



# Red Hat Virtualization 4.0

## 기술 자료

Red Hat Virtualization 환경의 기술 아키텍처



## Red Hat Virtualization 4.0 기술 자료

---

Red Hat Virtualization 환경의 기술 아키텍처

Enter your first name here. Enter your surname here.

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

Enter your email address here.

## 법적 공지

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

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

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

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

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

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

Linux<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 참조는 Red Hat Virtualization 환경에서 사용되는 개념, 구성 요소 및 기술을 설명합니다.

## 차례

<b>1장. 소개</b> .....	<b>4</b>
1.1. RED HAT VIRTUALIZATION MANAGER	4
1.2. RED HAT VIRTUALIZATION HOST	4
1.3. 관리자 액세스 인터페이스	6
1.4. 관리자를 지원하는 구성 요소	7
1.5. 스토리지	8
1.6. 네트워크	9
1.7. 데이터 센터	11
<b>2장. 스토리지</b> .....	<b>12</b>
2.1. 스토리지 도메인 개요	12
2.2. 스토리지 백엔드 스토리지 도메인 유형	12
2.3. 스토리지 도메인 유형	13
2.4. 가상 디스크 이미지의 스토리지 형식	13
2.5. 가상 디스크 이미지 스토리지 HIERADATA 정책	13
2.6. RED HAT VIRTUALIZATION의 스토리지 메타데이터 버전	14
2.7. RED HAT VIRTUALIZATION의 스토리지 도메인 자동 복구	15
2.8. 스토리지 풀 관리자	15
2.9. 스토리지 풀 관리자 선택 프로세스	16
2.10. RED HAT VIRTUALIZATION의 전용 리소스 및 SANLOCK	17
2.11. 셸 프로비저닝 및 스토리지 과다 할당	18
2.12. 논리 볼륨 확장	18
<b>3장. 네트워크</b> .....	<b>20</b>
3.1. 네트워크 아키텍처	20
3.2. 소개: 기본 네트워킹 약관	20
3.3. 네트워크 인터페이스 컨트롤러	20
3.4. BRIDGE	20
3.5. 본딩	21
본딩 모드	21
3.6. 본딩을 위한 스위치 구성	22
3.7. 가상 네트워크 인터페이스 카드	22
3.8. 가상 LAN(VLAN)	24
3.9. 네트워크 라벨	24
3.10. 클러스터 네트워킹	25
3.11. 논리 네트워크	26
3.12. 필수 네트워크, 선택적 네트워크 및 가상 머신 네트워크	29
3.13. 가상 머신 연결	29
3.14. 포트 미러링	29
3.15. 호스트 네트워킹 구성	30
3.16. 브릿지 설정	30
3.17. VLAN 설정	31
3.18. 브릿지 및 본딩 설정	32
3.19. 다중 브리지, 다중 VLAN 및 NIC 설정	33
3.20. 다중 브리지, 다중 VLAN 및 본딩 구성	33
<b>4장. 전원 관리</b> .....	<b>35</b>
4.1. 전원 관리 및 펜싱 소개	35
4.2. RED HAT VIRTUALIZATION의 PROXY에 의한 전원 관리	35
4.3. 전원 관리	35
4.4. 펜싱	37
4.5. 소프트 펜싱 호스트	38

4.6. 다중 전원 관리 펜싱 에이전트 사용	39
<b>5장. 로드 밸런싱, 스케줄링 및 마이그레이션</b>	<b>41</b>
5.1. 로드 밸런싱, 스케줄링 및 마이그레이션	41
5.2. 로드 밸런싱 정책	41
5.3. 로드 밸런싱 정책: VM_EVENLY_DISTRIBUTED	42
5.4. 로드 밸런싱 정책: EVENLY_DISTRIBUTED	42
5.5. 로드 밸런싱 정책: POWER_SAVING	42
5.6. 로드 밸런싱 정책: 없음	43
5.7. 로드 밸런싱 정책: INCLUSTERUPGRADE	43
5.8. 고가용성 가상 머신	44
5.9. 스케줄링	44
5.10. MIGRATION	44
<b>6장. 디렉터리 서비스</b>	<b>46</b>
6.1. 디렉터리 서비스	46
6.2. 로컬 인증: 내부 도메인	46
6.3. GSSAPI를 사용한 원격 인증	47
<b>7장. 템플릿 및 풀</b>	<b>48</b>
7.1. 템플릿 및 풀	48
7.2. 템플릿	48
7.3. 풀	49
<b>8장. 가상 머신 스냅샷</b>	<b>50</b>
8.1. 스냅샷	50
8.2. RED HAT VIRTUALIZATION의 실시간 스냅샷	51
8.3. 스냅샷 생성	51
8.4. 스냅샷 프리뷰	53
8.5. 스냅샷 삭제	54
<b>9장. 하드웨어 드라이버 및 장치</b>	<b>56</b>
9.1. VIRTUALIZED HARDWARE	56
9.2. RED HAT VIRTUALIZATION에서 안정적인 장치 주소	56
9.3. 중앙 처리 장치(CPU)	57
9.4. 시스템 장치	57
9.5. 네트워크 장치	58
9.6. 그래픽 장치	58
9.7. 스토리지 장치	58
9.8. 동영상 장치	59
9.9. 직렬 드라이버	59
9.10. BALLOON 드라이버	59
<b>10장. 최소 요구 사항 및 기술 제한</b>	<b>60</b>
10.1. 최소 요구 사항 및 지원 제한	60
10.2. 리소스 제한	60
10.3. 클러스터 제한	60
10.4. 스토리지 도메인 제한	61
10.5. RED HAT VIRTUALIZATION MANAGER 제한 사항	62
10.6. 하이퍼바이저 요구 사항	62
10.7. 게스트 요구 사항 및 지원 제한	66
10.8. SPICE 제한	66
10.9. 참고 자료	66

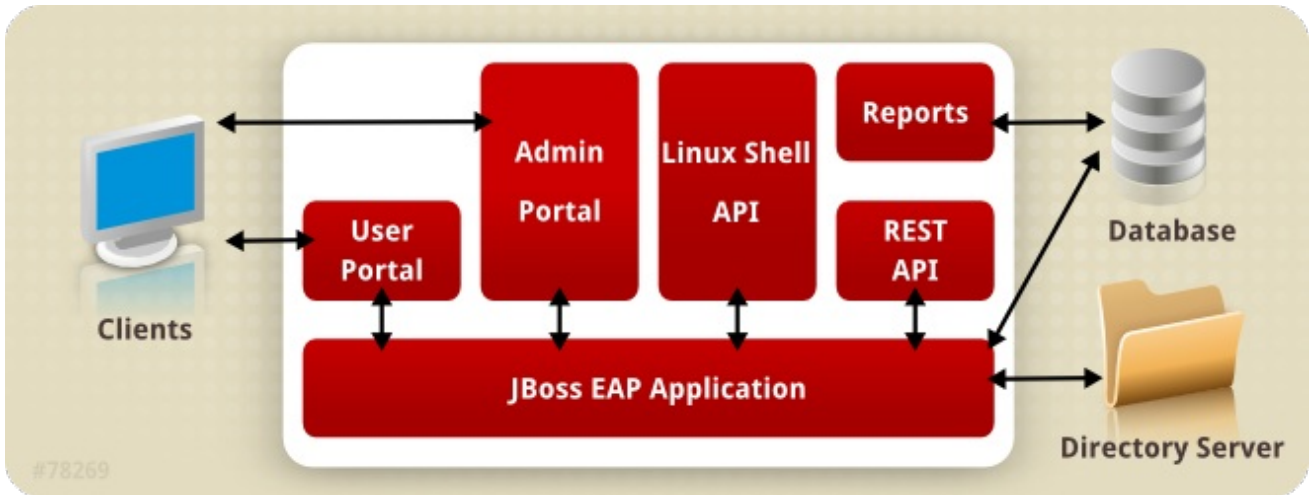


# 1장. 소개

## 1.1. RED HAT VIRTUALIZATION MANAGER

Red Hat Virtualization Manager는 가상화 환경을 위한 중앙 집중식 관리를 제공합니다. Red Hat Virtualization Manager에 액세스하는 데 다양한 인터페이스를 사용할 수 있습니다. 각 인터페이스를 사용하면 가상화된 환경에 다른 방식으로 액세스할 수 있습니다.

그림 1.1. Red Hat Virtualization Manager 아키텍처



Red Hat Virtualization Manager는 그래픽 인터페이스와 API( 애플리케이션 프로그래밍 인터페이스 )를 제공합니다. 각 인터페이스는 Red Hat JBoss Enterprise Application Platform 의 임베디드 인스턴스에서 제공하는 애플리케이션인 Manager에 연결됩니다. Red Hat JBoss Enterprise Application Platform 외에도 Red Hat Virtualization Manager를 지원하는 다른 구성 요소는 여러 가지가 있습니다.

## 1.2. RED HAT VIRTUALIZATION HOST

Red Hat Virtualization 환경에는 하나 이상의 호스트가 연결되어 있습니다. 호스트는 가상 시스템이 사용하는 물리적 하드웨어를 제공하는 서버입니다.

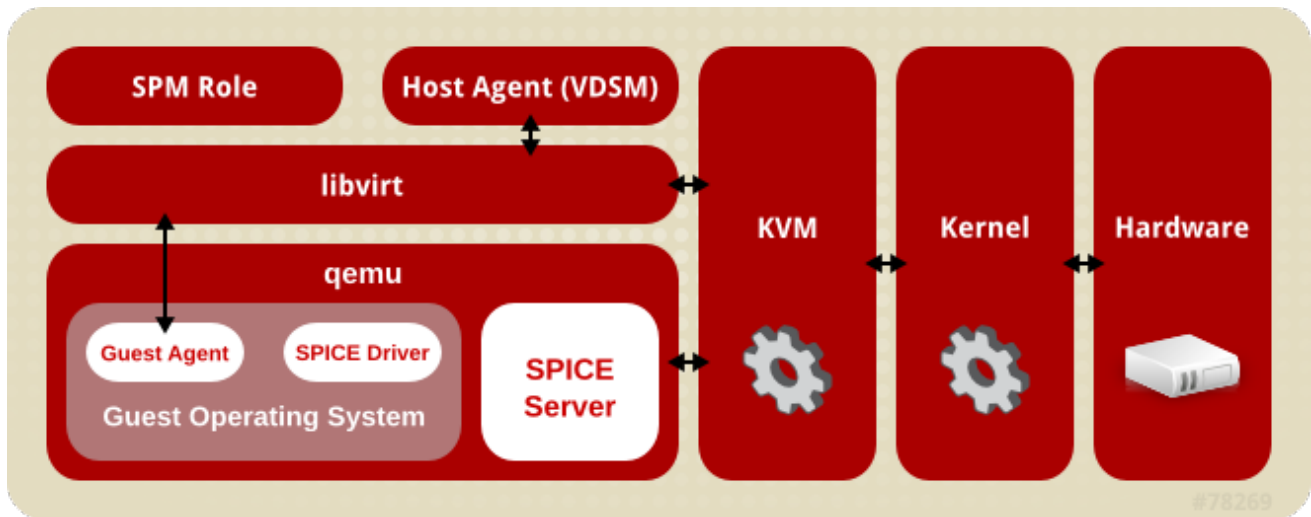
RHVH(Red Hat Virtualization Host)는 가상화 호스트를 생성하기 위해 특별히 사용자 지정된 특수 설치 미디어를 사용하여 설치된 최적화된 운영 체제를 실행합니다.

Red Hat Enterprise Linux 호스트는 설치 후 호스트로 사용할 수 있도록 구성된 표준 Red Hat Enterprise Linux 운영 체제를 실행하는 서버입니다.

두 가지 호스트 설치 방법 모두 호스트가 동일한 방식으로 나머지 가상화 환경과 상호 작용하는 호스트를 동일한 방식으로 호스트라고 합니다.



그림 1.2. 호스트 아키텍처



### KVM(커널 기반 가상 머신)

KVM(커널 기반 가상 시스템)은 Intel VT 또는 AMD-V 하드웨어 확장을 사용하여 완전한 가상화를 제공하는 로드 가능한 커널 모듈입니다. KVM 자체는 커널 공간에서 실행되지만 게스트는 사용자 공간에서 개별 QEMU 프로세스로 실행됩니다. KVM을 사용하면 가상 머신에서 물리적 하드웨어를 사용할 수 있습니다.

### QEMU

QEMU는 전체 시스템 에뮬레이션을 제공하는 데 사용되는 멀티 플랫폼 에뮬레이터입니다. QEMU는 하나 이상의 프로세서와 같은 전체 시스템을 에뮬레이션합니다. QEMU는 다른 운영 체제를 시작하거나 시스템 코드를 디버깅하는 데 사용할 수 있습니다. KVM 및 적절한 가상화 확장 기능이 있는 프로세서와 함께 작동하는 QEMU는 완전한 하드웨어 지원 가상화를 제공합니다.

### Red Hat Virtualization Manager 호스트 에이전트, VDSM

Red Hat Virtualization에서 **VDSM**은 가상 머신 및 스토리지에 대한 작업을 시작합니다. 또한 호스트 간 통신을 용이하게 합니다. VDSM은 메모리, 스토리지 및 네트워킹과 같은 호스트 리소스를 모니터링합니다. 또한 VDSM은 가상 머신 생성, 통계 누적 및 로그 수집과 같은 작업을 관리합니다. VDSM 인스턴스는 각 호스트에서 실행되며, 재구성 가능한 포트 **54321**을 사용하여 Red Hat Virtualization Manager에서 관리 명령을 받습니다.

### VDSM-REG

**VDSM**은 **VDSM-REG**를 사용하여 각 호스트를 Red Hat Virtualization Manager에 등록합니다. **VDSM-REG**는 포트 **80** 또는 포트 **443**을 사용하여 자체 및 호스트에 대한 정보를 제공합니다.

### libvirt

libvirt를 사용하면 가상 시스템 및 관련 가상 장치를 쉽게 관리할 수 있습니다. Red Hat Virtualization Manager가 가상 머신 라이프사이클 명령(시작, 중지, 재부팅)을 시작하면 VDSM에서 관련 호스트 머신에서 libvirt를 호출하여 실행합니다.

### 스토리지 풀 관리자(Storage Pool Manager)

스토리지 풀 관리자(SPM)는 데이터 센터에서 하나의 호스트에 할당된 역할입니다. DestinationRule 호스트는 데이터 센터의 모든 스토리지 도메인 구조 메타데이터 변경을 수행할 수 있는 유일한 권한을 갖습니다. 여기에는 가상 디스크 이미지, 스냅샷 및 템플릿 생성, 삭제 및 조작성이 포함됩니다. 또한 SAN(Storage Area Network)에서 스페스 블록 장치에 대한 스토리지 할당이 포함됩니다. ECDHE의 역할은 데이터 센터의 모든 호스트로 마이그레이션할 수 있습니다. 결과적으로 데이터 센터의 모든 호스트는 데이터 센터에 정의된 모든 스토리지 도메인에 액세스할 수 있어야 합니다.

Red Hat Virtualization Manager를 통해 항상 사용할 수 있는지 확인합니다. 스토리지 연결 오류가 발생하는 경우 Manager는 NSX 역할을 다른 호스트에 다시 할당합니다.

## 게스트 운영 체제

Red Hat Virtualization 환경에서 가상 머신을 수정하지 않고도 게스트 운영 체제를 설치할 수 있습니다. 게스트 운영 체제 및 게스트의 모든 애플리케이션은 가상화 환경을 인식하지 못하고 정상적으로 실행됩니다.

Red Hat은 가상화된 장치에 더 빠르고 효율적으로 액세스할 수 있는 향상된 장치 드라이버를 제공합니다. 또한 관리 콘솔에 향상된 게스트 정보를 제공하는 Red Hat Virtualization 게스트 에이전트를 게스트에 설치할 수도 있습니다.

## 1.3. 관리자 액세스 인터페이스

### 사용자 포털

데스크탑 가상화는 사용자가 개인용 컴퓨터의 데스크탑 환경과 유사한 데스크탑 환경을 제공합니다. 사용자 포털은 *가상 데스크탑 인프라*를 사용자에게 제공하기 위한 것입니다. 사용자는 웹 브라우저를 통해 사용자 포털에 액세스하여 할당된 가상 데스크탑을 표시하고 액세스합니다. 사용자 포털에서 사용자가 사용할 수 있는 작업은 시스템 관리자가 설정합니다. 표준 사용자는 시스템 관리자가 할당한 데스크탑을 시작, 중지 및 사용할 수 있습니다. 고급 사용자는 일부 관리 작업을 수행할 수 있습니다. 두 유형의 사용자 모두 동일한 URL에서 사용자 포털에 액세스하고 로그인 시 권한 수준에 적합한 옵션이 제공됩니다.

- **표준 사용자 액세스**

표준 사용자는 가상 데스크탑의 전원을 켜거나 끄고 사용자 포털을 통해 연결할 수 있습니다. 가상 머신에 대한 직접 연결은 *SPICE(Lightweight Computing Environment)* 또는 *VNC(Virtual Network Computing)* 클라이언트를 위한 간단한 프로토콜로 원활하게 연결됩니다. 두 프로토콜 모두 사용자에게 로컬로 설치된 데스크탑 환경과 유사한 환경을 제공합니다. 관리자는 가상 머신을 생성할 때 가상 머신에 연결하는 데 사용되는 프로토콜을 지정합니다.

사용자 포털 및 지원되는 브라우저 및 클라이언트에서 사용할 수 있는 작업에 대한 자세한 내용은 [사용자 포털 소개](#)에서 확인할 수 있습니다.

- **Power User Access**

Red Hat Virtualization 사용자 포털은 가상 리소스를 생성, 사용 및 모니터링할 수 있는 그래픽 사용자 인터페이스를 사용자에게 제공합니다. 시스템 관리자는 고급 사용자에게 액세스 권한을 부여하여 일부 관리 작업을 위임할 수 있습니다. 표준 사용자가 수행할 수 있는 작업 외에도 고급 사용자는 다음을 수행할 수 있습니다.

- 가상 머신을 생성, 편집 및 제거합니다.
- 가상 디스크 및 네트워크 인터페이스를 관리합니다.
- 사용자 권한을 가상 머신에 할당합니다.
- 템플릿을 생성하고 사용하여 가상 머신을 신속하게 배포합니다.
- 리소스 사용량 및 높은 심각도 이벤트를 모니터링합니다.
- 스냅샷을 생성하고 사용하여 가상 머신을 이전 상태로 복원합니다.

고급 사용자는 가상 시스템 관리 작업을 위임할 수 있습니다. 환경 관리자에게 데이터 센터 및 클러스터 수준 관리 작업이 저장됩니다.

## 관리 포털

관리 포털은 Red Hat Virtualization Manager 서버의 그래픽 관리 인터페이스입니다. 관리자는 웹 브라우저에서 이를 사용하여 가상화된 환경의 모든 요소를 모니터링, 생성 및 유지 관리할 수 있습니다. 관리 포털에서 수행할 수 있는 작업은 다음과 같습니다.

- 가상 인프라(네트워크, 스토리지 도메인) 생성 및 관리.
- 호스트 설치 및 관리
- 논리 엔티티(데이터 센터, 클러스터) 생성 및 관리.
- 가상 머신 생성 및 관리.
- Red Hat Virtualization 사용자 및 권한 관리.

관리 포털은 JavaScript를 사용하여 표시됩니다.

관리 포털 기능은 [Red Hat Virtualization 관리 가이드](#)에서 자세히 설명합니다. 관리 포털에서 지원하는 브라우저 및 플랫폼에 대한 정보는 [Red Hat Virtualization 설치 가이드](#)에서 참조하십시오.

## REST(Representational State Transfer) API

Red Hat Virtualization REST API는 Red Hat Virtualization 환경의 순환 및 제어를 위한 소프트웨어 인터페이스를 제공합니다. REST API는 HTTP 작업을 지원하는 모든 프로그래밍 언어에서 사용할 수 있습니다.

REST API 개발자 및 관리자는 다음을 수행할 수 있습니다.

- 엔터프라이즈 IT 시스템과 통합.
- 타사 가상화 소프트웨어와 통합.
- 자동화된 유지보수 및 오류 검사 작업을 수행합니다.
- 스크립트를 사용하여 Red Hat Virtualization 환경에서 반복적인 작업을 자동화합니다.

API 사양 및 사용 예제는 [Red Hat Virtualization REST API 가이드](#)를 참조하십시오.

## 1.4. 관리자를 지원하는 구성 요소

### Red Hat JBoss Enterprise Application Platform

Red Hat JBoss Enterprise Application Platform은 Java 애플리케이션 서버입니다. 플랫폼 간 Java 애플리케이션을 효율적으로 개발하고 제공할 수 있는 프레임워크를 제공합니다. Red Hat Virtualization Manager는 Red Hat JBoss Enterprise Application Platform을 사용하여 제공됩니다.



#### 중요

Red Hat Virtualization Manager와 함께 제공되는 Red Hat JBoss Enterprise Application Platform 버전은 다른 애플리케이션에 사용할 수 **없습니다**. Red Hat Virtualization Manager 서비스를 제공하기 위한 특정 목적으로 사용자 지정되었습니다. 추가 목적으로 Manager에 포함된 Red Hat JBoss Enterprise Application Platform을 사용하면 Red Hat Virtualization 환경을 서비스하는 데 부정적인 영향을 미칩니다.

Red Hat Virtualization Manager에는 호스트, 가상 머신 및 스토리지에 대한 모니터링 데이터를 수집하는 데이터 확보가 포함되어 있습니다. 여러 사전 정의 보고서를 사용할 수 있습니다. 고객은 환경을 분석하고 SQL을 지원하는 쿼리 도구를 사용하여 보고서를 생성할 수 있습니다.

Red Hat Virtualization Manager 설치 프로세스는 두 개의 데이터베이스를 생성합니다. 이러한 데이터베이스는 설치 중에 선택된 Postgres 인스턴스에서 생성됩니다.

- engine 데이터베이스는 Red Hat Virtualization Manager에서 사용하는 기본 데이터 저장소입니다. 상태, 구성 및 성능과 같은 가상화 환경에 대한 정보는 이 데이터베이스에 저장됩니다.
- ovirt\_engine\_history 데이터베이스에는 engine 작동 데이터베이스에서 시간이 지남에 따라 충돌하는 구성 정보 및 통계 메트릭이 포함되어 있습니다. engine 데이터베이스의 구성 데이터는 1분마다 검사되며 변경 사항은 ovirt\_engine\_history 데이터베이스에 복제됩니다. 데이터베이스에 대한 변경 사항을 추적하면 데이터베이스의 개체에 대한 정보가 제공됩니다. 이를 통해 Red Hat Virtualization 환경의 성능을 분석 및 개선하고 어려움을 해결할 수 있습니다.

ovirt\_engine\_history 데이터베이스를 기반으로 보고서를 생성하는 방법에 대한 자세한 내용은 *Red Hat Virtualization DataECDHE 가이드*의 [기록 데이터베이스](#)를 참조하십시오.



### 중요

ovirt\_engine\_history 데이터베이스의 데이터 복제는 **RHEVM 기록** 서비스인 ovirt-engine-dwhd에서 수행합니다.

## 디렉터리 서비스

디렉터리 서비스는 사용자 및 조직 정보에 대한 중앙 집중식 네트워크 기반 스토리지를 제공합니다. 저장된 정보 유형에는 애플리케이션 설정, 사용자 프로필, 그룹 데이터, 정책, 액세스 제어가 포함됩니다. Red Hat Virtualization Manager는 Active Directory, IdM(Identity Management), OpenLDAP 및 Red Hat Directory Server 9를 지원합니다. 관리 목적으로만 로컬, 내부 도메인도 있습니다. 이 내부 도메인에는 admin 사용자라는 하나의 사용자만 있습니다.

## 1.5. 스토리지

Red Hat Virtualization은 가상 디스크 이미지, 템플릿, 스냅샷 및 ISO 파일에 중앙 집중식 스토리지 시스템을 사용합니다. 스토리지는 스토리지 도메인으로 구성된 스토리지 풀로 논리적으로 그룹화됩니다. 스토리지 도메인은 스토리지의 내부 구조를 설명하는 스토리지 용량 및 메타데이터의 조합입니다. 스토리지 도메인은 데이터, 내보내기 및 ISO의 세 가지 유형이 있습니다.

데이터 스토리지 도메인은 각 데이터 센터에 필요한 유일한 도메인입니다. 데이터 스토리지 도메인은 단일 데이터 센터 전용입니다. 내보내기 및 ISO 도메인은 선택 사항입니다. 스토리지 도메인은 공유 리소스이며 데이터 센터의 모든 호스트에서 액세스할 수 있어야 합니다.

스토리지 네트워킹은 NFS(Network File System), iSCSI(Internet Small Computer System Interface), GlusterFS, FCP(Fibre Channel Protocol) 또는 POSIX 호환 네트워크 파일 시스템을 사용하여 구현할 수 있습니다.

NFS (및 기타 POSIX 호환 파일 시스템) 도메인에서 모든 가상 디스크, 템플릿 및 스냅샷은 간단한 파일입니다.

SAN (iSCSI/FCP) 도메인에서 블록 장치는 LVM(Logical Volume Manager)에서 볼륨 그룹(VG)에 의해 집계됩니다. 각 가상 디스크, 템플릿 및 스냅샷은 VG의 논리 볼륨(LV)입니다. LVM에 대한 자세한 내용은 [Red Hat Enterprise Linux Logical Volume Manager 관리 가이드](#)를 참조하십시오.

### 데이터 스토리지 도메인

데이터 도메인은 환경에서 실행되는 모든 가상 시스템의 가상 하드 디스크 이미지를 유지합니다. 가상 머신의 템플릿과 스냅샷도 데이터 도메인에 저장됩니다. 데이터 도메인은 데이터 센터 간에 공유할 수 없습니다.

### 스토리지 도메인 내보내기

내보내기 도메인은 데이터 센터와 Red Hat Virtualization 환경 간에 이미지를 복사하고 이동하는 데 사용되는 임시 스토리지 리포지토리입니다. 내보내기 도메인을 사용하여 가상 머신 및 템플릿을 백업할 수 있습니다. 내보내기 도메인은 데이터 센터 간에 이동할 수 있지만 한 번에 하나의 데이터 센터에서만 활성화할 수 있습니다.

