



Red Hat OpenStack Platform 17.1

OVN 메커니즘 드라이버로 마이그레이션

Red Hat OpenStack Platform Networking 서비스(neutron)를 ML2/OVS 메커니즘 드라이버에서 ML2/OVN 메커니즘 드라이버로 마이그레이션

Red Hat OpenStack Platform 17.1 OVN 메커니즘 드라이버로 마이그레이션

Red Hat OpenStack Platform Networking 서비스(neutron)를 ML2/OVS 메커니즘 드라이버에서 ML2/OVN 메커니즘 드라이버로 마이그레이션

OpenStack Team
rhos-docs@redhat.com

법적 공지

Copyright © 2024 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

Open vSwitch 메커니즘 드라이버(ML2/OVS)를 사용하는 Modular Layer 2 플러그인에서 ML2/OVN(Open Virtual Networking)을 사용하는 Modular Layer 2 플러그인으로 Red Hat OpenStack Platform Networking 서비스(neutron)를 마이그레이션하는 방법

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체	3
RED HAT 문서에 관한 피드백 제공	4
1장. OVS에서 OVN으로 ML2 메커니즘 드라이버 마이그레이션 계획	5
1.1. OVN 및 OVS 메커니즘 드라이버에서 기능 지원	6
1.2. ML2/OVS에서 ML2/OVN 인플레이스 마이그레이션: 검증 및 금지 시나리오	7
2장. OVS에서 OVN으로 ML2 메커니즘 드라이버 마이그레이션	9
2.1. OVN 메커니즘 드라이버로의 마이그레이션을 위한 환경 준비	9
2.2. OVS에서 OVN으로 ML2 메커니즘 드라이버 마이그레이션을 위한 컨테이너 이미지 준비	13
2.3. VXLAN OVS 배포에서 마이그레이션할 MTU 감소	14
2.4. OVS에서 OVN으로 ML2 메커니즘 드라이버 마이그레이션	16
3장. ML2/OVS에서 ML2/OVN으로 마이그레이션 되돌리기	18
3.1. ML2/OVN 마이그레이션을 되돌리도록 컨트롤러 노드 복원	18

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.

Jira에서 문서 피드백 제공

[Create Issue](#) 양식을 사용하여 OpenShift (RHOSO) 또는 이전 Red Hat OpenStack Platform (RHOSP)의 Red Hat OpenStack Services 문서에 대한 피드백을 제공합니다. RHOSO 또는 RHOSP 문서에 대한 문제를 생성할 때 RHOSO Jira 프로젝트에 문제가 기록되어 피드백의 진행 상황을 추적할 수 있습니다.

문제 생성 양식을 완료하려면 Jira에 로그인해야 합니다. Red Hat Jira 계정이 없는 경우 <https://issues.redhat.com>에서 계정을 생성할 수 있습니다.

1. 다음 링크를 클릭하여 **문제 생성** 페이지를 엽니다.
<https://issues.redhat.com/secure/CreateIssueDetails!init.jspx?pid=12336920&summary=Documentation%20feedback:%20%3CAdd%20summary%20here%3E&i:Include+the+documentation+URL,+the%20chapter+or+section+number,+and+a+detailed+descrip>
2. **요약** 및 **설명** 필드를 작성합니다. **설명** 필드에 문서 URL, 장 또는 섹션 번호, 문제에 대한 자세한 설명을 포함합니다. 양식의 다른 필드를 수정하지 마십시오.
3. **생성**을 클릭합니다.

1장. OVS에서 OVN으로 ML2 메커니즘 드라이버 마이그레이션 계획

Red Hat은 현재 대부분의 고객을 위해 ML2/OVS 메커니즘 드라이버를 통해 즉시 이점을 제공하기 때문에 RHOSP 15.0부터 모든 새로운 배포의 기본 메커니즘 드라이버로 ML2/OVN을 선택했습니다. 이러한 장점은 ML2/OVN 기능 세트를 지속적으로 개선하고 개선하는 동안 각 릴리스와 곱해집니다.

RHOSP 17.0에서는 ML2/OVS 메커니즘 드라이버가 더 이상 사용되지 않습니다. Red Hat은 여러 릴리스에서 ML2/OVS를 ML2/OVN으로 교체하고 있습니다.

RHOSP 17 릴리스를 통해 더 이상 사용되지 않는 ML2/OVS 메커니즘 드라이버에 대한 지원이 제공됩니다. 이 기간 동안 ML2/OVS 드라이버는 유지 관리 모드로 유지되어 버그 수정 및 정상적인 지원이 제공됩니다. 대부분의 새로운 기능 개발은 ML2/OVN 메커니즘 드라이버에서 수행됩니다.

RHOSP 18.0에서 Red Hat은 ML2/OVS 메커니즘 드라이버를 완전히 제거하고 지원을 중단할 계획입니다.

기존 RHOSP(Red Hat OpenStack Platform) 배포에서 ML2/OVS 메커니즘 드라이버를 사용하는 경우 지금 시작하여 ML2/OVS 메커니즘 드라이버를 ML2/OVN 메커니즘 드라이버로 교체할 수 있는 이점과 가능성을 평가하십시오. RHOSP 16.2 및 RHOSP 17.1에서 마이그레이션이 지원됩니다.



참고

Red Hat에서는 ML2/OVS에서 ML2/OVN으로 마이그레이션을 시도하기 전에 사전 지원 케이스를 제출해야 합니다. Red Hat은 사전 지원 케이스 없이는 마이그레이션을 지원하지 않습니다. [link:https://access.redhat.com/solutions/2186261](https://access.redhat.com/solutions/2186261)를 참조하십시오.

이 평가 초기에 Red Hat Technical Account Manager 또는 Red Hat Global Professional Services에 참여하십시오. 마이그레이션 결정 시 필요한 사전 예방 지원 케이스를 작성하는 것 외에도 Red Hat은 다음과 같은 기본 질문을 통해 계획 및 준비를 지원할 수 있습니다.

마이그레이션하려면 어떻게 해야 하나요?

