

OpenStack Neutron용 VMware NSX-T Data Center 플러그인 설치 가이드

2019년 9월 19일

VMware NSX-T Data Center 2.5



vmware®

다음 VMware 웹 사이트에서 최신 기술 문서를 확인할 수 있습니다.

<https://docs.vmware.com/kr/>

본 문서에 대한 의견이 있으시면 다음 주소로 피드백을 보내주십시오.

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

VMware 코리아
서울시 강남구
영동대로 517
아셈타워 13층
(우) 06164
전화: +82 2 3016 6500
팩스: +82 2 3016 6501
www.vmware.com/kr

Copyright © 2019 VMware, Inc. All rights reserved. [저작권 및 상표 정보](#)

목차

- 1** OpenStack Neutron용 VMware NSX-T Data Center 플러그인 설치 가이드 4
- 2** OpenStack용 NSX-T Data Center 플러그인 설치 준비 5
 - 사전 요구 사항 5
 - 시스템 요구 사항 5
 - Neutron 플러그인 비교 6
- 3** NSX-T Data Center 플러그인을 사용하여 Neutron 기본 서비스 설치 9
 - Ubuntu 시스템에 OpenStack용 NSX-T Data Center 플러그인 설치 9
 - Red Hat 시스템에 OpenStack용 NSX-T Data Center 플러그인 설치 10
- 4** NSX-T Data Center용 OpenStack 플러그인 구성 11
 - OpenStack Neutron 네트워크 노드 구성 11
 - neutron.conf 및 nsx.ini 파일 편집 12
 - 클라이언트 인증서 기반 인증 사용 13
 - DHCP 및 메타데이터 프록시 서비스 사용 13
 - OpenStack용 NSX-T Data Center 플러그인에 대한 샘플 구성 파일 15
 - OpenStack Nova 컨트롤러 구성 16
 - OpenStack Nova 계산 구성 파일 16
- 5** OpenStack용 NSX-T Data Center 플러그인을 사용하여 Neutron 고급 서비스 구성 17
 - OpenStack Octavia LBaaS(Load Balancer as a Service) 구성 18
 - OpenStack Neutron LBaaS(Load Balancer as a Service) 구성 18
 - OpenStack Neutron의 FWaaS(Firewall as a Service) 구성 19
 - OpenStack Neutron의 VPNaaS(IPSec VPN as a Service) 구성 20
 - Neutron 고급 서비스에 대한 샘플 구성 파일 21
- 6** 부록: OpenStack용 NSX-T Data Center 플러그인 구성 속성 24

OpenStack Neutron용 VMware NSX-T Data Center 플러그인 설치 가이드

1

이 가이드에서는 OpenStack Neutron용 NSX-T Data Center 플러그인을 설치 및 구성하는 방법을 설명합니다. 이 정보에는 단계별 구성 지침이 포함되어 있습니다.

VMware NSX-T Data Center 플러그인을 구성하고 실행하면 OpenStack Neutron이 NSX-T Data Center 배포의 가상 네트워킹 리소스를 인식하고 관리할 수 있습니다. 이 작업을 성공적으로 수행하려면 NSX-T Data Center 및 OpenStack 구성 요소와 기능에 익숙해야 합니다.

이 가이드에서는 NSX 정책 및 NSX-T Manager용 Openstack Neutron 플러그인에 대한 정보를 제공합니다. NSX-T 정책에 대해 Openstack 플러그인을 제공하는 첫 번째 NSX-T Data Center 릴리스입니다. 이 릴리스의 경우 NSX-T 정책 플러그인은 새 설치에만 사용할 수 있습니다.

이러한 항목에 대한 자세한 내용은 다음을 참조하십시오.

- “NSX-T Data Center 관리 가이드”
- OpenStack 설명서

OpenStack용 NSX-T Data Center 플러그인 설치 준비

2

본 장은 다음 항목을 포함합니다.

- 사전 요구 사항
- 시스템 요구 사항
- Neutron 플러그인 비교

사전 요구 사항

OpenStack Neutron 플러그인에 대해 VMware에서 제공하는 지원 패키지에는 NSX-T 특정 아티팩트만 포함되어 있습니다. 따라서 이 설치 프로세스를 시도하기 전에 선택한 OpenStack 서비스가 설치되어 있어야 합니다.

이 문서의 절차에 따라 NSX 배포와 통합할 OpenStack Neutron용 NSX-T Data Center 플러그인을 설치 및 구성합니다. 이 절차에서는 VMware NSX-T Data Center가 NSX-T 전송 노드에 설치되고 구성되었다고 가정합니다.

설치 프로세스의 일부로 적절한 종속성을 다운로드, 설치 및 구성할 수 있도록 하려면 Neutron 서비스 설치 중에 로컬 배포 저장소 미러에 인터넷으로 연결되거나 액세스할 수 있어야 합니다.

시스템 요구 사항

OpenStack용 NSX-T Data Center 플러그인 지원은 Neutron 플러그인으로 구현됩니다. Neutron을 구성할 때 사용되는 VMware NSX 플러그인 클래스는 사용 중인 NSX 버전에 따라 다릅니다.

vSphere KVM(Ubuntu, Red Hat Enterprise Linux, CentOS...)에 대해 지원되는 하이퍼바이저 버전은 "NSX-T Data Center 설치 가이드"에 나열되어 있습니다.

OpenStack용 NSX-T Data Center 플러그인은 호환되는 OpenStack 소프트웨어 버전과 관련하여 다음과 같은 특정 요구 사항을 갖습니다.

NSX-T 정책 플러그인에 대한 OpenStack 배포	버전
오픈 소스 버전	Stein

NSX-T Manager 플러그인에 대한 OpenStack 배포	버전
오픈 소스 버전	Rocky
오픈 소스 버전	Stein
Red Hat OpenStack Platform	Red Hat OpenStack 버전 13과 관련된 Red Hat Enterprise Linux 버전

Neutron 플러그인 비교

VMware NSX-T Data Center 2.5부터 Openstack Neutron을 NSX-T와 통합하는 데 다음 두 개의 플러그인을 사용할 수 있습니다.

- NSX-T 정책 플러그인은 의도 기반 API 추상화를 사용하여 NSX-T 정책 관리자와 상호 작용합니다. 이것은 새 플러그인으로, 새 설치에 권장되는 선택 사항입니다.
- NSX Manager 플러그인은 명령적 API를 사용하여 NSX Manager와 상호 작용합니다. 이것은 기존 NSX-T 플러그인으로, 기존 설치뿐만 아니라 NSX-T 정책 플러그인에서 아직 처리하지 못하는 사용 사례에도 사용할 수 있습니다.