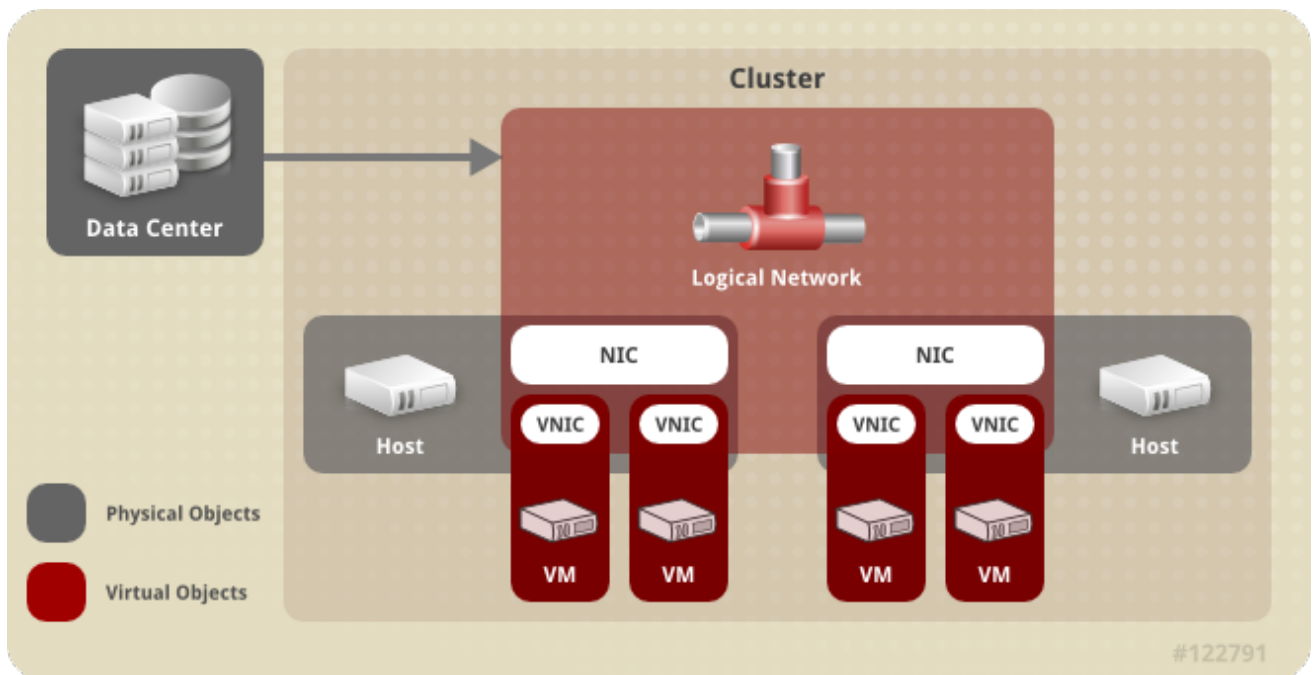
### ISO 스토리지 도메인

ISO 도메인은 가상 시스템의 운영 체제 및 애플리케이션을 설치하는 데 사용되는 논리적 CD-ROM인 ISO 파일을 저장합니다. 물리적 CD-ROM 또는 DVD 라이브러리를 대체하는 논리 엔티티로서 ISO 도메인은 물리적 미디어에 대한 데이터 센터의 필요성을 제거합니다. ISO 도메인은 다른 데이터 센터 간에 공유할 수 있습니다.

## 1.6. 네트워크

Red Hat Virtualization 네트워크 아키텍처를 사용하면 Red Hat Virtualization 환경의 다양한 요소를 연결할 수 있습니다. 네트워크 아키텍처는 네트워크 연결을 지원할 뿐만 아니라 네트워크 분리도 허용합니다.

그림 1.3. 네트워크 아키텍처



네트워크는 Red Hat Virtualization에 여러 계층으로 정의되어 있습니다. 기본 물리적 네트워킹 인프라를 마련하고 Red Hat Virtualization 환경의 논리 구성 요소와 하드웨어를 연결할 수 있도록 구성해야 합니다.

### 네트워킹 인프라 계층

Red Hat Virtualization 네트워크 아키텍처는 몇 가지 일반적인 하드웨어 및 소프트웨어 장치를 사용합니다.

- NIC (네트워크 인터페이스 컨트롤러)는 호스트를 네트워크에 연결하는 물리적 네트워크 인터페이스 장치입니다.

- *VNIC(가상NIC)*은 호스트의 물리적 NIC를 사용하여 작동하는 논리 NIC입니다. 가상 머신에 대한 네트워크 연결을 제공합니다.
- *본딩*은 여러 NIC를 단일 인터페이스에 바인딩합니다.
- *브리지*는 패킷 전환 네트워크의 패킷 전달 기술입니다. *가상 머신 논리적 네트워크*의 기반이 됩니다.

## 논리 네트워크

논리 네트워크를 사용하면 환경 요구 사항에 따라 네트워크 트래픽을 분리할 수 있습니다. 논리적 네트워크의 유형은 다음과 같습니다.

- 가상 머신 네트워크 트래픽을 전송하는 논리적 네트워크
- 가상 시스템 네트워크 트래픽을 전송하지 않는 논리 네트워크
- 선택적 논리적 네트워크
- 및 필수 네트워크

모든 논리 네트워크는 필수 또는 선택 사항일 수 있습니다.

가상 시스템 네트워크 트래픽을 전송하는 논리적 네트워크는 호스트 수준에서 소프트웨어 브릿지 장치로 구현됩니다. 기본적으로 Red Hat Virtualization Manager 설치 중에 하나의 논리 네트워크가 정의되어 있습니다(**ovirtmgmt** 관리 네트워크).

관리자가 추가할 수 있는 기타 논리적 네트워크는 전용 스토리지 논리 네트워크와 전용 디스플레이 논리 네트워크입니다. 가상 시스템 트래픽을 전송하지 않는 논리 네트워크에는 호스트에 연결된 브릿지 장치가 없습니다. 호스트 네트워크 인터페이스와 직접 연결됩니다.

Red Hat Virtualization은 마이그레이션 관련 네트워크 트래픽으로부터 관리 관련 네트워크 트래픽을 분리합니다. 따라서 실시간 마이그레이션에 전용 네트워크를 사용할 수 있으며(**ovirtmgmt**) 관리 네트워크(**ovirtmgmt**)가 마이그레이션 중에 하이퍼바이저에 대한 연결이 손실되지 않습니다.

## 서로 다른 계층의 논리적 네트워크에 대한 설명

논리 네트워크는 가상화 환경의 계층마다 다른 영향을 미칩니다.

### 데이터 센터 계층

논리적 네트워크는 데이터 센터 수준에서 정의됩니다. 각 데이터 센터에는 기본적으로 **ovirtmgmt** 관리 네트워크가 있습니다. 추가 논리 네트워크는 선택 사항이지만 권장됩니다. **VM 네트워크로 지정 및 사용자 지정 MTU**는 데이터 센터 수준에서 설정할 수 있습니다. 데이터 센터에 정의된 논리 네트워크도 논리 네트워크를 사용하는 클러스터에 추가해야 합니다.

### 클러스터 계층

논리적 네트워크는 데이터 센터에서 사용할 수 있도록 설정되며 이를 사용할 클러스터에 추가해야 합니다. 각 클러스터는 기본적으로 관리 네트워크에 연결되어 있습니다. 선택적으로 클러스터의 상위 데이터 센터에 정의된 클러스터 논리 네트워크에 추가할 수 있습니다. 필수 논리적 네트워크가 클러스터에 추가된 경우 클러스터의 각 호스트에 대해 구현해야 합니다. 선택적 논리적 네트워크는 필요에 따라 호스트에 추가할 수 있습니다.

### 호스트 계층

가상 시스템 논리 네트워크는 클러스터의 각 호스트에 대해 지정된 네트워크 인터페이스와 연결된 소프트웨어 브릿지 장치로 구현됩니다. 비가상 시스템 논리 네트워크에는 연결된 브릿지가 없으며 호스트 네트워크 인터페이스와 직접 연결됩니다. 각 호스트에는 Red Hat Virtualization 환경에 포함되어 있

기 때문에 네트워크 장치 중 하나를 사용하여 브리지로 구현되는 관리 네트워크가 있습니다. 클러스터에 추가된 필수 논리적 네트워크는 각 호스트의 네트워크 인터페이스와 연결되어 있어야 클러스터에 대해 작동할 수 있습니다.

## 가상 머신 계층

논리적 네트워크는 물리적 시스템에서 네트워크를 사용할 수 있는 것과 동일한 방식으로 가상 시스템에서 사용할 수 있도록 설정할 수 있습니다. 가상 머신은 가상 NIC를 실행하는 호스트에 구현된 모든 가상 머신 논리 네트워크에 연결할 수 있습니다. 그러면 가상 머신은 연결된 논리 네트워크에서 사용할 수 있는 다른 장치 또는 대상에 연결됩니다.

### 예 1.1. 관리 네트워크

**ovirtmgmt** 라는 관리 논리 네트워크는 Red Hat Virtualization Manager가 설치되면 자동으로 생성됩니다. **ovirtmgmt** 네트워크는 Red Hat Virtualization Manager와 호스트 간의 관리 트래픽 전용입니다. 특별히 설계된 다른 브릿지가 설정되지 않은 경우 **ovirtmgmt** 는 모든 트래픽의 기본 브릿지입니다.

## 1.7. 데이터 센터

데이터센터는 Red Hat Virtualization에서 최고 수준의 추상화를 제공합니다. 데이터 센터에는 세 가지 유형의 sub-containers로 구성된 컨테이너입니다.

- **스토리지 컨테이너**에는 스토리지 도메인의 연결 정보를 포함하여 스토리지 유형 및 스토리지 도메인에 대한 정보가 있습니다. 스토리지는 데이터 센터에 정의되며 데이터 센터의 모든 클러스터에서 사용할 수 있습니다. 데이터 센터 내의 모든 호스트 클러스터는 동일한 스토리지 도메인에 액세스할 수 있습니다.
- **네트워크 컨테이너**에는 데이터 센터의 논리적 네트워크에 대한 정보가 있습니다. 여기에는 네트워크 주소, VLAN 태그 및 STP 지원과 같은 세부 정보가 포함됩니다. 논리 네트워크는 데이터 센터에 정의되며, 선택적으로 클러스터 수준에서 구현됩니다.
- **클러스터 컨테이너**에는 클러스터가 있습니다. 클러스터는 호환 가능한 프로세서 코어(AMD 또는 Intel 프로세서)를 사용하는 호스트 그룹입니다. 클러스터는 마이그레이션 도메인입니다. 가상 머신은 다른 클러스터가 아닌 클러스터 내의 모든 호스트로 실시간으로 마이그레이션할 수 있습니다. 하나의 데이터 센터에는 여러 클러스터를 보유할 수 있으며 각 클러스터는 여러 호스트를 포함할 수 있습니다.



## 2장. 스토리지

### 2.1. 스토리지 도메인 개요

스토리지 도메인은 공통 스토리지 인터페이스가 있는 이미지 컬렉션입니다. 스토리지 도메인에는 템플릿 및 가상 머신(스냅샷 포함), ISO 파일 및 자체 메타데이터에 대한 완전한 이미지가 포함되어 있습니다. 스토리지 도메인은 블록 장치(SAN - iSCSI 또는 FCP) 또는 파일 시스템(NAS - NFS, GlusterFS 또는 기타 POSIX 호환 파일 시스템)으로 만들 수 있습니다.

NAS에서 모든 가상 디스크, 템플릿 및 스냅샷은 파일입니다.

SAN(iSCSI/FCP)에서 각 가상 디스크, 템플릿 또는 스냅샷은 논리 볼륨입니다. 블록 장치는 볼륨 그룹이라는 논리 엔티티로 집계된 다음 LVM(Logical Volume Manager)으로 분할되어 가상 하드 디스크로 사용됩니다. LVM에 대한 자세한 내용은 [Red Hat Enterprise Linux Logical Volume Manager 관리 가이드를 참조하십시오](#).

가상 디스크는 QCOW2 또는 RAW의 두 가지 형식 중 하나를 사용할 수 있습니다. 스토리지 유형은 Sparse 또는 Preallocated일 수 있습니다. 스냅샷은 항상 스파스이지만 RAW 또는 스파스로 생성된 디스크에 사용할 수 있습니다.

동일한 스토리지 도메인을 공유하는 가상 머신은 동일한 클러스터에 속한 호스트 간에 마이그레이션할 수 있습니다.

### 2.2. 스토리지 백엔드 스토리지 도메인 유형

스토리지 도메인은 블록 기반 및 파일 기반 스토리지를 사용하여 구현할 수 있습니다.

#### 파일 기반 스토리지

Red Hat Virtualization에서 지원하는 파일 기반 스토리지 유형은 NFS, GlusterFS, 기타 POSIX 호환 파일 시스템 및 호스트의 로컬 스토리지입니다.

파일 기반 스토리지는 Red Hat Virtualization 환경으로 외부에서 관리됩니다.

NFS 스토리지는 Red Hat Enterprise Linux NFS 서버 또는 기타 타사 네트워크 연결 스토리지 서버에서 관리합니다.

호스트는 자체 로컬 스토리지 파일 시스템을 관리할 수 있습니다.

#### 블록 기반 스토리지

블록 스토리지는 포맷되지 않은 블록 장치를 사용합니다. 블록 장치는 LVM(Logical Volume Manager)에 의해 *볼륨 그룹*으로 집계됩니다. LVM의 인스턴스는 모든 호스트에서 실행되며, 다른 호스트에서 실행되는 인스턴스를 인식하지 못합니다. VDSM은 변경 사항을 위해 볼륨 그룹을 스캔하여 LVM 상단에 클러스터링 논리를 추가합니다. 변경 사항이 감지되면 VDSM은 볼륨 그룹 정보를 새로 고침하도록 지시하여 개별 호스트를 업데이트합니다. 호스트는 볼륨 그룹을 논리 볼륨으로 분할하여 논리 볼륨 메타데이터를 디스크에 씁니다. 기존 스토리지 도메인에 스토리지 용량이 더 추가되면 Red Hat Virtualization Manager를 통해 각 호스트에서 VDSM이 볼륨 그룹 정보를 새로 고칩니다.

논리 단위 번호(LUN)는 개별 블록 장치입니다. 지원되는 블록 스토리지 프로토콜 중 하나인 iSCSI, FCoE 또는 SAS를 사용하여 LUN에 연결합니다. Red Hat Virtualization Manager는 LUN에 대한 소프트웨어 iSCSI 연결을 관리합니다. 다른 모든 블록 스토리지 연결은 Red Hat Virtualization 환경에서 외부에서 관리됩니다. 논리 볼륨 생성, 논리 볼륨 확장 또는 삭제, 새 LUN 추가와 같은 블록 기반 스토리지 환경의 모든 변경 사항은 스토리지 풀 관리자라는 특별히 선택한 호스트에서 LVM에 의해 처리됩니다. 그런 다음 VDSM에서 변경 사항을 동기화하여 클러스터의 모든 호스트에서 스토리지 메타데이터를 새로 고칩니다.



## 2.3. 스토리지 도메인 유형

Red Hat Virtualization은 각 스토리지 도메인에서 지원하는 스토리지 유형을 포함하여 세 가지 유형의 스토리지 도메인을 지원합니다.

- *Data Storage Domain* 은 Red Hat Virtualization 환경에 있는 모든 가상 머신의 하드 디스크 이미지를 저장합니다. 디스크 이미지에는 설치된 운영 체제 또는 가상 머신에서 저장되거나 생성된 데이터가 포함될 수 있습니다. 데이터 스토리지 도메인은 NFS, iSCSI, FCP, GlusterFS 및 POSIX 호환 스토리지를 지원합니다. 데이터 도메인은 여러 데이터 센터 간에 공유할 수 없습니다.
- *Export Storage Domain* 은 하드 디스크 이미지와 데이터 센터 간에 전송되는 가상 머신 템플릿을 위한 전송 스토리지를 제공합니다. 또한 내보내기 스토리지 도메인은 가상 머신 복사본을 백업합니다. 내보내기 스토리지 도메인은 NFS 스토리지를 지원합니다. 여러 데이터 센터에서 단일 내보내기 스토리지 도메인에 액세스할 수 있지만 한 번에 하나의 데이터 센터만 사용할 수 있습니다.
- *ISO 스토리지 도메인* 은 이미지라고도 하는 ISO 파일을 저장합니다. ISO 파일은 물리적 CD 또는 DVD의 표현입니다. Red Hat Virtualization 환경에서 일반적인 ISO 파일은 운영 체제 설치 디스크, 애플리케이션 설치 디스크 및 게스트 에이전트 설치 디스크입니다. 이러한 이미지는 가상 머신에 연결하고 물리적 디스크가 디스크 드라이브에 삽입되고 부팅되는 것과 동일한 방식으로 부팅될 수 있습니다. ISO 스토리지 도메인을 사용하면 데이터 센터 내의 모든 호스트가 ISO를 공유할 수 있으므로 물리적 미디어가 필요하지 않습니다.

## 2.4. 가상 디스크 이미지의 스토리지 형식

### QCOW2 포맷 가상 머신 스토리지

QCOW2는 가상 디스크 이미지의 스토리지 형식입니다. QCOW 는 쓰기 시 *QEMU* 사본입니다. QCOW2 형식은 논리 블록과 물리 블록 간에 매핑을 추가하여 가상 계층에서 물리 스토리지 계층을 분리합니다. 각 논리 블록은 물리적 오프셋에 매핑되어 스토리지 과다 커밋 및 가상 시스템 스냅샷을 활성화합니다. 각 QCOW 볼륨은 기본 디스크 이미지에 대한 변경 사항만 나타냅니다.

초기 매핑은 모든 논리 블록을 백업 파일 또는 볼륨의 오프셋으로 가리킵니다. 가상 머신이 스냅샷 후 QCOW2 볼륨에 데이터를 쓰는 경우 백업 볼륨에서 관련 블록을 읽고 새 정보를 사용하여 수정하고 새 스냅샷 QCOW2 볼륨에 기록됩니다. 그런 다음 맵이 새 위치를 가리키도록 업데이트됩니다.

### RAW

RAW 스토리지 형식은 QCOW2보다 성능 이점이 있으므로 RAW 형식으로 저장된 가상 디스크 이미지에 포맷이 적용되지 않습니다. RAW 형식으로 저장된 디스크 이미지의 가상 머신 데이터 작업에는 호스트의 추가 작업이 필요하지 않습니다. 가상 시스템이 가상 디스크의 지정된 오프셋에 데이터를 쓰는 경우 백업 파일 또는 논리 볼륨의 동일한 오프셋에 I/O가 기록됩니다.

원시 형식을 사용하려면 스토리지 어레이에서 외부에서 관리되는 썬 프로비저닝 LUN을 사용하지 않는 한 정의된 이미지의 전체 공간을 사전 할당해야 합니다.

## 2.5. 가상 디스크 이미지 스토리지 HIERADATA 정책

### 사전 할당된 스토리지

가상 머신 생성 전에 가상 디스크 이미지에 필요한 모든 스토리지가 할당됩니다. 가상 머신에 대해 20GB 디스크 이미지가 생성되는 경우 디스크 이미지는 20GB의 스토리지 도메인 용량을 사용합니다. 사전 할당된 디스크 이미지는 사용할 수 없습니다. 유연성을 고려하여 런타임 중에 스토리지 할당이 발생하지 않으므로 스토리지 사전 할당이 더 빠른 쓰기 시간을 의미할 수 있습니다. 따라서 스토리지를 할당하면 Red Hat Virtualization Manager의 용량을 줄여 스토리지를 과다 할당할 수 있습니다. 사전 할당된 스토리지는 스토리지의 대기 시간을 줄이기 위해 고가성 I/O 작업에 사용되는 가상 머신에 권장됩니다. 일반적으로 서버 가상 머신은 이 설명에 적합합니다.



## 참고

스토리지 백엔드에서 제공하는 썬 프로비저닝 기능을 사용하는 경우 가상 머신의 스토리지를 프로비저닝할 때 사전 할당된 스토리지를 관리 포털에서 계속 선택해야 합니다.

### sparsely Allocated Storage

가상 디스크 이미지의 상한 크기 제한은 가상 머신 생성 시 설정됩니다. 처음에는 디스크 이미지에서 스토리지 도메인 용량을 사용하지 않습니다. 대문자에 도달할 때까지 가상 머신에서 디스크에 데이터를 쓸 때 용량이 증가합니다. 디스크 이미지의 데이터가 제거되면 용량이 스토리지 도메인에 반환되지 않습니다. 스파스로 할당된 스토리지는 스토리지 대기 시간을 위해 다소 허용 오차가 낮거나 중간 수준의 I/O 작업이 있는 가상 머신에 적합합니다. 일반적으로 데스크탑 가상 머신은 이 설명에 적합합니다.



## 참고

스토리지 백엔드에서 썬 프로비저닝 기능을 제공하는 경우 썬 프로비저닝의 기본 구성으로 사용해야 합니다. 스토리지는 미리 할당된 그래픽 사용자 인터페이스에서 프로비저닝해야 하며 백엔드 솔루션으로 썬 프로비저닝을 유지해야 합니다.

## 2.6. RED HAT VIRTUALIZATION의 스토리지 메타데이터 버전

Red Hat Virtualization은 스토리지 도메인에 대한 정보를 스토리지 도메인 자체의 메타데이터로 저장합니다. Red Hat Virtualization의 각 주요 릴리스에서는 스토리지 메타데이터 구현이 개선되었습니다.

- *V1 메타데이터 (Red Hat Virtualization 2.x 시리즈)*

각 스토리지 도메인에는 자체 구조와 가상 디스크 이미지를 지원하는 데 사용되는 물리 볼륨의 이름을 설명하는 메타데이터가 포함되어 있습니다.

마스터 도메인에는 스토리지 풀의 모든 도메인 및 물리 볼륨 이름에 대한 메타데이터도 포함됩니다. 이 메타데이터의 총 크기는 2KB로 제한되므로 풀에 있을 수 있는 스토리지 도메인 수를 제한합니다.

템플릿 및 가상 머신 기본 이미지만 읽습니다.

V1 메타데이터는 NFS, iSCSI 및 FC 스토리지 도메인에 적용할 수 있습니다.

- *V2 메타데이터 (Red Hat Enterprise Virtualization 3.0)*

모든 스토리지 도메인 및 풀 메타데이터는 논리 볼륨에 기록되지 않고 논리 볼륨 태그로 저장됩니다. 가상 디스크 볼륨에 대한 메타데이터는 여전히 도메인의 논리 볼륨에 저장됩니다.

물리 볼륨 이름은 더 이상 메타데이터에 포함되지 않습니다.

템플릿 및 가상 머신 기본 이미지만 읽습니다.

V2 메타데이터는 iSCSI 및 FC 스토리지 도메인에 적용할 수 있습니다.

- *V3 메타데이터 (Red Hat Enterprise Virtualization 3.1 이상)*

모든 스토리지 도메인 및 풀 메타데이터는 논리 볼륨에 기록되지 않고 논리 볼륨 태그로 저장됩니다. 가상 디스크 볼륨에 대한 메타데이터는 여전히 도메인의 논리 볼륨에 저장됩니다.

가상 머신 및 템플릿 기본 이미지는 더 이상 읽기 전용이 아닙니다. 이러한 변경으로 인해 스냅샷에서 실시간 스냅샷, 실시간 스토리지 마이그레이션 및 복제가 가능합니다.

영어가 아닌 볼륨 이름에 대한 지원이 추가되었습니다.

V3 메타데이터는 NFS, GlusterFS, POSIX, iSCSI, FC 스토리지 도메인에 적용할 수 있습니다.

## 2.7. RED HAT VIRTUALIZATION의 스토리지 도메인 자동 복구

Red Hat Virtualization 환경의 호스트는 각 도메인의 메타데이터를 읽고 데이터 센터의 스토리지 도메인을 모니터링합니다. 스토리지 도메인에 액세스할 수 없는 데이터 센터 보고서의 모든 호스트가 스토리지 도메인에 액세스할 수 없는 경우 스토리지 도메인이 비활성화됩니다.

비활성 스토리지 도메인의 연결을 해제하는 대신 Manager는 예를 들어 임시 네트워크 중단으로 인해 스토리지 도메인이 일시적으로 비활성화되었다고 가정합니다. 5분마다 Manager는 비활성 스토리지 도메인을 다시 활성화하려고 합니다.

스토리지 연결 중단의 원인을 해결하기 위해 관리자의 개입이 필요할 수 있지만 Manager는 연결이 복원되면 스토리지 도메인을 다시 활성화하는 프로세스를 처리합니다.

## 2.8. 스토리지 풀 관리자

Red Hat Virtualization은 메타데이터를 사용하여 스토리지 도메인의 내부 구조를 설명합니다. 구조적 메타데이터는 각 스토리지 도메인의 세그먼트에 작성됩니다. 호스트는 단일 작성기 및 여러 리더 구성을 기반으로 스토리지 도메인 메타데이터와 함께 작동합니다. 스토리지 도메인 구조 메타데이터는 이미지 및 스냅샷 생성 및 삭제, 볼륨 및 도메인 확장을 추적합니다.

데이터 도메인 구조를 변경할 수 있는 호스트는 스토리지 풀 관리자(SPM)라고 합니다. FlexVolume은 디스크 이미지 생성 및 삭제, 스냅샷 생성 및 병합, 스토리지 도메인 간 이미지 복사, 블록 장치의 템플릿 및 스토리지 할당 생성과 같은 데이터 센터의 모든 메타데이터 변경 사항을 조정합니다. 모든 데이터 센터에는 1개의ECDHE가 있습니다. 다른 모든 호스트는 스토리지 도메인 구조 메타데이터만 읽을 수 있습니다.