타이밍은 비즈니스 요구 사항과 ML2/OVN 오픈링에 대한 지속적인 개선 상태를 포함한 여러 요인에 따라 달라집니다. [OVN 및 OVS 메커니즘 드라이버 및 ML2/OVS에서 ML2/OVN 인플레이스 마이그레이션: 검증된 시나리오 및 금지된 시나리오](#)를 참조하십시오.

인플레이스 마이그레이션 또는 병렬 마이그레이션?

다양한 요소에 따라 다음과 같은 마이그레이션 방법 중에서 선택할 수 있습니다.

- 병렬 마이그레이션. ML2/OVN을 사용하는 새 병렬 배포를 생성한 다음 작업을 해당 배포로 이동합니다.
- 인플레이스 마이그레이션. 이 문서에 설명된 대로 `ovn_migration.sh` 스크립트를 사용합니다. Red Hat은 RHOSP director에서 관리하는 배포에서만 **ovn_migration.sh** 스크립트를 지원합니다.



주의

ML2/OVS에서 ML2/OVN 마이그레이션은 완전히 되돌릴 수 없는 방식으로 환경을 변경합니다. 적절한 백업 단계를 따르고 지침을 되돌리면 실패 또는 중단된 마이그레이션을 되돌릴 수 있지만 복원된 OVS 환경이 원본에서 변경될 수 있습니다. 프로덕션 환경에서 마이그레이션하기 전에 사전 지원 케이스를 제출하십시오. 그런 다음 Red Hat Technical Account Manager 또는 Red Hat Global Professional Services와 협력하여 운영 환경과 밀접하게 유사한 단계 환경에서 백업 및 마이그레이션 계획을 작성하고 마이그레이션을 테스트합니다. 잠재적인 마이그레이션 복원을 위해 백업을 준비하도록 선택하는 경우 스테이지 환경에서 마이그레이션 반환도 테스트해야 합니다.

1.1. OVN 및 OVS 메커니즘 드라이버에서 기능 지원

RHOSP(Red Hat OpenStack Platform) 기능의 가용성을 OVS의 일부로 OVN 메커니즘 드라이버 마이그레이션 계획의 일부로 검토하십시오.

기능	OVN RHOS P 16.2	OVN RHOS P 17.1	OVS RHOS P 16.2	OVS RHOS P 17.1	추가 정보
OVN DHCP를 사용하여 베어 메탈 머신 프로비저닝	없음	없음	제공 됨	제공 됨	현재 OVN의 기본 제공 DHCP 서버는 baremetal 노드를 프로비저닝할 수 없습니다. provisioning 네트워크에 DHCP를 제공할 수 없습니다. Chainbooting iPXE에는 OVN DHCP 서버에서 지원되지 않는 태그 지정(dnsmasq에서 <code>--dhcp-match</code>)이 필요합니다. https://bugzilla.redhat.com/show_bug.cgi?id=1622154 을 참조하십시오.
VLAN 프로젝트(테넌트 네트워크)의 VF(direct) 포트에서 North/south 라우팅	없음	없음	제공 됨	제공 됨	코어 OVN 제한 사항. https://bugs.launchpad.net/neutron/+bug/1875852 을 참조하십시오.
내부 DNS 레코드의 역방향 DNS	없음	제공 됨	제공 됨	제공 됨	https://bugzilla.redhat.com/show_bug.cgi?id=2211426 을 참조하십시오.
격리된 네트워크의 내부 DNS 확인	없음	없음	제공 됨	제공 됨	OVN은 DNS 서비스에 포트를 할당하지 않기 때문에 격리된 네트워크에 대한 내부 DNS 확인을 지원하지 않습니다. OVS는 dnsmasq를 사용하므로 OVS 배포에 영향을 미치지 않습니다. https://issues.redhat.com/browse/OSP-25661 을 참조하십시오.

기능	OVN RHOS P 16.2	OVN RHOS P 17.1	OVS RHOS P 16.2	OVS RHOS P 17.1	추가 정보
보안 그룹 로깅	기술 프리뷰	예	없음	없음	RHOSP는 OVS 메커니즘 드라이버를 사용한 보안 그룹 로깅을 지원하지 않습니다.
상태 비저장 보안 그룹	없음	제공됨	아니요	아니요	보안 그룹 구성 을 참조하십시오.
DCS(Load-balancing service distributed virtual routing)	제공됨	제공됨	없음	아니요	OVS 메커니즘 드라이버는 DVR이 활성화된 경우에도 컨트롤러 또는 네트워크 노드를 통해 서비스 트래픽 로드 밸런싱을 라우팅합니다. OVN 메커니즘 드라이버는 컴퓨팅 노드를 통해 직접 서비스 트래픽 로드 밸런싱을 라우팅합니다.
IPv6 DVR	제공됨	제공됨	없음	아니요	OVS 메커니즘 드라이버를 사용하면 RHOSP에서 DVR이 활성화된 경우에도 IPv6 트래픽을 컴퓨팅 노드에 배포하지 않습니다. 모든 수신/추적 트래픽은 중앙 집중식 컨트롤러 또는 네트워크 노드를 통해 이루어집니다. IPv6 DVR이 필요한 경우 OVN 메커니즘 드라이버를 사용합니다.
DVR 및 계층 3 고가용성 (L3 HA)	제공됨	제공됨	없음	없음	OVS 메커니즘 드라이버를 사용한 RHOSP 배포는 L3 HA와 함께 DVR을 지원하지 않습니다. RHOSP director에서 DVR을 사용하는 경우 L3 HA가 비활성화됩니다. 즉, 네트워킹 서비스는 여전히 네트워크 노드에 라우터를 예약하고 L3 에이전트 간에 라우터를 로드합니다. 그러나 하나의 에이전트가 실패하면 이 에이전트에서 호스팅하는 모든 라우터도 실패합니다. 이는 SNAT 트래픽에만 영향을 미칩니다. Red Hat은 이러한 경우 allow_automatic_l3agent_failover 기능을 사용하는 것이 좋습니다. 따라서 하나의 네트워크 노드가 실패하면 라우터가 다른 노드로 다시 예약됩니다.

