드 재부팅</b> .....	<b>31</b>
5.1. 컨트롤러 노드 및 구성 가능 노드 재부팅 .....	31
5.2. CEPH STORAGE(OSD) 클러스터 재부팅 .....	32
5.3. 컴퓨팅 노드 재부팅 .....	34



## 보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 [CTO Chris Wright의 메시지](#)를 참조하십시오.

## RED HAT 문서에 관한 피드백 제공

문서 개선을 위한 의견을 보내 주십시오. Red Hat이 어떻게 더 나은지 알려주십시오.

### DDF(직접 문서 피드백) 기능 사용

피드백 추가 DDF 기능을 사용하여 특정 문장, 단락 또는 코드 블록에 대한 직접 의견을 제출할 수 있습니다.

1. *다중 페이지 HTML* 형식으로 문서를 봅니다.
2. 문서의 오른쪽 상단에 **피드백** 버튼이 표시되는지 확인합니다.
3. 주석 처리하려는 텍스트 부분을 강조 표시합니다.
4. **피드백 추가**를 클릭합니다.
5. 코멘트를 사용하여 **피드백 추가** 필드를 완료합니다.
6. 선택 사항: 문서 팀이 문제에 대한 자세한 설명을 위해 연락을 드릴 수 있도록 이메일 주소를 추가합니다.
7. **제출**을 클릭합니다.



## 1장. 소개

이 문서에서는 Red Hat OpenStack Platform 16.1 환경을 최신 패키지 및 컨테이너로 업데이트하는 데 도움이 되는 워크플로를 제공합니다.

이 가이드에서는 다음 버전을 통해 업그레이드 경로를 제공합니다.

이전 OpenStack 버전	새 OpenStack 버전
Red Hat OpenStack Platform 16.0	Red Hat OpenStack Platform 16.1.z
Red Hat OpenStack Platform 16.1	Red Hat OpenStack Platform 16.1.z

### 1.1. 상위 수준의 워크플로

다음 표에서는 업그레이드 프로세스에 필요한 단계를 간략하게 설명합니다.

Step	설명
언더클라우드 업데이트	언더클라우드를 최신 OpenStack Platform 16.1.z 버전으로 업데이트합니다.
오버클라우드 업데이트	오버클라우드를 최신 OpenStack Platform 16.1.z 버전으로 업데이트합니다.
Ceph Storage 노드 업데이트	모든 Ceph Storage 서비스를 업그레이드합니다.
업그레이드 종료	통합 명령을 실행하여 오버클라우드 스택을 새로 고칩니다.

다중 스택 인프라가 있는 경우 각 오버클라우드 스택을 한 번에 하나씩 완전히 업데이트합니다. DCN(Distributed Compute Node) 인프라가 있는 경우 중앙 위치에서 오버클라우드를 완전히 업데이트한 다음, 각 에지 사이트에서 한 번에 하나씩 오버클라우드를 업데이트합니다.

#### RHOSP 환경을 업데이트하기 전에 고려해야 할 사항

업데이트 프로세스 중에 안내하려면 다음 정보를 고려하십시오.

- Red Hat은 언더클라우드 및 오버클라우드 컨트롤 플레인을 백업하는 것이 좋습니다. 노드 백업에 대한 자세한 내용은 [언더클라우드 및 컨트롤 플레인 노드 백업 및 복원](#)을 참조하십시오.
- 업데이트를 차단할 수 있는 알려진 문제에 대해 숙지하십시오.
- 현재 유지 관리 릴리스를 확인하려면 **\$ cat /etc/rhosp-release** 를 실행합니다. 환경을 업데이트하여 업데이트를 검증한 후 이 명령을 실행할 수도 있습니다.

### 1.2. 업데이트를 차단할 수 있는 알려진 문제

성공적인 마이너 버전 업데이트에 영향을 줄 수 있는 다음과 같은 알려진 문제를 검토합니다.

**BZ#1973660 - (업데이트) 16.1에서 16.2의 중단으로, rabbitmq 서비스를 구성하려고 합니다.**

Pacemaker 버전 **2.0.3-5.el8\_2.4** 를 실행하는 오버클라우드 노드는 노드에서 클러스터를 종료할 때 발생하는 경쟁 조건으로 인해 업데이트되지 않을 수 있습니다.

Pacemaker 버전 **2.0.3-5.el8\_2.4** 가 현재 모든 오버클라우드 노드에 설치된 경우 **BZ#1973660** 을 방지하려면 먼저 Pacemaker를 업그레이드해야 오버클라우드 노드를 업데이트할 수 있습니다. 자세한 내용은 다음 Red Hat Knowledgebase 솔루션 업데이트를 **OSP16.1에서 OSP16.2로 업데이트하지 못할 수 있습니다** .

**BZ#1872404 - 쿼럼을 유지 관리하는 동안 노드를 병렬로 재시작하면 예기치 않은 노드 종료가 발생합니다.**

+ RHEL 8.2를 실행하고 구성 가능한 역할을 기반으로 하는 노드의 경우 다른 역할을 업데이트하기 전에 먼저 **데이터베이스** 역할을 업데이트해야 합니다.

**BZ#2117179 - osP 16.1.6의 마이너 업데이트로 인해 /etc/openldap의 이전 OSP13 구성이 있는 경우 Keystone LDAP 연결이 실패합니다.**

알려진 문제로 인해 릴리스 16.1.6 이하에서 16.1.7 이상으로 업데이트한 후 LDAP 연결이 실패합니다. RHOSP 16.1.7을 사용하면 ID 서비스(keystone) 컨테이너가 호스트 파일 시스템에 **/etc/openldap** 를 마운트합니다. 이전에 RHOSP 13에서 업데이트한 경우 ID 서비스 LDAP 연결이 실패할 수 있는 **/etc/openldap** 디렉토리에 이전 구성 파일이 있을 수 있습니다.

이 문제를 해결하려면 각 컨트롤러에서 다음 명령을 실행합니다.

```
$ sudo cp /etc/openldap/ldap.conf.rpmnew /etc/openldap/ldap.conf
$ sudo podman restart keystone
```

**BZ#2158626 - 공급자 패치 포트가 재시작 시 다시 생성됨**

다음 조건을 충족하는 경우 공급자 네트워크에서 트래버스하는 네트워크 트래픽이 ovn-controller를 다시 시작하는 동안 중단됩니다.

- 환경에는 공급자 네트워크가 공급자 네트워크의 유동 IP 또는 직접 포트에 연결되어 있습니다.
- 16.1 릴리스로 업데이트

다운타임은 기존 워크로드 수에 따라 다를 수 있습니다. 다운타임을 방지하려면 **모든 16.1 버전을 포함하여 16.2.2 이전의 OSP 업데이트/업그레이드 중에 Red Hat Knowledgebase 솔루션 데이터 플레인 중단** 의 해결 방법을 적용하십시오.

## 2장. 마이너 업데이트 준비

Red Hat OpenStack Platform 16.1을 최신 마이너 릴리스로 업데이트하는 프로세스를 시작하기 전에 언더클라우드 및 오버클라우드의 준비 단계를 수행해야 합니다.