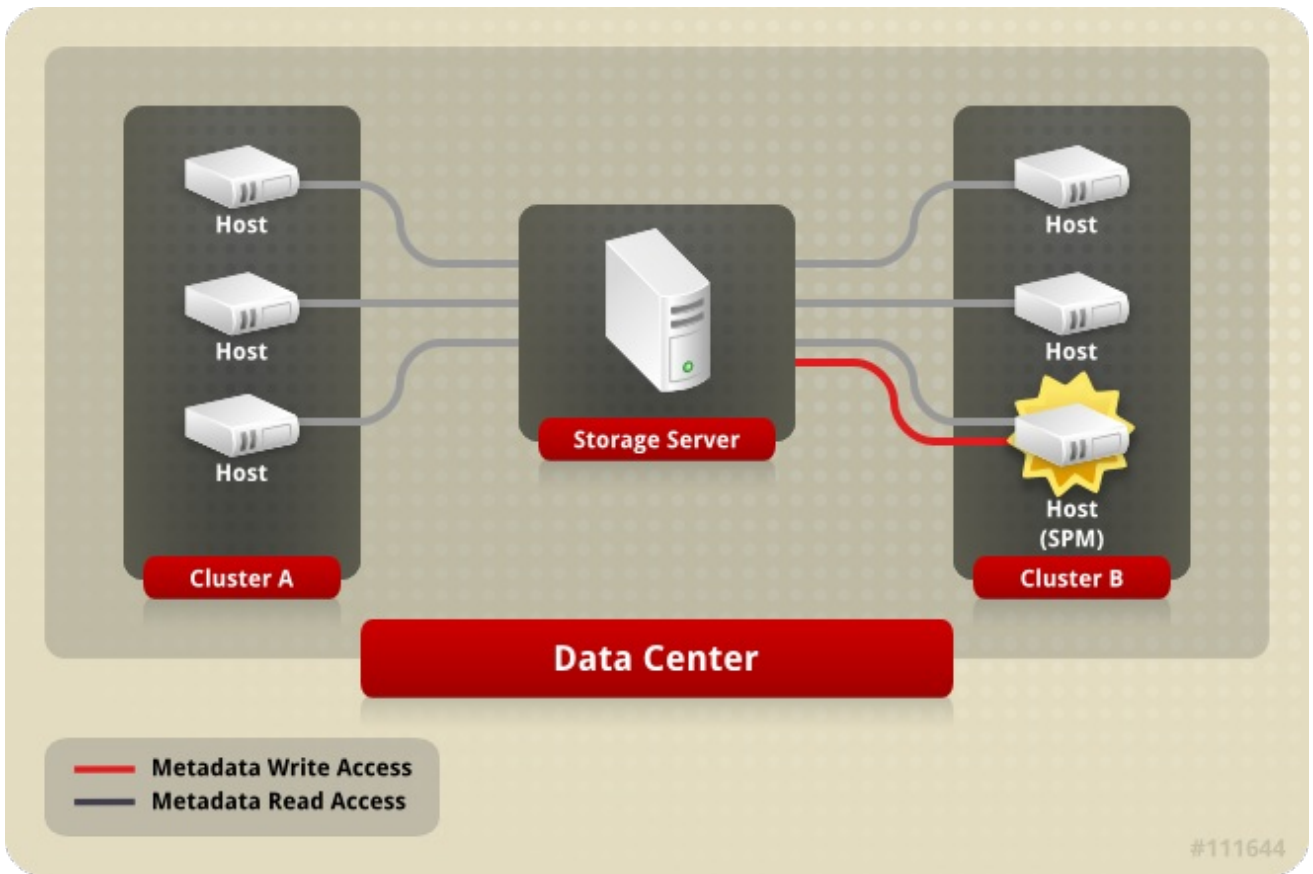
호스트는 FlexVolume으로 수동으로 선택하거나 Red Hat Virtualization Manager에서 할당할 수 있습니다. Manager는 잠재적인ECDHE 호스트가 *스토리지 중심 리스* 를 가정하도록 하여 DestinationRule 역할을 할당합니다. 임대를 통해 host는 스토리지 메타데이터를 작성할 수 있습니다. Manager 또는 호스트에서 추적하지 않고 스토리지 도메인에 작성되므로 스토리지 중심입니다. 스토리지 중심 리스는 *리스* 라는 마스터 스토리지 도메인의 특수 논리 볼륨에 작성됩니다. 데이터 도메인 구조에 대한 메타데이터는 **메타데이터** 라는 특수 논리 볼륨에 기록됩니다. **메타데이터** 논리 볼륨 변경 사항은 **임대** 논리 볼륨에 의해 보호됩니다.

Manager는 VDSM을 사용하여 호스트에 **spmStart** 명령을 실행하여 해당 호스트의 VDSM이 스토리지 중심 리스를 가정하려고 합니다. 호스트가 성공하면 Red Hat Virtualization Manager에서 새 호스트의 역할을 추측할 때까지 Red Hat Virtualization Manager에서 스토리지 중심 리스를 유지합니다.

다음과 같은 경우 Manager 역할을 다른 호스트로 이동합니다.

- ECDHE 호스트는 모든 스토리지 도메인에 액세스할 수 없지만 마스터 스토리지 도메인에 액세스할 수 있습니다.
- storage connection의 손실 또는 리스 볼륨이 가득 차 있고 쓰기 작업을 수행할 수 없기 때문에 host는 리스를 갱신할 수 없습니다.
- gradle 호스트가 충돌합니다.

그림 2.1. 스토리지 풀 관리자는 예외적으로 구조적 메타데이터를 작성합니다.



## 2.9. 스토리지 풀 관리자 선택 프로세스

호스트에 SPM(Storage Pool Manager) 역할이 수동으로 할당되지 않은 경우, Red Hat Virtualization Manager에서 선택 프로세스를 시작하고 관리합니다.

먼저 Red Hat Virtualization Manager에서 VDSM에 스토리지 중심 리스가 있는 호스트를 확인하도록 요청합니다.

Red Hat Virtualization Manager는 스토리지 도메인을 처음 생성할 때부터 FlexVolume 할당 내역을 추적합니다. DestinationRule 역할의 가용성은 다음 세 가지 방법으로 확인됩니다.

- "getSPMstatus" 명령: Manager는 VDSM을 사용하여 status가 last이고 "SPM", "Contending" 또는 "Free" 중 하나를 수신하는 호스트를 확인합니다.
- 스토리지 도메인의 메타데이터 볼륨에는 최신 상태가 있는 마지막 호스트가 포함되어 있습니다.
- 스토리지 도메인의 메타데이터 볼륨에는 status가 있는 마지막 호스트 버전이 포함되어 있습니다.

운영 가능한 응답 호스트에서 스토리지 중심 리스를 유지하는 경우 Red Hat Virtualization Manager는 관리자 포털에서 해당 호스트를 표시합니다. 더 이상의 작업은 수행되지 않습니다.

ECDHE 호스트가 응답하지 않는 경우 연결할 수 없는 것으로 간주됩니다. 전원 관리가 호스트에 대해 구성된 경우 자동으로 펜싱됩니다. 그렇지 않은 경우 수동 펜싱이 필요합니다. 이전 스토리지 풀 관리자를 펜싱할 때까지 스토리지 풀 관리자 역할을 새 호스트에 할당할 수 없습니다.

FlexVolume 역할 및 스토리지 중심 리스를 사용할 수 있는 경우 Red Hat Virtualization Manager는 데이터 센터에서 무작위로 선택한 운영 호스트에 할당합니다.

새 호스트에서 ECDHE 역할 할당이 실패하면 Red Hat Virtualization Manager는 작업이 실패한 호스트가 포함된 목록에 호스트를 추가하여 이러한 호스트를 ECDHE 역할에 적합하지 않음으로 표시합니다. 이 목록은 다음 BOOM 선택 프로세스의 시작 부분에 해제되어 모든 호스트를 다시 사용할 수 있습니다.

Red Hat Virtualization Manager는 BOOM 선택이 성공할 때까지 실패한 호스트 목록에 없는 무작위로 선택한 호스트에서 스토리지 풀 관리자 역할 및 스토리지 중심 리스를 가정하도록 계속 요청합니다.

현재 BOOM이 응답하지 않거나 책임을 이행할 수 없는 경우 Red Hat Virtualization Manager에서 Storage Pool Manager 선택 프로세스를 시작합니다.

## 2.10. RED HAT VIRTUALIZATION의 전용 리소스 및 SANLOCK

Red Hat Virtualization 환경의 특정 리소스는 독점적으로 액세스해야 합니다.

role은 이러한 리소스 중 하나입니다. 둘 이상의 호스트가 두 개 이상의 호스트가 되는 경우 동일한 데이터가 한 번에 두 위치에서 변경될 수 있으므로 데이터 손상 위험이 발생할 수 있습니다.

Red Hat Enterprise Virtualization 3.1 이전에는 *safe/lease* 라는 VDSM 기능을 사용하여 배상성을 유지하고 추적했습니다. 임대는 데이터 센터의 모든 스토리지 도메인의 특수 영역에 작성되었습니다. 환경의 모든 호스트는 네트워크 독립적인 방식으로 status를 추적할 수 있습니다. VDSM의 안전한 임대는 하나의 리소스에만 유지 관리(role)입니다.

Sanlock은 동일한 기능을 제공하지만 잠길 수 있는 리소스 중 하나로 이 역할을 처리합니다. Sanlock은 추가 리소스를 잠글 수 있으므로 더 유연합니다.

리소스 잠금이 필요한 애플리케이션은 Sanlock에 등록할 수 있습니다. 등록된 애플리케이션은 다른 애플리케이션이 액세스할 수 없도록 해당 리소스를 대신하여 Sanlock 잠금을 요청할 수 있습니다. 예를 들어 VDSM 상태 잠금을 사용하는 대신 VDSM에서 이제 Sanlock이 이를 요청합니다.

잠금은 잠금 공간의 디스크에서 추적됩니다. 각 스토리지 도메인에는 하나의 잠금 공간이 있습니다. BOOM 리소스에 대한 잠금의 경우, 각 호스트의 활성 상태가 호스트가 스토리지에 연결될 때 수신한 호스트를 갱신하고 일정한 간격으로 잠금 공간에 타임스탬프를 쓰는 기능을 통해 잠금 공간에 추적됩니다. *ids* 논리 볼륨은 각 호스트의 고유 식별자를 추적하고 호스트가 *hostid*를 업데이트할 때마다 업데이트됩니다. *DestinationRule* 리소스는 라이브 호스트에서만 보유할 수 있습니다.

리소스는 임대 논리 볼륨의 디스크에서 추적됩니다. 디스크에 대한 표현이 해당 프로세스의 고유 식별자로 업데이트될 때 리소스는 수행됩니다. A resource is said to be *taken* when its representation on disk has been updated with the unique identifier of the process that has taken it. ECDHE 역할의 경우, 해당 리소스를 가져온 *hostid*로 리소스가 업데이트됩니다.

각 호스트의 Sanlock 프로세스는 리소스를 한 번만 확인하면 됩니다. 초기 점검 후 Sanlock은 잠긴 리소스를 가진 호스트의 타임 스탬프가 될 때까지 잠금 공간을 모니터링할 수 있습니다.

Sanlock은 리소스를 사용하는 애플리케이션을 모니터링합니다. 예를 들어 VDSM 상태 및 *hostid*에 대해 모니터링됩니다. 호스트가 Manager에서 *hostid*를 갱신할 수 없는 경우 잠금 공간에 있는 모든 리소스에서 배타성이 손실됩니다. Sanlock은 리소스를 업데이트하여 더 이상 사용되지 않음을 표시합니다.

ECDHE 호스트에서 지정된 시간 동안 스토리지 도메인의 잠금 공간에 타임스탬프를 쓸 수 없는 경우 VDSM 프로세스에서 리소스를 해제하도록 호스트의 SANlock 인스턴스를 요청합니다. VDSM 프로세스가 응답하는 경우 해당 리소스가 해제되고 잠금 공간의 resources는 다른 호스트에서 사용할 수 있습니다.

*DestinationRule* 호스트의 VDSM이 리소스 릴리스 요청에 응답하지 않는 경우 호스트의 Sanlock은 VDSM 프로세스를 종료합니다. kill 명령이 실패하면 Sanlock은 sigkill을 사용하여 VDSM을 종료하여 에스컬레이션됩니다. sigkill이 실패하면 Sanlock은 위치득 때문에 따라 호스트를 재부팅합니다.

호스트의 VDSM이 호스트 ID를 갱신하고 타임 스탬프를 잠금 공간에 쓸 때마다 watchdog 데몬은 반려 동을 수신합니다. VDSM이 이 작업을 수행할 수 없는 경우 watchdog 데몬이 더 이상 차단되지 않습니다. 위

치독 데몬에서 지정된 시간 동안 반려동을 수신하지 않으면 호스트를 재부팅합니다. 이 최종 수준의 에스컬레이션은 (이미 도달되는 경우) resources 리소스가 해제되고 다른 호스트에서 수행할 수 있습니다.

## 2.11. 썬 프로비저닝 및 스토리지 과다 할당

Red Hat Virtualization Manager는 가상화 환경 내에서 스토리지 사용량을 최적화하기 위한 프로비저닝 정책을 제공합니다. 썬 프로비저닝 정책을 사용하면 가상화 환경의 실제 스토리지 사용량에 따라 스토리지 리소스를 과도하게 커밋하고 스토리지를 프로비저닝할 수 있습니다.

스토리지 과다 할당은 스토리지 풀에서 물리적으로 사용 가능한 것보다 더 많은 스토리지를 가상 머신에 할당하는 것입니다. 일반적으로 가상 머신은 할당된 스토리지보다 적은 스토리지를 사용합니다. 썬 프로비저닝을 사용하면 가상 머신에 대해 정의된 스토리지가 완전히 할당된 것처럼 작동할 수 있습니다. 실제로는 스토리지의 일부만 할당된 경우입니다.



### 참고

Red Hat Virtualization Manager는 자체 썬 프로비저닝 기능을 제공하지만 스토리지 백엔드의 썬 프로비저닝 기능을 사용해야 합니다.

스토리지 과다 할당을 지원하기 위해 VDSM에 임계값이 정의되어 있으며 논리 스토리지 할당을 실제 스토리지 사용량과 비교합니다. 이 임계값은 디스크 이미지에 기록된 데이터가 이를 지원하는 논리 볼륨보다 작는지 확인하는 데 사용됩니다. QEMU는 논리 볼륨에서 작성된 가장 높은 오프셋을 식별합니다. 이는 가장 큰 스토리지 사용 지점을 나타냅니다. VDSM은 QEMU에서 표시된 가장 높은 오프셋을 모니터링하여 사용량이 정의된 임계값을 초과하지 않는지 확인합니다. VDSM이 가장 높은 오프셋이 임계값 아래로 유지됨을 나타내는 한 Red Hat Virtualization Manager는 해당 논리 볼륨이 계속 작업을 수행할 수 있는 충분한 스토리지가 있음을 알고 있습니다.

QEMU가 사용량이 임계값 제한을 초과했음을 나타내는 경우 VDSM은 디스크 이미지가 곧 논리 볼륨 크기에 도달할 것이라고 관리자에게 통신합니다. Red Hat Virtualization Manager에서 host가 논리 볼륨을 확장하도록 요청합니다. 데이터 센터의 데이터 스토리지 도메인에 사용 가능한 공간이 있는 한 이 프로세스를 반복할 수 있습니다. 데이터 스토리지 도메인이 사용 가능한 공간이 부족하면 확장할 스토리지 용량을 수동으로 추가해야 합니다.

## 2.12. 논리 볼륨 확장

Red Hat Virtualization Manager는 썬 프로비저닝을 사용하여 스토리지 풀에서 사용 가능한 스토리지를 과다 할당하고, 실제 사용 가능한 것보다 더 많은 스토리지를 할당합니다. 가상 머신은 작동 시 데이터를 작성합니다. 썬 프로비저닝된 디스크 이미지가 있는 가상 머신은 결국 디스크 이미지를 지원하는 논리 볼륨보다 더 많은 데이터를 작성합니다. 이러한 상황이 발생하면 논리 볼륨 확장 기능을 사용하여 추가 스토리지를 제공하고 가상 시스템의 지속적인 작업을 원활하게 수행할 수 있습니다.

Red Hat Virtualization은 LVM에 비해 썬 프로비저닝 메커니즘을 제공합니다. QCOW2 포맷 스토리지를 사용하는 경우 Red Hat Virtualization은 호스트 시스템 프로세스 *qemu-kvm* 을 사용하여 디스크의 스토리지 블록을 순차적으로 논리 블록에 매핑합니다. 예를 들어 논리 100GB 디스크의 정의가 1GB 논리 볼륨에서 지원됩니다. *qemu-kvm*이 VDSM에 설정된 사용 임계값을 초과하면 로컬 VDSM 인스턴스에서 논리 볼륨의 다른 1GB에 의해 확장되도록 request합니다. 볼륨 확장이 필요한 가상 머신을 실행하는 호스트의 VDSM에 더 많은 공간이 필요하다는 것을 알릴 수 있습니다. FlexVolume은 논리 볼륨과 VDSM 인스턴스를 확장하면 호스트 VDSM에서 볼륨 그룹 정보를 새로 고치고 확장 작업이 완료되었음을 인식합니다. 호스트는 작업을 계속할 수 있습니다.

논리 볼륨 확장에서는 호스트에서 다른 호스트가 어떤 호스트인지 알 필요가 없으며, 자체일 수도 있습니다. 스토리지 확장 통신은 스토리지 6.4를 통해 수행됩니다. 스토리지 6.4는 데이터 스토리지 도메인의 전용 논리 볼륨입니다. 논리 볼륨을 확장하기 위해 ECDHE이 필요한 호스트는 스토리지 ECDHE의 특정 호스트에 지정된 영역에 메시지를 씁니다. BOOM은 수신 이메일을 주기적으로 읽고 요청된 논리 볼륨 확장을

수행하며 발신 메일에 응답을 씁니다. 요청을 보낸 후 호스트는 들어오는 메일에서 2초마다 응답을 모니터링합니다. 호스트에서 논리 볼륨 확장 요청에 성공적으로 응답을 수신하면 장치 매핑에서 논리 볼륨 맵을 새로 할당하여 새로 할당된 스토리지를 인식합니다.

스토리지 풀에서 사용 가능한 물리적 스토리지가 거의 소모되면 리소스를 보충할 수 있는 방법으로 여러 이미지가 사용 가능한 스토리지가 부족해질 수 있습니다. 스토리지를 소모하는 스토리지 풀로 인해 QEMU에서 **enospc** 오류를 반환하는데, 이는 장치에 더 이상 사용 가능한 스토리지가 없음을 나타냅니다. 이 시점에는 실행 중인 가상 머신이 자동으로 일시 중지되며 볼륨 그룹에 새 LUN을 추가하는 데 수동 개입이 필요합니다.

새 LUN이 볼륨 그룹에 추가되면 스토리지 풀 관리자는 추가 스토리지를 필요한 논리 볼륨에 자동으로 배포합니다. 추가 리소스를 자동으로 할당하면 관련 가상 머신이 자동으로 작업이 중단되거나 중지된 경우 작업을 재개할 수 있습니다.

## 3장. 네트워크

### 3.1. 네트워크 아키텍처

Red Hat Virtualization 네트워크는 기본 네트워킹, 클러스터 내 네트워킹 및 호스트 네트워킹 구성의 관점에서 설명할 수 있습니다. 기본 네트워킹 용어에는 네트워킹을 용이하게 하는 기본 하드웨어 및 소프트웨어 요소가 포함됩니다. 클러스터 내의 네트워킹에는 호스트, 논리적 네트워크 및 가상 시스템과 같은 클러스터 수준 개체 간의 네트워크 상호 작용이 포함됩니다. 호스트 네트워킹 구성은 호스트 내의 네트워킹에 지원되는 구성을 다룹니다.

예를 들어, 잘 설계되고 구축된 네트워크를 통해 높은 대역폭 작업이 적절한 대역폭, 사용자 상호 작용이 대기 시간에 영향을 받지 않고 마이그레이션 도메인 내에서 가상 머신을 성공적으로 마이그레이션할 수 있습니다. 예기치 않게 구축된 네트워크는 예를 들어 대기 시간 및 네트워크 플러드로 인한 마이그레이션 및 복제 실패가 발생할 수 있습니다.

### 3.2. 소개: 기본 네트워킹 약관

Red Hat Virtualization은 다음을 사용하여 가상 머신, 가상화 호스트 및 광범위한 네트워크 간의 네트워킹 기능을 제공합니다.

- NIC(네트워크 인터페이스 컨트롤러)
- 브리지
- 관련 리뷰: A Bond
- A Virtual NIC
- 가상 LAN(VLAN)

NIC, 브리지 및 VNIC는 호스트, 가상 머신, 로컬 영역 네트워크 및 인터넷 간의 네트워크 통신을 허용합니다. 본딩 및 VLAN은 보안, 내결함성 및 네트워크 용량을 개선하기 위해 선택적으로 구현됩니다.

### 3.3. 네트워크 인터페이스 컨트롤러

*NIC(Network Interface Controller)* 는 컴퓨터를 컴퓨터 네트워크에 연결하는 네트워크 어댑터 또는 LAN 어댑터입니다. NIC는 시스템의 물리적 링크 계층과 데이터 링크 계층 모두에서 작동하며 네트워크 연결을 허용합니다. Red Hat Virtualization 환경의 모든 가상화 호스트에는 NIC가 하나 이상 있지만 호스트에 두 개 이상의 NIC가 있는 것이 더 일반적입니다.

하나의 물리적 NIC에는 논리적으로 연결된 여러 개의 가상 NIC(VNIC)가 있을 수 있습니다. 가상 NIC는 가상 머신의 물리적 네트워크 인터페이스 역할을 합니다. VNIC와 NIC를 구분하기 위해 Red Hat Virtualization Manager는 각 VNIC에 고유한 MAC 주소를 할당합니다.

### 3.4. BRIDGE

*브리지* 는 패킷 전환 네트워크에서 패킷 전달을 사용하는 소프트웨어 장치입니다. 브리징을 사용하면 여러 네트워크 인터페이스 장치가 하나의 NIC 연결을 공유하고 별도의 물리적 장치로 네트워크에 표시될 수 있습니다. 브릿지는 패킷의 소스 주소를 검사하여 관련 대상 주소를 결정합니다. 대상 주소가 결정되면 브릿지는 향후 참조를 위해 위치를 테이블에 추가합니다. 이를 통해 호스트는 네트워크 트래픽을 브리지의 멤버인 가상 머신 관련 VNIC로 리디렉션할 수 있습니다.

Red Hat Virtualization에서는 브리지를 사용하여 논리적 네트워크를 구현합니다. IP 주소를 수신하는 호스트의 물리적 인터페이스가 아닌 브릿지입니다. 브릿지와 연결된 IP 주소는 연결에 브리지를 사용하는



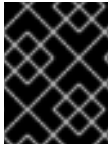
가상 시스템과 동일한 서브넷에 있을 필요가 없습니다. 브릿지에 사용하는 가상 시스템과 동일한 서브넷에 있는 IP 주소가 할당되면 호스트는 논리 네트워크 내에서 가상 시스템에 의해 주소를 지정할 수 있습니다. 일반적으로 가상화 호스트에서 네트워크 노출된 서비스를 실행하지 않는 것이 좋습니다. 게스트는 VNIC에 의해 논리 네트워크에 연결되어 있으며 호스트는 NIC를 사용하여 논리 네트워크의 원격 요소에 연결됩니다. 각 게스트는 DHCP 또는 정적으로 설정된 VNIC의 IP 주소를 독립적으로 가질 수 있습니다. 브릿지는 호스트 외부의 오브젝트에 연결할 수 있지만 이러한 연결은 필수가 아닙니다.

브리지 및 이더넷 연결에 대해 사용자 지정 속성을 정의할 수 있습니다. VDSM은 네트워크 정의 및 사용자 지정 속성을 설정 네트워크 후크 스크립트에 전달합니다.

### 3.5. 본딩

본딩은 여러 네트워크 인터페이스 카드를 단일 소프트웨어 정의 장치에 집계하는 것입니다. 결합된 네트워크 인터페이스는 본딩에 포함된 네트워크 인터페이스 카드의 전송 기능을 결합하여 단일 네트워크 인터페이스로 작동하므로 단일 네트워크 인터페이스 카드보다 더 큰 전송 속도를 제공할 수 있습니다. 또한 본딩 자체의 모든 네트워크 인터페이스 카드가 실패해야 하므로 본딩을 통해 내결함성이 향상됩니다. 그러나 본딩된 네트워크 인터페이스를 구성하는 네트워크 인터페이스 카드는 본딩의 모든 네트워크 인터페이스 카드가 동일한 옵션과 모드를 지원하도록 동일한 make 및 model이어야 한다는 것입니다.

본딩의 패킷 분산 알고리즘은 사용된 본딩 모드에 따라 결정됩니다.



#### 중요

모드 1, 2, 3 및 4는 가상 머신 (bridged) 및 비가상 머신 (bridgeless) 네트워크 유형을 모두 지원합니다. 모드 0, 5 및 6은 비가상 머신(bridgeless) 네트워크만 지원합니다.

#### 본딩 모드

Red Hat Virtualization에서는 기본적으로 모드 4를 사용하지만 다음과 같은 일반적인 본딩 모드를 지원합니다.

##### 모드 0 (round-robin policy)

네트워크 인터페이스 카드를 순차순으로 전송합니다. 패킷은 본딩에서 사용 가능한 첫 번째 네트워크 인터페이스 카드로 시작하고 본딩에서 사용 가능한 마지막 네트워크 인터페이스 카드로 끝나는 루프로 전송됩니다. 이후의 모든 루프는 사용 가능한 첫 번째 네트워크 인터페이스 카드로 시작합니다. 모드 0은 내결함성을 제공하며 본딩의 모든 네트워크 인터페이스 카드의 부하를 조정합니다. 그러나 모드 0은 브리지와 함께 사용할 수 없으므로 가상 머신 논리 네트워크와 호환되지 않습니다.

##### 모드 1 (active-backup 정책)

하나의 네트워크 인터페이스 카드가 활성 상태로 유지되는 동안 모든 네트워크 인터페이스 카드를 백업 상태로 설정합니다. 활성 네트워크 인터페이스 카드의 오류가 발생하는 경우 백업 네트워크 인터페이스 카드 중 하나가 해당 네트워크 인터페이스 카드를 본딩에서 유일한 활성 네트워크 인터페이스 카드로 대체합니다. 모드 1의 MAC 주소는 활성 네트워크 인터페이스 카드의 MAC 주소를 반영하도록 본딩의 MAC 주소가 변경된 경우 발생할 수 있는 혼동을 방지하기 위해 하나의 포트에만 표시됩니다. 모드 1은 내결함성을 제공하며 Red Hat Virtualization에서 지원됩니다.

##### 모드 2 (XOR 정책)

소스 및 대상 MAC 주소 modulo 네트워크 인터페이스 카드 슬레이브 수에 대한 XOR 작업 결과에 따라 패킷을 전송할 네트워크 인터페이스 카드를 선택합니다. 이 계산을 통해 사용되는 각 대상 MAC 주소에 대해 동일한 네트워크 인터페이스 카드가 선택됩니다. 모드 2는 내결함성과 로드 밸런싱을 제공하며 Red Hat Virtualization에서 지원됩니다.

##### 모드 3 (브로드캐스트 정책)

모든 패킷을 모든 네트워크 인터페이스 카드에 전송합니다. 모드 3은 내결함성을 제공하며 Red Hat Virtualization에서 지원됩니다.

#### 모드 4 (IEEE 802.3ad 정책)

인터페이스가 동일한 속도와 듀플렉스 설정을 공유하는 집계 그룹을 만듭니다. 모드 4는 IEEE 802.3ad 사양에 따라 활성 집계 그룹의 모든 네트워크 인터페이스 카드를 사용하며 Red Hat Virtualization에서 지원됩니다.

#### 모드 5 (Active 전송 로드 밸런싱 정책)

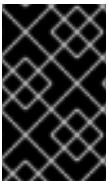
본딩의 각 네트워크 인터페이스 카드에 대한 발신 트래픽 계정을 배포하고 현재 네트워크 인터페이스 카드가 들어오는 모든 트래픽을 수신하는지 확인합니다. 트래픽을 수신하도록 할당된 네트워크 인터페이스 카드가 실패하면 수신되는 트래픽을 수신하는 역할에 다른 네트워크 인터페이스 카드가 할당됩니다. 모드 5는 브리지와 함께 사용할 수 없으므로 가상 시스템 논리 네트워크와 호환되지 않습니다.

#### 모드 6 (Active 로드 밸런싱 정책)

모드 5(Active 전송 로드 밸런싱 정책)를 특정 스위치 요구 사항 없이 IPv4 트래픽에 대한 수신 부하 분산과 결합합니다. ARP 협상은 수신 로드의 균형을 조정하는 데 사용됩니다. 모드 6은 브리지와 함께 사용할 수 없으므로 가상 시스템 논리 네트워크와 호환되지 않습니다.