1.2. ML2/OVS에서 ML2/OVN 인플레이스 마이그레이션: 검증 및 금지 시나리오

Red Hat은 마이그레이션 시나리오를 지속적으로 테스트하고 개선합니다. Red Hat Technical Account Manager 또는 Global Professional Services와 협력하여 OVS 배포가 유효한 인플레이스 마이그레이션 시나리오의 기준을 충족하는지 확인합니다.

1.2.1. 검증된 ML2/OVS에서 ML2/OVN 마이그레이션 시나리오

Red Hat은 다음 마이그레이션 경로를 테스트했습니다.

- DVR에 분산 가상 라우팅(DVR)

- no-DVR(No-DVR)에 대한 중앙 집중식 라우팅
- no-DVR to DVR

테스트 결과 다음 포트 구성으로 워크로드가 포함되었습니다.

- 표준 포트
- SR-IOV 포트
- 트렁크 포트

성공적인 테스트에는 iptables_hybrid 및 Open vSwitch 방화벽 드라이버도 포함되어 있었습니다.



참고

각 테스트에서 마이그레이션 전 환경이 greenfield ML2/OVS 배포로 생성되었습니다.

1.2.2. 검증되지 않은 ML2/OVS에서 ML2/OVN 인플레이스 마이그레이션 시나리오

Red Hat에서 기본 문제가 해결되었음을 알릴 때까지 다음 시나리오에서는 인플레이스 ML2/OVS를 ML2/OVN으로 마이그레이션할 수 없습니다.

- GRE 네트워크를 사용한 OVS 사전 마이그레이션 배포
- VXLAN을 사용하여 VXLAN을 OVN으로 사용하는 OVS
- VLAN 프로젝트 네트워크 및 DVR이 있는 OVS 사전 마이그레이션 네트워크
- SR-IOV 및 DVR을 사용하여 OVN에 OVS 사전 마이그레이션 배포

1.2.3. ML2/OVS에서 ML2/OVN 인플레이스 마이그레이션 및 보안 그룹 규칙

원래 ML2/OVS 배포의 사용자 지정 보안 그룹 규칙이 대상 ML2/OVN 배포와 호환되는지 확인합니다.

2장. OVS에서 OVN으로 ML2 메커니즘 드라이버 마이그레이션

2.1. OVN 메커니즘 드라이버로의 마이그레이션을 위한 환경 준비

성공적인 마이그레이션에 환경 평가 및 준비는 매우 중요합니다. Red Hat 기술 계정 관리자 또는 글로벌 전문 서비스는 다음 단계를 안내합니다.

참고

비SR-IOV 환경에서 **tripleo_controller_group_name**의 기본값은 **Controller**입니다. SR-IOV 환경에서 **tripleo_controller_group_name** 값은 **ControllerSriov** 일 수 있습니다. 선택적 사전 마이그레이션 백업 단계에서 사용되는 Ansible 플레이북에서는 값 **Controller**를 예상합니다. 값이 **ControllerSriov** 또는 **Controller** 대신 다른 값인 경우 백업 작업이 실패할 수 있습니다.

선택적 백업 단계를 수행하려는 경우

/usr/share/ansible/roles/backup_and_restore/defaults/main.yml 및 /

usr/share/ansible/ovn-migration/playbooks/roles/recovery-

backup/defaults/main.yml 파일을 확인합니다. **Controller** 이외의

tripleo_controller_group_name이 포함된 경우 해결 방법은 Red Hat 지원에 문의하십시오.

이 문제의 수정 상태를 추적하려면 https://bugzilla.redhat.com/show_bug.cgi?id=2158396을 참조하십시오.

사전 요구 사항

- 배포는 최신 RHOSP 17.1 버전입니다. 즉, OpenStack 버전을 업그레이드하거나 업데이트해야 하는 경우 먼저 업그레이드 또는 업데이트를 수행한 다음 ML2/OVS to ML2/OVN 마이그레이션을 수행합니다.
- 각 서버넷 풀에는 하나 이상의 IP 주소를 사용할 수 있습니다. OVN 메커니즘 드라이버는 각 서버넷에 대한 메타데이터 포트를 생성합니다. 각 메타데이터 포트는 IP 주소 풀의 IP 주소를 클레임합니다.
- Red Hat Technical Account Manager 또는 Global Professional Services와 협력하여 마이그레이션을 계획하고 사전 대응 지원 사례를 제출했습니다. 진행 중인 [케이스를 제출하는 방법](#)을 참조하십시오.
- ML2/OVS 배포에서 VXLAN 프로젝트 네트워크를 사용하는 경우 [2.3절. "VXLAN OVS 배포에서 마이그레이션할 MTU 감소"](#)에 설명된 잠재적인 조정을 검토하십시오.

프로세스

1. ML2/OVN 단계 배포를 생성하여 대상 ML2/OVN 배포의 기본 구성을 가져오고 대상 배포의 가능성을 테스트합니다.
예정된 마이그레이션 후 프로덕션 배포와 동일한 기본 역할, 라우팅 및 토폴로지 스테이지 배포를 설계합니다. 모든 배포 인수와 함께 전체 **openstack overcloud deploy** 명령을 **overcloud-deploy.sh** 라는 파일에 저장합니다. 또한 **openstack overcloud deploy** 명령에서 참조하는 파일 (예: 환경 파일)을 저장합니다. 마이그레이션의 대상 ML2/OVN 환경을 구성하려면 이 절차의 뒷 부분에서 이러한 파일이 필요합니다.



참고

이러한 파일은 스테이지 배포 및 마이그레이션 생성에만 사용됩니다. 마이그레이션 후 다시 사용하지 마십시오.