### 2.1. RED HAT ENTERPRISE LINUX 릴리스에 대한 환경 잠금

Red Hat OpenStack Platform 16.1은 Red Hat Enterprise Linux 8.2에서 지원됩니다. 업데이트를 실행하기 전에 운영 체제를 최신 마이너 릴리스로 업그레이드하지 않도록 언더클라우드 및 오버클라우드 리포지토리를 Red Hat Enterprise Linux 8.2 릴리스에 고정합니다.

#### 절차

1. **stack** 사용자로 언더클라우드에 로그인합니다.
2. **stackrc** 파일을 소싱합니다.

```
$ source ~/stackrc
```

3. **RhsmVars** 매개변수가 포함된 파일인 오버클라우드 서브스크립션 관리 환경 파일을 편집합니다. 이 파일의 기본 이름은 usually **rhsm.yml** 입니다.
4. 서브스크립션 관리 구성에 the **rhsm\_release** 매개변수가 있는지 확인합니다. 이 매개변수를 설정하지 않으면 이 매개변수를 추가하고 매개변수를 8.2로 설정합니다.

```
parameter_defaults:
  RhsmVars:
    ...
    rhsm_username: "myusername"
    rhsm_password: "p@55w0rd!"
    rhsm_org_id: "1234567"
    rhsm_pool_ids: "1a85f9223e3d5e43013e3d6e8ff506fd"
    rhsm_method: "portal"
    rhsm_release: "8.2"
```

5. Overcloud 서브스크립션 관리 환경 파일을 저장합니다.
6. 오버클라우드의 정적 인벤토리 파일을 생성합니다.

```
$ tripleo-ansible-inventory --ansible_ssh_user heat-admin --static-yaml-inventory
~/inventory.yaml
```

오버클라우드 이름을 오버클라우드의 기본 오버클라우드 이름에 다르게 사용하는 경우 **--plan** 옵션을 사용하여 오버클라우드 이름을 설정합니다.

7. 모든 노드에서 운영 체제 버전을 Red Hat Enterprise Linux 8.2에 고정하는 작업이 포함된 플레이북을 생성합니다.

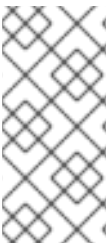
```
$ cat > ~/set_release.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: set release to 8.2
```

```
command: subscription-manager release --set=8.2
become: true
EOF
```

8. **set\_release.yaml** 플레이북을 실행합니다.

```
$ ansible-playbook -i ~/inventory.yaml -f 25 ~/set_release.yaml --limit <undercloud>,
<Controller>,<Compute>
```

- **--limit** 옵션을 사용하여 모든 RHOSP 노드에 콘텐츠를 적용합니다. < **undercloud** >, < **Controller** >, < **Compute** >를 해당 노드를 포함하는 사용자 환경의 Ansible 그룹으로 바꿉니다.
- 이러한 노드에 다른 서브스크립션을 사용하는 경우 Ceph Storage 노드에서 이 플레이북을 실행할 수 없습니다.



**참고**

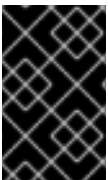
노드를 버전으로 수동으로 잠그려면 노드에 로그인하고 **subscription-manager release** 명령을 실행합니다.

```
$ sudo subscription-manager release --set=8.2
```

## 2.2. EUS 리포지토리에서 TUS 리포지토리로 변경

Red Hat OpenStack Platform 서브스크립션에는 Red Hat Enterprise Linux 8.2 Extended Update Support (EUS)용 리포지토리가 포함되어 있습니다. 2022년 4월 30일 이후 유지 관리 지원을 위해 RHEL 8.2 TUS(Telecommunications Update Service) 리포지토리를 활성화해야 합니다. TUS 리포지토리에는 Red Hat Enterprise Linux 8.2의 최신 보안 패치 및 버그 수정이 포함되어 있습니다. 업데이트를 수행하기 전에 다음 리포지토리로 전환합니다.

EUS 리포지토리	TUS Repository
rhel-8-for-x86_64-baseos-eus-rpms	rhel-8-for-x86_64-baseos-tus-rpms
rhel-8-for-x86_64-appstream-eus-rpms	rhel-8-for-x86_64-appstream-tus-rpms
rhel-8-for-x86_64-highavailability-eus-rpms	rhel-8-for-x86_64-highavailability-tus-rpms



**중요**

특정 버전의 Podman과의 호환성을 유지하려면 TUS 리포지토리를 사용해야 합니다. 이후 버전의 Podman은 Red Hat Open Stack Platform 16.1 릴리스에서 테스트되지 않았으며 예기치 않은 결과를 초래할 수 있습니다.

**사전 요구 사항**

- RHOSP 16.1 EUS 서브스크립션

**절차**

1. **stack** 사용자로 언더클라우드에 로그인합니다.

2. **stackrc** 파일을 소싱합니다.

```
$ source ~/stackrc
```

3. **RhsmVars** 매개변수가 포함된 파일인 오버클라우드 서브스크립션 관리 환경 파일을 편집합니다. 이 파일의 기본 이름은 **usually rhsm.yml** 입니다.

4. 서브스크립션 관리 구성의 **rhsm\_repos** 매개변수를 확인합니다. 이 매개변수에 TUS 리포지토리가 포함되어 있지 않은 경우 관련 리포지토리를 TUS 버전으로 변경합니다.

```
parameter_defaults:
  RhsmVars:
    rhsm_repos:
      - rhel-8-for-x86_64-baseos-tus-rpms
      - rhel-8-for-x86_64-appstream-tus-rpms
      - rhel-8-for-x86_64-highavailability-tus-rpms
      - ansible-2.9-for-rhel-8-x86_64-rpms
      - advanced-virt-for-rhel-8-x86_64-rpms
      - openstack-16.1-for-rhel-8-x86_64-rpms
      - rhceph-4-tools-for-rhel-8-x86_64-rpms
      - fast-datapath-for-rhel-8-x86_64-rpms
```

5. Overcloud 서브스크립션 관리 환경 파일을 저장합니다.

6. 오버클라우드의 정적 인벤토리 파일을 생성합니다.

```
$ tripleo-ansible-inventory --ansible_ssh_user heat-admin --static-yaml-inventory
~/inventory.yaml
```

기본 오버클라우드 이름 오버클라우드 이름과 다른 오버클라우드 이름을 사용하는 경우 **--plan** 옵션을 사용하여 오버클라우드 이름을 설정합니다.

7.

모든 노드에서 리포지토리를 **Red Hat Enterprise Linux 8.2 TUS**로 설정하는 작업이 포함된 플레이북을 생성합니다.

```
$ cat > ~/change_tus.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: change to tus repos
      command: subscription-manager repos --disable=rhel-8-for-x86_64-baseos-eus-rpms --
disable=rhel-8-for-x86_64-appstream-eus-rpms --disable=rhel-8-for-x86_64-highavailability-
eus-rpms --enable=rhel-8-for-x86_64-baseos-tus-rpms --enable=rhel-8-for-x86_64-
appstream-tus-rpms --enable=rhel-8-for-x86_64-highavailability-tus-rpms
      become: true
EOF
```

8.