### 3.6. 본딩을 위한 스위치 구성

스위치 구성은 하드웨어 요구 사항에 따라 다릅니다. 운영 체제에 대한 배포 및 네트워킹 구성 가이드를 참조하십시오.



#### 중요

모든 유형의 스위치에 대해 CiscoPAGP( *Port Aggregation Protocol* ) 프로토콜이 아닌 LACP( *Link Aggregation Control Protocol* ) 프로토콜과 결합된 스위치를 설정하는 것이 중요합니다.

### 3.7. 가상 네트워크 인터페이스 카드

가상 네트워크 인터페이스 카드는 호스트의 물리적 네트워크 인터페이스 카드를 기반으로 하는 가상 네트워크 인터페이스입니다. 각 호스트에는 여러 네트워크 인터페이스 카드가 있을 수 있으며 각 네트워크 인터페이스 카드는 여러 가상 네트워크 인터페이스 카드의 기반이 될 수 있습니다.

가상 네트워크 인터페이스 카드를 가상 머신에 연결하면 Red Hat Virtualization Manager는 가상 네트워크 인터페이스 카드가 연결된 가상 머신, 가상 네트워크 인터페이스 카드 자체 및 가상 네트워크 인터페이스 카드가 기반으로 하는 물리적 호스트 네트워크 인터페이스 카드 간에 여러 연결을 생성합니다. 특히 가상 네트워크 인터페이스 카드를 가상 머신에 연결하면 가상 네트워크 인터페이스 카드가 기반이 되는 물리적 호스트 네트워크 인터페이스 카드에 새 가상 네트워크 인터페이스 카드와 MAC 주소가 생성됩니다. 그런 다음 해당 가상 네트워크 인터페이스 카드를 연결한 후 가상 시스템을 처음 시작하면 **libvirt** 에서 가상 네트워크 인터페이스 카드에 PCI 주소를 할당합니다. 그런 다음 MAC 주소와 PCI 주소는 가상 머신에서 가상 네트워크 인터페이스 카드(예: **eth0**)의 이름을 가져오는 데 사용됩니다.

MAC 주소를 할당하고 해당 MAC 주소를 PCI 주소와 연결하는 프로세스는 템플릿 또는 스냅샷을 기반으로 가상 머신을 생성할 때 약간 다릅니다. PCI 주소가 템플릿 또는 스냅샷에 대해 이미 생성된 경우 해당 템플릿 또는 스냅샷을 기반으로 생성된 가상 시스템의 가상 네트워크 인터페이스 카드가 해당 순서로 할당된 PCI 주소 및 MAC 주소에 따라 정렬됩니다. PCI 주소가 템플릿에 대해 생성되지 않은 경우 해당 템플릿을 기반으로 생성된 가상 머신의 가상 네트워크 인터페이스 카드가 가상 네트워크 인터페이스 카드 이

를 지정 순서에 따라 할당됩니다. PCI 주소가 스냅샷에 대해 생성되지 않은 경우 Red Hat Virtualization Manager는 해당 스냅샷을 기반으로 가상 머신의 가상 네트워크 인터페이스 카드에 새 MAC 주소를 할당합니다.

가상 네트워크 인터페이스 카드가 생성되면 네트워크 브리지 장치에 추가됩니다. 네트워크 브리지 장치는 가상 시스템이 가상 시스템 논리적 네트워크에 연결되는 방법입니다.

가상화 호스트에서 **ip addr show** 명령을 실행하면 해당 호스트의 가상 머신에 연결된 모든 가상 네트워크 인터페이스 카드가 표시됩니다. 또한 논리적 네트워크를 백업하기 위해 생성된 모든 네트워크 브리지와 호스트에서 사용하는 모든 네트워크 인터페이스 카드도 표시됩니다.

```
[root@rhev-host-01 ~]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:21:86:a2:85:cd brd ff:ff:ff:ff:ff:ff
    inet6 fe80::221:86ff:fea2:85cd/64 scope link
        valid_lft forever preferred_lft forever
3: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN qlen
1000
    link/ether 00:21:6b:cc:14:6c brd ff:ff:ff:ff:ff:ff
5: ;vdsmdummy,: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN
    link/ether 4a:d5:52:c2:7f:4b brd ff:ff:ff:ff:ff:ff
6: bond0: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
7: bond4: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
8: bond1: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
9: bond2: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
10: bond3: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
11: ovirtmgmt: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UNKNOWN
    link/ether 00:21:86:a2:85:cd brd ff:ff:ff:ff:ff:ff
    inet 10.64.32.134/23 brd 10.64.33.255 scope global ovirtmgmt
    inet6 fe80::221:86ff:fea2:85cd/64 scope link
        valid_lft forever preferred_lft forever
```

명령의 콘솔 출력은 하나의 루프 백 장치(**lo**), 하나의 이더넷 장치(**eth0**), 하나의 무선 장치(**wlan0**), 1개의 VDSM 더미 장치(**;vdsmdummy;**), 5개의 본딩 장치(**bond0,bond4,bond1, bond2, bond2**), 하나의 네트워크 브리지(**ovirtmgmt**)를 표시합니다.

가상 네트워크 인터페이스 카드는 네트워크 브리지 장치 및 논리 네트워크의 모든 멤버입니다. **brctl show** 명령을 사용하여 브리지 멤버십을 표시할 수 있습니다.

```
[root@rhev-host-01 ~]# brctl show
bridge name bridge id STP enabled interfaces
ovirtmgmt 8000.e41f13b7fdd4 no vnet002
    vnet001
    vnet000
    eth0
```

`brctl show` 명령의 콘솔 출력은 `virtio` 가상 네트워크 인터페이스 카드가 `ovirtmgmt` 브리지의 멤버임을 보여줍니다. 가상 네트워크 인터페이스 카드가 연결된 모든 가상 머신은 `ovirtmgmt` 논리적 네트워크에 연결됩니다. `eth0` 네트워크 인터페이스 카드는 `ovirtmgmt` 브리지의 멤버이기도 합니다. `eth0` 장치는 호스트 이외의 연결을 제공하는 스위치로 연결됩니다.

### 3.8. 가상 LAN(VLAN)

**VLAN(Virtual LAN)**은 네트워크 패킷에 적용할 수 있는 속성입니다. 네트워크 패킷은 번호가 지정된 VLAN으로 "tagged"될 수 있습니다. VLAN은 스위치 수준에서 네트워크 트래픽을 완전히 분리하는 데 사용되는 보안 기능입니다. VLAN은 완전히 분리되어 함께 사용할 수 없습니다. **Red Hat Virtualization Manager**는 VLAN을 인식하고 VLAN 트래픽을 태그하고 리디렉션할 수 있지만 VLAN 구현에서는 VLAN을 지원하는 스위치가 필요합니다.

스위치 수준에서 포트에는 VLAN 지정이 할당됩니다. 스위치는 특정 포트에서 시작된 트래픽에 VLAN 태그를 적용하여 트래픽을 VLAN의 일부로 표시하고 응답이 동일한 VLAN 태그를 갖도록 합니다. VLAN은 여러 스위치에서 확장할 수 있습니다. 스위치에서 태그된 VLAN 네트워크 트래픽은 올바른 VLAN으로 지정된 포트에 연결된 시스템을 제외하고 완전히 감지할 수 없습니다. 지정된 포트를 여러 VLAN에 태그를 지정하여 여러 VLAN에서 단일 포트로 전송되는 트래픽을 단일 포트로 보낼 수 있습니다. 이 경우 트래픽을 수신하는 시스템의 소프트웨어를 사용하여 암호화를 해제할 수 있습니다.

### 3.9. 네트워크 라벨

네트워크 레이블을 사용하면 논리 네트워크를 생성 및 제공하고 해당 논리 네트워크를 물리적 호스트 네트워크 인터페이스 및 본딩과 연결하는 것과 관련된 여러 관리 작업을 크게 단순화할 수 있습니다.

네트워크 레이블은 사람이 읽을 수 있는 일반 텍스트로, 논리 네트워크 또는 물리적 호스트 네트워크 인터페이스에 연결할 수 있습니다. 레이블 길이에 엄격한 제한은 없지만 소문자와 대문자, 밑줄 및 하이픈의 조합을 사용해야 합니다. 공백이나 특수 문자는 허용되지 않습니다.

논리 네트워크 또는 물리적 호스트 네트워크 인터페이스에 레이블을 연결하면 다음과 같이 동일한 레이블이 연결된 다른 논리 네트워크 또는 물리적 호스트 네트워크 인터페이스와 연결됩니다.

## 네트워크 라벨"

- 논리 네트워크에 레이블을 연결하면 해당 논리 네트워크가 지정된 레이블과 물리적 호스트 네트워크 인터페이스와 자동으로 연결됩니다.
- 물리적 호스트 네트워크 인터페이스에 레이블을 연결하면 지정된 레이블이 있는 모든 논리적 네트워크가 해당 물리적 호스트 네트워크 인터페이스와 자동으로 연결됩니다.
- 논리 네트워크 또는 물리적 호스트 네트워크 인터페이스에 연결된 레이블을 변경하면 레이블을 제거하고 새 레이블을 추가하는 것과 동일한 방식으로 작동합니다. 관련 논리적 네트워크 또는 물리적 호스트 네트워크 인터페이스 간의 연결이 업데이트됩니다.

## 네트워크 라벨 및 클러스터

- 레이블이 지정된 논리 네트워크가 클러스터에 추가되고 해당 클러스터에 동일한 레이블이 있는 물리적 호스트 네트워크 인터페이스가 있으면 논리 네트워크가 해당 물리적 호스트 네트워크 인터페이스에 자동으로 추가됩니다.
- 레이블이 지정된 논리 네트워크가 클러스터에서 분리되고 해당 클러스터에 동일한 레이블이 있는 물리적 호스트 네트워크 인터페이스가 있으면 논리 네트워크가 해당 물리적 호스트 네트워크 인터페이스에서 자동으로 분리됩니다.

## 역할을 사용하는 네트워크 라벨 및 논리 네트워크

- 레이블이 지정된 논리 네트워크가 디스플레이 네트워크 또는 마이그레이션 네트워크로 작동하도록 할당되면 해당 논리 네트워크가 **DHCP** 를 사용하여 물리적 호스트 네트워크 인터페이스에 구성되므로 논리적 네트워크에 **IP** 주소가 할당될 수 있습니다.

역할 네트워크(예: "마이그레이션 네트워크" 또는 "포시 네트워크")에 레이블을 설정하면 모든 호스트에 해당 네트워크를 대량으로 배포합니다. 이러한 네트워크 추가는 **DHCP**를 사용하여 수행됩니다. 이 대량 배포 방법은 많은 고정 **IP** 주소 입력 작업의 확장 불가능한 특성으로 인해 정적 주소 입력 방법을 통해 선택되었습니다.

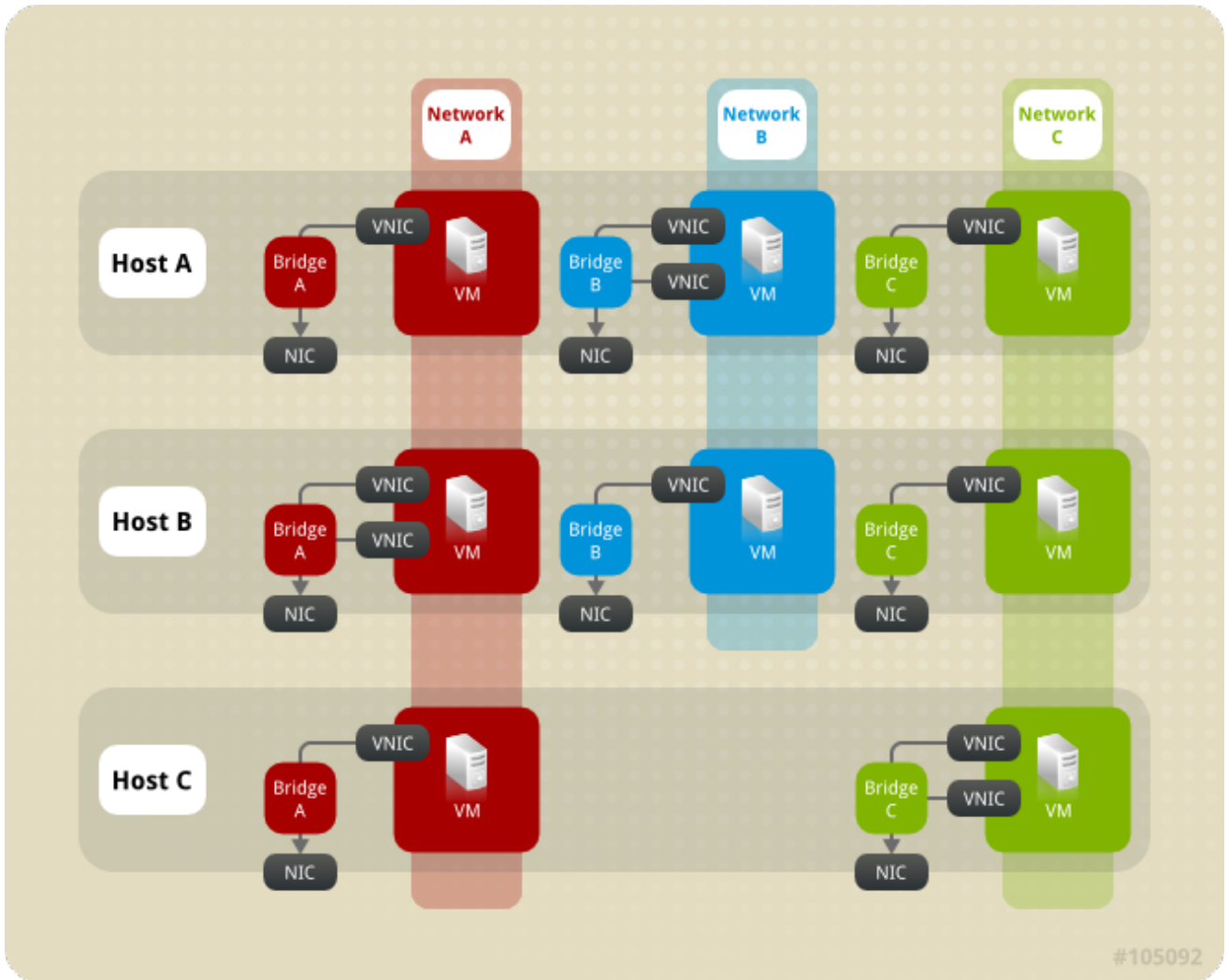
### 3.10. 클러스터 네트워킹

클러스터 수준 네트워킹 오브젝트는 다음과 같습니다.

- 클러스터

• 논리 네트워크

그림 3.1. 클러스터 내의 네트워킹



데이터센터는 여러 클러스터의 논리적 그룹이며 각 클러스터는 여러 호스트의 논리 그룹입니다. 그림 3.1. “클러스터 내의 네트워킹” 단일 클러스터의 콘텐츠를 표시합니다.

클러스터의 호스트는 모두 동일한 스토리지 도메인에 액세스할 수 있습니다. 클러스터의 호스트에도 클러스터 수준에서 적용되는 논리 네트워크도 있습니다. 가상 머신의 논리적 네트워크가 작동하도록 하려면 Red Hat Virtualization Manager를 사용하여 클러스터의 각 호스트에 대해 네트워크를 정의하고 구현해야 합니다. 다른 논리적 네트워크 유형은 이를 사용하는 호스트에서만 구현할 수 있습니다.

다중 호스트 네트워크 구성은 네트워크가 할당된 데이터 센터 내의 모든 호스트에 업데이트된 네트워크 설정을 자동으로 적용합니다.

3.11. 논리 네트워크

논리 네트워킹을 사용하면 **Red Hat Virtualization** 환경에서 유형별로 네트워크 트래픽을 분리할 수 있습니다. 예를 들어 **ovirtmgmt** 네트워크는 **Manager**와 호스트 간의 관리 통신에 사용할 **Red Hat Virtualization**을 설치하는 동안 기본적으로 생성됩니다. 일반적으로 논리적 네트워크에 사용하는 것은 유사한 요구 사항 및 사용으로 네트워크 트래픽을 그룹화하는 것입니다. 대부분의 경우 관리자가 스토리지 네트워크 및 디스플레이 네트워크를 생성하여 최적화 및 문제 해결을 위해 각 유형의 트래픽을 분리합니다.

논리적 네트워크의 유형은 다음과 같습니다.

- 가상 머신 네트워크 트래픽을 전송하는 논리적 네트워크
- 가상 시스템 네트워크 트래픽을 전송하지 않는 논리 네트워크
- 선택적 논리적 네트워크
- 및 필수 네트워크

모든 논리 네트워크는 필수 또는 선택 사항일 수 있습니다.

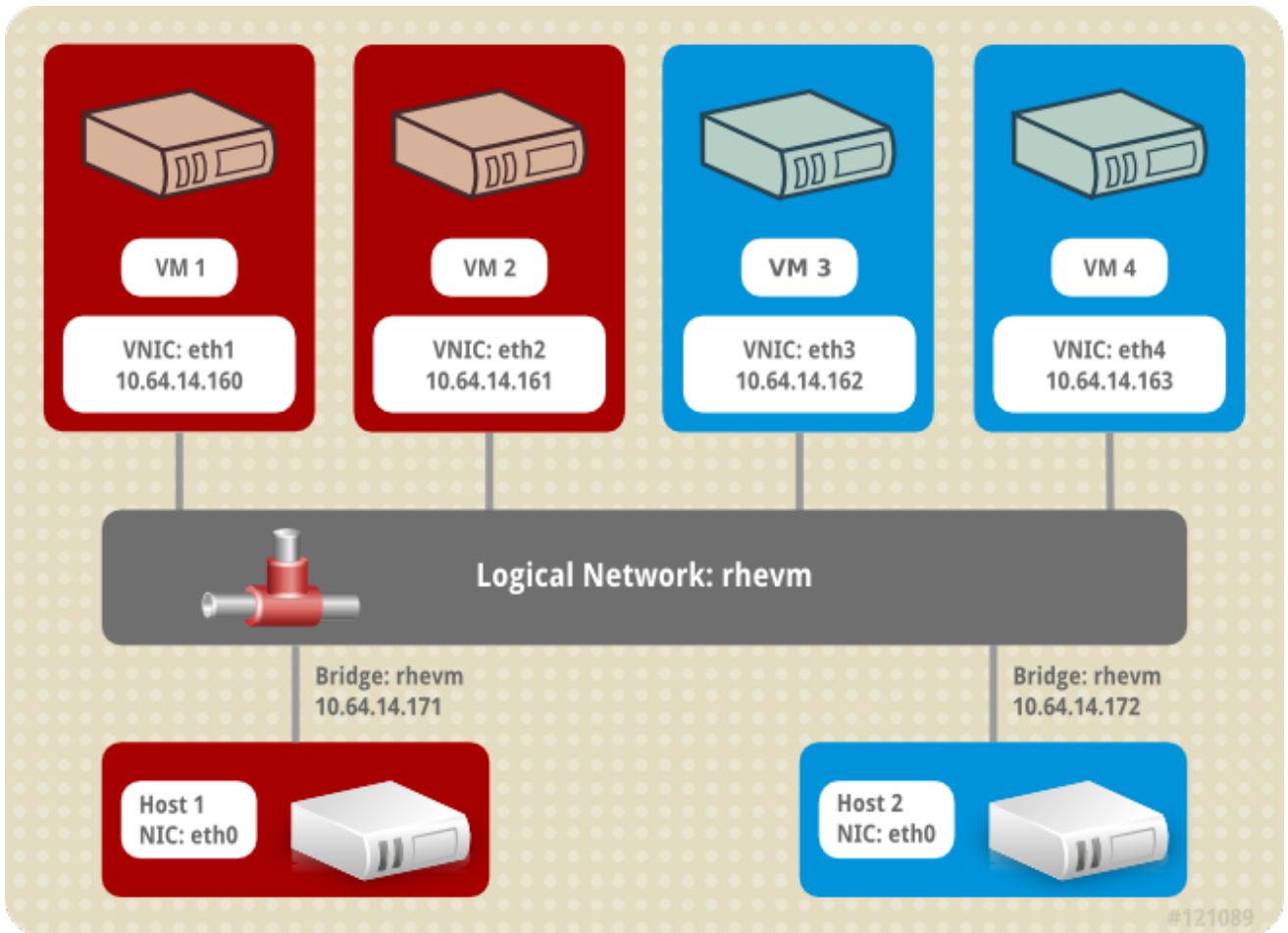
논리 네트워크는 데이터 센터 수준에서 정의되며 호스트에 추가됩니다. 필수 논리적 네트워크가 작동하려면 지정된 클러스터의 모든 호스트에 대해 이 네트워크를 구현해야 합니다.

**Red Hat Virtualization** 환경의 각 가상 머신 논리적 네트워크는 호스트의 네트워크 브리지 장치에서 지원됩니다. 따라서 클러스터에 대해 새 가상 시스템 논리적 네트워크를 정의할 때 논리적 네트워크가 가상 머신에서 사용되도록 하려면 클러스터의 각 호스트에서 일치하는 브리지 장치를 생성해야 합니다. **Red Hat Virtualization Manager**는 가상 머신 논리 네트워크에 필요한 브릿지를 자동으로 생성합니다.

가상 시스템 논리 네트워크를 백업하기 위해 **Red Hat Virtualization Manager**에서 생성한 브리지 장치는 호스트 네트워크 인터페이스와 연결됩니다. 브리지의 일부인 호스트 네트워크 인터페이스에 네트워크 연결이 있는 경우 나중에 브리지에 포함된 모든 네트워크 인터페이스가 브리지의 네트워크 연결을 공유합니다. 가상 머신이 생성되어 특정 논리 네트워크에 배치되면 해당 가상 네트워크 카드가 해당 논리 네트워크의 브리지에 포함됩니다. 그런 다음 해당 가상 머신은 서로 그리고 브리지에 연결된 다른 오브젝트와 통신할 수 있습니다.

가상 시스템 네트워크 트래픽에 사용되지 않는 논리적 네트워크는 호스트 네트워크 인터페이스와 직접 연결됩니다.

그림 3.2. ovirtmgmt 논리적 네트워크입니다.



예 3.1. 논리 네트워크의 사용 예.

Purple이라는 데이터 센터의 Pink라는 클러스터에는 Red 및 White라는 두 개의 호스트가 있습니다. Red 및 White 모두 모든 네트워킹 기능에 대해 기본 논리 네트워크인 ovirtmgmt 를 사용하고 있습니다. Pink를 담당하는 시스템 관리자는 웹 서버와 일부 클라이언트 가상 머신을 별도의 논리적 네트워크에 배치하여 웹 서버에 대한 네트워크 테스트를 분리하기로 결정합니다. 새 논리적 네트워크 network\_testing 이라고 합니다.

먼저 Purple 데이터 센터의 논리적 네트워크를 정의합니다. 그런 다음 사용자 이름을 Pink 클러스터에 적용합니다. 논리 네트워크는 유지 관리 모드에서 호스트에서 구현되어야 합니다. 따라서 관리자는 먼저 실행 중인 모든 가상 머신을 Red Hat으로 마이그레이션하고 유지 관리 모드로 White를 배치합니다. 그런 다음 브리지에 포함될 물리적 네트워크 인터페이스와 연결된 네트워크를 편집합니다. 선택한 네트워크 인터페이스에 대한 링크 상태가 Down 에서 Non-Operational 로 변경됩니다. Pink 클러스터의 각 호스트에 있는 물리적 네트워크 인터페이스를 network\_testing 네트워크에 추가하여 해당 브릿지를 클러스터의 모든 호스트에서 설정해야 하기 때문입니다. 다음으로 화이트를 활성화하고 실행 중인 모든 가상 머신을 Red Hat에서 마이그레이션한 후 Red Hat의 프로세스를 반복합니다.

White와 Red 둘 다 network\_testing 논리 네트워크를 물리적 네트워크 인터페이스에 브리지하는 경우 network\_testing 논리 네트워크가 operation이 되어 가상 시스템에서 사용할 수 있습니다.



### 3.12. 필수 네트워크, 선택적 네트워크 및 가상 머신 네트워크

필수 네트워크는 클러스터의 모든 호스트에서 사용할 수 있어야 하는 논리적 네트워크입니다. 호스트의 필수 네트워크가 작동하지 않게 되면 해당 호스트에서 실행 중인 가상 시스템이 다른 호스트로 마이그레이션됩니다. 이 마이그레이션의 범위는 선택한 스케줄링 정책에 따라 달라집니다. 이는 미션 크리티컬 워크로드를 실행하는 가상 머신이 있는 경우 유용합니다.

선택적 네트워크는 **Required** 로 명시적으로 선언되지 않은 논리 네트워크입니다. 선택적 네트워크는 해당 네트워크를 사용하는 호스트에서만 구현할 수 있습니다. 선택적 네트워크의 존재 여부 또는 부재는 호스트의 운영 상태에 영향을 미치지 않습니다. 필수가 아닌 네트워크가 작동하지 않게 되면 네트워크에서 실행 중인 가상 시스템이 다른 호스트로 마이그레이션되지 않습니다. 이렇게 하면 대량 마이그레이션으로 인한 불필요한 I/O 과부하가 발생하지 않습니다. 논리 네트워크가 생성되어 클러스터에 추가되면 기본적으로 필수 박스가 선택됩니다.

네트워크의 필수 지정을 변경하려면 관리 포털에서 네트워크를 선택하고 클러스터 탭을 클릭하고 **Manage Networks** (네트워크 관리) 버튼을 클릭합니다.

가상 머신 네트워크(사용자 인터페이스의 **VM 네트워크** 라고 함)는 가상 머신 네트워크 트래픽만 전달하도록 지정된 논리 네트워크입니다. 가상 머신 네트워크는 필수 또는 선택 사항일 수 있습니다. 선택적 가상 머신 네트워크를 사용하는 가상 머신은 해당 네트워크가 있는 호스트에서만 시작됩니다.

### 3.13. 가상 머신 연결

**Red Hat Virtualization**에서 가상 머신의 **NIC**는 가상 머신이 생성될 때 논리 네트워크에 배치됩니다. 이 시점에서 가상 머신은 동일한 네트워크의 다른 대상과 통신할 수 있습니다.

호스트 관점에서 가상 시스템이 논리 네트워크에 배치되면 가상 시스템의 **NIC**를 지원하는 **VNIC**가 논리 네트워크의 브리지 장치에 멤버로 추가됩니다. 예를 들어 가상 시스템이 **ovirtmgmt** 논리 네트워크에 있는 경우 해당 가상 시스템이 실행되는 호스트의 **ovirtmgmt** 브리지 멤버로 **VNIC**가 추가됩니다.

### 3.14. 포트 미러링

포트 미러링은 지정된 논리적 네트워크 및 호스트의 3 계층 네트워크 트래픽을 가상 시스템의 가상 인터페이스에 복사합니다. 이 가상 머신은 네트워크 디버깅 및 튜닝, 침입 탐지 및 동일한 호스트 및 논리적 네트워크에서 다른 가상 시스템의 동작을 모니터링하는 데 사용할 수 있습니다.