표 2-1. 플러그인 기능 비교

네트워킹 및 보안 기능	NSX-T MP 플러그인	NSX-T 정책	설명
스위칭			
접치는 IP 서브넷 지원	예	예	각 프로젝트는 프로젝트 전용 네트워크를 동적으로 생성할 수 있습니다. 이러한 네트워크에는 서로 접치는 IP 서브넷이 있을 수 있습니다.
DHCP	예	예	인스턴스에 DHCP를 통한 자동 주소 지정이 있습니다.
정적 IPv6 주소 바인딩	아니요	예	
라우팅			
논리적 라우팅	예	예	논리적 네트워크와 외부 네트워크 간의 라우팅뿐 아니라 여러 개인 논리적 네트워크 간의 라우팅을 사용하도록 설정합니다.
IPv6 논리적 라우팅	아니요	예	논리적 네트워크와 외부 네트워크 간의 라우팅뿐 아니라 여러 개인 IPv6 논리적 네트워크 간의 라우팅을 사용하도록 설정합니다.

표 2-1. 플러그인 기능 비교 (계속)

네트워킹 및 보안 기능	NSX-T MP 플러그인	NSX-T 정책	설명
외부 네트워크	예	예	인스턴스에 대한 외부 액세스를 제공하는 네트워크입니다. 개인 네트워크는 라우터를 통해 외부 네트워크에 업링크되어 개인 네트워크의 인스턴스에 대한 외부 액세스를 제공합니다.
IPv6 외부 네트워크	예	예	IPv6을 사용하는 외부 네트워크
정적 경로	예	예	정적 경로를 삽입합니다.
IPv6 정적 경로	아니요	예	IPv6을 사용하는 외부 네트워크
인스턴스에 대한 부동 IP	예	예	인스턴스에 대한 외부 액세스를 사용하도록 설정하려면 공용 라우팅 가능 IP 주소를 인스턴스에 할당합니다.
No-NAT 라우터	예	예	No-NAT 라우팅 토폴로지
IPv6 no-NAT 라우터	아니요	예	No-NAT 토폴로지는 IPv6을 사용하는 OpenStack에서 지원하는 유일한 라우팅 토폴로지입니다. IPv6을 사용하는 NAT는 지원되지 않습니다.
Neutron 라우터 이중 스택 인터페이스	아니요	예	Neutron 라우터의 동일한 인터페이스에서 IPv4 및 IPv6 이중 스택 지원
IPv6 SLAAC	아니요	예	상태 비저장 주소 자동 구성 지원
보안			
방화벽 - 보안 그룹	예	예	OpenStack 보안 그룹(NSX 포함, 보안 그룹이 사용되고, 해당 SG를 사용하여 DFW 규칙이 생성됨. 마이크로 세분화가 허용됨)
IPv6 방화벽(보안 그룹)	아니요	예	IPv6을 사용하는 Neutron 보안 그룹
포트 보안	예	예	Neutron 포트 보안은 NSX SpoofGuard 기능을 사용하여 구현됩니다.

표 2-1. 플러그인 기능 비교 (계속)

네트워킹 및 보안 기능	NSX-T MP 플러그인	NSX-T 정책	설명
IPv6 포트 보안	아니요	예	Neutron 포트 보안은 NSX SpoofGuard 기능을 사용하여 구현됩니다. 이를 통해 allowed_address_pairs 및 포트에 대한 IPv6 서브넷 매핑이 허용됩니다.
방화벽 (L3 FWaaS)	예	예	
IPv6 방화벽 (L3 FWaaS)	아니요	예	
기타 서비스			
로드 밸런싱	예	예	
서비스 품질	예	예	
DNS	예	예	
VPNaas	예	아니요	

업그레이드

NSX-T Manager 플러그인이 있는 Openstack Neutron에서 NSX-T 정책 플러그인이 있는 Openstack Neutron으로의 마이그레이션 경로는 없습니다. 업그레이드 시 기존 설치된 NSX-T Manager 플러그인을 계속 실행하게 됩니다. NSX-T Manager에서 NSX-T 정책으로의 마이그레이션 경로는 향후 릴리스에서 사용할 수 있게 됩니다. NSX-T 정책 플러그인은 고유한 기능(IPv6)을 포함하기 때문에 새 설치에 권장되는 솔루션입니다. 또한 NSX-T 플러그인의 경우만 새 기능으로 업그레이드할 수 있습니다.

NSX-T Data Center 플러그인을 사용하여 Neutron 기본 서비스 설치

3

본 장은 다음 항목을 포함합니다.

- Ubuntu 시스템에 OpenStack용 NSX-T Data Center 플러그인 설치
- Red Hat 시스템에 OpenStack용 NSX-T Data Center 플러그인 설치

Ubuntu 시스템에 OpenStack용 NSX-T Data Center 플러그인 설치

NSX-T Data Center 플러그인은 Ubuntu 기반 Linux 배포를 위한 debian(.deb) 패키지로 배포됩니다.

사전 요구 사항

NSX-T Data Center 플러그인은 Ubuntu 기반 Linux 배포를 위한 debian(.deb) 패키지로 배포됩니다.

다음 지침은 NSX-T Manager 및 NSX-T 정책 플러그인 둘 다에 적용됩니다.

- 종속성 오류로 인해 debian 패키지 설치가 실패하는 경우 python-tooz 및 python-oslo.vmware 패키지를 설치해야 할 수 있습니다. Tooz는 분산 조정 기본 형식에 대한 추상화를 제공하는 Python 라이브러리입니다. 주요 목표는 분산 시스템에서 그룹과 이러한 그룹의 멤버 자격을 처리하는 것입니다. Oslo VMware 라이브러리는 일반 VMware 작업 및 API에 대한 지원을 제공합니다. 예: `sudo apt-get install python-oslo.vmware`.

절차

- 1 .deb 파일: NSX Neutron 플러그인 및 NSX Neutron 공용 라이브러리를 다운로드합니다.
- 2 파일을 Neutron 네트워크 노드에 복사합니다.
- 3 dpkg 명령을 사용하여 .deb 파일이 있는 동일한 디렉토리에 패키지를 설치합니다.