**change\_tus.yaml** 플레이북을 실행합니다.

```
$ ansible-playbook -i ~/inventory.yaml -f 25 ~/change_tus.yaml --limit <undercloud>,
<Controller>,<Compute>
```

- **--limit** 옵션을 사용하여 모든 **RHOSP** 노드에 콘텐츠를 적용합니다. **< undercloud >** , **< Controller >** , **< Compute >**를 해당 노드를 포함하는 사용자 환경의 **Ansible** 그룹으로 바꿉니다.
- 이러한 노드에 다른 서브스크립션을 사용하는 경우 **Ceph Storage** 노드에서 이 플레이북을 실행할 수 없습니다.

### 2.3. RED HAT OPENSTACK PLATFORM 및 ANSIBLE 리포지토리 업데이트

Red Hat OpenStack Platform 16.1 및 Ansible 2.9 패키지를 사용하도록 리포지토리를 업데이트합니다.

#### 절차

1. **stack** 사용자로 언더클라우드에 로그인합니다.
2. **stackrc** 파일을 소싱합니다.

```
$ source ~/stackrc
```

3. **RhsmVars** 매개변수가 포함된 파일인 오버클라우드 서브스크립션 관리 환경 파일을 편집합니다. 이 파일의 기본 이름은 **usually rhsm.yml** 입니다.
4. 서브스크립션 관리 구성의 **rhsm\_repos** 매개변수를 확인합니다. **rhsm\_repos** 매개변수에서 **Red Hat OpenStack Platform 16.0** 및 **Ansible 2.8** 리포지토리를 사용하는 경우 리포지토리를 올바른 버전으로 변경합니다.

```
parameter_defaults:
  RhsmVars:
    rhsm_repos:
      - rhel-8-for-x86_64-baseos-tus-rpms
      - rhel-8-for-x86_64-appstream-tus-rpms
      - rhel-8-for-x86_64-highavailability-tus-rpms
      - ansible-2.9-for-rhel-8-x86_64-rpms
      - advanced-virt-for-rhel-8-x86_64-rpms
      - openstack-16.1-for-rhel-8-x86_64-rpms
      - fast-datapath-for-rhel-8-x86_64-rpms
```

5. **Overcloud** 서브스크립션 관리 환경 파일을 저장합니다.

6. 오버클라우드의 정적 인벤토리 파일을 생성합니다.

```
$ tripleo-ansible-inventory --ansible_ssh_user heat-admin --static-yaml-inventory
~/inventory.yaml
```

기본 오버클라우드 이름 오버클라우드 이름과 다른 오버클라우드 이름을 사용하는 경우 **--plan** 옵션을 사용하여 오버클라우드 이름을 설정합니다.

7. 모든 노드에서 리포지토리를 **Red Hat OpenStack Platform 16.1**로 설정하는 작업이 포함된 플레이북을 생성합니다.

```
$ cat > ~/update_rhosp_repos.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: change osp repos
      command: subscription-manager repos --disable=openstack-16-for-rhel-8-x86_64-rpms --enable=openstack-16.1-for-rhel-8-x86_64-rpms --disable=ansible-2.8-for-rhel-8-x86_64-rpms --enable=ansible-2.9-for-rhel-8-x86_64-rpms
      become: true
EOF
```

8. **update\_rhosp\_repos.yaml** 플레이북을 실행합니다.

```
$ ansible-playbook -i ~/inventory.yaml -f 25 ~/update_rhosp_repos.yaml --limit <undercloud>,
<Controller>,<Compute>
```

- **--limit** 옵션을 사용하여 모든 **RHOSP** 노드에 콘텐츠를 적용합니다. **< undercloud > , < Controller > , < Compute >**를 해당 노드를 포함하는 사용자 환경의 **Ansible** 그룹으로 바꿉니다.
- 이러한 노드에 다른 서브스크립션을 사용하는 경우 **Ceph Storage** 노드에서 이 플레이북을 실행할 수 없습니다.

9. 모든 노드에서 리포지토리를 **Red Hat OpenStack Platform 16.1**로 설정하는 작업이 포함된 플레이북을 생성합니다.

```
$ cat > ~/update_ceph_repos.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: change ceph repos
      command: subscription-manager repos --disable=openstack-16-deployment-tools-for-rhel-8-x86_64-rpms --enable=openstack-16.1-deployment-tools-for-rhel-8-x86_64-rpms --disable=ansible-2.8-for-rhel-8-x86_64-rpms --enable=ansible-2.9-for-rhel-8-x86_64-rpms
      become: true
EOF
```

10.

**update\_ceph\_repos.yaml** 플레이북을 실행합니다.

```
$ ansible-playbook -i ~/inventory.yaml -f 25 ~/update_ceph_repos.yaml --limit CephStorage
```

**--limit** 옵션을 사용하여 콘텐츠를 **Ceph Storage** 노드에 적용합니다.

## 2.4. CONTAINER-TOOLS 버전 설정

모든 노드에서 올바른 패키지 버전을 사용하도록 **container-tools** 모듈을 버전 **2.0** 으로 설정합니다.

### 절차

1. **stack** 사용자로 언더클라우드에 로그인합니다.

2. **stackrc** 파일을 소싱합니다.

```
$ source ~/stackrc
```

3. 오버클라우드의 정적 인벤토리 파일을 생성합니다.

```
$ tripleo-ansible-inventory --ansible_ssh_user heat-admin --static-yaml-inventory ~/inventory.yaml
```

오버클라우드 이름을 오버클라우드의 기본 오버클라우드 이름에 다르게 사용하는 경우 **--plan** 옵션을 사용하여 오버클라우드 이름을 설정합니다.

4. 작업이 포함된 플레이북을 생성하여 모든 노드에서 **container-tools** 모듈을 버전 **2.0** 으로 설정합니다.



```
$ cat > ~/container-tools.yaml <<'EOF'
- hosts: all
  gather_facts: false
  tasks:
    - name: disable default dnf module for container-tools
      command: dnf module reset -y container-tools
      become: true
    - name: set dnf module for container-tools:2.0
      command: dnf module enable -y container-tools:2.0
      become: true
EOF
```

5.

모든 노드에 대해 **container-tools.yaml** 플레이북을 실행합니다.

```
$ ansible-playbook -i ~/inventory.yaml -f 25 ~/container-tools.yaml
```

## 2.5. 컨테이너 이미지 준비 파일 업데이트

컨테이너 준비 파일은 **ContainerImagePrepare** 매개변수가 포함된 파일입니다. 이 파일을 사용하여 언더클라우드 및 오버클라우드의 컨테이너 이미지를 가져오는 규칙을 정의합니다. 환경을 업데이트하기 전에 파일을 확인하여 이미지 버전이 올바른지 확인합니다.

### 절차

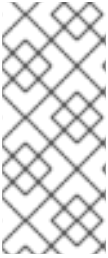
1.

컨테이너 준비 파일을 편집합니다. 이 파일의 기본 이름은 일반적으로 **containers-prepare-parameter.yaml**입니다.