복사된 유일한 트래픽은 한 호스트에서 하나의 논리적 네트워크에 있습니다. 호스트 외부의 네트워크에서 트래픽을 늘리지 않지만, 포트 미러링이 활성화된 가상 머신은 다른 가상 머신보다 더 많은 호스트 **CPU**와 **RAM**을 사용합니다.

논리 네트워크의 vNIC 프로필에서 포트 미러링이 활성화되거나 비활성화되어 있으며 다음과 같은 제한 사항이 있습니다.

- 포트 미러링이 활성화된 프로필로 vNIC를 핫플러그하는 것은 지원되지 않습니다.
- vNIC 프로필이 가상 머신에 연결된 경우 포트 미러링을 변경할 수 없습니다.

위의 제한 사항에 따라 추가 전용 vNIC 프로필에서 포트 미러링을 활성화하는 것이 좋습니다.



**중요**

포트 미러링을 사용하면 다른 네트워크 사용자의 개인 정보가 줄어듭니다.

### 3.15. 호스트 네트워킹 구성

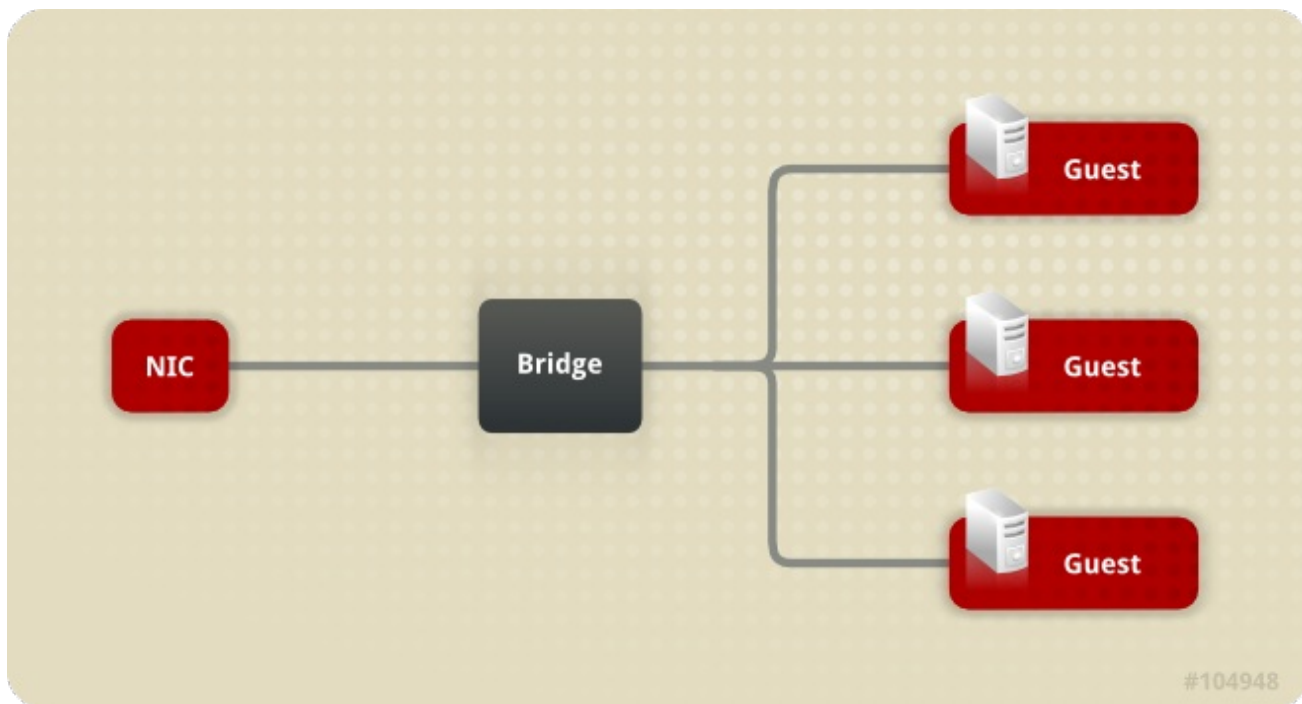
가상화 호스트의 일반적인 유형의 네트워킹 구성은 다음과 같습니다.

- 브릿지 및 NIC 구성.
- 브리지, VLAN 및 NIC 구성입니다.
- 브리지, 본딩 및 VLAN 구성.
- 다중 브릿지, 여러 VLAN, NIC 구성

### 3.16. 브릿지 설정

Red Hat Virtualization에서 가장 간단한 호스트 구성은 Bridge 및 NIC 설정입니다. [그림 3.3. “브릿지 및 NIC 구성”](#)에서 볼 수 있듯이 이 구성은 브리지를 사용하여 하나 이상의 가상 머신(또는 게스트)을 호스트의 NIC에 연결합니다.

그림 3.3. 브릿지 및 NIC 구성

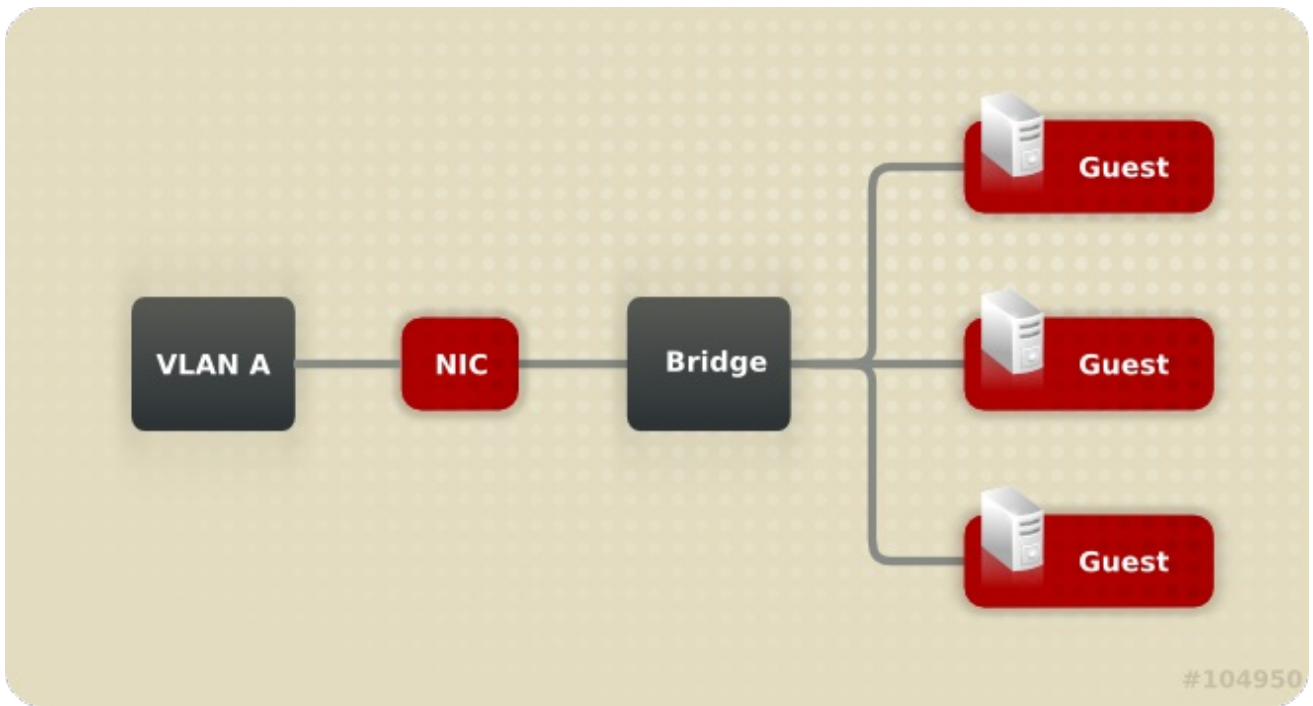


이 구성의 예로는 **Red Hat Virtualization Manager**가 설치될 때 **ovirtmgmt** 브릿지 자동 생성이 있습니다. 설치 시 **Red Hat Virtualization Manager**는 호스트에 **VDSM** 을 설치합니다. **VDSM** 설치 프로세스는 **ovirtmgmt** 브리지를 생성합니다. **ovirtmgmt** 브리지는 호스트의 **IP** 주소를 가져와 호스트에 대한 관리 통신을 활성화합니다.

### 3.17. VLAN 설정

그림 3.4. “브리지, VLAN 및 NIC 구성” 호스트 NIC 및 브릿지를 연결하는 VLAN(가상 LAN)이 포함된 대체 구성을 보여줍니다.

그림 3.4. 브리지, VLAN 및 NIC 구성

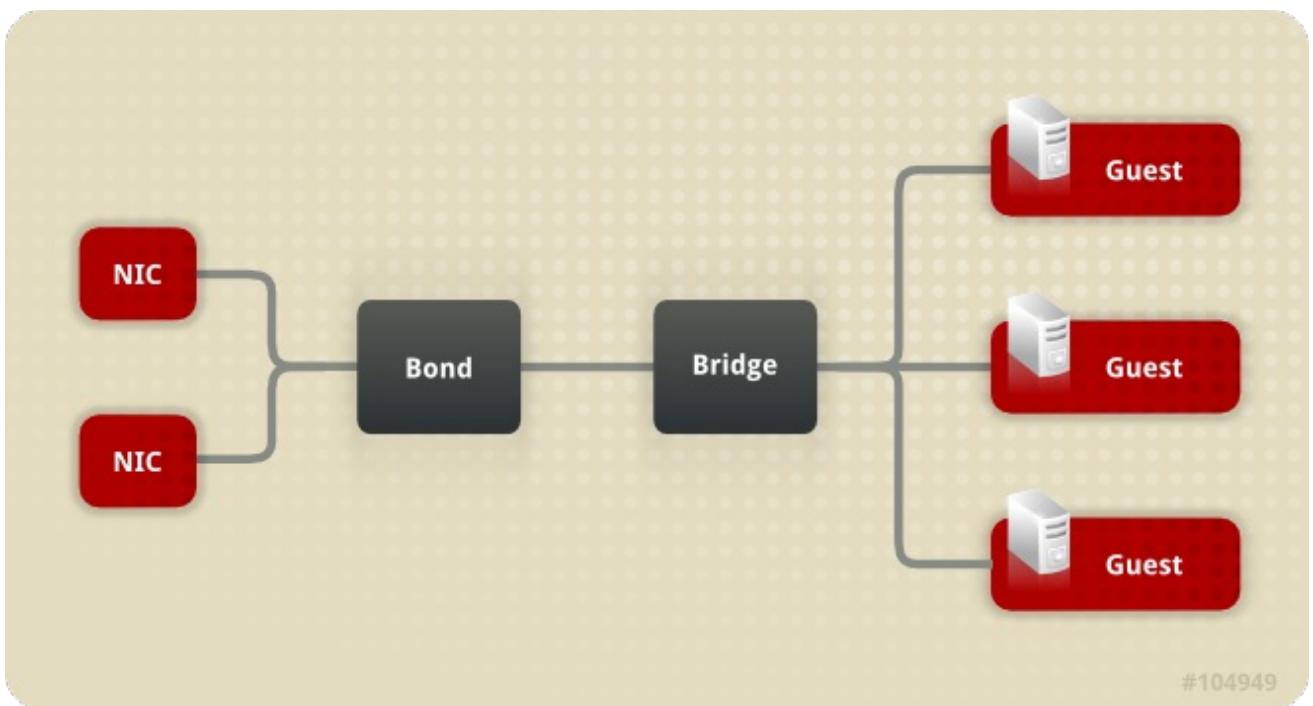


이 네트워크를 통해 데이터 전송을 위한 보안 채널을 제공하고 여러 VLAN을 사용하여 여러 브릿지를 단일 NIC에 연결하는 옵션도 지원하는 VLAN이 포함되어 있습니다.

### 3.18. 브릿지 및 본딩 설정

그림 3.5. “Bridge, Bond 및 NIC 설정” 여러 호스트 NIC를 동일한 브릿지 및 네트워크에 연결하는 본딩이 포함된 구성을 표시합니다.

그림 3.5. Bridge, Bond 및 NIC 설정

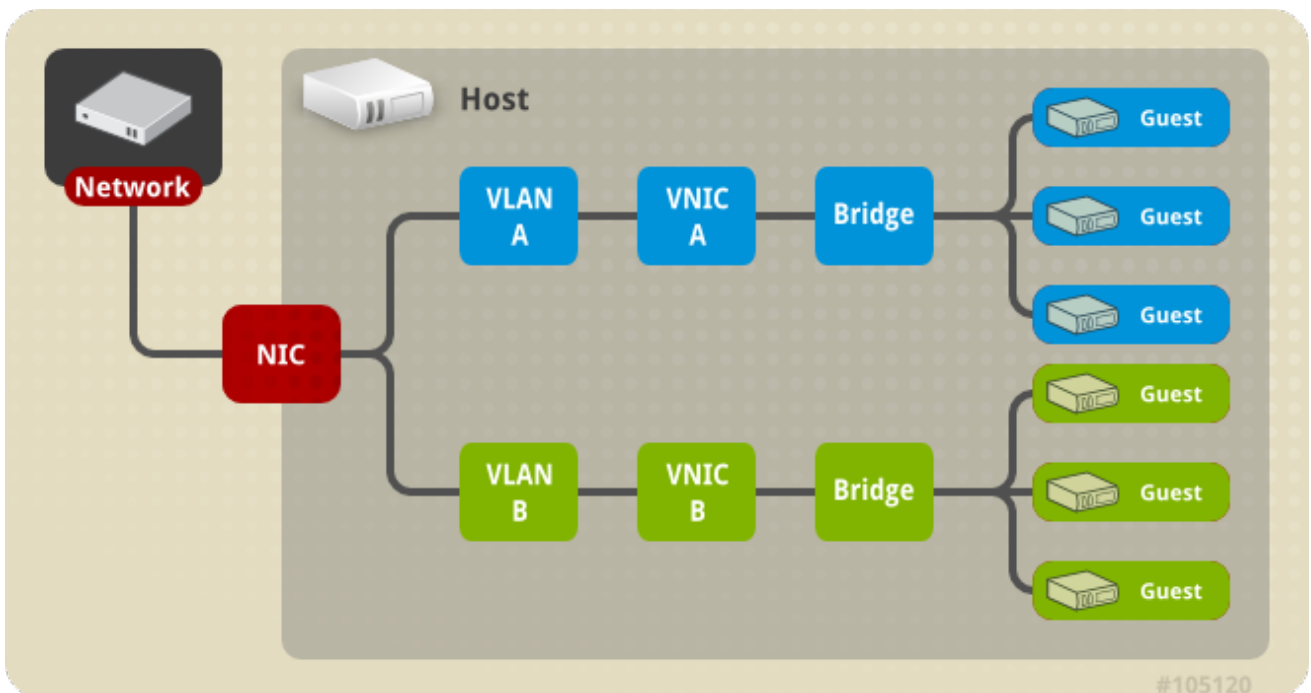


포함된 본딩은 둘 이상의 물리적 이더넷 링크를 결합하는 논리 링크를 생성합니다. 결과적으로 NIC 내 결합성 및 잠재적 대역폭 확장은 본딩 모드에 따라 달라집니다.

### 3.19. 다중 브리지, 다중 VLAN 및 NIC 설정

그림 3.6. “다중 브리지, 다중 VLAN 및 NIC 구성” 단일 NIC를 두 개의 VLAN에 연결하는 구성을 보여줍니다. 이렇게 하면 네트워크 스위치가 두 개의 VLAN 중 하나로 태그된 네트워크 트래픽을 호스트의 하나의 NIC로 전달하도록 구성되어 있다고 가정합니다. 호스트는 두 개의 VNIC를 사용하여 각 VLAN에 하나씩 VLAN 트래픽을 구분합니다. 그런 다음 두 VLAN에 태그된 트래픽이 적절한 VNIC를 브리지 멤버로 지정하여 별도의 브리지에 연결됩니다. 각 브릿지는 차례로 여러 가상 머신에 의해 연결됩니다.

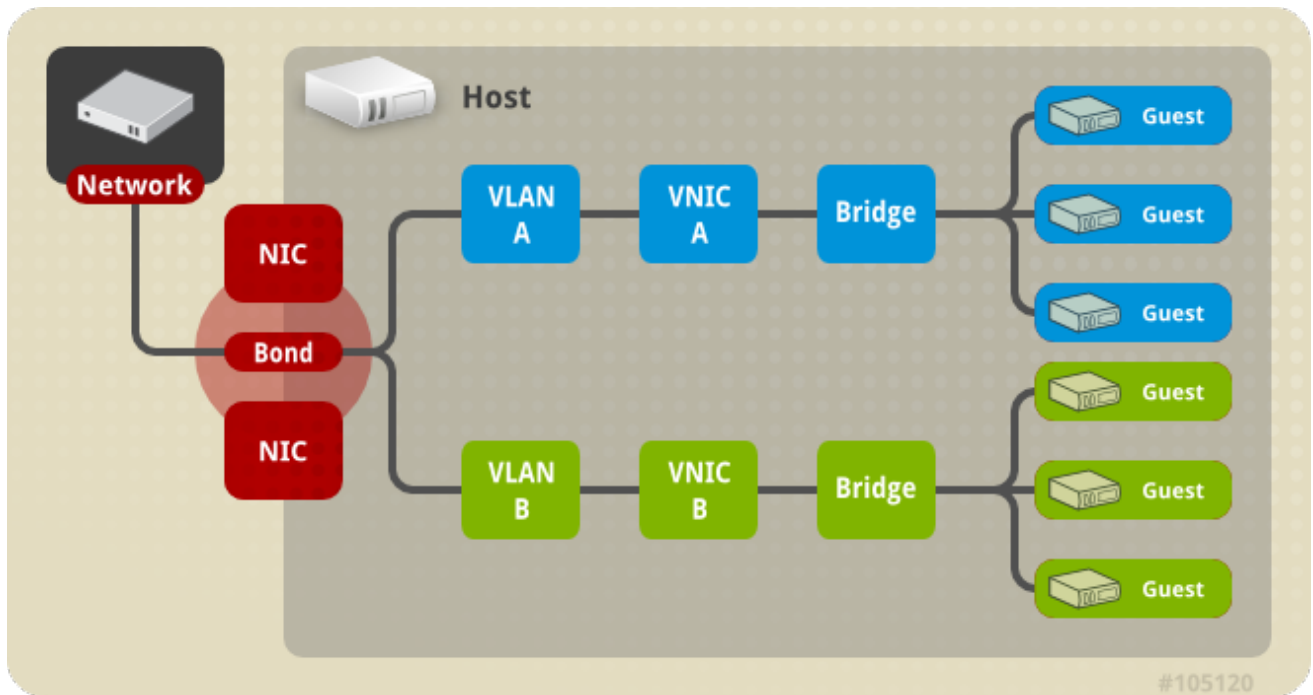
그림 3.6. 다중 브리지, 다중 VLAN 및 NIC 구성



### 3.20. 다중 브리지, 다중 VLAN 및 본딩 구성

그림 3.7. “여러 브릿지, 여러 VLAN, 본딩 연결을 통한 다중 NIC” 여러 VLAN과 쉽게 연결할 수 있도록 여러 NIC를 결합하는 구성을 표시합니다.

그림 3.7. 여러 브릿지, 여러 VLAN, 본딩 연결을 통한 다중 NIC



이 구성의 각 VLAN은 NIC를 연결하는 본딩을 통해 정의됩니다. 각 VLAN은 개별 브릿지에 연결되고 각 브릿지는 하나 이상의 게스트에 연결됩니다.

## 4장. 전원 관리

### 4.1. 전원 관리 및 펜싱 소개

**Red Hat Virtualization** 환경은 전원 관리 및 펜싱이 구성된 경우 가장 유연하고 탄력적입니다. 전원 관리를 통해 **Red Hat Virtualization Manager**는 호스트 전원 사이클 작업을 제어할 수 있으며, 가장 중요한 것은 문제가 감지된 호스트를 재부팅하는 것입니다. 펜싱은 성능 저하를 방지하기 위해 시스템을 재부팅하여 문제 호스트를 **Red Hat Virtualization** 환경에서 분리하는 데 사용됩니다. 그런 다음 펜싱된 호스트를 관리자 작업을 통해 응답 상태로 반환하고 환경으로 다시 통합할 수 있습니다.

전원 관리 및 펜싱은 호스트 운영 체제와 독립적으로 호스트를 다시 시작하기 위해 특수 전용 하드웨어를 사용합니다. **Red Hat Virtualization Manager**는 네트워크 IP 주소 또는 호스트 이름을 사용하여 전원 관리 장치에 연결됩니다. **Red Hat Virtualization**의 컨텍스트에서 전원 관리 장치와 펜싱 장치는 동일합니다.

### 4.2. RED HAT VIRTUALIZATION의 PROXY에 의한 전원 관리

**Red Hat Virtualization Manager**는 차단 에이전트와 직접 통신하지 않습니다. 대신 **Manager**는 프록시를 사용하여 호스트 전원 관리 장치에 전원 관리 명령을 보냅니다. **Manager**는 **VDSM**을 사용하여 전원 관리 장치 작업을 실행하므로 환경의 다른 호스트가 펜싱 프록시로 사용됩니다.

다음 중 하나를 선택할 수 있습니다.

- 펜싱이 필요한 호스트와 동일한 클러스터에 있는 호스트입니다.
- 펜싱이 필요한 호스트와 동일한 데이터 센터에 있는 모든 호스트입니다.

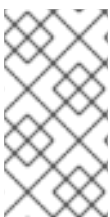
실행 가능한 펜싱 프록시 호스트의 상태는 **UP** 또는 **Maintenance** 중 하나입니다.

### 4.3. 전원 관리

**Red Hat Virtualization Manager**는 비작동 또는 비응답(**non-responsive**) 상태로 진입한 호스트를 재부팅할 수 있을 뿐 만 아니라 전력을 절감하기 위해 활용도가 낮은 호스트의 전원을 끄도록 준비할 수 있습니다. 이 기능은 올바르게 구성된 전원 관리 장치에 따라 다릅니다. **Red Hat Virtualization** 환경은 다음과 같은 전원 관리 장치를 지원합니다.

- **APC( American Power Conversion )**입니다.

- **BladeCenter.**
- **Cisco Unified Computing System (ciisco\_ucs).**
- **Dell 원격 액세스 카드 5 (drac5).**
- **Dell 원격 액세스 카드 7 (drac7).**
- **전자 전원 스위치 (eps)**
- **HP BladeSystem (하루드).**
- **통합 Lights Out (ilo ilo2,ilo3,ilo4).**
- **지능형 플랫폼 관리 인터페이스 (ipmilan).**
- **원격 Supervisor Adapter (rsa)**
- **rsb.**
- **Western Telematic, Inc (Wti).**



#### 참고

APC 5.x 전원 관리 장치는 **apc** 차단 에이전트에서 지원되지 않습니다. 대신 **apc\_snmp** 차단 에이전트를 사용합니다.



나열된 전원 관리 장치와 통신하기 위해 **Red Hat Virtualization Manager**는 차단 에이전트를 사용합니다. **Red Hat Virtualization Manager**를 사용하면 관리자가 장치가 수락하고 응답할 매개 변수를 사용하여 해당 환경에서 전원 관리 장치에 맞게 차단 에이전트를 구성할 수 있습니다. 기본 설정 옵션은 그래픽 사용자 인터페이스를 사용하여 구성할 수 있습니다. 특수 구성 옵션을 입력할 수도 있으며, 펜싱 장치에 구문 분석되지 않습니다. 특수 설정 옵션은 지정된 펜싱 장치에 한정되며 기본 설정 옵션은 지원되는 모든 전원 관리 장치에서 제공하는 기능입니다. 모든 전원 관리 장치에서 제공하는 기본 기능은 다음과 같습니다.

- **status:** 호스트의 상태를 점검합니다.
- **Start:** 호스트의 전원을 켭니다.
- **stop:** 호스트의 전원을 끕니다.
- **다시 시작:** 호스트를 다시 시작하십시오. 실제로 **stop, wait, status, start, wait, status**로 구현됩니다.

가장 좋은 방법은 처음 구성할 때 전원 관리 구성을 한 번 테스트하여 지속적인 기능을 보장하는 것입니다.

복원력은 환경에 있는 모든 호스트에서 올바르게 구성된 전원 관리 장치를 통해 제공됩니다. 펜싱 에이전트를 사용하면 **Red Hat Virtualization Manager**에서 호스트 전원 관리 장치와 통신하여 문제 호스트의 운영 체제를 무시하고 재부팅하여 나머지 환경에서 호스트를 분리할 수 있습니다. 그런 다음 **Manager**는 문제가 있는 경우 **Manager** 역할을 다시 할당하고 고가용성 가상 머신을 다른 호스트에서 안전하게 다시 시작할 수 있습니다.

#### 4.4. 펜싱

**Red Hat Virtualization** 환경의 컨텍스트에서 펜싱은 **Manager**에서 시작된 호스트 재부팅이며 전원 관리 장치에서 수행합니다. 펜싱을 사용하면 클러스터가 예기치 않은 호스트 장애에 대응하고 절전, 로드 밸런싱, 가상 시스템 가용성 정책을 적용할 수 있습니다.

펜싱을 수행하면 스토리지 풀 관리자(ECDHE)의 역할이 항상 기능 호스트에 할당됩니다. 펜싱된 호스트가ECDHE인 경우ECDHE 역할이 다시 조정되고 응답 호스트에 다시 할당됩니다. IRQ 역할이 있는 호스트는 데이터 도메인 구조 메타데이터, 무응답, 비-해상할 수 있는 유일한 호스트이므로, 환경에서 가상 디스크를 생성 및 제거하고, 스냅샷을 만들고, 논리 볼륨을 확장하며, 데이터 도메인 구조 메타데이터를 변경해야 하는 기타 모든 작업을 수행할 수 있습니다.

호스트가 응답하지 않게 되면 현재 해당 호스트에서 실행 중인 모든 가상 머신도 응답하지 않을 수 있습니다. 그러나 무응답 호스트는 실행 중인 가상 머신의 가상 머신 하드 디스크 이미지에 대한 잠금을 유지합니다. 두 번째 호스트에서 가상 머신을 시작하고 가상 머신 하드 디스크 이미지에 대한 두 번째 호스트 쓰기 권한을 할당하면 데이터가 손상될 수 있습니다.

펜싱을 사용하면 Red Hat Virtualization Manager에서 가상 머신 하드 디스크 이미지의 잠금이 해제되었다고 가정할 수 있습니다. Manager는 펜싱 에이전트를 사용하여 문제가 발생한 호스트를 재부팅했는지 확인할 수 있습니다. 이 확인이 수신되면 Red Hat Virtualization Manager는 데이터 손상 위험을 초래하지 않고 다른 호스트의 문제 호스트에서 가상 머신을 시작할 수 있습니다. 펜싱은고가용성 가상 시스템의 기반입니다.고가용성으로 표시된 가상 머신은 다른 호스트에서 안전하게 시작할 수 없으므로 데이터 손상이 발생하지 않습니다.