2. **openstack-neutron-ovn-migration-tool** 을 설치합니다.

```
sudo dnf install openstack-neutron-ovn-migration-tool
```

3. 1단계에서 생성한 **overcloud-deploy.sh** 스크립트를 복사하고 복사본의 이름을 **overcloud-migrate-ovn.sh**로 변경합니다. **overcloud-migrate-ovn.sh** 내부의 **overcloud deploy** 명령의 모든 경로가 여전히 올바른지 확인합니다. 후속 단계에서 **overcloud-migrate-ovn.sh** 스크립트에서 일부 인수를 사용자 지정합니다.
4. 다음 목록에서 마이그레이션 시나리오를 찾아 **overcloud-migrate-ovn.sh** 에서 **openstack deploy** 명령을 사용자 지정하는 데 적절한 단계를 수행합니다.



주의

배포 명령에서 환경 파일을 추가하는 **-e** 인수 순서에 주의하십시오. 일반 기본값(예: **neutron-ovn-dvr-ha.yaml**)을 사용하는 환경 파일은 브리지 매핑과 같은 사용자 지정 네트워크 환경 설정으로 파일을 지정하는 **-e** 인수 앞에야 합니다.

시나리오 1: DVR에서 DVR, 컴퓨팅 노드가 외부 네트워크에 연결되어 있음

- **overcloud-migrate-ovn.sh** 에서 **openstack overcloud deploy** 명령에 사용자 지정 heat 템플릿 파일 인수를 추가합니다. 코어 템플릿 파일 인수 뒤에 추가합니다.
다음 명령 예제에서는 기본 **neutron-ovn-dvr-ha.yaml** heat 템플릿 파일을 사용합니다. 배포에서 여러 heat 파일을 사용하여 OVN 환경을 정의할 수 있습니다. 별도의 **-e** 인수를 사용하여 각각을 추가합니다.

```
openstack overcloud deploy \
--templates /usr/share/openstack-tripleo-heat-templates \
...
-e /usr/share/openstack-tripleo-heat-templates/environments/services/ \
neutron-ovn-dvr-ha.yaml
```

시나리오 2: 중앙 집중식 라우팅으로의 중앙 라우팅 (DVR 없음)

- 배포에서 SR-IOV 및 기타 NFV 기능을 사용하는 경우 **overcloud-migrate-ovn.sh** 에서 **-e** 인수를 사용하여 **openstack overcloud deploy** 명령에 SR-IOV 환경 매개변수를 추가합니다. 코어 템플릿 환경 파일 인수 및 기타 사용자 지정 환경 파일 인수 뒤에 SR-IOV 환경 파일을 추가합니다. SR-IOV 환경 파일의 예는 **/usr/share/openstack-tripleo-heat-templates/environments/services/neutron-ovn-sriov.yaml** 을 참조하십시오.
- 사용자 지정 네트워크 수정 사항은 마이그레이션 전과 동일하게 둡니다.

시나리오 3: DVR으로의 중앙 라우팅 및 **br-ex**를 통해 외부 네트워크에 연결된 컴퓨팅 노드

- **br-ex** 브리지를 통해 컴퓨팅 노드가 외부 네트워크에 연결되어 있는지 확인합니다. 예를 들어 **compute-dvr.yaml** 과 같은 환경 파일에서 다음 매개변수를 설정합니다. 그런 다음 **-e** 를 사용하여 **overcloud-migrate-ovn.sh** 스크립트의 **openstack overcloud deploy** 명령에 환경 파일을 추가합니다.

```
type: ovs_bridge
# Defaults to br-ex, anything else requires specific # bridge mapping entries for it to
be used.
name: bridge_name
use_dhcp: false
members:
-
  type: interface
  name: nic3
  # force the MAC address of the bridge to this interface
  primary: true
```

5. **overcloud-migrate-ovn.sh** 의 **overcloud deploy** 명령 끝에 다음 인수를 추가합니다.

```
-e /usr/share/openstack-tripleo-heat-templates/environments/disable-container-manage-
clean-orphans.yaml \
-e $HOME/ovn-extras.yaml
```

6. 라우터 가 환경 파일 또는 템플릿에서 **NeutronServicePlugins** 또는 **NeutronPluginExtensions** 의 값으로 표시되는 경우 값 라우터를 **ovn- router** 로 교체합니다. 예를 들어 **tripleo-heat-templates/environments/services/neutron-ovn-dvr-ha.yaml**:

```
parameter_defaults:
  NeutronServicePlugins: "ovn-router,trunk,qos,placement"
```

7. 모든 사용자에게 **overcloud-migrate-ovn.sh** 파일에 대한 실행 권한이 있는지 확인합니다. 이 스크립트에는 마이그레이션 프로세스 중에 실행 권한이 필요합니다.

```
$ chmod a+x ~/overcloud-migrate-ovn.sh
```

8. **내보내기** 명령을 사용하여 다음 마이그레이션 관련 환경 변수를 설정합니다. 예를 들면 다음과 같습니다.

```
$ export OVERCLOUDRC_FILE=~/.myovercloudrc
```

STACKRC_FILE

언더클라우드의 stackrc 파일입니다.
기본값: ~/stackrc

OVERCLOUDRC_FILE

언더클라우드의 overcloudrc 파일입니다.
기본값: ~/overcloudrc

OVERCLOUD_OVN_DEPLOY_SCRIPT

배포 스크립트입니다.
기본값: ~/overcloud-migrate-ovn.sh

DHCP_RENEWAL_TIME

DHCP 에이전트 구성 파일에서 구성하는 DHCP 갱신 시간(초)입니다.
기본값: 30

BACKUP_MIGRATION_IP

백업이 저장되는 서버의 IP 주소입니다.
기본값: 192.168.24.1

BACKUP_MIGRATION_CTL_PLANE_CIDRS

백업할 모든 노드에 대해 CIDR 표기법에서 쉼표로 구분된 컨트롤 플레인 서브넷 문자열입니다.
기본값: 192.168.24.0/24

/usr/bin/ovn_migraton.sh 파일의 시작 부분에서 모든 관련 환경 변수 목록을 볼 수 있습니다.