2.

**tag** 매개변수가 각 규칙 세트에 대해 **16.1**로 설정되어 있는지 확인합니다.

```
parameter_defaults:
  ContainerImagePrepare:
    - push_destination: true
      set:
        ...
        tag: '16.1'
        tag_from_label: '{version}-{release}'
```



## 참고

**16.1** 또는 **16.1.2** 와 같은 업데이트에 특정 태그를 사용하지 않으려면 태그 키-값 쌍을 제거하고 `tag_from_label` 만 지정합니다. 업데이트 프로세스의 일부로 사용할 태그의 값을 결정할 때 설치된 **Red Hat OpenStack Platform** 버전이 사용됩니다.

1. 이 파일을 저장합니다.

## 2.6. SSL/TLS 구성 업데이트

`resource_registry` 에서 `NodeTLSData` 리소스를 제거하여 **SSL/TLS** 구성을 업데이트합니다.

### 절차

1. `stack` 사용자로 언더클라우드에 로그인합니다.
2. `stackrc` 파일을 소싱합니다.

```
$ source ~/stackrc
```

3. 사용자 지정 오버클라우드 **SSL/TLS** 공용 엔드포인트 파일을 편집합니다. 이 파일은 일반적으로 `~/templates/enable-tls.yaml` 입니다.
4. `'resource_registry'`에서 `NodeTLSData` 리소스를 제거합니다.

```
resource_registry:
  OS::TripleO::NodeTLSData: /usr/share/openstack-tripleo-heat-
  templates/puppet/extraconfig/tls/tls-cert-inject.yaml
  ...
```

**Overcloud** 배포에서는 **HAProxy**의 새 서비스를 사용하여 **SSL/TLS**가 활성화되어 있는지 확인합니다.



## 참고

`enable-tls.yaml` 파일의 `resource_registry` 섹션에 있는 유일한 리소스인 경우 전체 `resource_registry` 섹션을 제거합니다.

5. **SSL/TLS 공용 엔드포인트 파일** 파일을 저장합니다.

## 2.7. 오버클라우드에서 펜싱 비활성화

오버클라우드를 업데이트하기 전에 펜싱이 비활성화되었는지 확인합니다.

컨트롤러 노드 업데이트 프로세스 중에 펜싱을 환경에 배포하면 오버클라우드에서 특정 노드를 비활성화한 것으로 탐지하고 펜싱 작업을 시도하여 의도하지 않은 결과가 발생할 수 있습니다.

오버클라우드에서 펜싱을 활성화한 경우 의도하지 않은 결과를 피하려면 업데이트 기간 동안 펜싱을 일시적으로 비활성화해야 합니다.

### 절차

1. **stack** 사용자로 언더클라우드에 로그인합니다.

2. **stackrc** 파일을 소싱합니다.

```
$ source ~/stackrc
```

3. 컨트롤러 노드에 로그인하고 **Pacemaker** 명령을 실행하여 펜싱을 비활성화합니다.

```
$ ssh heat-admin@CONTROLLER_IP "sudo pcs property set stonith-enabled=false"
```

4. **fencing.yaml** 환경 파일에서 **EnableFencing** 매개변수를 **false** 로 설정하여 업데이트 프로세스 중에 펜싱이 비활성화되었는지 확인합니다.

### 추가 리소스

- **STONITH**를 사용하여 컨트롤러 노드 펜싱

### 3장. 언더클라우드 업데이트

이 프로세스에서는 언더클라우드 및 해당 오버클라우드 이미지를 최신 **Red Hat OpenStack Platform 16.1** 버전으로 업데이트합니다.

#### 3.1. 컨테이너화된 언더클라우드의 마이너 업데이트 수행

**director**는 언더클라우드 노드의 기본 패키지를 업데이트하는 명령을 제공합니다. 이를 통해 현재 버전의 **OpenStack Platform** 환경에서 마이너 업데이트를 수행할 수 있습니다.

절차

1. **stack** 사용자로 **director**에 로그인합니다.

2. **dnf** 를 실행하여 **director** 기본 패키지를 업그레이드합니다.

```
$ sudo dnf update -y python3-tripleoclient* tripleo-ansible ansible
```

3. **director**는 **openstack undercloud upgrade** 명령을 사용하여 언더클라우드 환경을 업데이트합니다. 명령을 실행합니다.

```
$ openstack undercloud upgrade
```

4. 언더클라우드 업그레이드 프로세스가 완료될 때까지 기다립니다.

5. 언더클라우드를 재부팅하여 운영 체제의 커널 및 기타 시스템 패키지를 업데이트합니다.

```
$ sudo reboot
```

6. 노드가 부팅될 때까지 기다립니다.

#### 3.2. 오버클라우드 이미지 업데이트

현재 오버클라우드 이미지를 새 버전으로 교체해야 합니다. 새 이미지를 사용하면 최신 버전의 **OpenStack Platform** 소프트웨어를 사용하여 **director**가 노드를 세부 검사하고 프로비저닝할 수 있습니다.

다.

#### 사전 요구 사항

- 언더클라우드를 최신 버전으로 업데이트했습니다.

#### 절차

1.

**stackrc** 파일을 소싱합니다.

```
$ source ~/stackrc
```

2.

**stack** 사용자 홈(/home/stack/images)의 **images** 디렉터리에서 기존 이미지를 제거합니다.

```
$ rm -rf ~/images/*
```

3.

아카이브를 추출합니다.

```
$ cd ~/images
$ for i in /usr/share/rhosp-director-images/overcloud-full-latest-16.1.tar /usr/share/rhosp-director-images/ironic-python-agent-latest-16.1.tar; do tar -xvf $i; done
$ cd ~
```

4.

최신 이미지를 **director**로 가져옵니다.

```
$ openstack overcloud image upload --update-existing --image-path /home/stack/images/
```

5.

새 이미지를 사용하도록 노드를 구성합니다.

```
$ openstack overcloud node configure $(openstack baremetal node list -c UUID -f value)
```

6.

새 이미지가 있는지 확인합니다.

```
$ openstack image list
$ ls -l /var/lib/ironic/httpboot
```

## 중요

- 오버클라우드 노드를 배포할 때 오버클라우드 이미지 버전이 해당 **heat** 템플릿 버전에 해당하는지 확인합니다. 예를 들어 **RHOSP 16.1 heat** 템플릿이 있는 **RHOSP 16.1** 이미지만 사용합니다.
- Red Hat Customer Portal** 또는 **Red Hat Satellite Server**를 사용하는 연결된 환경을 배포한 경우 오버클라우드 이미지 및 패키지 리포지토리 버전이 동기화되지 않을 수 있습니다. 오버클라우드 이미지 및 패키지 리포지토리 버전이 일치하는지 확인하려면 **virt-customize** 툴을 사용할 수 있습니다. 자세한 내용은 **virt-customize**를 사용하여 **Red Hat Linux OpenStack Platform Overcloud** 이미지를 수정하는 **Red Hat Knowledgebase** 솔루션을 참조하십시오.
- 새 **overcloud-full** 이미지가 이전 **overcloud-full** 이미지를 대체합니다. 이전 이미지를 변경한 경우 특히 나중에 새 노드를 배포하려는 경우 새 이미지의 변경 사항을 반복해야 합니다.