호스트가 무응답이 되면 Red Hat Virtualization Manager를 사용하면 30초 동안 유예 기간이 완료되기 전에 전달되어 호스트가 일시적인 오류로부터 복구할 수 있습니다. 유예 기간이 경과한 시점에 호스트가 응답하지 않는 경우 Manager는 무응답 호스트에서 부정적인 영향을 자동으로 완화하기 시작합니다. Manager는 호스트의 전원 관리 카드에 펜싱 에이전트를 사용하여 호스트를 중지하고, 중지되었는지 확인하고, 호스트를 시작하고, 호스트가 시작되었는지 확인합니다. 호스트 부팅이 완료되면 호스트의 일부였던 클러스터에 다시 참여하려고 합니다. 재부팅을 통해 호스트가 응답하지 않는 문제가 해결되면 호스트는 자동으로 Up 상태로 설정되고 가상 머신을 다시 시작하고 호스팅할 수 있습니다.

#### 4.5. 소프트 펜싱 호스트

예기치 않은 문제로 인해 호스트가 무응답이 될 수 있으며 VDSM에 요청에 응답할 수 없는 경우에도 VDSM에 의존하는 가상 머신은 활성 상태로 유지되어 액세스할 수 있습니다. 이러한 경우 VDSM 재시작은 VDSM을 응답 상태로 반환하고 이 문제를 해결합니다.

"SSHECDHE Fencing"은 관리자가 무응답 호스트에서 SSH를 통해 VDSM을 다시 시작하려고 시도하는 프로세스입니다. Manager가 SSH를 통해 VDSM을 재시작하지 못하는 경우 외부 펜싱 에이전트가 구성된 경우 펜싱에 대한 책임은 외부 펜싱 에이전트에 종속됩니다.

SSH를 통한 소프트 펜싱은 다음과 같이 작동합니다. 호스트에서 펜싱을 구성하고 활성화해야 하며 유효한 프록시 호스트(데이터 센터의 UP 상태의 두 번째 호스트)가 있어야 합니다. 관리자와 호스트 간의 연결이 시간 초과되면 다음이 발생합니다.

1. 첫 번째 네트워크 오류에서 호스트 상태가 "연결"으로 변경됩니다.
2. 그러면 **Manager**에서 **VDSM**에 상태를 요청하거나 호스트의 부하에 따라 결정되는 간격을 기다립니다. 간격의 길이를 결정하는 공식은 **TimeoutToResetVdsInSeconds** (기본값은 60초) + **[DelayResetPerVmInSeconds) \* (호스트에서 가상 머신 실행 횟수) + [DelayResetForSpmlnSeconds) \* [DelayResetForSpmlnSeconds (기본값) \* (기본값은 20초) \* (기본값은 0초) \* (초) \* (기본값: 0초) \* (초)**에 의해 구성됩니다. **VDSM**에 응답 시간을 최대로 제공하기 위해 **Manager**는 위에서 언급한 두 가지 옵션 중 더 긴 기간(양식 3개 또는 위 공식에 따라 결정된 간격)을 선택합니다.
3. 해당 간격이 경과할 때 호스트가 응답하지 않으면 **SSH**를 통해 **vds restart** 가 실행됩니다.
4. **vds** 재시작 이 호스트와 **Manager** 간 연결을 다시 설정하지 않으면 호스트의 상태가 **Non Responsive** 로 변경되고 전원 관리가 구성된 경우 펜싱이 외부 펜싱 에이전트로 전달됩니다.



#### 참고

**SSH**를 통한 소프트웨어 펜싱은 전원 관리가 구성되지 않은 호스트에서 실행할 수 있습니다. 이는 "연결"과 다릅니다. 펜싱은 전원 관리가 구성된 호스트에서만 실행할 수 있습니다.

#### 4.6. 다중 전원 관리 펜싱 에이전트 사용

단일 에이전트는 기본 에이전트로 취급됩니다. 보조 에이전트는 두 개의 펜싱 에이전트가 있는 경우 유효합니다(예: 각 전원 스위치에 두 개의 에이전트가 동일한 전원 스위치에 연결된 듀얼 전원 호스트의 경우). 에이전트는 동일하거나 다른 유형일 수 있습니다.

호스트에 펜싱 에이전트가 여러 개 있으면 펜싱 절차의 안정성이 향상됩니다. 예를 들어 호스트의 단일 펜싱 에이전트가 실패하면 호스트는 수동으로 재부팅될 때까지 작동하지 않는 상태로 유지됩니다. 이전에 호스트에서 실행 중인 가상 머신은 일시 중지되며 원래 호스트를 수동으로 펜싱한 후 클러스터의 다른 호스트로만 장애 조치됩니다. 여러 에이전트가 있으면 첫 번째 에이전트가 실패하면 다음 에이전트를 호출할 수 있습니다.

호스트에 두 개의 펜싱 에이전트가 정의되면 동시 또는 순차 흐름을 사용하도록 구성할 수 있습니다.



**Concurrent:** 주 및 보조 에이전트 모두 호스트를 중지하려면 **Stop** 명령에 응답해야 합니다.

하나의 에이전트가 **Start** 명령에 응답하는 경우 호스트는 가동됩니다.

- 순차적: 호스트를 중지하거나 시작하기 위해 기본 에이전트가 먼저 사용되며 실패하는 경우 보조 에이전트가 사용됩니다.

## 5장. 로드 밸런싱, 스케줄링 및 마이그레이션

### 5.1. 로드 밸런싱, 스케줄링 및 마이그레이션

개별 호스트에는 유한 하드웨어 리소스가 있으며 오류가 발생하기 쉽습니다. 실패 및 리소스 소진을 완화하기 위해 호스트는 기본적으로 공유 리소스 그룹입니다. **Red Hat Virtualization** 환경은 로드 밸런싱 정책, 스케줄링 및 마이그레이션을 사용하여 호스트 리소스의 수요 변화에 응답합니다. **Manager**는 클러스터의 단일 호스트가 해당 클러스터의 모든 가상 시스템을 처리하지 않도록 할 수 있습니다. 반대로 **Manager**는 활용도가 낮은 호스트를 인식하고 모든 가상 시스템을 마이그레이션하여 관리자가 해당 호스트를 종료하여 전원을 절약할 수 있습니다.

세 개의 이벤트로 인해 사용 가능한 리소스를 확인합니다.

- 가상 머신 시작 - 가상 시스템이 시작할 호스트를 결정하기 위해 리소스를 확인합니다.
- 가상 머신 마이그레이션 - 적절한 대상 호스트를 결정하기 위해 리소스를 확인합니다.
- 시간 경과 - 리소스를 정기적으로 확인하여 개별 호스트 로드가 클러스터 로드 밸런싱 정책을 준수하는지 확인합니다.

**Manager**는 클러스터의 한 호스트에서 다른 호스트로 가상 머신 마이그레이션을 예약하기 위해 클러스터에 대한 로드 밸런싱 정책을 사용하여 사용 가능한 리소스의 변경 사항에 응답합니다. 로드 밸런싱 정책, 스케줄링 및 가상 머신 마이그레이션 간의 관계에 대해서는 다음 섹션에서 설명합니다.

### 5.2. 로드 밸런싱 정책

로드 밸런싱 정책은 각각 다른 하드웨어 매개변수와 사용 가능한 메모리를 가질 수 있는 하나 이상의 호스트를 포함하는 클러스터에 대해 설정됩니다. **Red Hat Virtualization Manager**는 로드 밸런싱 정책을 사용하여 클러스터에서 가상 머신을 시작하는 호스트를 결정합니다. 또한 로드 밸런싱 정책을 사용하면 **Manager**에서 과도하게 사용되는 호스트에서 활용도가 낮은 호스트로 가상 머신을 이동할 시기를 결정할 수 있습니다.

부하 분산 프로세스는 데이터 센터의 각 클러스터에 대해 1분마다 한 번씩 실행됩니다. 활용도가 낮은 호스트 및 가상 시스템 마이그레이션에 유효한 대상인 호스트를 과도하게 활용하는 호스트를 결정합니다.

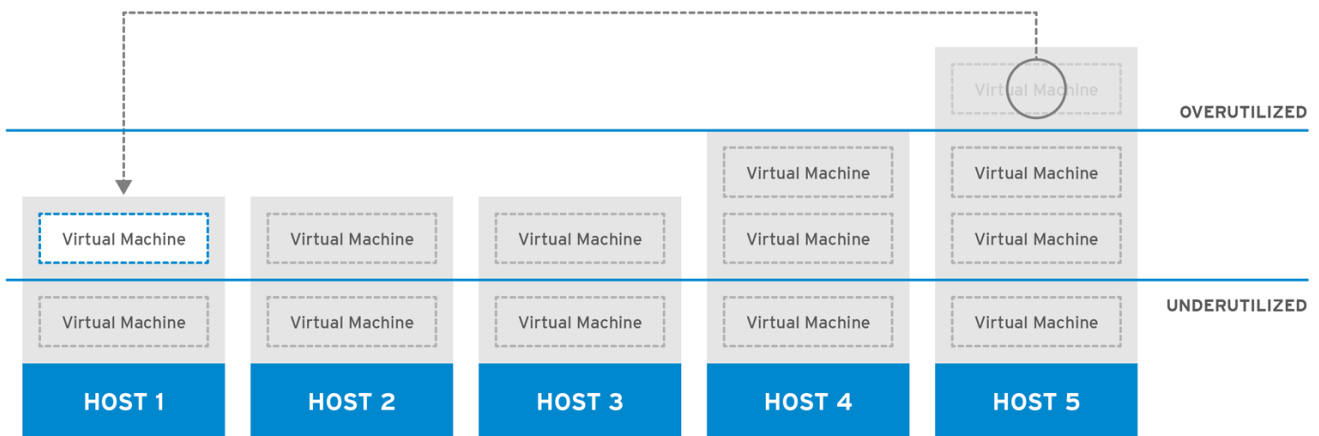
이는 지정된 클러스터에 대해 관리자가 설정한 로드 밸런싱 정책에 따라 결정됩니다. 로드 밸런싱 정책의 옵션은 **VM\_Evenly\_Distributed, Evenly\_Distributed, Power\_Saving, None** 입니다.

### 5.3. 로드 밸런싱 정책: VM\_EVENLY\_DISTRIBUTED

가상 머신의 균등하게 분산 로드 밸런싱 정책은 가상 머신 수에 따라 호스트 간에 가상 머신을 균등하게 분배합니다. 높은 가상 머신 수는 각 호스트에서 실행할 수 있는 최대 가상 머신 수이며, 그 이상으로 호스트가 과부하로 분류됩니다. **VM\_Evenly\_Distributed** 정책을 사용하면 관리자가 호스트의 높은 가상 머신 수를 설정할 수 있습니다. 가장 많이 사용되는 호스트와 최소 사용률 호스트 간의 가상 시스템 수의 최대 포함 차이도 관리자가 설정합니다. 클러스터의 모든 호스트에 이 마이그레이션 임계값에 속하는 가상 머신 수가 있을 때 클러스터가 균형을 유지합니다. 또한 관리자는 **ECDHE** 호스트에서 예약할 가상 머신의 슬롯 수를 설정합니다. **ECDHE** 호스트는 다른 호스트보다 부하가 낮기 때문에 이 변수는 실행할 수 있는 호스트보다 가상 머신 수를 정의합니다. 호스트가 높은 가상 머신 수보다 많은 가상 머신을 실행 중이고 하나 이상의 호스트에 마이그레이션 임계값을 벗어나는 가상 머신 수가 있는 경우 가상 머신은 **CPU** 사용률이 가장 낮은 클러스터의 호스트로 마이그레이션됩니다. 클러스터의 모든 호스트에 마이그레이션 임계값에 속하는 가상 머신 수가 한 번에 하나씩 마이그레이션됩니다.

### 5.4. 로드 밸런싱 정책: EVENLY\_DISTRIBUTED

그림 5.1. 균등하게 분산 스케줄링 정책

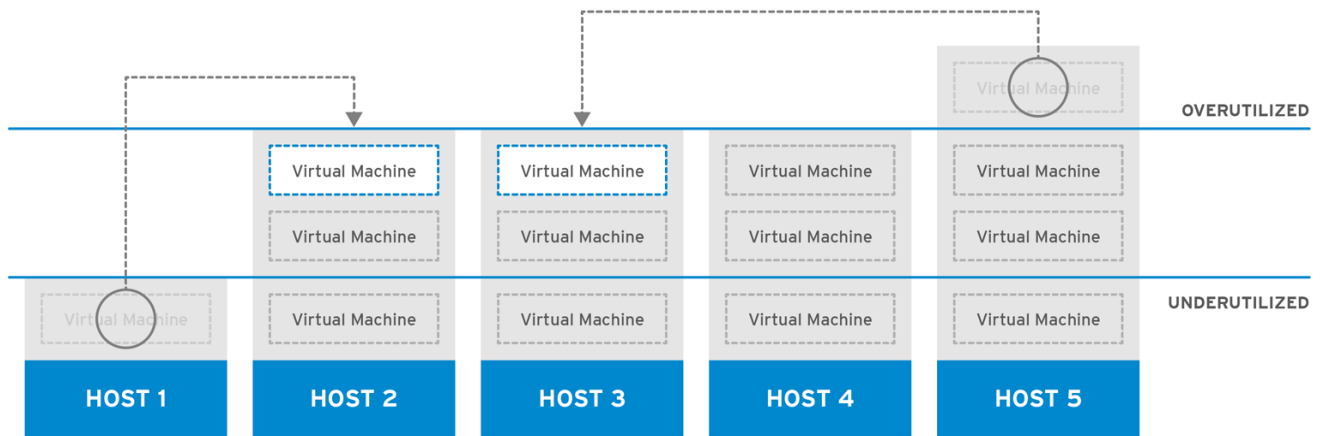


RHV\_444396\_0417

균등하게 분산 로드 밸런싱 정책은 가장 낮은 **CPU** 부하 또는 사용 가능한 최고 메모리에 따라 새 가상 시스템의 호스트를 선택합니다. 설정된 시간 동안 클러스터의 호스트에 허용되는 최대 **CPU** 로드 및 최소 사용 가능한 메모리는 균등하게 분산 스케줄링 정책의 매개변수로 정의됩니다. 이 외에도 환경 성능이 저하됩니다. 관리자는 균등하게 분산 정책을 통해 가상 머신 실행을 위해 이러한 수준을 설정할 수 있습니다. 호스트가 정의된 최대 **CPU** 부하 또는 최소 사용 가능한 메모리에 도달했으며 호스트가 설정된 시간보다 오래 남아 있는 경우 해당 호스트의 가상 머신은 사용 중인 매개변수에 따라 가장 낮은 **CPU** 또는 사용 가능한 메모리가 있는 클러스터의 호스트로 마이그레이션됩니다. 호스트 리소스는 분당 한 번 확인되며 호스트 **CPU** 로드가 정의된 제한보다 작거나 호스트 사용 가능한 메모리가 정의된 제한을 초과할 때까지 한 번에 하나의 가상 시스템이 마이그레이션됩니다.

### 5.5. 로드 밸런싱 정책: POWER\_SAVING

그림 5.2. Power Saving Scheduling Policy



RHV\_444396\_0417

절전 로드 밸런싱 정책은 가장 낮은 **CPU** 또는 사용 가능한 최고 메모리에 따라 새 가상 시스템의 호스트를 선택합니다. 설정된 시간 동안 클러스터의 호스트에 허용되는 최대 **CPU** 로드 및 최소 사용 가능한 메모리는 절전 스케줄링 정책의 매개변수로 정의됩니다. 이 외에도 환경 성능이 저하됩니다. 또한 절전 매개변수는 호스트의 지속적인 작동을 비효율적으로 간주하기 전에 클러스터의 호스트에 허용되는 최소 **CPU** 로드 및 사용 가능한 최대 메모리를 정의합니다. 호스트가 최대 **CPU** 부하 또는 최소 사용 가능한 메모리에 도달하여 설정된 시간보다 많은 경우 해당 호스트의 가상 시스템이 사용 중인 매개변수에 따라 가장 낮은 **CPU** 또는 사용 가능한 메모리가 있는 호스트로 마이그레이션됩니다. 호스트 리소스는 분당 한 번 확인되며 호스트 **CPU** 로드가 정의된 제한보다 작거나 호스트 사용 가능한 메모리가 정의된 제한을 초과할 때까지 한 번에 하나의 가상 시스템이 마이그레이션됩니다. 호스트의 **CPU** 부하가 정의된 최솟 수준 이상으로 떨어지거나 호스트의 사용 가능한 메모리가 정의된 최대 수준 이상으로 떨어지면 해당 호스트의 가상 머신이 클러스터의 다른 호스트로 마이그레이션됩니다. 활용도가 낮은 호스트가 나머지 가상 시스템을 지우면 **Manager**는 호스트 시스템의 전원을 자동으로 끄고 로드 밸런싱이 필요하거나 클러스터에 사용 가능한 호스트가 충분하지 않은 경우 다시 다시 시작합니다.

## 5.6. 로드 밸런싱 정책: 없음

로드 밸런싱 정책을 선택하지 않으면 **CPU** 사용률과 사용 가능한 메모리가 가장 낮은 클러스터 내의 호스트에서 가상 머신이 시작됩니다. **CPU** 사용률을 확인하기 위해 가상 **CPU** 수 및 **CPU** 사용량 백분율을 고려하는 결합된 매트릭이 사용됩니다. 이 접근 방식은 새 가상 시스템이 시작될 때 유일한 호스트 선택 지점이기 때문에 가장 덜 동적입니다. 호스트의 수요가 증가해도 가상 머신은 자동으로 마이그레이션되지 않습니다.

관리자는 지정된 가상 시스템의 적절한 마이그레이션 대상인 호스트를 결정해야 합니다. 고정을 사용하여 가상 머신을 특정 호스트와 연결할 수도 있습니다. 고정을 수행하면 가상 머신이 자동으로 다른 호스트로 마이그레이션되지 않습니다. 리소스를 많이 사용하는 환경의 경우 수동 마이그레이션이 최상의 방법입니다.

## 5.7. 로드 밸런싱 정책: INCLUSTERUPGRADE

**InClusterUpgrade** 스케줄링 정책은 호스트 운영 체제 버전을 기반으로 가상 머신을 배포합니다. 최신 운영 체제가 현재 실행되는 가상 머신보다 최신 운영 체제가 있는 호스트는 동일한 운영 체제를 사용하는 호스트보다 우선 순위가 부여됩니다. 최신 운영 체제가 있는 호스트로 마이그레이션한 가상 머신은 이전

운영 체제로 다시 마이그레이션되지 않습니다. 클러스터의 모든 호스트에서 가상 머신을 재시작할 수 있습니다. 이 정책을 사용하면 클러스터에 운영 체제 버전이 혼합될 수 있으므로 클러스터의 호스트를 업그레이드할 수 있습니다. 정책을 활성화하기 전에 사전 조건을 충족해야 합니다. **Red Hat Enterprise Virtualization 3.6 업그레이드 가이드의 클러스터 업그레이드에서 Red Hat Enterprise Linux 6에서 Red Hat Enterprise Linux 7로 호스트 업그레이드를 참조하십시오.**



## 중요

**InClusterUpgrade** 스케줄링 정책은 주요 버전 간 업그레이드에만 사용됩니다. 예를 들어 **Red Hat Enterprise Linux 6에서 Red Hat Enterprise Linux 7로 업그레이드합니다.**

## 5.8. 고가용성 가상 머신

고가용성(HA) 가상 머신 예약 정책을 통해 **Red Hat Virtualization Manager**는 고가용성 가상 머신의 클러스터 용량을 모니터링할 수 있습니다. **Manager**에는 고가용성을 위해 개별 가상 머신에 플래그를 지정할 수 있는 기능이 있습니다. 즉, 호스트 장애 시 이러한 가상 머신이 대체 호스트에서 재부팅됩니다. 이 정책은 클러스터의 호스트 전체에서 고가용성 가상 시스템의 균형을 유지합니다. 클러스터의 호스트가 실패하면 나머지 호스트는 클러스터 성능에 영향을 주지 않고 고가용성 가상 머신의 마이그레이션 로드를 지원할 수 있습니다. 고가용성 가상 머신 예약이 활성화되면 **Manager**에서 기존 호스트가 예기치 않게 실패하는 경우 **HA** 가상 머신을 마이그레이션할 수 있도록 클러스터 내에 적절한 용량이 있는지 확인합니다.

## 5.9. 스케줄링

**Red Hat Virtualization**에서 스케줄링은 **Red Hat Virtualization Manager**가 새로운 또는 마이그레이션된 가상 시스템의 대상으로 클러스터에서 호스트를 선택하는 방식을 나타냅니다.

호스트가 가상 머신을 시작하거나 다른 호스트에서 마이그레이션된 가상 시스템을 수락하려면 가상 머신의 요구 사항을 지원하거나 마이그레이션하는 데 충분한 여유 메모리와 **CPU**가 있어야 합니다. 과부하된 **CPU**가 있는 호스트에서 가상 머신이 시작되지 않습니다. 기본적으로 호스트의 **CPU**는 5분 동안 **80%**를 초과하는 부하가 있는 경우 과부하로 간주되지만 스케줄링 정책을 사용하여 이러한 값을 변경할 수 있습니다. 여러 호스트가 적합한 대상인 경우 클러스터의 로드 밸런싱 정책에 따라 하나가 선택됩니다. 예를 들어 **Evenly\_Distributed** 정책이 적용되는 경우 **Manager**에서 **CPU** 사용률이 가장 낮은 호스트를 선택합니다. **Power\_Saving** 정책이 적용되는 경우 최대 및 최소 서비스 수준 간에 **CPU** 사용률이 가장 낮은 호스트가 선택됩니다. 지정된 호스트의 스토리지 풀관리자(**ECDHE**) 상태는 가상 머신 또는 가상 머신 마이그레이션 시작 대상의 자격에도 영향을 미칩니다. **SPM**이 아닌 호스트는 기본 대상 호스트입니다. 예를 들어 클러스터에서 시작된 첫 번째 가상 시스템은 해당 클러스터의 호스트에 의해 보유되는 경우 클러스터에서 시작된 첫 번째 가상 시스템이 **ECDHE** 호스트에서 실행되지 않습니다.

## 5.10. MIGRATION

**Red Hat Virtualization Manager**는 마이그레이션을 사용하여 클러스터에 대한 로드 밸런싱 정책을 적용합니다. 가상 머신 마이그레이션은 클러스터의 로드 밸런싱 정책과 클러스터 내의 호스트에 대한 현재 요구에 따라 수행됩니다. 호스트를 펜싱하거나 유지 관리 모드로 이동할 때 마이그레이션이 자동으로 발생하도록 구성할 수도 있습니다. **Red Hat Virtualization Manager**는 먼저 **CPU** 사용률이 가장 낮은 가상



머신을 마이그레이션합니다. 이는 백분율로 계산되며 I/O 작업이 CPU 사용률에 영향을 미치는 경우를 제외하고 RAM 사용량 또는 I/O 작업을 고려하지 않습니다. 동일한 CPU 사용량이 있는 가상 머신이 두 개 이상 있는 경우 먼저 마이그레이션되는 가상 머신이 Red Hat Virtualization Manager에서 실행하는 첫 번째 가상 머신으로, 가상 머신 CPU 사용량을 결정합니다.

가상 머신 마이그레이션에는 기본적으로 다음과 같은 제한 사항이 있습니다.

- 52MiBps(초당 메가바이트)의 대역폭 제한이 각 가상 시스템 마이그레이션에 적용됩니다.
- 마이그레이션 시간은 GB의 가상 머신 메모리당 64초 후에 시간 초과됩니다.
- 진행이 2초 동안 중단되면 마이그레이션이 중단됩니다.
- 동시 발신 마이그레이션은 호스트당 CPU 코어당 1개 또는 2개 중 작은 것으로 제한됩니다.

마이그레이션 설정 튜닝에 대한 자세한 내용은 을 참조하십시오  
<https://access.redhat.com/solutions/744423>.

## 6장. 디렉터리 서비스

### 6.1. 디렉터리 서비스

**Red Hat Virtualization** 플랫폼은 사용자 인증 및 권한 부여를 위해 디렉터리 서비스에 의존합니다. 사용자 포털, 관리 포털, **REST API**를 포함한 모든 **Manager** 인터페이스와의 상호 작용은 인증된 인증된 사용자로 제한됩니다. **Red Hat Virtualization** 환경의 가상 머신은 동일한 디렉터리 서비스를 사용하여 인증 및 권한 부여를 제공할 수 있지만 이를 구성하려면 구성해야 합니다. 현재 **Red Hat Virtualization Manager**에서 사용되는 디렉터리 서비스 공급자는 **IdM( Identity Management )**, **RHDS( Red Hat Directory Server 9)**, **AD( Active Directory Server)**, **OpenLDAP**입니다. **Red Hat Virtualization Manager**는 다음을 위해 디렉터리 서버와 상호 작용합니다.

- 포털 로그인(사용자, 전원 사용자, 관리자, **REST API**).
- 사용자 정보를 표시하는 쿼리입니다.
- 도메인에 **Manager** 추가.

인증은 일부 데이터를 생성한 당사자의 확인 및 확인이며 생성된 데이터의 무결성입니다. 보안 주체는 해당 **ID**가 확인되는 당사자입니다. 검증자는 본원의 신원을 보장할 것을 요구하는 당사자입니다. **Red Hat Virtualization**의 경우 **Manager**는 검증자이며 사용자는 보안 주체입니다. 데이터 무결성은 수신된 데이터가 주체가 생성한 데이터와 동일함을 보장하는 것입니다.

기밀성 및 권한 부여는 인증과 밀접하게 관련되어 있습니다. 기밀성은 데이터를 수신하려는 의도가 아닌 공개로부터 보호합니다. 강력한 인증 방법은 선택적으로 기밀성을 제공할 수 있습니다. 권한 부여는 사용자가 작업을 수행할 수 있는지 여부를 결정합니다. **Authorization determines whether a principal is allowed to perform an operation.** **Red Hat Virtualization**은 디렉터리 서비스를 사용하여 사용자와 역할을 연결하고 그에 따라 권한 부여를 제공합니다. 권한 부여는 일반적으로 보안 주체가 인증된 후에 수행되며 검증자에게 로컬 또는 원격 정보를 기반으로 할 수 있습니다.

설치하는 동안 **Red Hat Virtualization** 환경 관리를 위해 로컬 내부 도메인이 자동으로 구성됩니다. 설치가 완료되면 더 많은 도메인을 추가할 수 있습니다.