9. ovn-migration 디렉터리에 있고 **ovn_migration.sh generate-inventory** 명령을 실행하여 인벤토리 파일 **hosts_for_migration** 및 **ansible.cfg** 파일을 생성합니다.

```
$ ovn_migration.sh generate-inventory | sudo tee -a /var/log/ovn_migration_output.txt
```

10. **hosts_for_migration** 파일을 검토하여 정확성을 확인합니다.

- a. 목록이 사용자 환경과 일치하는지 확인합니다.
- b. 각 노드에 ovn 컨트롤러가 있는지 확인합니다.
- c. 목록 항목이 없는 목록 제목(예: [ovn-controllers])이 없는지 확인합니다.
- d. ovn 마이그레이션 디렉터리에서 **ansible -i hosts_for_migration -m ping** 명령을 모두 실행합니다.

11. (선택 사항) 마이그레이션 중에 예기치 않은 상황이 발생하는 경우 잠재적인 마이그레이션을 준비하기 위해 배포를 백업합니다.

- a. 파일 **setup_rear_extra_vars.yaml** 파일이 있는 경우 다음 행을 제거합니다.

```
USER_INPUT_TIMEOUT: 5
KERNEL_CMDLINE: unattended
ISO_RECOVER_MODE: unattended
```

- b. 컨트롤 플레인을 백업합니다. 선택한 백업 메커니즘을 사용하여 컨트롤러 노드를 백업합니다. 지원되는 옵션은 ReaR 백업 도구를 사용하는 기본 **ovn-migration.sh** 백업 명령입니다.

```
$ ovn_migration.sh backup
```

- c. 오버클라우드 배포에 사용된 템플릿 및 환경 파일을 백업합니다. **ovn-migration.sh** 백업 명령에서 오버클라우드를 백업하지 않습니다. 부분적인 마이그레이션 또는 실패한 후 컨트롤러 노드를 복원해야 하는 경우 OVS 오버클라우드를 복원하려면 이 백업이 필요합니다.
- d. 원래 RHOSP 16.2 ML2/OVS 배포를 배포하는 데 사용한 스크립트를 복사합니다. 예를 들어 원본 스크립트의 이름은 **overcloud_deploy.sh** 일 수 있습니다. 복사 이름을 **overcloud-revert-ovs.sh** 로 지정합니다.

- e. 다음 콘텐츠를 사용하여 **/home/stack/ovs-extra.yml** 파일을 생성합니다.

```
parameter_defaults:
  ForceNeutronDriverUpdate: true
```

- f. **overcloud-revert-ovs.sh** 의 최종 환경 파일 인수가 다음과 같은지 확인합니다.

```
-e /home/stack/ovs-extra.yml
```

- g. **overcloud-revert-ovs.sh** 를 안전하게 저장합니다. 실패한 마이그레이션을 되돌리는 경우 이 작업이 필요합니다. .

12. 2.2절. “OVS에서 OVN으로 ML2 메커니즘 드라이버 마이그레이션을 위한 컨테이너 이미지 준비”으로 이동합니다.

2.2. OVS에서 OVN으로 ML2 메커니즘 드라이버 마이그레이션을 위한 컨테이너 이미지 준비

성공적인 마이그레이션에 환경 평가 및 준비는 매우 중요합니다. Red Hat 기술 계정 관리자 또는 글로벌 전문 서비스는 다음 단계를 안내합니다.

사전 요구 사항

- OVS에서 OVN으로 ML2 메커니즘 드라이버를 마이그레이션하기 위한 환경 준비 단계를 완료했습니다.

프로세스

1. ML2/OVN으로 마이그레이션한 후 사용할 새 컨테이너 이미지를 준비합니다.

- a. **containers-prepare-parameter.yaml** 파일이 없는 경우 홈 디렉터리에 생성합니다.

```
$ test -f $HOME/containers-prepare-parameter.yaml || sudo openstack tripleo container
image prepare default \
--output-env-file $HOME/containers-prepare-parameter.yaml
```

- b. **containers-prepare-parameter.yaml** 이 \$HOME/overcloud-migrate-ovn.sh 및 \$HOME/overcloud-deploy.sh 파일의 끝에 있는지 확인합니다.

- c. **containers-prepare-parameter.yaml** 파일에서 neutron_driver를 ovn으로 변경합니다.

```
$ sed -i -E 's/neutron_driver:([ ]w+)/neutron_driver: ovn/' $HOME/containers-prepare-
parameter.yaml
```

- d. neutron_driver에 대한 변경 사항을 확인합니다.

```
$ grep neutron_driver $HOME/containers-prepare-parameter.yaml
neutron_driver: ovn
```

- e. 이미지를 업데이트합니다.

```
$ sudo openstack tripleo container image prepare \
--environment-file /home/stack/containers-prepare-parameter.yaml
```



참고

containers-prepare-parameter.yaml 파일의 전체 경로를 제공합니다. 그렇지 않으면 이미지 목록을 업데이트하거나 오류 메시지를 제공하지 않고 명령이 매우 빠르게 완료됩니다.

2. 언더클라우드에서 업데이트된 이미지를 검증합니다.

```
. Log in to the undercloud as the user `stack` and source the stackrc file.
$ source ~/stackrc
$ openstack tripleo container image list | grep '\-ovn'
```

목록은 다음 예와 유사해야 합니다. OVN 데이터베이스, OVN 컨트롤러, 메타데이터 에이전트 및 neutron 서버 에이전트용 컨테이너가 포함됩니다.

```
docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp17-openstack-ovn-
northd:17.1_20240725.1
docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp17-openstack-ovn-sb-db-
server:17.1_20240725.1
docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp17-openstack-ovn-
controller:17.1_20240725.1
docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp17-openstack-neutron-server-
ovn:17.1_20240725.1
docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp17-openstack-ovn-nb-db-
server:17.1_20240725.1
docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp17-openstack-neutron-
metadata-agent-ovn:17.1_20240725.1
```