다음 예제의 버전 번호는 다운로드 시 선택한 릴리스에 따라 다를 수 있습니다.

- `sudo dpkg -i python-vmware-nsxlib_12.0.0.9797177-1_all.deb`
- `sudo dpkg -i openstack-vmware-nsx_12.0.0.9797177-1_all.deb`

- 4 FWaaS(Firewall as a Service) 패키지를 설치합니다. 설치 후에 이 기능을 사용하도록 설정할 필요가 없습니다.

다음 예제의 버전 번호는 다운로드 시 선택한 릴리스에 따라 다를 수 있습니다.

- `sudo apt-get install python-neutron-fwaas`

Red Hat 시스템에 OpenStack용 NSX-T Data Center 플러그인 설치

OpenStack용 NSX-T Data Center 플러그인은 Red Hat 기반 Linux 배포를 위해 .rpm 파일로 패키징됩니다.

.rpm 패키지는 NSX-T Data Center 다운로드 페이지의 **드라이버 및 도구**에서 찾을 수 있습니다. 이 지침은 TripleO를 활용하지 않는 설치에만 해당되며, 이외의 경우에는 Red Hat OpenStack 전용 가이드를 참조하십시오.

절차

- 1 .rpm 파일: NSX Neutron 플러그인 및 NSX Neutron 공용 라이브러리를 다운로드합니다.
- 2 이러한 파일을 플러그인을 설치하려는 Neutron 네트워크 노드에 복사합니다.
- 3 rpm 명령을 사용하여 .rpm 파일이 있는 동일한 디렉토리에 패키지를 설치합니다.

다음 예제의 버전 번호는 다운로드 시 선택한 릴리스에 따라 다를 수 있습니다.

- `sudo rpm -i python-vmware-nsxlib_12.0.0.9797177-1_all.rpm`
 - `sudo rpm -i vmware-nsx-12.0.0.9797177-1.noarch.rpm`
- 4 FWaaS(Firewall as a Service) 패키지를 설치합니다. 설치 후에 이 기능을 사용하도록 설정할 필요가 없습니다.
- `sudo yum install python-neutron-fwaas`

NSX-T Data Center용 OpenStack 플러그인 구성

4

- **OpenStack Neutron 네트워크 노드 구성**

이 섹션에 설명된 구성은 Neutron 네트워크 노드 구성에 대해 설명합니다.

- **OpenStack용 NSX-T Data Center 플러그인에 대한 샘플 구성 파일**

구성 파일은 일반적으로 `/etc/neutron/plugins/vmware/nsx.ini`에 있습니다.

- **OpenStack Nova 컨트롤러 구성**

이 섹션에 설명된 구성은 NSX-T Data Center에 대한 정보로 Nova 컨트롤러 노드 구성을 보완합니다.

- **OpenStack Nova 계산 구성 파일**

Nova 구성 파일을 편집하려면 NSX-T를 사용하십시오.

OpenStack Neutron 네트워크 노드 구성

이 섹션에 설명된 구성은 Neutron 네트워크 노드 구성에 대해 설명합니다.

이 설명서는 NSX-T 정책 플러그인을 설명합니다. 그러나 이 섹션에서는 NSX-T Manager 플러그인에 관련된 설정도 강조해서 설명합니다.

다음과 같은 두 가지 기본 구성 파일 경로가 사용됩니다.

- `/etc/neutron/neutron.conf` -- Neutron의 구성 파일

- `/etc/neutron/plugin/vmware/nsx.ini` -- VMware NSX Neutron 플러그인의 구성 파일

- **neutron.conf 및 nsx.ini 파일 편집**

Neutron 플러그인이 NSX-T 배포와 상호 작용할 수 있도록 하려면 이러한 파일을 NSX-T 환경과 관련된 정보로 편집해야 합니다.

- **클라이언트 인증서 기반 인증 사용**

NSX Manager에 대한 Neutron 클라이언트 인증서 기반 인증이 지원됩니다.

- **DHCP 및 메타데이터 프록시 서비스 사용**

NSX-T Data Center 플러그인을 사용하는 경우 OpenStack 참조 DHCP 구현이 NSX-T Data Center 기본 DHCP 서버로 교체됩니다. 또한 NSX-T Data Center 플랫폼은 Nova 메타데이터에 액세스하는 프록시 서버를 제공합니다.

neutron.conf 및 nsx.ini 파일 편집

Neutron 플러그인이 NSX-T 배포와 상호 작용할 수 있도록 하려면 이러한 파일을 NSX-T 환경과 관련된 정보로 편집해야 합니다.

절차

- 1 neutron.conf 파일을 편집하여 코어 Neutron 플러그인 **[DEFAULT] core_plugin = vmware_nsxp**를 설정합니다. NSX-T Manager 플러그인을 사용하도록 설정하려면 **[DEFAULT] core_plugin = vmware_nsxv3**를 사용합니다.

- 2 nsx.ini 구성 파일을 편집하여 NSX 배포에 맞게 플러그인을 구성합니다.

NSX-T OpenStack 플러그인 속성은 nsx.ini 구성 파일의 [nsx_p] 섹션 아래에 있습니다.

아래의 구성 속성은 NSX Manager 플러그인에도 적용되며 [nsx_v3] 구성 섹션에 지정됩니다.

정의해야 하는 최소 구성 속성 집합은 다음과 같습니다.

변수	설명
nsx_api_managers	이 매개 변수를 사용하면 쉼표로 구분된 Manager 끝점 목록을 사용할 수 있습니다.
nsx_api_user	관리자 NSX-T Manager 사용자 이름(일반적으로 admin)입니다.
nsx_api_password	관리자 NSX-T Manager 암호입니다.
insecure	NSX Manager 서버 인증서의 확인을 적용하려면 False로 설정합니다. 기본값은 True입니다.
ca_file	NSX Manager 서버 인증서를 확인하는 데 사용할 CA 번들 파일입니다. 이 옵션은 "비보안"이 True로 설정된 경우 무시됩니다. "비보안"을 False로 설정하고 이 옵션을 설정 해제한 경우 시스템 루트 CA가 서버 인증서를 확인하는 데 사용됩니다.
nsx_api_managers	Neutron 네트워크를 생성하는 데 사용되는 기본 NSX 오버레이 전송 영역의 이름 또는 UUID입니다. Neutron을 시작하기 전에 NSX에서 생성해야 합니다.
default_tier0_router	향후 OpenStack 논리적 라우터(NSX-T Tier1)가 연결될 정책 관리자 NSX-T Tier0 이름 게이트웨이 라우터 또는 UUID여야 합니다("라우팅/라우터" 아래에 있음).
dhcp_profile	UUID 또는 이름을 입력합니다. NSX Manager에서 DHCP 프로파일 생성 의 내용을 참조하십시오.
metadata_proxy	UUID 또는 이름을 입력합니다. 메타데이터 프록시 생성 의 내용을 참조하십시오.