### 6.2. 로컬 인증: 내부 도메인

**Red Hat Virtualization Manager**는 설치 중에 제한된 내부 관리 도메인을 생성합니다. 이 도메인은 디

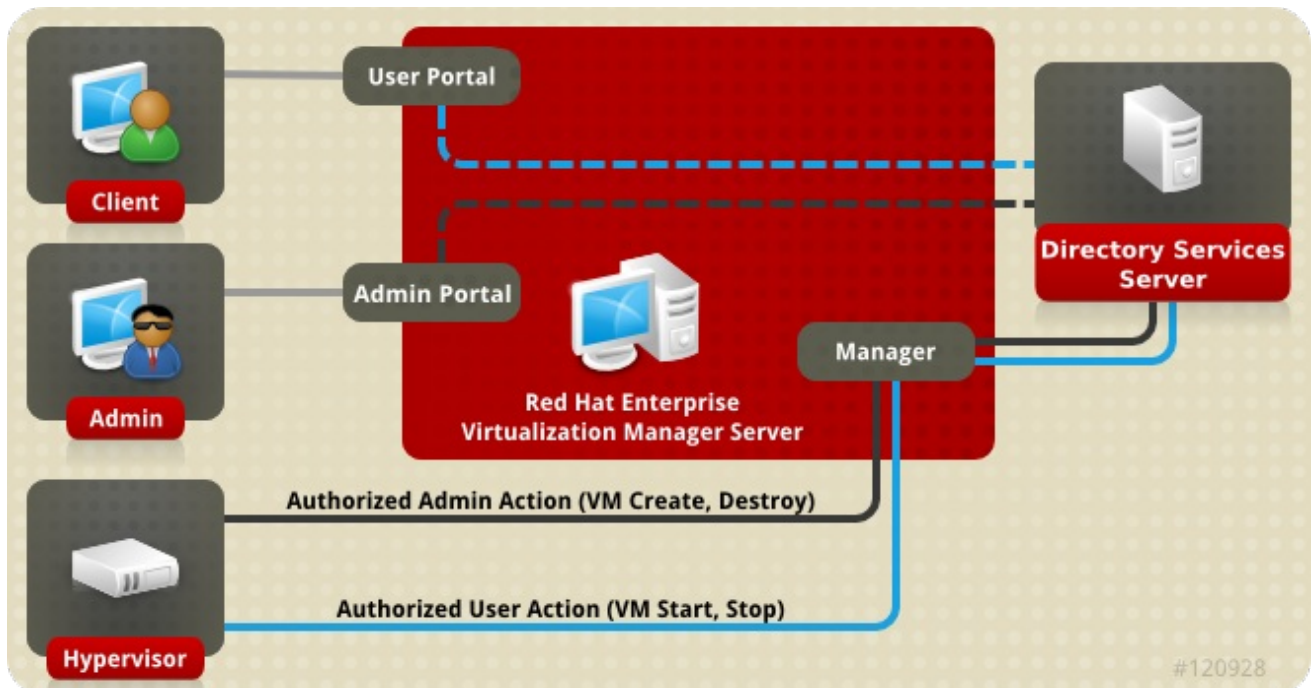
렉터리 서버에서 디렉터리 서비스 사용자가 아닌 **Red Hat Virtualization PostgreSQL** 데이터베이스의 키를 기반으로 하므로 **AD** 또는 **IdM** 도메인과 동일하지 않습니다. 내부 도메인은 **admin@internal** 사용자 중 하나만 보유하므로 내부 도메인은 외부 도메인과도 다릅니다. 이 접근 방식을 초기 인증을 통해 완전한 기능 디렉터리 서버 없이도 **Red Hat Virtualization**을 평가할 수 있으며 관리 계정을 사용하여 외부 디렉터리 서비스의 문제를 해결할 수 있습니다.

**admin@internal** 사용자는 환경의 초기 구성을 위한 것입니다. 여기에는 호스트 설치 및 수락, 외부 **AD** 또는 **IdM** 인증 도메인 추가, 외부 도메인에서 사용자에게 권한을 위임하는 작업이 포함됩니다.

### 6.3. GSSAPI를 사용한 원격 인증

**Red Hat Virtualization**의 컨텍스트에서 원격 인증은 **Red Hat Virtualization Manager**에서 원격으로 처리되는 인증을 나타냅니다. 원격 인증은 **AD**, **IdM** 또는 **RHDS** 도메인 내에서 **Manager**로 들어오는 사용자 또는 **API** 연결에 사용됩니다. **Red Hat Virtualization Manager**는 **engine-manage-domains** 툴을 사용하여 **RHDS**, **AD** 또는 **IdM** 도메인의 일부로 관리자가 구성해야 합니다. 이를 위해서는 시스템을 도메인에 결합할 수 있는 충분한 권한이 있는 도메인의 **RHDS**, **AD** 또는 **IdM** 디렉터리 서버에서 계정에 대한 자격 증명을 관리자에게 제공해야 합니다. 도메인을 추가한 후 암호를 사용하여 디렉터리 서버에 대해 **Red Hat Virtualization Manager**에서 도메인 사용자를 인증할 수 있습니다. 관리자는 **GSSAPI**(**Generic Security Services Application Program Interface**)를 사용하여 사용자의 ID를 안전하게 확인하고 사용자가 사용할 수 있는 권한 부여 수준을 확인하는 데 **GSSAPI**(**Simple Authentication and Security Layer**)라는 프레임워크를 사용합니다.

그림 6.1. GSSAPI 인증



## 7장. 템플릿 및 풀

### 7.1. 템플릿 및 풀

**Red Hat Virtualization** 환경을 통해 관리자는 가상 머신의 프로비저닝을 사용자에게 단순화할 수 있는 틀을 제공합니다. 이는 **템플릿 및 풀**입니다. 템플릿은 관리자가 사전 구성된 기존 가상 머신을 기반으로 새 가상 머신을 신속하게 생성하여 운영 체제 설치 및 구성을 우회할 수 있는 바로 가기입니다. 이 기능은 특히 웹 서버 가상 시스템(예: 웹 서버 가상 시스템)과 같이 사용되는 가상 시스템에 유용합니다. 조직에서 특정 웹 서버의 여러 인스턴스를 사용하는 경우 관리자는 템플릿으로 사용될 가상 머신을 생성하고, 운영 체제, 웹 서버, 지원 패키지를 설치하고, 고유한 구성 변경 사항을 적용할 수 있습니다. 그런 다음 관리자는 필요에 따라 새 가상 머신을 생성하는 데 사용할 작업 가상 머신을 기반으로 템플릿을 생성할 수 있습니다.

가상 머신 풀은 지정된 템플릿을 기반으로 하여 사용자에게 빠르게 프로비저닝할 수 있는 가상 머신 그룹입니다. 풀에서 가상 머신을 사용할 수 있는 권한은 풀 수준에서 부여됩니다. 풀을 사용할 수 있는 권한이 부여된 사용자에게는 풀의 모든 가상 시스템이 할당됩니다. 가상 머신 풀의 고유성은 해당 가상 머신 내의 가상 머신의 전송 특성입니다. 과거에는 풀에 있는 가상 머신에 관계없이 사용자에게 가상 머신이 할당되므로 풀은 데이터 지속성이 필요한 목적에 적합하지 않습니다. 가상 머신 풀은 사용자 데이터가 중앙 위치에 저장되고 가상 머신이 해당 데이터에 액세스하여 사용하거나 데이터 지속성이 중요하지 않은 시나리오에 가장 적합합니다. 풀을 생성하면 풀을 정지된 상태로 채우는 가상 시스템이 생성됩니다. 그런 다음 사용자 요청 시 시작됩니다.

### 7.2. 템플릿

관리자는 템플릿을 생성하기 위해 가상 시스템을 생성하고 사용자 지정합니다. 원하는 패키지가 설치되고 사용자 지정 구성이 적용되므로 배포 후 수행해야 하는 변경 사항을 최소화하기 위해 가상 시스템이 의도한 대로 준비됩니다. 가상 머신에서 템플릿을 생성하기 전에 선택 사항이지만 권장 단계는 **일반화**입니다. 일반화는 배포 시 변경될 시스템 사용자 이름, 암호 및 시간대 정보와 같은 세부 정보를 제거하는 데 사용됩니다. 표준화는 사용자 지정 구성에 영향을 미치지 않습니다. **Red Hat Virtualization** 환경에서 **Windows** 및 **Linux** 게스트를 일반화하는 방법은 **가상 머신 관리 가이드의 템플릿에서** 자세히 설명합니다. <https://access.redhat.com/documentation/en/red-hat-virtualization/4.0/single/virtual-machine-management-guide#chap-Templates> Red Hat Enterprise Linux 게스트는 `sys-unconfig` 를 사용하여 일반화됩니다. Windows 게스트는 `sys-prep` 을 사용하여 일반화됩니다.

템플릿의 기반을 제공하는 가상 머신이 원하는 경우 일반화되고 중지되면 관리자가 가상 머신에서 템플릿을 생성할 수 있습니다. 가상 머신에서 템플릿을 생성하면 특별히 구성된 가상 디스크 이미지의 읽기 전용 사본만 생성됩니다. 읽기 전용 이미지는 해당 템플릿을 기반으로 하는 모든 가상 머신에 대한 백업 이미지를 형성합니다. 즉, 템플릿은 기본적으로 관련 가상 하드웨어 구성이 있는 사용자 지정 읽기 전용 디스크 이미지입니다. 하드웨어는 템플릿에서 생성된 가상 머신에서 변경할 수 있습니다(예: **1GB의 RAM**이 있는 템플릿에서 생성된 가상 머신에 대해 **2GB의 RAM**)을 프로비저닝할 수 있습니다. 그러나 템플릿 디스크 이미지를 변경하면 템플릿을 기반으로 모든 가상 시스템이 변경될 수 있습니다.

템플릿을 생성하면 여러 가상 시스템의 기반으로 사용할 수 있습니다. 가상 머신은 **Thin** 프로비저닝 방법 또는 **Clone** 프로비저닝 방법을 사용하여 지정된 템플릿에서 생성됩니다. 템플릿에서 복제된 가상 머신은 템플릿 기본 이미지의 전체 쓰기 가능한 사본을 사용하여 더 이상 템플릿의 존재 여부에 따라 교환되

는 썬 생성 방법의 공간 절감을 저해합니다. **thin** 방법을 사용하여 템플릿에서 생성된 가상 머신은 템플릿의 읽기 전용 이미지를 기본 이미지로 사용하므로 해당 템플릿과 생성된 모든 가상 시스템을 동일한 스토리지 도메인에 저장해야 합니다. 데이터 변경 사항 및 새로 생성된 데이터는 쓰기 이미지의 복사본에 저장됩니다. 템플릿을 기반으로 하는 각 가상 머신은 동일한 기본 읽기 전용 이미지와 가상 머신에 고유한 쓰기 이미지에 복사본을 사용합니다. 이를 통해 동일한 데이터 저장 횟수를 제한하여 스토리지 절약을 제공합니다. 또한 읽기 전용 백업 이미지를 자주 사용하면 데이터가 캐시되어 성능이 저하될 수 있습니다.

### 7.3. 풀

가상 머신 풀을 사용하면 데스크탑과 같이 수많은 가상 머신을 사용자에게 빠르게 프로비저닝할 수 있습니다. 풀에서 가상 머신에 액세스하고 사용할 수 있는 권한이 부여된 사용자는 요청 대기열의 위치에 따라 사용 가능한 가상 머신을 수신합니다. 풀의 가상 머신은 데이터 지속성을 허용하지 않습니다. 풀에서 가상 시스템이 할당될 때마다 기본 상태로 할당됩니다. 이는 사용자 데이터가 중앙에서 저장되는 상황에서 사용되는 것이 가장 적합합니다.

가상 머신 풀은 템플릿에서 생성됩니다. 풀의 각 가상 머신은 동일한 백업 읽기 전용 이미지를 사용하며 쓰기 이미지에 임시 복사본을 사용하여 변경되고 새로 생성된 데이터를 유지합니다. 풀의 가상 머신은 생성된 사용자를 보유하고 있는 쓰기 계층의 복사본이 종료 시 변경된 데이터를 보유하는 쓰기 계층의 다른 가상 시스템과 다릅니다. 이는 가상 시스템 풀에 백업하는 템플릿보다 스토리지가 더 이상 필요하지 않으며 사용 중 생성되거나 변경된 데이터 공간이 필요하다는 것을 의미합니다. 가상 머신 풀은 각 사용자에게 전용 가상 데스크탑을 제공하는 데 드는 스토리지 비용 없이 일부 작업에 대해 사용자에게 컴퓨팅 기능을 제공하는 효율적인 방법입니다.

#### 예 7.1. 풀 사용량의 예

기술 지원 회사에서는 10개의 지원 담당자를 두고 있습니다. 그러나 한 번에 5개 만 작업하고 있습니다. 10개의 가상 머신을 생성하는 대신 각 지원 센터 작업당 하나씩 5개의 가상 머신 풀을 생성할 수 있습니다. 지원 센터 직원은 전환이 시작될 때 가상 머신을 자동으로 할당한 후 마지막 풀로 돌아갑니다.

## 8장. 가상 머신 스냅샷

### 8.1. 스냅샷

스냅샷은 관리자가 특정 시점에서 가상 머신의 운영 체제, 애플리케이션 및 데이터에 대한 복원 지점을 생성할 수 있는 스토리지 기능입니다. 스냅샷은 가상 머신 하드 디스크 이미지에 현재 존재하는 데이터를 **COW** 볼륨으로 저장하고 스냅샷을 만들 때 존재하는 대로 데이터를 복구할 수 있습니다. 스냅샷을 사용하면 현재 계층에 새 **COW** 계층이 생성됩니다. 스냅샷을 만든 후 수행되는 모든 쓰기 작업은 새 **COW** 계층에 기록됩니다.

가상 머신 하드 디스크 이미지가 하나 이상의 볼륨 체인임을 이해하는 것이 중요합니다. 가상 머신의 관점에서 이러한 볼륨이 단일 디스크 이미지로 표시됩니다. 가상 머신은 디스크가 여러 볼륨으로 구성되어 있다는 점에 유의해야 합니다.

**COW** 볼륨 및 **COW** 계층이라는 용어는 서로 바꿔 사용할 수 있지만 계층에서 스냅샷의 시간 특성을 더 명확하게 인식합니다. 각 스냅샷은 스냅샷을 만든 후 관리자가 데이터에 대한 불만족스러운 변경 사항을 삭제할 수 있도록 생성됩니다. 스냅샷은 여러 단어 프로세서에 있는 **Undo** 함수와 유사한 기능을 제공합니다.



#### 참고

공유 가능으로 표시된 가상 머신 하드 디스크의 스냅샷과 직접 **LUN** 연결을 기반으로 하는 가상 머신의 스냅샷은 지원되지 않습니다.

세 가지 주요 스냅샷 작업은 다음과 같습니다.

- 가상 머신에 대해 생성된 첫 번째 스냅샷을 생성하는 생성.
- 미리 보기로 스냅샷을 미리 보고 스냅샷을 만든 시점으로 시스템 데이터를 복원할지 여부를 결정합니다.
- 더 이상 필요하지 않은 복원 지점을 삭제해야 합니다.

스냅샷 작업에 대한 자세한 내용은 *Red Hat Virtualization 가상 머신 관리 가이드*의 [스냅샷](#) 을 참조하십시오.

## 8.2. RED HAT VIRTUALIZATION의 실시간 스냅샷

공유 가능으로 표시된 가상 머신 하드 디스크의 스냅샷과 직접 LUN 연결을 기반으로 하는 가상 머신의 스냅샷은 지원되지 않습니다.

복제 또는 마이그레이션되지 않은 다른 가상 머신에 실행, 일시 중지 또는 중지 시 스냅샷이 있을 수 있습니다.

가상 머신의 실시간 스냅샷이 시작되면 **Manager**에서 **Restic** 호스트에서 사용할 가상 머신의 새 볼륨을 생성하도록 요청합니다. 새 볼륨이 준비되면 **Manager**는 **VDSM**을 사용하여 가상 머신 쓰기 작업에 새 볼륨을 사용하기 시작하는 가상 시스템을 실행하는 호스트의 **libvirt** 및 **qemu**와 통신합니다. 가상 시스템이 새 볼륨에 쓸 수 있는 경우 스냅샷 작업이 성공으로 간주되고 가상 머신은 이전 볼륨에 쓰기를 중지합니다. 가상 머신을 새 볼륨에 쓸 수 없는 경우 스냅샷 작업이 실패로 간주되고 새 볼륨이 삭제됩니다.

가상 머신은 새 볼륨이 준비될 때까지 라이브 스냅샷이 시작될 때부터 현재 볼륨과 새 볼륨에 모두 액세스해야 하므로 두 볼륨 모두 읽기-쓰기 액세스 권한으로 열립니다.

**quiescing**을 지원하는 설치된 게스트 에이전트가 있는 가상 머신은 스냅샷 전체에 파일 시스템의 일관성을 보장할 수 있습니다. 등록된 **Red Hat Enterprise Linux** 게스트는 **qemu-guest-agent**를 설치하여 스냅샷 전에 정지할 수 있습니다.

스냅샷을 작성할 때 가상 머신에 호환되는 게스트 에이전트가 정지되는 경우 **VDSM**에서는 **libvirt**를 사용하여 에이전트와 통신하여 스냅샷을 준비합니다. 미결 쓰기 작업이 완료된 후 스냅샷을 생성하기 전에 파일 시스템이 동결됩니다. 스냅샷이 완료되고 **libvirt**가 디스크 쓰기 작업을 위해 가상 시스템을 새 볼륨으로 전환하면 파일 시스템이 제거되고 디스크 재개에 씁니다.

모든 라이브 스냅샷은 **quiescing**이 활성화된 상태에서 시도되었습니다. 호환되는 게스트 에이전트가 없기 때문에 스냅샷 명령이 실패하면 **use-quiescing** 플래그 없이 실시간 스냅샷이 다시 시작됩니다. **quiesced Filesystems**를 사용하여 가상 머신을 **pre-snapshot** 상태로 되돌리면 파일 시스템 검사가 필요하지 않고 정상적으로 부팅됩니다. 정지되지 않은 파일 시스템을 사용하여 이전 스냅샷을 되돌리려면 부팅 시 파일 시스템을 확인해야 합니다.

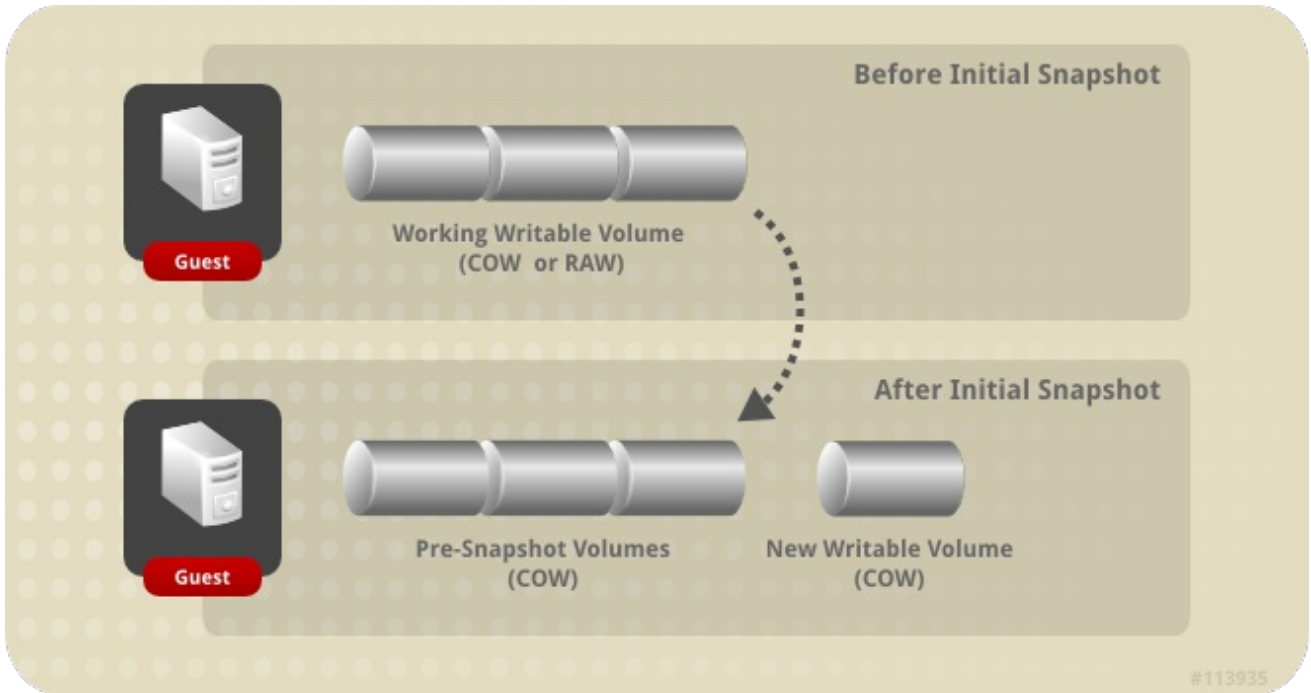
## 8.3. 스냅샷 생성

**Red Hat Virtualization**에서는 초기 스냅샷이 **QCOW2** 또는 **RAW** 형식 중 하나를 유지한다는 점에서 가상 머신의 초기 스냅샷은 후속 스냅샷과 다릅니다. 가상 머신의 첫 번째 스냅샷은 기존 볼륨을 기본 이미지로 지정합니다. 추가 스냅샷은 이전 스냅샷 이후 이미지에 저장된 데이터를 추적하는 추가 **COW** 계층입니다.



Red Hat Virtualization에서 게스트 가상 머신은 일반적으로 이미지가 썬 프로비저닝된 이미지로 생성되거나 사용자가 QCOW2로 요청하지 않는 한 RAW 디스크 이미지와 상호 작용합니다. **그림 8.1. “초기 스냅샷 생성”**에 표시된 대로 스냅샷 생성으로 인해 가상 디스크 이미지를 구성하는 볼륨이 후속 스냅샷의 기본 이미지 역할을 합니다.

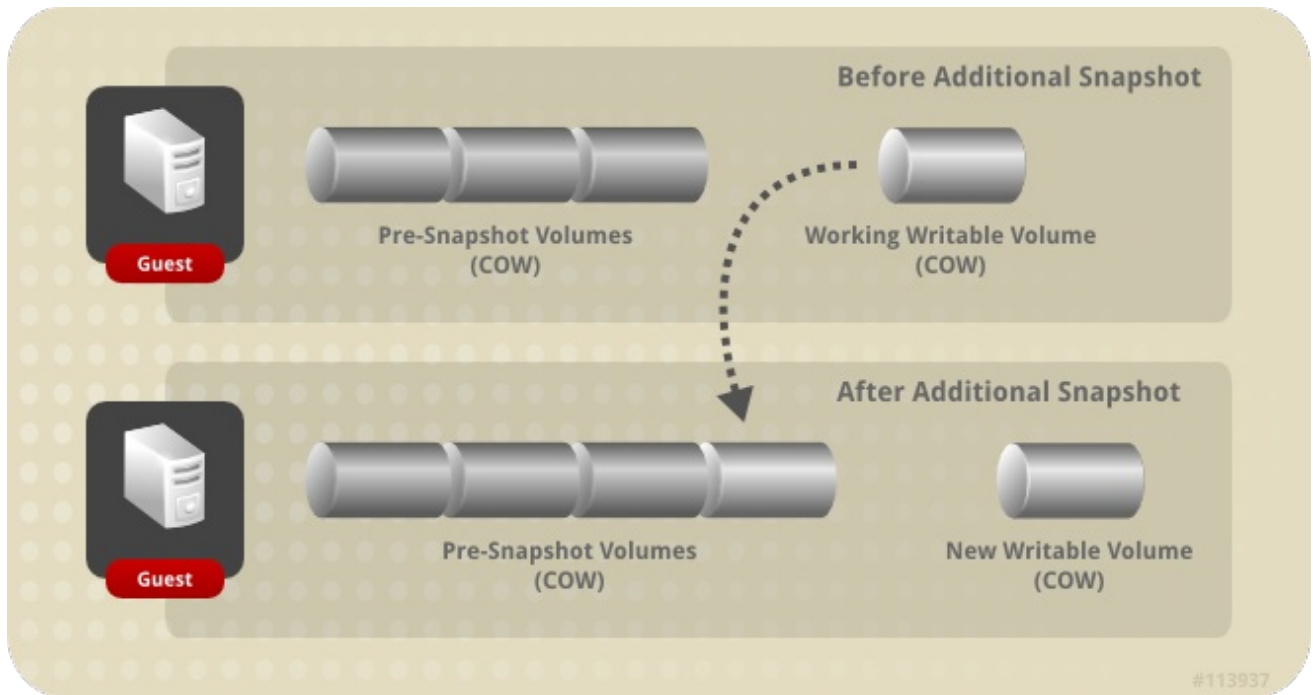
그림 8.1. 초기 스냅샷 생성



초기 스냅샷 이후 수행된 스냅샷은 스냅샷을 만든 후 생성되거나 변경된 새 COW 볼륨을 생성합니다. 각 새로운 COW 계층에는 COW 메타데이터만 포함됩니다. 스냅샷이 새 COW 계층에 기록된 후 가상 머신 사용 및 작업을 통해 생성된 데이터입니다. 이전 COW 계층에 있는 데이터를 수정하는 데 가상 머신을 사용하면 이전 계층에서 데이터를 읽고 최신 계층에 기록됩니다. 가상 머신은 가장 최근부터 가장 오래된 것, 가상 머신에 투명하게 각 COW 계층을 확인하여 데이터를 찾습니다.



그림 8.2. 추가 스냅샷 생성



#### 8.4. 스냅샷 프리뷰

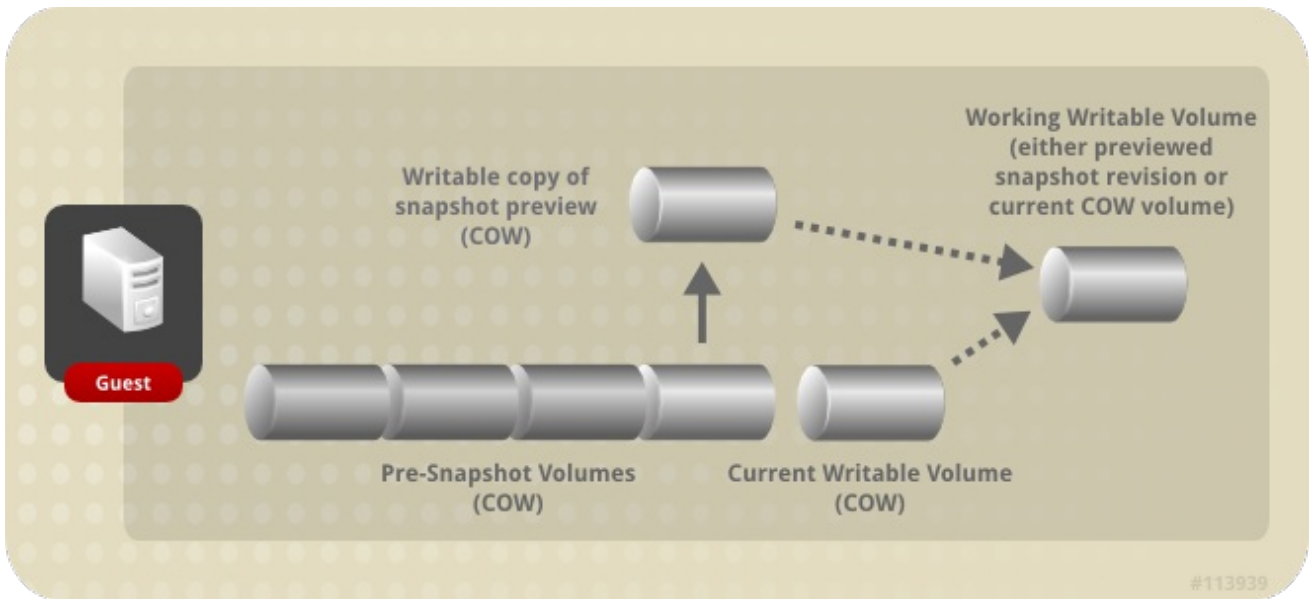
가상 디스크 이미지를 복원할 스냅샷을 선택하기 위해 관리자는 이전에 생성된 모든 스냅샷을 미리 볼 수 있습니다.