## 3.3. 언더클라우드 업그레이드 후 참고 사항

- 스택 사용자 홈 디렉터리에서 코어 템플릿의 로컬 세트를 사용하는 경우 **Advanced Overcloud Customization** 가이드에서 사용자 지정 코어 **heat** 템플릿을 사용하여 권장 워크플로를 사용하여 템플릿을 업데이트해야 합니다. 오버클라우드를 업그레이드하기 전에 로컬 복사본을 업데이트해야 합니다.

## 4장. 오버클라우드 업데이트

언더클라우드를 업데이트한 후 오버클라우드 및 컨테이너 이미지 준비 명령을 실행하고 노드를 업데이트한 다음 오버클라우드 업데이트 컨버 합 명령을 실행하여 오버클라우드를 업데이트할 수 있습니다. 마이너 업데이트 중에 컨트롤 플레인 API를 완전히 사용할 수 있습니다.

### 사전 요구 사항

- 언더클라우드를 최신 버전으로 업데이트했습니다.

### 4.1. 오버클라우드 업데이트 준비 실행

업데이트 프로세스를 위해 오버클라우드를 준비하려면 다음 작업을 수행하는 `openstack overcloud update prepare` 명령을 실행합니다.

- 오버클라우드 계획을 **OpenStack Platform 16.1**로 업데이트
- 업데이트를 위해 노드를 준비합니다

### 사전 요구 사항

- **Ceph** 서브스크립션을 사용하고 **Ceph** 스토리지 노드에 **overcloud-minimal** 이미지를 사용하도록 **director**가 구성된 경우 **roles\_data.yaml** 역할 정의 파일에서 **rhsm\_enforce** 매개변수가 **False**로 설정되어 있는지 확인해야 합니다.
- 사용자 정의 **NIC** 템플릿을 렌더링한 경우 오버클라우드 버전과 호환되지 않도록 업데이트된 버전의 **openstack-tripleo-heat-templates** 컬렉션을 사용하여 템플릿을 다시 생성해야 합니다. 사용자 지정 **NIC** 템플릿에 대한 자세한 내용은 **Advanced Overcloud Customization** 가이드의 [사용자 지정을 위해 기본 네트워크 인터페이스 템플릿 렌더링](#)을 참조하십시오.

### 절차

1. **stackrc** 파일을 소싱합니다.

```
$ source ~/stackrc
```



2.

**update** 준비 명령을 실행합니다.

```
$ openstack overcloud update prepare \
  --templates \
  --stack <stack_name> \
  -r <roles_data_file> \
  -n <network_data_file> \
  -e <environment_file> \
  -e <environment_file> \
  ...
```

환경과 관련된 다음 옵션을 포함합니다.

- 오버클라우드 스택 이름이 기본 오버클라우드와 다른 경우 업데이트 준비 명령에 **--stack** 옵션을 포함하고 **<stack\_name>** 을 스택 이름으로 교체합니다.
- 고유한 사용자 지정 역할을 사용하는 경우 사용자 지정 역할(**<roles\_data>**) 파일 (**-r**)을 포함합니다.
- 사용자 지정 네트워크를 사용하는 경우 구성 가능 네트워크(**<network\_data>**) 파일 (**-n**)을 포함합니다.
- 고가용성 클러스터를 배포하는 경우 업데이트 준비 명령에 **--ntp-server** 옵션을 포함하거나 환경 파일에 **NtpServer** 매개변수 및 값을 포함합니다.
- 모든 사용자 지정 구성 환경 파일(**-e**).

3.

업데이트가 완료될 때까지 기다립니다.

#### 4.2. 컨테이너 이미지 준비 실행

업데이트를 수행하기 전에 오버클라우드에 최신 **OpenStack Platform 16.1** 컨테이너 이미지가 필요합니다. 여기에는 **container\_image\_prepare** 외부 업데이트 프로세스를 실행해야 합니다. 이 프로세스를 실행하려면 **container\_image\_prepare** 태그가 지정된 작업에 대해 **openstack overcloud external-update run** 명령을 실행해야 합니다. 이러한 작업은 다음 작업을 수행합니다.

- 환경과 관련된 모든 컨테이너 이미지 구성을 자동으로 준비합니다.
- 이전에 이 옵션을 비활성화한 경우를 제외하고 관련 컨테이너 이미지를 언더클라우드로 가져옵니다.



참고

기본 스택 이름(**Overcloud**)을 사용하지 않는 경우 스택 이름을 `--stack <stack_name>` 옵션으로 교체하여 `<stack_name>` 을 스택 이름으로 설정합니다.

절차

1. **stackrc** 파일을 소싱합니다.

```
$ source ~/stackrc
```

2. **container\_image\_prepare** 태그가 지정된 작업에 대해 **openstack overcloud external-update run** 명령을 실행합니다.

```
$ openstack overcloud external-update run --stack <stack_name> --tags container_image_prepare
```

4.3. 선택 사항: 모든 오버클라우드 서버에서 **OVN-CONTROLLER** 컨테이너 업데이트

**Modular Layer 2 Open Virtual Network** 메커니즘 드라이버(**ML2/OVN**)를 사용하여 오버클라우드를 배포한 경우 **ovn-controller** 컨테이너를 최신 **RHOSP 16.1** 버전으로 업데이트합니다. 업데이트는 **ovn-controller** 컨테이너를 실행하는 모든 오버클라우드 서버에서 수행됩니다.



참고

오버클라우드 인 기본 스택 이름을 사용하지 않는 경우 스택 이름을 `--stack <stack_name>` 옵션으로 설정하고 `<stack_name>` 을 스택 이름으로 교체합니다.

절차

1. **stack** 사용자로 언더클라우드에 로그인합니다.

2. **stackrc** 파일을 소싱합니다.

```
$ source ~/stackrc
```

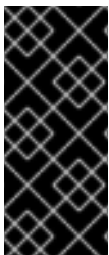
3. **ovn** 태그가 있는 작업에 대해 **openstack overcloud external-update run** 명령을 실행합니다.

```
$ openstack overcloud external-update run --stack <stack_name> --tags ovn
```

4. **ovn-controller** 컨테이너 업데이트가 완료될 때까지 기다립니다.

#### 4.4. 모든 컨트롤러 노드 업데이트

이 프로세스에서는 모든 컨트롤러 노드를 최신 **OpenStack Platform 16.1** 버전으로 업데이트합니다. 프로세스에는 **openstack overcloud update run** 명령을 실행하고 컨트롤러 노드로만 작업을 제한하는 **-limit Controller** 옵션을 포함합니다. 컨트롤 플레인 API는 마이너 업데이트 중에 완전히 사용할 수 있습니다.



##### 중요

구성 가능 역할을 기반으로 하는 노드의 경우 [BZ#1872404](#)가 해결될 때까지 컨트롤러, 메시징, 계산, Ceph 및 기타 역할을 업데이트하기 전에 **Database** 역할을 먼저 업데이트해야 합니다.