- 3 ps-aux |grep neutron 명령을 실행하여 Neutron을 다시 시작하고 nsx.ini 파일의 변경 내용을 선택합니다.

- 4 출력에 `nsx.ini` 및 `neutron.conf`가 있는지 확인합니다. Neutron은 명령줄에서 하나 이상의 구성 파일을 수락합니다. 이러한 파일은 구성 파일 구조가 특정 사용자의 기본 설정을 반영할 수 있도록 구성이 구문 분석될 때 병합됩니다.

```
ps -aux |grep neutron
stack      7688  0.0  1.8 311332 148904 ?        Ss   Nov26  21:10
/usr/bin/python /usr/local/bin/neutron-server --config-file
/etc/neutron/neutron.conf --config-file
/etc/neutron/plugins/vmware/nsx.ini
```

클라이언트 인증서 기반 인증 사용

NSX Manager에 대한 Neutron 클라이언트 인증서 기반 인증이 지원됩니다.

클라이언트 인증서 기반 인증을 사용하면 Neutron 플러그인에서 엔터프라이즈 관리자 역할이 있는 주체 ID로 로그인할 수 있습니다. 다른 주체 ID는 neutron 주체 ID로 생성된 리소스를 편집할 수 없으므로 neutron 라우터에 연결된 논리적 라우터의 삭제와 같은 실수로 인한 오류를 방지할 수 있습니다. 자세한 내용은 NSX-T Data Center 관리 가이드의 "주체 ID 보기"를 참조하십시오.

절차

- 클라이언트 인증서 인증을 사용하도록 설정하려면 `nsx.ini` 파일에서 다음을 정의합니다.
 - `nsx_use_client_auth = True`
 - `nsx_client_cert_storage = nsx-db`
 - `nsx_client_cert_file = <file to store certificate and private key>`
- `service neutron-server restart` 명령을 실행하여 Neutron을 다시 시작하고 `nsx.ini` 파일의 변경 내용을 선택합니다.

다음 명령을 실행하여 Neutron 서버가 `neutron.conf` 및 `nsx.ini` 파일을 둘 다 사용하고 있는지 확인합니다.

- `ps -aux |grep neutron`

출력에 `nsx.ini` 및 `neutron.conf`가 있는지 확인합니다. 예:

```
ps -aux |grep neutron
stack      7688  0.0  1.8 311332 148904 ?        Ss   Nov26  21:10
/usr/bin/python /usr/local/bin/neutron-server --config-file
/etc/neutron/neutron.conf --config-file
/etc/neutron/plugins/vmware/nsx.ini
```

DHCP 및 메타데이터 프록시 서비스 사용

NSX-T Data Center 플러그인을 사용하는 경우 OpenStack 참조 DHCP 구현이 NSX-T Data Center 기본 DHCP 서버로 교체됩니다. 또한 NSX-T Data Center 플랫폼은 Nova 메타데이터에 액세스하는 프록시 서버를 제공합니다.

이전 단계에서 구성된 NSX-T 플러그인과 관계없이 이러한 작업을 수행해야 합니다.

■ NSX Manager에서 DHCP 프로파일 생성

DHCP 서버 프로파일은 NSX Edge 클러스터 또는 NSX Edge 클러스터의 멤버를 지정합니다. 이 프로파일이 있는 DHCP 서버는 프로파일에 지정된 NSX Edge 노드에 연결되어 있는 논리 스위치에서 VM의 DHCP 요청을 처리합니다.

■ 메타데이터 프록시 생성

메타데이터 프록시 서버를 사용하면 VM 인스턴스는 OpenStack Nova API 서버에서 인스턴스별 메타데이터를 검색할 수 있습니다.

■ nsx.ini 파일 편집

이러한 변수 `native_dhcp_metadata`, `metadata_proxy` 및 `dhcp_profile`은 해당 프로파일을 사용하는 `nsx.ini`에 지정해야 합니다.

NSX Manager에서 DHCP 프로파일 생성

DHCP 서버 프로파일은 NSX Edge 클러스터 또는 NSX Edge 클러스터의 멤버를 지정합니다. 이 프로파일이 있는 DHCP 서버는 프로파일에 지정된 NSX Edge 노드에 연결되어 있는 논리 스위치에서 VM의 DHCP 요청을 처리합니다.

사전 요구 사항

NSX-T Data Center의 기본 DHCP 서버를 사용하도록 설정하려면 NSX-T Data Center에서 DHCP 프로파일을 생성하고 `nsx.ini`의 Neutron 플러그인 구성에 전달해야 합니다. `neutron dhcp` 서비스(`devstack`의 `q-dhcp`) 및 메타데이터 에이전트(`devstack`의 `q-meta`)가 실행되고 있지 않은지 확인합니다. `neutron.conf` 파일에서 `False`로 설정합니다.

절차

- 1 브라우저에서 관리자 권한으로 NSX Manager(<https://nsx-manager-ip-address>)에 로그인합니다.
- 2 탐색 패널에서 **고급 네트워킹 및 보안 > DHCP**를 선택합니다.
- 3 **서버 프로파일**을 선택하고 **추가**를 클릭합니다.
- 4 이름과 설명(선택 사항)을 입력합니다.
- 5 드롭다운 메뉴에서 **Edge 클러스터**를 선택합니다.
- 6 **추가**를 클릭합니다.

메타데이터 프록시 생성

메타데이터 프록시 서버를 사용하면 VM 인스턴스는 OpenStack Nova API 서버에서 인스턴스별 메타데이터를 검색할 수 있습니다.

NSX 플랫폼은 Nova 메타데이터에 액세스할 수 있는 프록시 서버를 제공합니다. 이 프록시는 169.254.269.254 주소의 모든 요청을 캡처하여 NSX-T 메타데이터 프록시 구성에 지정된 nova 메타데이터 서버 끝점으로 전달합니다.

사전 요구 사항

메타데이터 프록시에 사용되는 Edge 노드는 관리 IP 주소에서 메타데이터 서버에 대한 IP 연결이 있어야 합니다.