RHOSP 16.2에서 OVN 메커니즘 드라이버로 마이그레이션하는 경우 목록은 다음과 유사합니다.

```
docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp16-openstack-ovn-
northd:16.2_20211110.2
docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp16-openstack-ovn-sb-db-
server:16.2_20211110.2
docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp16-openstack-ovn-
controller:16.2_20211110.2
docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp16-openstack-neutron-server-
ovn:16.2_20211110.2
docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp16-openstack-ovn-nb-db-
server:16.2_20211110.2
docker://undercloud-0.ctlplane.redhat.local:8787/rh-osbs/rhosp16-openstack-neutron-
metadata-agent-ovn:16.2_20211110.2
```

3. 원래 배포에서 VXLAN을 사용하는 경우 최대 전송 단위(MTU) 값을 조정해야 할 수 있습니다. [2.3절. "VXLAN OVS 배포에서 마이그레이션할 MTU 감소"](#) 으로 이동합니다.
원래 배포에서 VLAN 네트워크를 사용하는 경우 MTU 조정을 건너뛰고 [2.4절. "OVS에서 OVN으로 ML2 메커니즘 드라이버 마이그레이션"](#) 로 진행할 수 있습니다.

2.3. VXLAN OVS 배포에서 마이그레이션할 MTU 감소

마이그레이션 전 OVS 배포에서 VXLAN 터널링 프로토콜을 사용하는 경우 Geneve 터널링 프로토콜을 사용하는 OVN으로 마이그레이션하기 전에 네트워크 최대 전송 단위(MTU)를 8바이트로 줄여야 할 수 있습니다.



참고

마이그레이션 전 전용 유지 관리 기간에서 이 절차를 수행하는 것이 좋습니다.

VXLAN 패킷은 헤더 콘텐츠에 대해 50바이트의 데이터를 예약합니다. 여기에는 42바이트의 표준 외부 헤더와 8바이트 VXLAN 헤더가 포함됩니다. 물리적 네트워크에서 1500바이트의 표준 이더넷 MTU를 사용하는 경우 VXLAN 네트워크의 MTU를 1450으로 설정하고 트래픽은 조각화 없이 전달할 수 있습니다.

Geneve 패킷은 헤더 콘텐츠에 대해 58바이트의 데이터를 예약합니다. 여기에는 표준 외부 헤더의 42바이트와 16바이트 Geneve 헤더가 포함됩니다. 따라서 물리적 네트워크에 1508 미만의 MTU가 있는 경우 조각화할 필요가 없도록 사전 마이그레이션 OpenStack VXLAN 네트워크의 MTU를 8바이트까지 줄여야 합니다.



참고

물리적 네트워크가 조각화 없이 OpenStack VXLAN 네트워크 MTU보다 58바이트 이상을 전송할 수 있는 경우 이 절차를 건너뛰고 2.4절. ["OVS에서 OVN으로 ML2 메커니즘 드라이버 마이그레이션"](#) 로 이동합니다. 예를 들어 물리적 네트워크가 9000바이트 점보 프레임으로 구성되어 있고 openstack network MTU가 8942 이하인 경우 이 절차를 건너뛸 수 있습니다.

RHOSP OVN 마이그레이션 툴은 VXLAN 및 GRE 오버클라우드 네트워크에서 MTU를 8바이트로 자동으로 줄입니다. 다음 절차에서는 도구를 사용하여 다음을 수행합니다.

- DHCP T1 타이머를 30초로 줄여 DHCP 갱신 빈도를 늘립니다.
- 기존 VXLAN 네트워크의 MTU 크기를 8바이트로 줄입니다.

배포에서 모든 VM 인스턴스를 구성하는 데 DHCP를 사용하지 않는 경우 제외된 인스턴스에서 MTU를 수동으로 줄여야 합니다.

사전 요구 사항

- 다음 단계를 완료했습니다. 2.1절. ["OVN 메커니즘 드라이버로의 마이그레이션을 위한 환경 준비"](#)
- 2.2절. ["OVS에서 OVN으로 ML2 메커니즘 드라이버 마이그레이션을 위한 컨테이너 이미지 준비"](#)의 단계를 완료했습니다.
- 마이그레이션 전 배포는 VXLAN을 사용한 RHOSP(Red Hat OpenStack Platform) 17.1 이상입니다.

프로세스

1. `ovn_migration.sh 'reduce-dhcp-t1'` 을 실행합니다. 이렇게 하면 DHCP 에이전트가 실행 중인 모든 노드에서 `dhcp_renewal_time` 을 `/var/lib/config-data/puppet-generated/neutron/etc/neutron/dhcp_agent.ini`에서 구성하는 내부 neutron DHCP 서버의 T1 매개변수가 줄어듭니다.

```
$ ovn_migration.sh reduce-dhcp-t1 | sudo tee -a /var/log/ovn_migration_output.txt
```

2. T1 매개변수가 기존 VM에 전파되었는지 확인합니다. 이 프로세스는 최대 4시간이 걸릴 수 있습니다.
 - 컴퓨팅 노드 중 하나에 로그인합니다.

- 프로젝트 네트워크에 연결된 VM 중 하나 이상을 **tcpdump** 를 실행합니다. T1 전파가 성공하면 요청이 약 30초마다 발생하는 것으로 예상됩니다.

```
[heat-admin@overcloud-novacompute-0 ~]$ sudo tcpdump -i tap52e872c2-e6 port 67 or port 68 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tap52e872c2-e6, link-type EN10MB (Ethernet), capture size 262144 bytes
13:17:28.954675 IP 192.168.99.5.bootpc > 192.168.99.3.bootps: BOOTP/DHCP, Request from fa:16:3e:6b:41:3d, length 300
13:17:28.961321 IP 192.168.99.3.bootps > 192.168.99.5.bootpc: BOOTP/DHCP, Reply, length 355
13:17:56.241156 IP 192.168.99.5.bootpc > 192.168.99.3.bootps: BOOTP/DHCP, Request from fa:16:3e:6b:41:3d, length 300
13:17:56.249899 IP 192.168.99.3.bootps > 192.168.99.5.bootpc: BOOTP/DHCP, Reply, length 355
```



참고

이 확인은 cirros VM에서는 불가능합니다. cirros **udhcpd**의 구현은 DHCP 옵션 58(T1)에 응답하지 않습니다. 전체 Linux VM에 속하는 포트에서 이 확인을 시도합니다. Windows 및 Linux 배포판의 변형과 같이 워크로드에 표시되는 모든 다양한 운영 체제를 확인하는 것이 좋습니다.