##### 참고

기본 스택 이름(**Overcloud**)을 사용하지 않는 경우 스택 이름을 **--stack <stack\_name>** 옵션으로 교체하여 **<stack\_name>**을 스택 이름으로 설정합니다.

#### 절차

1. **stackrc** 파일을 소싱합니다.

```
$ source ~/stackrc
```

2. **update** 명령을 실행합니다.

```
$ openstack overcloud update run --stack <stack_name> --limit Controller
```

3.

컨트롤러 노드 업데이트가 완료될 때까지 기다립니다.

#### 4.5. 모든 컴퓨팅 노드 업데이트

이 프로세스에서는 모든 컴퓨팅 노드가 최신 **OpenStack Platform 16.1** 버전으로 업데이트됩니다. 프로세스에는 **openstack overcloud update run** 명령을 실행하고 작업을 **Compute** 노드로만 제한하는 **--limit Compute** 옵션을 포함해야 합니다.

##### 병렬 처리 고려 사항

성능을 향상시키기 위해 다수의 컴퓨팅 노드를 업데이트하는 경우 노드 배치에서 **--limit Compute** 옵션을 사용하여 **openstack overcloud update run** 명령을 실행할 수 있습니다. 예를 들어 배포에 80개의 컴퓨팅 노드가 있는 경우 다음 명령을 실행하여 컴퓨팅 노드를 병렬로 업데이트할 수 있습니다.

```
$ openstack overcloud update run -y --limit 'Compute[0:19]' > update-compute-0-19.log 2>&1 &
$ openstack overcloud update run -y --limit 'Compute[20:39]' > update-compute-20-39.log 2>&1 &
$ openstack overcloud update run -y --limit 'Compute[40:59]' > update-compute-40-59.log 2>&1 &
$ openstack overcloud update run -y --limit 'Compute[60:79]' > update-compute-60-79.log 2>&1 &
```

'**Compute[0:19]**', '**Compute[20:39]**', '**Compute[40:59]**', '**Compute[60:79]**' 노드 공간을 임의로 분할하는 방법은 임의로 업데이트되는 노드를 제어할 수 없습니다.

특정 컴퓨팅 노드를 업데이트하려면 쉼표로 구분하여 배치에서 업데이트할 노드를 나열합니다.

```
$ openstack overcloud update run --limit <Compute0>,<Compute1>,<Compute2>,<Compute3>
```



##### 참고

기본 스택 이름(**Overcloud**)을 사용하지 않는 경우 스택 이름을 **--stack <stack\_name>** 옵션으로 교체하여 **<stack\_name>** 을 스택 이름으로 설정합니다.

##### 절차

1.

**stackrc** 파일을 소싱합니다.

```
$ source ~/stackrc
```

2. **update** 명령을 실행합니다.

```
$ openstack overcloud update run --stack <stack_name> --limit Compute
```

3. 컴퓨팅 노드 업데이트가 완료될 때까지 기다립니다.

#### 4.6. 모든 HCI 컴퓨팅 노드 업데이트

이 프로세스에서는 **HCI(Hyperconverged Infrastructure)** 컴퓨팅 노드를 업데이트합니다. 프로세스에는 다음이 포함됩니다.

- **openstack overcloud update run** 명령을 실행하고 **--limit ComputeHCI** 옵션을 포함하여 **HCI** 노드로만 작업을 제한합니다.
- **openstack overcloud external-update**를 실행하면 **--tags ceph** 명령을 실행하여 컨테이너화된 **Red Hat Ceph Storage 4** 클러스터에 대한 업데이트를 수행합니다.



#### 참고

기본 스택 이름(**Overcloud**)을 사용하지 않는 경우 스택 이름을 **--stack <stack\_name>** 옵션으로 교체하여 **<stack\_name>** 을 스택 이름으로 설정합니다.

#### 사전 요구 사항

- **ceph-mon** 서비스를 실행하는 **Ceph Monitor** 또는 컨트롤러 노드에서 **Red Hat Ceph Storage** 클러스터 상태가 정상이고 **pg** 상태가 **active+clean** 인지 확인합니다.

```
$ sudo podman exec -it ceph-mon-controller-0 ceph -s
```

**Ceph** 클러스터가 정상이면 **HEALTH\_OK** 상태를 반환합니다.

**Ceph** 클러스터 상태가 비정상인 경우 **HEALTH\_WARN** 또는 **HEALTH\_ERR** 의 상태를 반환합니다. 문제 해결 지침은 [Red Hat Ceph Storage 4 문제 해결 가이드를 참조하십시오.](#)

#### 절차

1. **stackrc** 파일을 소싱합니다.

```
$ source ~/stackrc
```

2. **update** 명령을 실행합니다.

```
$ openstack overcloud update run --stack <stack_name> --limit ComputeHCI
```

3. 노드 업데이트가 완료될 때까지 기다립니다.

4. **Ceph Storage update** 명령을 실행합니다. 예를 들면 다음과 같습니다.

```
$ openstack overcloud external-update run --stack <stack_name> --tags ceph
```

5. 컴퓨팅 HCI 노드 업데이트가 완료될 때까지 기다립니다.

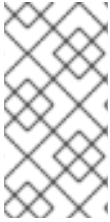
#### 4.7. 모든 CEPH STORAGE 노드 업데이트

이 프로세스는 **Ceph Storage** 노드를 업데이트합니다. 프로세스에는 다음이 포함됩니다.

- **openstack overcloud update run** 명령을 실행하고 **--limit CephStorage** 옵션을 포함하여 작업을 **Ceph Storage** 노드로만 제한합니다.
- **openstack overcloud external-update run** 명령을 실행하여 **ceph-ansible** 을 외부 프로세스로 실행하고 **Red Hat Ceph Storage 3** 컨테이너를 업데이트합니다.

#### 중요

**RHOSP 16.1**은 **RHEL 8.2**에서 지원됩니다. 그러나 **Ceph Storage** 역할 업데이트에 매핑된 호스트는 최신 주요 **RHEL** 릴리스로 업데이트합니다. 자세한 내용은 [Red Hat Ceph Storage](#)를 참조하십시오. 지원되는 구성.



## 참고

기본 스택 이름(**Overcloud**)을 사용하지 않는 경우 스택 이름을 `--stack <stack_name>` 옵션으로 교체하여 `<stack_name>` 을 스택 이름으로 설정합니다.

## 사전 요구 사항

- **ceph-mon** 서비스를 실행하는 **Ceph Monitor** 또는 컨트롤러 노드에서 **Red Hat Ceph Storage** 클러스터 상태가 정상이고 **pg** 상태가 **active+clean** 인지 확인합니다.

```
$ sudo podman exec -it ceph-mon-controller-0 ceph -s
```

**Ceph** 클러스터가 정상이면 **HEALTH\_OK** 상태를 반환합니다.

**Ceph** 클러스터 상태가 비정상인 경우 **HEALTH\_WARN** 또는 **HEALTH\_ERR** 의 상태를 반환합니다. 문제 해결 지침은 [Red Hat Ceph Storage 4 문제 해결 가이드](#)를 참조하십시오.