절차

- 1 브라우저에서 관리자 권한으로 NSX Manager(<https://nsx-manager-ip-address>)에 로그인합니다.
- 2 탐색 패널에서 **고급 네트워킹 > DHCP**를 선택합니다.
- 3 **메타데이터 프록시**를 선택하고 **추가**를 클릭합니다.
- 4 **이름**과 **설명**(선택 사항)을 입력합니다.
- 5 **Nova 서버 URL**을 `http://<openstack_controller>:8775`로 입력합니다. 메타데이터 프록시 서버가 기본 8775 이외의 포트에서 수신 대기하는 경우 올바른 포트에 URL을 업데이트합니다. 포트는 Nova API 구성 파일 `/etc/nova.conf`의 컨트롤러 노드에서 `metadata_listen_port` 매개 변수를 조회하여 찾을 수 있습니다. 구성을 변경해야 하는 경우 `n-api` 또는 `nova` 서버를 다시 시작합니다.
- 6 **Secret** 매개 변수를 입력합니다.
- 7 드롭다운 메뉴에서 **Edge 클러스터**를 선택합니다.
- 8 **추가**를 클릭합니다.

nsx.ini 파일 편집

이러한 변수 `native_dhcp_metadata`, `metadata_proxy` 및 `dhcp_profile`은 해당 프로파일을 사용하는 `nsx.ini`에 지정해야 합니다.

- `dhcp_profile = <UUID or name - DHCP>`
- `native_dhcp_metadata = True`
- `metadata_proxy = <UUID or name - MetaData Proxy>`
- `native_metadata_route = 169.254.169.254/31`

OpenStack용 NSX-T Data Center 플러그인에 대한 샘플 구성 파일

구성 파일은 일반적으로 `/etc/neutron/plugins/vmware/nsx.ini`에 있습니다.

다음은 샘플 구성 파일입니다.

```
[nsx_p]
# NSX-T credentials
nsx_api_managers = 192.168.10.5
nsx_api_user = admin
nsx_api_password = VMware1!
insecure = True
# NSX-T objects information
default_tier0_router = 0fd8b97f-315d-4461-a80b-adb489b6cfbc
```

```
default_overlay_tz_ = 4d3fcd4f-0946-4b08-ab6b-5463c571463d
default_vlan_tz_ = f74b5dab-dad3-47d2-b46e-57a1eeb5fde3
# DHCP and Metadata Proxy offered by NSX-T
dhcp_profile = 153637ce-657a-4ff9-a2f2-ffab62441abc
metadata_proxy = 32cf4708-7b1f-4932-b4ca-9f7029c9a7a2
```

```
[nsx_v3]
# NSX-T credentials
nsx_api_managers = 192.168.10.5
nsx_api_user = admin
nsx_api_password = VMware1!
insecure = True
# NSX-T objects information
default_tier0_router_uuid = 0fd8b97f-315d-4461-a80b-adb489b6cfbc
default_overlay_tz_uuid = 4d3fcd4f-0946-4b08-ab6b-5463c571463d
# DHCP and Metadata Proxy offered by NSX-T
dhcp_profile = 153637ce-657a-4ff9-a2f2-ffab62441abc
metadata_proxy = 32cf4708-7b1f-4932-b4ca-9f7029c9a7a2
```

OpenStack Nova 컨트롤러 구성

이 섹션에 설명된 구성은 NSX-T Data Center에 대한 정보로 Nova 컨트롤러 노드 구성을 보완합니다.

다음은 Nova 컨트롤러 샘플 구성 파일로, 일반적으로 제어 노드의 `/etc/nova/nova.conf`에 있습니다.

```
[DEFAULT]
firewall_driver = nova.virt.firewall.NoopFirewallDriver
use_neutron = True

[neutron]
metadata_proxy_shared_secret = VMware1!
service_metadata_proxy = True
```

OpenStack Nova 계산 구성 파일

Nova 구성 파일을 편집하려면 NSX-T를 사용하십시오.

이 섹션에 설명된 구성은 NSX-T Data Center에 대한 정보로 Nova 계산 노드 구성을 보완합니다.

다음은 Nova 계산 샘플 구성 파일로, 일반적으로 계산 노드의 `/etc/nova/nova.conf`에 있습니다.

```
[DEFAULT]
firewall_driver = nova.virt.firewall.NoopFirewallDriver
use_neutron = True

[neutron]
#for KVM
ovs_bridge = nsx-managed
```


OpenStack용 NSX-T Data Center 플러그인을 사용하여 Neutron 고급 서비스 구성

5

OpenStack Neutron의 LBaaS(Load Balancer as a Service), FWaaS(Firewall as a service) 및 VPNaaS(IPSec VPN as a Service)를 Neutron 고급 서비스라고도 합니다.

NSX-T Data Center 2.5부터 NSX-T 플러그인(정책 및 Manager 모두)은 더 이상 사용되지 않는 Neutron LBaaS 서비스를 대신하는 Octavia 로드 밸런싱 서비스를 지원합니다.

다음 정보에서는 사용자에게 NSX-T Data Center 2.4 및 OpenStack Stein 14.0이 있다고 가정합니다.

지원되는 현재 버전은 [시스템 요구 사항](#)에 나열됩니다.

Neutron 고급 서비스 NSX-T Data Center를 사용하도록 설정하려면 Neutron 구성에서 다음을 지정해야 합니다.

- 특정 서비스에 대해 서비스 플러그인 사용
- 서비스에 대한 서비스 제공자 구성
- NSX-T Data Center 드라이버(해당하는 경우) 및 NSX-T Data Center에 관련된 구성 항목을 제공합니다.

이러한 옵션은 neutron 구성 파일에 지정되며, 일반적으로 /etc/neutron에 있습니다(참고: 플러그인 관련 구성 파일은 일반적으로 /etc/neutron/plugins/vmware에 있음). Neutron은 명령줄에서 하나 이상의 구성 파일을 수락합니다. 이러한 파일은 구성 파일 구조가 특정 사용자의 기본 설정을 반영할 수 있도록 구성이 구문 분석될 때 병합됩니다. 일반적으로 다음과 같은 구조가 사용됩니다.