- DHCP의 T1 매개변수를 반영하도록 VM 인스턴스가 업데이트되지 않은 경우 재부팅합니다.
- 마이그레이션 전 VXLAN 네트워크의 MTU를 줄입니다.

```
$ ovn_migration.sh reduce-mtu | sudo tee -a /var/log/ovn_migration_output.txt
```

이 단계에서는 네트워크별로 MTU 네트워크를 줄이고 완료된 네트워크에 `adapted_mtu`를 태그합니다. 이 톨은 VXLAN 네트워크에서만 작동합니다. 배포에 VLAN 프로젝트 네트워크만 있는 경우 이 단계에서는 값을 변경하지 않습니다.

- VXLAN 프로젝트 네트워크에 고정 IP 할당이 있는 인스턴스가 있는 경우 인스턴스 MTU를 8바이트로 수동으로 줄입니다. 예를 들어 VXLAN 기반 MTU가 1450이면 1442로 변경합니다.



참고

VXLAN 프로젝트 네트워크에 수동으로 고정 IP 할당 및 MTU 설정이 제공된 경우에만 이 단계를 수행합니다. 기본적으로 DHCP는 IP 할당 및 MTU 설정을 제공합니다.

- 2.4절. "OVS에서 OVN으로 ML2 메커니즘 드라이버 마이그레이션" 으로 이동합니다.

2.4. OVS에서 OVN으로 ML2 메커니즘 드라이버 마이그레이션

`ovn-migration` 스크립트는 OVS에서 OVN으로 ML2 메커니즘 드라이버의 인플레이스 마이그레이션과 관련된 환경 설정, 마이그레이션 및 정리 작업을 수행합니다.

사전 요구 사항

- OVS에서 OVN으로 ML2 메커니즘 드라이버를 마이그레이션하기 위한 환경 준비 단계를 완료했습니다.

- 원래 배포에서 VXLAN 또는 GRE를 사용하는 경우 **OVS 메커니즘 드라이버에서 OVN 메커니즘 드라이버로 마이그레이션하기 위해 MTU 조정** 단계를 완료했습니다.
- 또한 **OVS 메커니즘 드라이버에서 OVN 메커니즘 드라이버로 마이그레이션하기 위해 컨테이너 이미지를 준비하여 필요한 모든 마이그레이션** 단계를 완료했습니다.

프로세스

1. 새 네트워크, 서브넷, 라우터 또는 인스턴스 생성 또는 컴퓨팅 노드 간에 인스턴스 마이그레이션과 같은 Networking 서비스(neutron) API와 상호 작용하는 모든 작업을 중지합니다.
마이그레이션 중 Networking API와 상호 작용하면 정의되지 않은 동작이 발생할 수 있습니다. 마이그레이션을 완료한 후 API 작업을 다시 시작할 수 있습니다.
2. **ovn_migration.sh start-migration** 을 실행하여 마이그레이션 프로세스를 시작합니다. **tee** 명령은 문제 해결을 위해 스크립트 출력 사본을 생성합니다.

```
$ ovn_migration.sh start-migration | sudo tee -a /var/log/ovn_migration_output.txt
```

결과

이 스크립트는 다음 작업을 수행합니다.

- br-int 대신 임시 브리지 br-migration을 사용하여 OVN과 함께 OVN을 배포하도록 오버클라우드 스택을 업데이트합니다. 임시 브릿지는 마이그레이션 중에 다운타임을 제한하는 데 도움이 됩니다.
- neutron-ovn-db-sync-util을 실행하여 OVN northbound 데이터베이스를 생성합니다. 유틸리티는 Neutron 데이터베이스를 검사하여 OVN northbound 데이터베이스에 동일한 리소스를 생성합니다.
- br-migration 대신 ovn-controller를 br-int에 다시 할당합니다.
- 다음을 포함하여 ML2/OVN에서 사용하지 않는 노드 리소스를 제거합니다.
 - 네트워크 네임스페이스(fip, snat, qrouter, qdhcp)를 정리합니다.
 - **br-int** 에서 불필요한 패치 포트를 제거합니다.
 - **br-tun** 및 **br-migration** ovs 브리지를 제거합니다.
 - **qr-, ha-, qg-** 로 시작하는 **br-int** 에서 포트를 삭제합니다(neutron-netns-cleanup 사용).
- Networking 서비스(neutron) 에이전트 및 네트워킹 서비스 HA 내부 네트워크를 데이터베이스에서 Networking 서비스 API를 통해 삭제합니다.

3장. ML2/OVS에서 ML2/OVN으로 마이그레이션 되돌리기

ovn-migration.sh 백업을 사용하여 OVN 메커니즘 드라이버로 마이그레이션하기 위한 환경 준비의 선택적 백업 단계에 설명된 대로 컨트롤러 노드를 백업하고 ML2/OVS에서 ML2/OVN으로 마이그레이션을 시작하기 전에 오버클라우드 템플릿을 백업한 경우 이러한 기본 단계를 수행하여 마이그레이션을 되돌릴 수 있습니다.

1. 컨트롤러 노드를 복원합니다.
2. revert.yaml을 실행하여 ML2/OVN 아티팩트를 제거합니다.
3. 백업한 ML2/OVS 템플릿을 사용하여 오버클라우드를 재배포합니다.



참고

마이그레이션을 되돌리고 컴퓨팅 노드를 복원한 후 20분 이상 네트워크 다운 타임이 발생할 수 있습니다.



주의

ML2/OVS에서 ML2/OVN 마이그레이션은 완전히 되돌릴 수 없는 방식으로 환경을 변경합니다.