## 절차

1. **stackrc** 파일을 소싱합니다.

```
$ source ~/stackrc
```

2. 그룹 노드를 업데이트합니다.

그룹의 모든 노드를 업데이트하려면 다음을 수행합니다.

```
$ openstack overcloud update run --limit <GROUP_NAME>
```

그룹의 단일 노드를 업데이트하려면 다음을 수행합니다.

```
$ openstack overcloud update run --limit <GROUP_NAME> [NODE_INDEX]
```



참고

노드를 개별적으로 업데이트하도록 선택하는 경우 모든 노드를 업데이트해야 합니다.

그룹에서 첫 번째 노드의 인덱스는 0(0)입니다. 예를 들어 **CephStorage** 그룹의 첫 번째 노드를 업데이트하려면 다음을 수행합니다.

**OpenStack overcloud update run --limit CephStorage[0]**

3. 노드 업데이트가 완료될 때까지 기다립니다.

4. **Ceph Storage** 컨테이너 업데이트 명령을 실행합니다.

```
$ openstack overcloud external-update run --tags ceph
```

5. **Ceph Storage** 컨테이너 업데이트가 완료될 때까지 기다립니다.

4.8. 온라인 데이터베이스 업데이트 수행

일부 오버클라우드 구성 요소에는 데이터베이스 테이블의 온라인 업그레이드(또는 마이그레이션)가 필요합니다. 여기에는 **online\_upgrade** 외부 업데이트 프로세스를 실행해야 합니다. 이 프로세스를 실행하려면 **online\_upgrade** 태그가 지정된 작업에 대해 **openstack overcloud external-update run** 명령을 실행합니다. 이는 다음 구성 요소에 대한 온라인 데이터베이스 업데이트를 수행합니다.

- **OpenStack Block Storage(cinder)**
- **OpenStack Compute(nova)**

절차

1. **stackrc** 파일을 소싱합니다.

```
$ source ~/stackrc
```



2. **online\_upgrade** 태그를 사용하는 작업에 대해 **openstack overcloud external-update run** 명령을 실행합니다.

```
$ openstack overcloud external-update run --tags online_upgrade
```

#### 4.9. 업데이트 종료

업데이트하려면 오버클라우드 스택을 업데이트하기 위한 최종 단계가 필요합니다. 이렇게 하면 스택의 리소스 구조가 **OpenStack Platform 16.1**의 일반 배포와 일치할 수 있으며 향후 표준 **openstack overcloud deploy** 기능을 수행할 수 있습니다.

##### 절차

1. **stackrc** 파일을 소싱합니다.

```
$ source ~/stackrc
```

2. 오버클라우드에서 펜싱을 다시 활성화하려면 **fencing.yaml** 환경 파일에서 **EnableFencing** 매개변수를 **true** 로 설정합니다.

3. 업데이트 완료 명령을 실행합니다.

```
$ openstack overcloud update converge \
  --templates \
  --stack <stack_name> \
  -r <roles_data_file> \
  -n <network_data_file> \
  -e <environment_file> \
  -e <environment_file> \
  ...
  ...
```

환경과 관련된 다음 옵션을 포함합니다.

- **EnableFencing** 매개변수가 **true** 로 설정된 **fencing.yaml** 환경 파일.
- 오버클라우드 스택 이름이 기본 오버클라우드와 다른 경우 업데이트 준비 명령에 **--stack** 옵션을 포함하고 **<stack\_name>** 을 스택 이름으로 교체합니다.

- 고유한 사용자 지정 역할을 사용하는 경우 사용자 정의 역할 (<roles\_data>) 파일 (-r)을 포함하십시오.
  
  - 사용자 지정 네트워크를 사용하는 경우 구성 가능한 네트워크 (<network\_data>) 파일 (-n)을 포함합니다.
  
  - 모든 사용자 지정 구성 환경 파일(-e).
4. 업데이트 완료가 완료될 때까지 기다립니다.

## 5장. 오버클라우드 재부팅

마이너 **Red Hat OpenStack** 버전 업데이트 후 오버클라우드를 재부팅합니다. 재부팅은 연결된 커널, 시스템 수준 및 컨테이너 구성 요소 업데이트를 통해 노드를 새로 고칩니다. 이러한 업데이트는 성능 및 보안상의 이점을 제공할 수 있습니다.

다음 재부팅 절차를 수행하기 위해 다운타임을 계획합니다.

### 5.1. 컨트롤러 노드 및 구성 가능 노드 재부팅

컴퓨팅 노드와 **Ceph Storage** 노드를 제외하고, 구성 가능 역할을 기반으로 컨트롤러 노드 및 독립형 노드를 재부팅하려면 다음 단계를 완료합니다.

#### 절차

1. 재부팅하려는 노드에 로그인합니다.
2. 선택 사항: 노드에서 **Pacemaker** 리소스를 사용하는 경우 클러스터를 중지합니다.

```
[heat-admin@overcloud-controller-0 ~]$ sudo pcs cluster stop
```

3. 노드를 재부팅합니다.

```
[heat-admin@overcloud-controller-0 ~]$ sudo reboot
```

4. 노드가 부팅될 때까지 기다립니다.
5. 서비스를 확인합니다. 예를 들면 다음과 같습니다.

- a. 노드에서 **Pacemaker** 서비스를 사용하는 경우 노드가 클러스터에 다시 가입했는지 확인합니다.

```
[heat-admin@overcloud-controller-0 ~]$ sudo pcs status
```

- b. 노드에서 **Systemd** 서비스를 사용하는 경우 모든 서비스가 활성화되었는지 확인합니다.

```
[heat-admin@overcloud-controller-0 ~]$ sudo systemctl status
```

- c. 노드에서 컨테이너화된 서비스를 사용하는 경우 노드의 모든 컨테이너가 활성화되었는지 확인합니다.

```
[heat-admin@overcloud-controller-0 ~]$ sudo podman ps
```

## 5.2. CEPH STORAGE(OSD) 클러스터 재부팅

**Ceph Storage(OSD)** 노드 클러스터를 재부팅하려면 다음 단계를 완료합니다.

### 사전 요구 사항

- **ceph-mon** 서비스를 실행하는 **Ceph Monitor** 또는 컨트롤러 노드에서 **Red Hat Ceph Storage** 클러스터 상태가 정상이고 **pg** 상태가 **active+clean** 인지 확인합니다.

```
$ sudo podman exec -it ceph-mon-controller-0 ceph -s
```

**Ceph** 클러스터가 정상이면 **HEALTH\_OK** 상태를 반환합니다.

**Ceph** 클러스터 상태가 비정상인 경우 **HEALTH\_WARN** 또는 **HEALTH\_ERR**의 상태를 반환합니다. 문제 해결 지침은 [Red Hat Ceph Storage 4 문제 해결 가이드](#)를 참조하십시오.