- neutron.conf - 코어 neutron 옵션, 기본 구성 매개 변수(예: API 관리자 끝점, 전송 영역 식별자), 서비스 플러그인 목록
- neutron_lbass.conf - 로드 밸런싱 서비스 제공자 및 옵션
- Octavia.conf - 로드 밸런싱 제공자 및 MQ 항목 Octavia 로드 밸런서 서비스에만 해당합니다. 이 파일은 일반적으로 /etc/octavia/octavia.conf에 있으며 Octavia 서비스에서 로드됩니다.
- neutron_fwass.conf - 방화벽 서비스 제공자, 드라이버 및 드라이버 옵션
- neutron_vpnaas.conf - VPN 서비스 제공자 및 기타 옵션

서비스 제공자는 service_provider 옵션을 사용하여 지정됩니다. 이 옵션은 다른 종류의 서비스에 대한 neutron 구성에서 여러 번 반복될 수 있지만 지정된 서비스 유형에 대해 기본 서비스 제공자가 둘 이상일 수는 없습니다.

Service_provider 구성 특성 구조: <SERVICE_TYPE>:<PROVIDER_CLASS>:[<DEFAULT>]

- **OpenStack Octavia LBaaS(Load Balancer as a Service) 구성**

이러한 지침은 Octavia를 통해 NSX-T Data Center 로드 밸런서의 OpenStack 사용을 구성하기 위한 것입니다.

- **OpenStack Neutron LBaaS(Load Balancer as a Service) 구성**

- **OpenStack Neutron의 FWaaS(Firewall as a Service) 구성**

이러한 지침은 FWaaS v2 구성과 관련된 내용입니다.

- **OpenStack Neutron의 VPNaaS(IPSec VPN as a Service) 구성**

이 서비스를 사용하여 OpenStack 사용자는 보안 VPN 터널을 통해 원격 사이트에서 Neutron 네트워크에 액세스할 수 있도록 할 수 있습니다.

- **Neutron 고급 서비스에 대한 샘플 구성 파일**

OpenStack Octavia LBaaS(Load Balancer as a Service) 구성

이러한 지침은 Octavia를 통해 NSX-T Data Center 로드 밸런서의 OpenStack 사용을 구성하기 위한 것입니다.

동일한 명령이 NSX-T 정책 및 NSX-T Manager 플러그인 둘 다에 적용됩니다.

Octavia 로드 밸런서 서비스에 대한 일반 정보는 공식 설명서를 참조하십시오.

절차

- 1 /etc/neutron/neutron.conf에서 NSX-T Data Center용 로드 밸런싱 서비스 플러그인이 구성되지 않았는지 확인합니다. 즉, vmware_nsx_lbaasv2가 service_plugins 목록에 없어야 합니다. neutron.conf 파일이 변경되었으면 neutron 서비스를 다시 시작합니다.
- 2 /etc/octavia/octavia.conf에서 다음을 지정합니다.
 - a [api_settings] 구성 설정에서 다음과 같이 NSX-T 제공자를 지정합니다.
default_provider_driver = vmwareedge enabled_provider_drivers = vmwareedge:NSX
 - b [oslo_messaging] 섹션에서 Octavia와 neutron 프로세스 공간에서 실행되는 NSX-T 드라이버 간의 통신에 대한 항목을 지정합니다.
- 3 Octavia API 서비스를 다시 시작합니다.

OpenStack Neutron LBaaS(Load Balancer as a Service) 구성

이러한 지침은 LBaaSv2를 통해 NSX-T Data Center 로드 밸런서의 OpenStack 사용을 구성하기 위한 것입니다. 동일한 명령이 NSX-T 정책 및 NSX-T Manager 플러그인 둘 다에 적용됩니다.

절차

- 1 /etc/neutron/neutron.conf를 편집하여 기본 구성 섹션 `service_plugins = vmware_nsx_lbaasv2, [...]`에 NSX-T Data Center용 로드 밸런싱 서비스 플러그인을 추가합니다.

`service_plugins`는 목록 옵션입니다. 전체 클래스 이름 또는 바로 가기를 쉼표로 구분하여 여러 서비스 플러그인을 지정할 수 있습니다.

- 2 다음을 사용하여 /etc/neutron/neutron-lbaas.conf 파일을 편집합니다.

- a `service_providers` 구성 섹션 `service_plugins =`
`LOADBALANCERV2:VMwareEdge:neutron_lbaas.drivers.vmware.edge_driver_v2.EdgeLoadBalancerDriverV2:default`에서 `service_provider` 옵션을 설정하여 NSX-T Data Center용 로드 밸런서 서비스 드라이버를 설정합니다.
- b 아직 구성하지 않은 경우 Keystone 인증 매개 변수를 구성합니다. 이러한 항목은 `neutron-lbaas` 서비스에서 사용되며 NSX-T Data Center 통합과는 관련이 없습니다. Keystone 끝점의 위치를 제공해야 합니다.

```
[service_auth]
auth_version = 3
admin_password = password
admin_user = admin
admin_tenant_name = admin
auth_url = http://<keystone_endpoint>/identity/v3
```

- 3 /etc/neutron/neutron-lbaas.conf 파일이 neutron 서버 명령줄에 추가되었는지 확인합니다. 이것은 `ps -aux | grep neutron` 명령을 실행하고 출력에 /etc/neutron/neutron-lbaas.conf가 있는지 확인하면 알 수 있습니다.

이 파일이 포함되지 않은 경우 neutron 서비스 시작 관리자를 편집해야 합니다. 서비스 시작 관리자의 위치와 구조는 사용되는 특정 OpenStack 배포에 따라 다릅니다.

- 4 neutron 서비스를 다시 시작합니다. 정확한 서비스 이름은 사용되는 OpenStack 배포에 따라 다릅니다.

OpenStack Neutron의 FWaaS(Firewall as a Service) 구성

이러한 지침은 FWaaS v2 구성과 관련된 내용입니다.

절차

- 1 /etc/neutron/neutron.conf를 편집하여 기본 구성 섹션 `service_plugins = firewall_v2`에 NSX-T Data Center용 Firewall as a Service 플러그인을 추가합니다.

`Service_plugins`는 목록 옵션입니다. 전체 클래스 이름 또는 바로 가기를 쉼표로 구분하여 여러 서비스 플러그인을 지정할 수 있습니다.

2 다음을 사용하여 /etc/neutron/neutron-fwaas.conf 파일을 편집합니다.