적절한 백업 단계를 따르고 지침을 되돌리는 경우 실패 또는 중단된 마이그레이션을 되돌릴 수 있지만 복원된 OVS 환경이 원본에서 변경될 수 있습니다. 예를 들어 OVN 메커니즘 드라이버로 마이그레이션한 다음 인스턴스를 다른 컴퓨팅 노드로 마이그레이션한 다음 OVN 마이그레이션을 되돌리면 인스턴스가 원래 컴퓨팅 노드에 있습니다. 또한 되돌리기 작업은 데이터 플레인에 대한 연결을 중단합니다.

프로덕션 환경에서 마이그레이션하기 전에 사전 지원 케이스를 제출하십시오. 그런 다음 Red Hat Technical Account Manager 또는 Red Hat Global Professional Services와 협력하여 운영 환경과 밀접하게 유사한 단계 환경에서 백업 및 마이그레이션 계획을 작성하고 마이그레이션을 테스트합니다.

잠재적인 마이그레이션 복원을 위해 백업을 준비하도록 선택하는 경우 스테이지 환경에서 마이그레이션 반환도 테스트해야 합니다.

마이그레이션 되돌리기는 역방향 마이그레이션이 아닙니다. 마이그레이션에 실패한 경우 마지막 수단으로만 사용하기 위한 것입니다. 마이그레이션이 검증을 통과하고 나중에 마이그레이션 후 환경에서 운영 문제를 식별하는 경우 이러한 문제를 마이그레이션 후 환경에서 버그로 해결합니다.

3.1. ML2/OVN 마이그레이션을 되돌리도록 컨트롤러 노드 복원

ovn-migration.sh 백업을 사용하여 마이그레이션 전에 컨트롤러 노드를 백업한 경우 Relax 및 Recover 툴(ReaR)을 사용하여 마이그레이션 후 환경의 경우 이를 복원할 수 있습니다.

사전 요구 사항

- 실패한 마이그레이션에 대한 지원 케이스를 제출했으며 Red Hat에서 지침을 받았습니다.

- **ovn-migration.sh** 백업을 사용하여 마이그레이션 전에 컨트롤러 노드를 백업했습니다.
- 백업 노드에 액세스할 수 있습니다.
- ML2/OVS 오버클라우드를 다시 배포하는 데 오버클라우드 템플릿을 백업해야 합니다.

프로세스

1. 새 네트워크, 서브넷 또는 라우터 생성 또는 컴퓨팅 노드 간에 가상 머신 인스턴스 마이그레이션과 같은 컨트롤 플레인 API와 상호 작용하는 모든 작업을 중지합니다.
2. 각 컨트롤 플레인 노드의 전원을 끕니다. 진행하기 전에 컨트롤 플레인 노드의 전원이 완전히 꺼졌는지 확인합니다.
3. 해당 백업 ISO 이미지로 각 컨트롤 플레인 노드를 부팅합니다.
4. **Relax-and-Recover** 부팅 메뉴가 표시되면 각 컨트롤 플레인 노드에서 **Recover** `<control_plane_node>`를 선택합니다. `<control_plane_node>`를 해당 컨트롤 플레인 노드의 이름으로 바꿉니다.



참고

시스템에서 UEFI를 사용하는 경우 **Relax-and-Recover (Secure Boot 없음)** 옵션을 선택합니다.

5. 각 컨트롤 플레인 노드에서 **root** 사용자로 로그인하여 노드를 복원합니다. 다음 메시지가 표시됩니다.

```
Welcome to Relax-and-Recover. Run "rear recover" to restore your system!
RESCUE <control_plane_node>:~ # rear recover
```

컨트롤 플레인 노드 복원 프로세스가 완료되면 콘솔에 다음 메시지가 표시됩니다.

```
Finished recovering your system
Exiting rear recover
Running exit tasks
```

6. 노드의 전원을 끕니다.

```
RESCUE <control_plane_node>:~ # poweroff
```

7. 부팅 순서를 일반 부팅 장치로 설정합니다. 부팅 시 노드가 이전 상태를 다시 시작합니다.
8. 서비스가 올바르게 실행되고 있는지 확인하려면 **pacemaker**의 상태를 확인합니다. 컨트롤러 노드에 **root** 사용자로 로그인하고 다음 명령을 입력합니다.

```
# pcs status
```

9. Ansible을 사용하여 **revert.yml** 파일을 실행합니다.

```
ansible-playbook -vv /usr/share/ansible/\
neutron-ovn-migration/playbooks/revert.yml \
-i hosts_for_migration
```

10. **ovn-router** 가 모든 환경 파일 또는 템플릿에서 **NeutronServicePlugins** 또는 **NeutronPluginExtensions** 의 값으로 표시되는 경우 **ovn-router** 값을 라우터로 바꿉니다. 매개 변수는 둘 이상의 파일에 표시될 수 있습니다. 예를 들어 하나의 파일은 **tripleo-heat templates/environments/services/neutron-ovn-dvr-ha.yaml** 입니다.

```
parameter_defaults:
  NeutronServicePlugins: "router,trunk,qos,placement"
```

11. OVS 템플릿을 사용하도록 오버클라우드를 업데이트합니다.

```
bash ~/overcloud-revert-ovs.sh
```

12. 원래 ML2/OVS 배포에 트렁크 포트를 사용하는 인스턴스가 포함된 경우 다음 명령을 실행하여 해당 인스턴스에 대한 연결을 복원합니다.

```
ansible-playbook -vv /usr/share/ansible/neutron-ovn-migration/playbooks/revert-rewire.yml \
-i hosts_for_migration
```

문제 해결

- 다음 명령을 실행하여 **pcs 상태**에 의해 표시되는 리소스 알람을 지웁니다.

```
# pcs resource clean
```

- 다음 명령을 실행하여 **pcs 상태**에 따라 표시되는 STONITH 펜싱 작업 오류를 지웁니다.

```
# pcs resource clean
# pcs stonith history cleanup
```