### 절차

1. **ceph-mon** 서비스를 실행하는 **Ceph Monitor** 또는 컨트롤러 노드에 로그인하고 **Ceph Storage** 클러스터 재조정을 일시적으로 비활성화합니다.

```
$ sudo podman exec -it ceph-mon-controller-0 ceph osd set noout
$ sudo podman exec -it ceph-mon-controller-0 ceph osd set norebalance
```



## 참고

다중 스택 또는 DCN(Distributed Compute Node) 아키텍처가 있는 경우 **noout** 및 **norebalance** 플래그를 설정할 때 클러스터 이름을 지정해야 합니다. 예: `sudo podman exec -it ceph-mon-controller-0 ceph osd set noout --cluster <cluster_name>`

2. 재부팅할 첫 번째 **Ceph Storage** 노드를 선택하고 노드에 로그인합니다.

3. 노드를 재부팅합니다.

```
$ sudo reboot
```

4. 노드가 부팅될 때까지 기다립니다.

5. 노드에 로그인하고 클러스터 상태를 확인합니다.

```
$ sudo podman exec -it ceph-mon-controller-0 ceph status
```

**pgmap**이 모든 **pgs**를 정상(**active+clean**)으로 보고하는지 확인합니다.

6. 노드에서 로그아웃하고, 다음 노드를 재부팅한 후 상태를 확인합니다. 모든 **Ceph Storage** 노드가 재부팅될 때까지 이 프로세스를 반복합니다.

7. 완료되면 **ceph-mon** 서비스를 실행하는 **Ceph Monitor** 또는 컨트롤러 노드에 로그인하고 클러스터 재조정을 다시 활성화합니다.

```
$ sudo podman exec -it ceph-mon-controller-0 ceph osd unset noout
$ sudo podman exec -it ceph-mon-controller-0 ceph osd unset norebalance
```



## 참고

다중 스택 또는 DCN(Distributed Compute Node) 아키텍처가 있는 경우 **noout** 및 **norebalance** 플래그를 설정 해제할 때 클러스터 이름을 지정해야 합니다. 예: `sudo podman exec -it ceph-mon-controller-0 ceph osd set noout --cluster <cluster_name>`

8.

최종 상태 검사를 수행하여 클러스터가 **HEALTH\_OK**를 보고하는지 확인합니다.

```
$ sudo podman exec -it ceph-mon-controller-0 ceph status
```

### 5.3. 컴퓨팅 노드 재부팅

컴퓨팅 노드를 재부팅하려면 다음 단계를 완료합니다. **Red Hat OpenStack Platform** 환경에서 인스턴스 다운 타임을 최소화할 수 있도록 다음 절차에서는 재부팅할 컴퓨팅 노드에서 인스턴스를 마이그레이션하는 방법도 설명합니다. 이 작업은 다음 워크플로우에 따라 수행됩니다.

- 노드를 재부팅하기 전에 인스턴스를 다른 컴퓨팅 노드로 마이그레이션할지 여부 결정
- 새 인스턴스를 프로비저닝하지 않도록 재부팅할 컴퓨팅 노드를 선택한 뒤 비활성화
- 인스턴스를 다른 컴퓨팅 노드로 마이그레이션
- 빈 컴퓨팅 노드 재부팅
- 빈 컴퓨팅 노드 활성화

#### 사전 요구 사항

컴퓨팅 노드를 재부팅하기 전에 노드가 재부팅되는 동안 인스턴스를 다른 컴퓨팅 노드로 마이그레이션할지 여부를 결정해야 합니다.

컴퓨팅 노드 간에 가상 머신 인스턴스를 마이그레이션할 때 실행할 수 있는 마이그레이션 제약 조건 목록을 검토합니다. 자세한 내용은 *Configuring the Compute Service for Instance Creation*에서 [마이그레이션 제한 조건](#)을 참조하십시오.

인스턴스를 마이그레이션할 수 없는 경우 다음과 같은 코어 템플릿 매개변수를 설정하여 컴퓨팅 노드를 재부팅한 후의 인스턴스 상태를 제어할 수 있습니다.

#### NovaResumeGuestsStateOnHostBoot

재부팅한 후에 컴퓨팅 노드에서 인스턴스를 동일한 상태로 되돌릴지 여부를 결정합니다. **False**로

설정하면 인스턴스가 다운된 상태로 유지되며 수동으로 시작해야 합니다. 기본값은 다음과 같습니다.  
**False**

### NovaResumeGuestsShutdownTimeout

재부팅하기 전에 인스턴스가 종료될 때까지 대기하는 시간(초)입니다. 이 값을 **0**으로 설정하지 않는 것이 좋습니다. 기본값은 다음과 같습니다. **300**

오버클라우드 매개변수 및 사용법에 관한 자세한 내용은 [Overcloud Parameters](#)를 참조하십시오.

### 절차

1. **stack** 사용자로 언더클라우드에 로그인합니다.

2. 모든 컴퓨팅 노드 및 해당 **UUID**를 나열합니다.

```
$ source ~/stackrc
(undercloud) $ openstack server list --name compute
```

재부팅할 컴퓨팅 노드의 **UUID**를 확인합니다.

3. 언더클라우드에서 컴퓨팅 노드를 선택합니다. 노드를 비활성화합니다.

```
$ source ~/overcloudrc
(overcloud) $ openstack compute service list
(overcloud) $ openstack compute service set <hostname> nova-compute --disable
```

4. 컴퓨팅 노드에 모든 인스턴스를 나열합니다.

```
(overcloud) $ openstack server list --host <hostname> --all-projects
```

5. 인스턴스를 마이그레이션하지 않으려면 [이 단계](#)로 건너뛩니다.

6. 인스턴스를 다른 컴퓨팅 노드로 마이그레이션하려면 다음 명령 중 하나를 사용합니다.

- 인스턴스를 다른 호스트로 마이그레이션합니다.

```
(overcloud) $ openstack server migrate <instance_id> --live <target_host> --wait
```

- **nova-scheduler**에서 대상 호스트를 자동으로 선택하도록 합니다.

```
(overcloud) $ nova live-migration <instance_id>
```

- 한 번에 모든 인스턴스를 실시간 마이그레이션합니다.

```
$ nova host-evacuate-live <hostname>
```



참고

**nova** 명령으로 인해 몇 가지 사용 중단 경고가 표시될 수 있으며, 이러한 경고는 무시해도 됩니다.

7. 마이그레이션이 완료될 때까지 기다립니다.

8. 마이그레이션을 성공적으로 완료했음을 확인합니다.

```
(overcloud) $ openstack server list --host <hostname> --all-projects
```

9. 선택한 컴퓨팅 노드에 남은 항목이 없을 때까지 인스턴스를 계속 마이그레이션합니다.

10. 컴퓨팅 노드에 로그인하고 노드를 재부팅합니다.

```
[heat-admin@overcloud-compute-0 ~]$ sudo reboot
```

11. 노드가 부팅될 때까지 기다립니다.

12. 컴퓨팅 노드를 다시 활성화합니다.

```
$ source ~/overcloudrc
(overcloud) $ openstack compute service set <hostname> nova-compute --enable
```



13.

컴퓨팅 노드가 활성화되었는지 확인합니다.

```
(overcloud) $ openstack compute service list
```