- a Service_providers 구성 섹션에서 service_provider service_provider =FIREWALL_V2:fwaas_db:neutron_fwaas.services.firewall.service_drivers.agents.agents.FirewallAgentDriver:default 옵션을 설정하여 NSX-T Data Center에 대해 Firewall as a Service 드라이버를 설정합니다.

```
[service_auth]
auth_version = 3
admin_password = password
admin_user = admin
admin_tenant_name = admin
auth_url = http://<keystone_endpoint>/identity/v3
```

이 옵션의 값은 특정 구조 <service_type>:<service_name>:<driver_class>:[<default>]를 갖습니다. service_provider는 "다중 문자열" 옵션입니다. 지정될 때마다 옵션 값이 목록에 추가됩니다. 각각에 대해 service_provider 옵션을 설정하여 여러 서비스 제공자를 지정할 수 있습니다.

- b FWaaS 구성 섹션에서 enabled = True를 설정하여 Firewall as a Service를 켭니다.
- c FWaaS 구성 섹션에서 driver = vmware_nsxp_edge_v2를 설정하여 NSX-T Data Center에 대한 FWaaS 디바이스 드라이버를 설정합니다.
- d FWaaS 구성 섹션에서 driver = vmware_nsxv3_edge_v2를 설정하여 NSX Manager 플러그인에 대한 FWaaS 디바이스 드라이버를 설정합니다.

3 /etc/neutron/neutron-fwaas.conf 파일이 neutron 서버 명령줄에 추가되었는지 확인합니다. 이것은 ps -aux | grep neutron을 실행하고 출력에 /etc/neutron/neutron-fwaas.conf가 있는지 확인하면 알 수 있습니다.

이 파일이 포함되지 않은 경우 neutron 서비스 시작 관리자를 편집해야 합니다. 서비스 시작 관리자의 위치와 구조는 사용되는 특정 OpenStack 배포에 따라 다릅니다.

4 neutron 서비스를 다시 시작합니다. 구체적인 서비스 이름은 사용되는 OpenStack 배포에 따라 다릅니다.

OpenStack Neutron의 VPNaaS(IPSec VPN as a Service) 구성

이 서비스를 사용하여 OpenStack 사용자는 보안 VPN 터널을 통해 원격 사이트에서 Neutron 네트워크에 액세스할 수 있도록 할 수 있습니다.

VPNaaS 드라이버는 NSX-T 정책 플러그인에 사용할 수 없습니다. 다음 지침은 NSX-T Manager 플러그인에만 적용됩니다.

절차

- 1 /etc/neutron/neutron.conf를 편집하여 기본 구성 섹션 `service_plugins = vmware_nsx_vpnaas, [...]`에 NSX-T Data Center용 IPsec VPN as a Service 플러그인을 추가합니다.

`service_plugins`는 목록 옵션입니다. 전체 클래스 이름 또는 바로 가기를 쉼표로 구분하여 여러 서비스 플러그인을 지정할 수 있습니다.
- 2 /etc/neutron/neutron-vpnaas.conf 파일을 다음으로 편집합니다. `service_providers` 구성 섹션에서 `service_provider` 옵션을 설정하여 NSX-T Data Center에 대한 VPNservice 로드 드라이버를 설정합니다. `service_provider = VPN:vmware:vmware_nsx.services.vpnaas.nsxv3.ipsec_driver.NSXv3IPsecVpnDriver:default`

이 옵션의 값은 특정 구조 `<service_type>:<service_name>:<driver_class>:[<default>]`를 갖습니다. `service_provider`는 "다중 문자열" 옵션입니다. 지정될 때마다 옵션 값이 목록에 추가됩니다. 각각에 대해 `service_provider` 옵션을 설정하여 여러 서비스 제공자를 지정할 수 있습니다.
- 3 /etc/neutron/neutron-vpnaas.conf 파일이 neutron 서버 명령줄에 추가되었는지 확인합니다. 이것은 `ps -aux | grep neutron`을 실행하고 출력에 /etc/neutron/neutron-vpnaas.conf가 있는지 확인하여 확인할 수 있습니다.

이 파일이 포함되지 않은 경우 neutron 서비스 시작 관리자를 편집해야 합니다. 서비스 시작 관리자의 위치와 구조는 사용되는 특정 OpenStack 배포에 따라 다릅니다.
- 4 neutron 서비스를 다시 시작합니다. 구체적인 서비스 이름은 사용되는 OpenStack 배포에 따라 다릅니다.

Neutron 고급 서비스에 대한 샘플 구성 파일

```
[DEFAULT]
ovs_integration_bridge = nsxvswitch
dhcp_agent_notification = False
notify_nova_on_port_data_changes = True
notify_nova_on_port_status_changes = True
core_plugin = vmware_nsxv3
service_plugins =
vmware_nsx_lbaasv2,vmware_nsx_vpnaas,neutron_fwaas.services.firewall.fwaas_plugin_v2.FirewallPluginV2
[...]
neutron_vpnaas.conf
[DEFAULT]
[service_providers]
service_provider = VPN:vmware:vmware_nsx.services.vpnaas.nsxv3.ipsec_driver.NSXv3IPsecVpnDriver:default
neutron_fwaas.conf
[DEFAULT]
[quotas]
# Number of firewalls allowed per tenant. A negative value means unlimited.
# (integer value)
#quota_firewall = 10
# Number of firewall policies allowed per tenant. A negative value means
# unlimited. (integer value)
#quota_firewall_policy = 10
# Number of firewall rules allowed per tenant. A negative value means
```

```
# unlimited. (integer value)
#quota_firewall_rule = 100

[service_providers]
service_provider =
FIREWALL_V2:fwaas_db:neutron_fwaas.services.firewall.service_drivers.agents.agents.FirewallAgentDriver:default
[fwaas]
enabled = True
driver = vmware_nsxv3_edge_v2
neutron_lbaas.conf
[DEFAULT]
[quotas]
# Number of LoadBalancers allowed per tenant. A negative value
# means unlimited. (integer value)
#quota_loadbalancer = 10

# Number of Loadbalancer Listeners allowed per tenant. A negative
# value means unlimited. (integer value)
#quota_listener = -1
# Number of pools allowed per tenant. A negative value means
# unlimited. (integer value)
#quota_pool = 10
# Number of pool members allowed per tenant. A negative value means
# unlimited. (integer value)
#quota_member = -1
# Number of health monitors allowed per tenant. A negative value
# means unlimited. (integer value)
#quota_healthmonitor = -1
[service_auth]
auth_version = 3
admin_password = password
admin_user = admin
admin_tenant_name = admin
auth_url = http://<keystone_ip>/identity/v3
[service_providers]
service_provider =
LOADBALANCERV2:VMWareEdge:neutron_lbaas.drivers.vmware.edge_driver_v2.EdgeLoadBalancerDriverV2:default
```

```
Octavia.conf
[DEFAULT]
verbose = True
transport_url = rabbit://<amqp_user>:<amqp_password>@<amqp_node>:5672/
debug = True

[api_settings]
default_provider_driver = vmwareedge
enabled_provider_drivers = vmwareedge:NSX
bind_port = 9875
api_handler = queue_producer
bind_host = 0.0.0.0

[database]
connection = mysql+pymysql://root:<db_password>@<db_node>:3306/octavia
```

```

[keystone_authtoken]
signing_dir =
memcached_servers = <memcached_node>:11211
cafile = <cabundle_path>
project_domain_name = Default
project_name = service
user_domain_name = Default
password = <password>
username = octavia
auth_url = http://<keystone_node>/identity
auth_type = password

[certificates]
server_certs_key_passphrase = insecure-key-do-not-use-this-key
ca_private_key_passphrase = foobar
ca_private_key = /etc/octavia/certs/private/cakey.pem
ca_certificate = /etc/octavia/certs/ca_01.pem

[controller_worker]
amp_ssh_key_name = octavia_ssh_key
amp_image_tag = amphora
network_driver = allowed_address_pairs_driver
compute_driver = compute_nova_driver
amphora_driver = amphora_haproxy_rest_driver
workers = 2
amp_active_retries = 100
amp_active_wait_sec = 2

[oslo_messaging]
topic = vmwarensxv_edge_lb
rpc_thread_pool_size = 2

[house_keeping]
load_balancer_expiry_age = 3600

[service_auth]
memcached_servers = <memcached_node>:11211
cafile = <cabundle_path>
project_domain_name = Default
project_name = admin
user_domain_name = Default
password = openstack
username = admin

```

부록: OpenStack용 NSX-T Data Center 플러그인 구성 속성

6

표 6-1. 구성 속성

섹션	변수	설명
nsx_p	nsx_api_managers	쉽포로 구분된 하나 이상의 NSX Manager IP 주소입니다. IP 주소는 [<scheme>://]<ip_adress>[:<port>] 형식이어야 합니다. 체계를 제공하지 않으면 https가 사용됩니다. 포트를 제공하지 않으면 http에는 포트 80이, https에는 포트 443이 사용됩니다.
	nsx_use_client_auth	부울. 클라이언트 인증서 인증을 사용하려면 True로 설정합니다.
	nsx_client_cert_file	클라이언트 인증서 및 개인 키가 포함된 파일의 경로로 PEM 형식으로 표시합니다.
	nsx_client_cert_pk_password	(옵션) 개인 키를 해독하기 위한 암호입니다.
	nsx_api_user	NSX Manager API에 액세스하는 데 사용되는 사용자 이름입니다.
	nsx_api_password	NSX Manager API에 액세스하는 데 사용되는 암호입니다.
	dns_domain	호스트 이름을 작성하는 데 사용할 도메인입니다.
	default_overlay_tz	default_edge_cluster
	default_vlan_tz	필수 VLAN 또는 플랫 제공자 네트워크를 생성할 때만 필요합니다. 물리적 네트워크를 지정하지 않은 경우 Neutron 네트워크 간 브리징에 사용되는 기본 NSX VLAN 전송 영역의 UUID 또는 이름입니다.
	edge_cluster	(옵션) 연결된 Edge 클러스터 외에, Tier1 라우터에서 연결할 Edge 클러스터를 지정합니다.
	재시도	(옵션) 부실 개정 오류 시 API 요청을 재시도하는 최대 횟수입니다.

표 6-1. 구성 속성 (계속)

섹션	변수	설명
	ca_file	(옵션) NSX Manager 서버 인증서를 확인하는 데 사용할 CA 번들 파일을 지정합니다. 이 옵션은 "비보안"이 True로 설정된 경우 무시됩니다. "비보안"을 False로 설정하고 ca_file을 설정 해제한 경우 시스템 루트 CA가 서버 인증서를 확인하는 데 사용됩니다.
	비보안	(옵션) true이면 NSX Manager 서버 인증서가 확인되지 않습니다. false이면 "ca_file"을 통해 지정된 CA 번들이 사용되고, 설정 해제되면 기본 시스템 루트 CA가 사용됩니다.
	http_timeout	(옵션) NSX Manager에 대한 HTTP 연결을 중단하기까지의 시간(초)입니다.
	http_read_timeout	(옵션) NSX Manager에 대한 HTTP 읽기 응답을 중단하기까지의 시간(초)입니다.
	http_retries	(옵션) HTTP 연결을 다시 시도할 수 있는 최대 횟수입니다.
	concurrent_connections	(옵션) 각 NSX Manager에 대한 최대 연결 수입니다.
	conn_idle_timeout	(옵션) Manager 연결을 사용하지 않은 경우 NSX Manager에 대한 연결을 확인하기 전에 대기할 시간(초)입니다.
	default_tier0_router	(옵션) Tier 1 논리적 라우터에 연결하고 외부 네트워크를 구성하는 데 사용되는 기본 Tier 0 라우터의 UUID 또는 이름입니다.
	metadata_on_demand	(옵션) True인 경우 라우터가 DHCP를 사용하지 않는 서브넷에 연결된 경우에만 라우터에 대한 내부 메타데이터 네트워크가 생성됩니다.
	dhcp_profile	(옵션) 네이티브 DHCP 서비스를 사용하도록 설정하는 데 사용되는 NSX DHCP 프로파일의 UUID입니다. NSX 플러그인으로 Neutron를 시작하기 전에 NSX에서 생성해야 합니다.
	locking_coordinator_url	(옵션) 잠금 관리자에 대한 분산 잠금 조정 리소스의 URL입니다. 이 값은 tooz 조정자에 매개 변수로 전달됩니다. 기본적으로 이 값은 None이고 oslo_concurrency는 단일 노드 잠금 관리에 사용됩니다.

표 6-1. 구성 속성 (계속)

섹션	변수	설명
	realization_max_attempts	(옵션) 리소스가 실현되기를 기다리는 동안 허용되는 최대 재시도 횟수입니다. 기본값: 50
	realization_wait_sec	(옵션) 리소스의 인식 시도 간격(초)입니다. 기본값: 1초