게스트당 사용 가능한 스냅샷에서 관리자는 스냅샷 볼륨을 선택하여 내용을 미리 볼 수 있습니다. [그림 8.3. “스냅샷 프리뷰”](#)에 표시된 대로 각 스냅샷은 COW 볼륨으로 저장되고 미리 보기되면 스냅샷이 미리 보기 중인 스냅샷에서 새 프리뷰 계층이 복사됩니다. 게스트는 실제 스냅샷 볼륨 대신 미리 보기와 상호 작용합니다.

관리자가 선택한 스냅샷을 미리 확인한 후 프리뷰를 커밋하여 게스트 데이터를 스냅샷에 캡처된 상태로 복원할 수 있습니다. 관리자가 프리뷰를 커밋하면 게스트가 프리뷰 계층에 연결됩니다.

스냅샷을 미리 볼 때 관리자는 **Undo**를 선택하여 확인한 스냅샷의 프리뷰 계층을 삭제할 수 있습니다. 스냅샷 자체가 포함된 계층은 프리뷰 계층이 삭제됨에도 불구하고 보존됩니다.

그림 8.3. 스냅샷 프리뷰



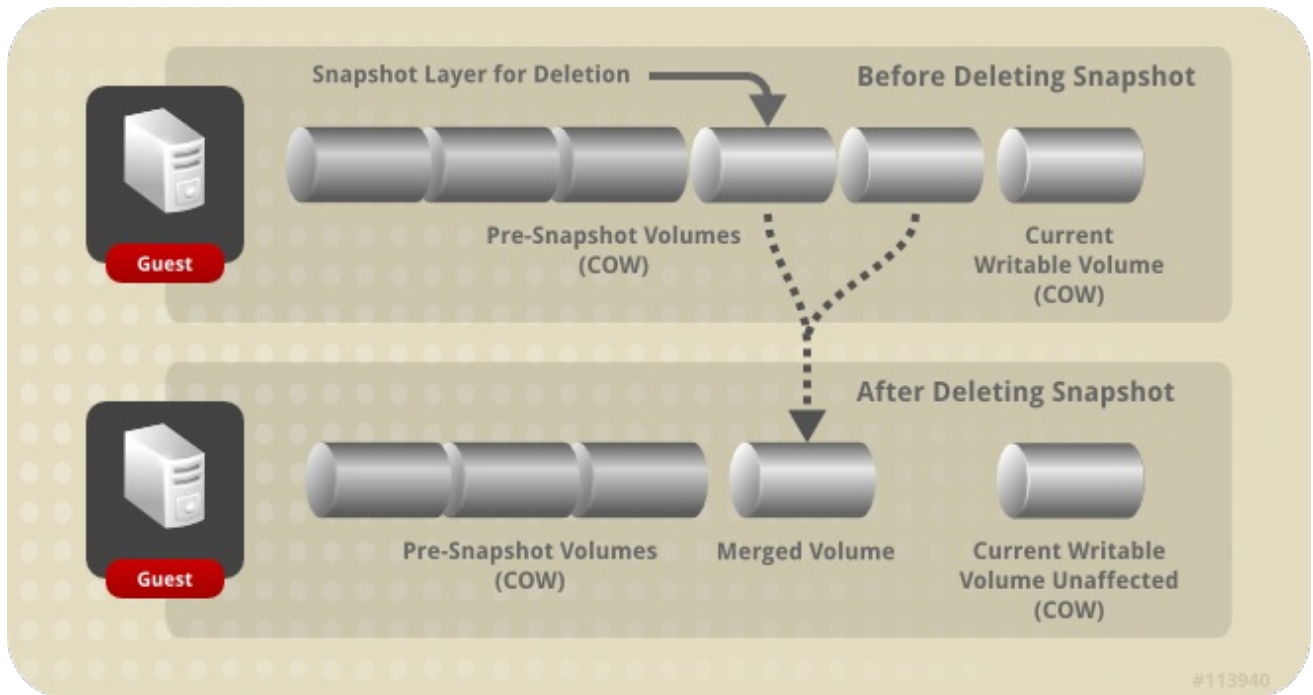
### 8.5. 스냅샷 삭제

개별 스냅샷 또는 더 이상 필요하지 않은 일련의 스냅샷을 삭제할 수 있습니다. 스냅샷을 삭제하면 가상 디스크 이미지를 특정 복원 지점으로 복원하는 기능이 없습니다. 스냅샷에서 사용하는 디스크 공간을 회수할 필요도 없으며 데이터를 삭제하지도 않습니다. 후속 스냅샷이 삭제된 스냅샷의 데이터를 덮어쓰는 경우에만 디스크 공간이 회수됩니다. 예를 들어, 5개의 스냅샷에서 세 번째 스냅샷이 삭제된 경우 세 번째 스냅샷의 변경되지 않은 데이터는 네 번째 및 5번째 스냅샷을 사용할 수 있도록 디스크에 보존되어야 하지만 네 번째 또는 다섯 번째 스냅샷이 세 번째의 데이터를 덮어쓰면 세 번째 스냅샷이 중복되고 디스크 공간을 회수할 수 있습니다. 잠재적인 디스크 공간 재요청 외에도 스냅샷을 삭제하면 가상 머신의 성능도 향상될 수 있습니다.

삭제를 위해 스냅샷을 선택하면 QEMU는 동일한 크기의 새 논리 볼륨을 생성하여 후속 스냅샷과 삭제되는 스냅샷을 병합합니다. 이 새 논리 볼륨의 크기가 두 스냅샷 간의 모든 차이를 수용하도록 크기가 조정됩니다. 새 논리 볼륨은 두 스냅샷의 총 결합된 크기일 수 있습니다. 두 개의 스냅샷이 병합되면 후속 스냅샷의 이름이 변경되고 삭제 플래그가 지정되고 이름이 지정된 새 논리 볼륨으로 교체됩니다. 원래 삭제용으로 플래그가 지정된 스냅샷과 후속 스냅샷은 모두 삭제되며, 그 대신 병합된 단일 스냅샷입니다.

예를 들어 Delete\_snapshot 스냅샷은 200GB이고 후속 스냅샷인 Next\_snapshot 은 100GB입니다. Delete\_snapshot 이 삭제되고 새 논리 볼륨이 일시적으로 이름이 Snapshot\_merge 이고 크기가 200GB 인 새 논리 볼륨이 생성됩니다. Snapshot\_merge 는 결국 Delete\_snapshot 및 Next\_snapshot 의 병합된 총 콘텐츠를 수용하기 위해 300GB로 크기를 조정합니다. 그런 다음 Next\_snapshot 의 이름이 Delete\_me\_too\_snapshot 으로 변경되어 Snapshot\_merge 의 이름을 Next\_snapshot 으로 변경할 수 있습니다. 마지막으로 Delete\_snapshot 및 Delete\_me\_too\_snapshot 이 삭제됩니다.

그림 8.4. 스냅샷 삭제



실행 중인 가상 머신에서 스냅샷을 삭제하는 데 사용되는 논리는 종료된 가상 머신의 논리와 약간 다릅니다. 실시간 스냅샷 삭제는 VDSM 이 가상 머신의 복구 파일에서 작업 레코드를 유지 관리하는 비동기 블록 작업으로 처리되므로, 작업 중 VDSM 이 다시 시작되거나 가상 머신이 종료되는 경우에도 작업을 추적할 수 있습니다. 작업이 시작되면 작업이 실패하거나 중단되어도 삭제되는 스냅샷을 복원 지점으로 미리 보거나 사용할 수 없습니다. 활성 계층을 부모와 병합하는 작업에서 작업은 활성 계층에서 상위 계층으로 복사되는 2단계 프로세스로 분할되며 디스크 쓰기는 활성 계층과 상위 계층 모두에 미리링됩니다. 마지막으로 삭제된 스냅샷의 데이터가 상위 스냅샷과 병합되고 VDSM 이 이미지 체인 전체에서 변경 사항을 동기화하면 작업이 완료된 것으로 간주됩니다.

## 9장. 하드웨어 드라이버 및 장치

### 9.1. VIRTUALIZED HARDWARE

**Red Hat Virtualization**은 가상화 게스트에 세 가지 유형의 시스템 장치를 제공합니다. 이러한 하드웨어 장치는 모두 가상화된 게스트에 물리적으로 연결된 하드웨어 장치로 표시되지만 장치 드라이버는 다른 방식으로 작동합니다.

#### 에뮬레이션된 장치

*가상 장치*라고도 하는 *에뮬레이션된 장치*는 전적으로 소프트웨어에 존재합니다. *에뮬레이션된 장치 드라이버*는 호스트에서 실행되는 운영 체제(소스 장치를 관리하는) 및 게스트에서 실행되는 운영 체제 간의 변환 계층입니다. 에뮬레이션된 장치에서 및 로 들어오는 장치 수준 지침은 하이퍼바이저에 의해 가로채기 및 변환됩니다. **Linux** 커널에서 에뮬레이션 및 인식하는 것과 동일한 유형의 장치를 에뮬레이션된 드라이버의 백업 소스 장치로 사용할 수 있습니다.

#### 반 가상화된 장치

반가상화된 장치에는 호스트 머신의 하이퍼바이저와 통신할 수 있는 인터페이스를 제공하는 게스트 운영 체제에 장치 드라이버를 설치해야 합니다. 이 인터페이스는 가상화된 환경 외부에서 디스크 I/O와 같은 집약적인 작업을 수행할 수 있도록 하는 데 사용됩니다. 이러한 방식으로 가상화에 내재된 오버헤드를 낮추는 것은 게스트 운영 체제 성능이 물리적 하드웨어에서 직접 실행될 때 예상하는 것보다 더 가깝게 허용하기 위한 것입니다.

#### 물리적 공유 장치

특정 하드웨어 플랫폼을 통해 가상화 게스트는 다양한 하드웨어 장치 및 구성 요소에 직접 액세스할 수 있습니다. 가상화의 이 프로세스를 *passthrough* 또는 *장치 할당*이라고 합니다. 통과를 통해 장치가 게스트 운영 체제에 물리적으로 연결된 것처럼 보이고 작동할 수 있습니다.

### 9.2. RED HAT VIRTUALIZATION에서 안정적인 장치 주소

가상 하드웨어 **PCI** 주소 할당은 **ovirt-engine** 데이터베이스에 유지됩니다.

**PCI** 주소는 가상 머신 생성 시 **QEMU**에 의해 할당되며 **libvirt**를 통해 **VDSM**에 보고됩니다. **VDSM**을 **Manager**에 다시 보고합니다. 여기서 **ovirt-engine** 데이터베이스에 저장됩니다.

가상 머신이 시작되면 **Manager**에서 **VDSM**에 장치 주소를 데이터베이스 외부로 보냅니다. **VDSM**은 가상 머신이 처음 실행될 때 할당된 **PCI** 장치 주소를 사용하여 가상 머신을 시작하는 **libvirt**에 전달합니다.

가상 머신에서 장치를 제거하면 안정적인 PCI 주소를 포함한 모든 참조도 제거됩니다. 제거된 장치를 교체하기 위해 장치가 추가되면 PCI 주소가 QEMU 로 할당되며 이는 교체된 장치와 동일하지 않을 수 있습니다.

### 9.3. 중앙 처리 장치(CPU)

클러스터 내의 각 호스트에는 여러 가상 CPU (vCPU)가 있습니다. 가상 CPU는 호스트에서 실행되는 게스트에 차례로 노출됩니다. 클러스터 내에서 호스트에서 노출하는 모든 가상 CPU는 Red Hat Virtualization Manager를 통해 클러스터를 처음 생성할 때 선택한 유형입니다. 클러스터 내에서 가상 CPU 유형 혼합을 수행할 수 없습니다.

사용 가능한 각 가상 CPU 유형에는 동일한 이름의 물리적 CPU에 따라 기능이 있습니다. 가상 CPU는 실제 CPU와 게스트 운영 체제의 구분이 불가능합니다.



참고

**x2APIC 지원 :**

Red Hat Enterprise Linux 7 호스트에서 제공하는 모든 가상 CPU 모델에는 x2APIC 지원이 포함되어 있습니다. 이는 하드웨어 인터럽트를 더 잘 처리할 수 있는 고급 프로그램 가능 APIC(Interrupt Controller)를 제공합니다.

### 9.4. 시스템 장치

게스트를 실행하는 데 시스템 장치가 중요하므로 제거할 수 없습니다. 게스트에 연결된 각 시스템 장치는 사용 가능한 PCI 슬롯도 필요로 합니다. 기본 시스템 장치는 다음과 같습니다.

- 호스트 브리지
- ISA 브리지 및 USB 브리지(USB 및 ISA 브리지는 동일한 장치임)
- 그래픽 카드(Cirrus 또는 qxl 드라이버 사용)
- 메모리 balloon 장치입니다.

## 9.5. 네트워크 장치

**Red Hat Virtualization**은 세 가지 유형의 네트워크 인터페이스 컨트롤러를 게스트에 노출할 수 있습니다. 게스트에 노출할 네트워크 인터페이스 컨트롤러 유형은 게스트를 생성할 때 선택되지만 **Red Hat Virtualization Manager**에서 변경할 수 있습니다.

- **e1000** 네트워크 인터페이스 컨트롤러는 가상화된 **Intel PRO/1000(e1000)**을 게스트에 노출합니다.
- **virtio** 네트워크 인터페이스 컨트롤러는 게스트에 반가상화 네트워크 장치를 노출합니다.
- **rtl8139** 네트워크 인터페이스 컨트롤러는 가상화된 **Realtek Semiconductor Corp RTL8139**를 게스트에 노출합니다.

게스트당 여러 네트워크 인터페이스 컨트롤러가 허용됩니다. 각 컨트롤러가 추가된 각 컨트롤러는 게스트에서 사용 가능한 **PCI** 슬롯을 사용합니다.

## 9.6. 그래픽 장치

에뮬레이션된 그래픽 장치가 두 개 제공됩니다. 이러한 장치는 **SPICE** 프로토콜 또는 **VNC**를 사용하여 연결할 수 있습니다.

- **ac97** 은 **Cirrus CLGD 5446 PCI VGA** 카드를 에뮬레이션합니다.
- **vga** 는 **BochsVESA** 확장 (모든 비표준 모드 포함)으로 더미 **VGA** 카드를 에뮬레이션합니다.

## 9.7. 스토리지 장치

스토리지 장치 및 스토리지 풀은 블록 장치 드라이버를 사용하여 스토리지 장치를 가상화된 게스트에 연결할 수 있습니다. 스토리지 드라이버는 스토리지 장치가 아닙니다. 드라이버는 백업 스토리지 장치, 파

일 또는 스토리지 풀 볼륨을 가상화된 게스트에 연결하는 데 사용됩니다. 백업 스토리지 장치는 지원되는 모든 유형의 스토리지 장치, 파일 또는 스토리지 풀 볼륨일 수 있습니다.

- **IDE** 드라이버는 게스트에 에뮬레이션된 블록 장치를 노출합니다. 에뮬레이션된 **IDE** 드라이버는 최대 4개의 가상화된 **IDE** 하드 디스크 또는 가상화된 **IDE CD-ROM** 드라이브를 각 가상화 게스트에 연결하는 데 사용할 수 있습니다. 에뮬레이션된 **IDE** 드라이버는 가상화된 **DVD-ROM** 드라이브를 제공하는 데도 사용됩니다.
- **VirtIO** 드라이버는 반 가상화된 블록 장치를 게스트에 노출합니다. 반 가상화된 블록 드라이버는 가상화된 게스트에 연결된 하이퍼바이저에서 지원하는 모든 스토리지 장치에 대한 드라이버입니다(미러링해야 하는 유동 디스크 드라이브 제외).

## 9.8. 동영상 장치

두 개의 에뮬레이션된 음성 장치를 사용할 수 있습니다.

- **ac97** 은 Intel 82801AA AC97의 AC97 호환 가능한 표시 카드를 에뮬레이션합니다.
- **es1370** 은 ENSONIQ audioPCI ES1370 표시 카드를 에뮬레이션합니다.

## 9.9. 직렬 드라이버

반가상화 직렬 드라이버(**virtio-serial**)는 바이트 스트림 지향적 문자 스트림 드라이버입니다. 반 가상화된 직렬 드라이버는 네트워킹을 사용할 수 없거나 사용할 수 없는 게스트의 사용자 공간과 호스트의 사용자 공간 간의 간단한 통신 인터페이스를 제공합니다.

## 9.10. BALLOON 드라이버

**balloon** 드라이버를 사용하면 게스트가 필요한 메모리 양을 하이퍼바이저에 표시할 수 있습니다. **balloon** 드라이버를 사용하면 호스트가 게스트에 효율적으로 및 메모리를 할당하고 여유 메모리를 다른 게스트 및 프로세스에 할당할 수 있습니다.

**balloon** 드라이버를 사용하는 게스트는 게스트 **RAM**의 섹션을 사용하지 않는 것으로 표시할 수 있습니다(**balloon inflation**). 하이퍼바이저는 메모리를 비우고 해당 호스트의 다른 호스트 프로세스 또는 기타 게스트에 메모리를 사용할 수 있습니다. 게스트에 사용 가능한 메모리가 다시 필요한 경우 하이퍼바이저는 **RAM**을 게스트에 재할당할 수 있습니다(**balloon deflation**).

## 10장. 최소 요구 사항 및 기술 제한

### 10.1. 최소 요구 사항 및 지원 제한

Red Hat Virtualization 환경에 적용되는 물리적 및 논리적 제한 사항이 많이 있습니다. 이러한 제한 사항 이외의 구성이 있는 환경은 현재 지원되지 않습니다.

### 10.2. 리소스 제한

특정 제한 사항은 스토리지 도메인 및 호스트와 같은 리소스에 적용됩니다.

표 10.1. 리소스 제한

항목	제한
스토리지 도메인	데이터 센터당 최소 2개의 스토리지 도메인을 사용하는 것이 좋습니다. <ul style="list-style-type: none"> <li>● 데이터 스토리지 도메인이 필요합니다.</li> <li>● ISO 스토리지 도메인을 사용하는 것이 좋습니다.</li> </ul>
호스트	Red Hat은 Red Hat Virtualization Manager당 최대 200개의 호스트를 지원합니다.

### 10.3. 클러스터 제한

클러스터는 가상 머신 세트의 리소스 풀로 취급되는 물리적 호스트 집합입니다. 클러스터의 호스트는 동일한 네트워크 인프라와 동일한 스토리지를 공유합니다. 클러스터는 가상 머신을 호스트에서 호스트로 이동할 수 있는 마이그레이션 도메인입니다. 안정성을 보장하기 위해 각 클러스터에 몇 가지 제한 사항이 적용됩니다.

- 모든 관리형 하이퍼바이저는 클러스터에 있어야 합니다.
- 클러스터 내의 모든 관리형 하이퍼바이저는 동일한 CPU 유형을 보유해야 합니다. Intel 및 AMD CPU는 동일한 클러스터 내에서 공존할 수 없습니다.





## 참고

클러스터에 대한 자세한 내용은 *관리 가이드*의 클러스터를 참조하십시오.

## 10.4. 스토리지 도메인 제한

스토리지 도메인은 가상 디스크 이미지 및 ISO 이미지의 스토리지와 가상 시스템의 가져오기 및 내보내기를 위한 공간을 제공합니다. 지정된 데이터 센터 내에 많은 스토리지 도메인을 생성할 수 있지만 각 스토리지 도메인에 적용되는 여러 제한 사항 및 권장 사항이 있습니다.

표 10.2. 스토리지 도메인 제한

항목	제한
스토리지 유형	<p>지원되는 스토리지 유형은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>● 파이버 채널 프로토콜(FCP)</li> <li>● iSCSI(Internet Small Computer System Interface)</li> <li>● 네트워크 파일 시스템(NFS)</li> <li>● POSIX 호환 파일 시스템(POSIX)</li> <li>● Red Hat Gluster Storage(GlusterFS)</li> </ul> <p>Red Hat Virtualization 4.0의 새로운 ISO 및 내보내기 스토리지 도메인은 모든 파일 기반 스토리지(NFS, Posix 또는 GlusterFS)에서 제공할 수 있습니다.</p>
논리 단위 번호(LUN)	<p>iSCSI 또는 FCP에서 제공하는 각 스토리지 도메인에 대해 300개 이상의 LUN이 허용되지 않습니다.</p>
논리 볼륨(LV)	<p>Red Hat Virtualization에서 논리 볼륨은 가상 머신, 템플릿 및 가상 머신 스냅샷의 가상 디스크를 나타냅니다.</p> <p>iSCSI 또는 FCP에서 제공하는 각 스토리지 도메인에 350개 이상의 논리 볼륨이 권장됩니다. 지정된 스토리지 도메인의 논리 볼륨 수가 이 수를 초과하면 각각 350개 이상의 논리 볼륨이 없는 별도의 스토리지 도메인으로 사용 가능한 스토리지를 분할하는 것이 좋습니다.</p> <p>이 제한의 근본 원인은 LVM 메타데이터의 크기입니다. 논리 볼륨의 수가 증가함에 따라 해당 논리 볼륨과 연결된 LVM 메타데이터도 증가합니다. 이 메타데이터의 크기가 1MB를 초과하면 새 디스크 또는 스냅샷 생성과 같은 프로비저닝 작업의 성능이 저하되고 QCOW 디스크를 실행하는 데 시간이 오래 걸리는 경우 논리 볼륨을 쉼 프로비저닝하는 데 필요한 시간이 단축됩니다.</p> <p>논리 볼륨에 대한 자세한 내용은 에서 <a href="https://access.redhat.com/solutions/441203">https://access.redhat.com/solutions/441203</a> 확인할 수 있습니다.</p>



참고

[스토리지](#) 도메인에 대한 자세한 내용은 [관리 가이드](#)의 스토리지를 참조하십시오.

10.5. RED HAT VIRTUALIZATION MANAGER 제한 사항

Red Hat Virtualization Manager 서버는 Red Hat Enterprise Linux 7을 실행해야 합니다. 또한 여러 추가 하드웨어 요구 사항도 충족해야 합니다.

표 10.3. Red Hat Virtualization Manager 제한 사항

항목	제한
RAM	<ul style="list-style-type: none"> <li>최소 4GB의 RAM이 필요합니다.</li> </ul>
PCI 장치	<ul style="list-style-type: none"> <li>최소 대역폭이 1Gbps인 하나 이상의 네트워크 컨트롤러가 권장됩니다.</li> </ul>
스토리지	<ul style="list-style-type: none"> <li>최소 25GB의 사용 가능한 로컬 디스크 공간이 권장됩니다.</li> </ul>



참고

Red Hat Virtualization Manager에 대한 자세한 내용은 [설치 가이드](#)를 참조하십시오.

10.6. 하이퍼바이저 요구 사항

RHVH(Red Hat Virtualization Host)에는 여러 하드웨어 요구 사항과 지원되는 제한이 있습니다. Red Hat Enterprise Linux 호스트의 스토리지 요구 사항은 기존 구성에서 사용하는 디스크 공간의 크기에 따라 다르지만 RHVH보다 커야 합니다.

표 10.4. Red Hat Virtualization 호스트 요구 사항 및 지원 제한

항목	지원 제한
----	-------

항목	지원 제한
CPU	<p>최소 1개의 물리적 CPU가 필요합니다. Red Hat Virtualization은 호스트에서 이러한 CPU 모델의 사용을 지원합니다.</p> <ul style="list-style-type: none"> <li>● AMD Opteron G1</li> <li>● AMD Opteron G2</li> <li>● AMD Opteron G3</li> <li>● AMD Opteron G4</li> <li>● AMD Opteron G5</li> <li>● Intel Conroe</li> <li>● Intel Penryn</li> <li>● Intel Nehalem</li> <li>● Intel Westmere</li> <li>● Intel Haswell</li> <li>● Intel SandyBridge 제품군</li> <li>● IBM POWER 8</li> </ul> <p>모든 CPU 는 Intel® 64 또는 AMD64 CPU 확장 기능을 지원하고 AMD-V™ 또는 Intel VT® 하드웨어 가상화 확장 기능이 활성화되어 있어야 합니다. <b>No eXecute</b> 플래그 (NX) 지원도 필요합니다.</p>
RAM	<p>각 가상 시스템에 필요한 RAM의 크기는 다음 사항에 따라 다릅니다.</p> <ul style="list-style-type: none"> <li>● 게스트 운영 체제 요구 사항</li> <li>● 게스트 애플리케이션 요구 사항 및</li> <li>● 가상 시스템의 메모리 활동 및 사용량.</li> </ul> <p>또한 KVM 은 가상 머신에 대해 물리적 RAM 을 초과 커밋할 수 있습니다. 필요에 따라 가상 머신의 RAM 을 할당하고 활용도가 낮은 가상 머신을 스왑으로 전환하여 이를 수행합니다.</p> <p>지원되는 최대 RAM 및 최소 RAM은 을 참조하십시오 <a href="https://access.redhat.com/articles/rhel-limits">https://access.redhat.com/articles/rhel-limits</a>.</p>

항목	지원 제한
스토리지	<p>호스트에 지원되는 최소 내부 스토리지는 다음 목록의 합계입니다.</p> <ul style="list-style-type: none"> <li>● 루트(/) 파티션에는 최소 6GB의 스토리지가 필요합니다.</li> <li>● /boot 파티션에는 최소 1GB의 스토리지가 필요합니다.</li> <li>● /var 파티션에는 최소 15GB의 스토리지가 필요합니다. 셀프 호스트 엔진 배포의 경우 최소 60GB여야 합니다.</li> <li>● 스왑 파티션에는 최소 8MB의 스토리지가 필요합니다. 스왑 파티션의 권장 크기는 호스트가 설치된 시스템과 환경에 대해 예상되는 오버 커밋 수준에 따라 다릅니다. 자세한 내용은 <a href="https://access.redhat.com/solutions/15244">https://access.redhat.com/solutions/15244</a> 을 참조하십시오.</li> </ul> <p>호스트 설치를 위한 최소 스토리지 요구 사항은 다음과 같습니다. 스토리지 공간을 더 많이 사용하는 기본 할당을 사용하는 것이 좋습니다.</p>
PCI 장치	<p>최소 대역폭이 1Gbps인 경우 하나 이상의 네트워크 컨트롤러가 필요합니다.</p>

## 중요

**Red Hat Virtualization Host**가 부팅되면 다음과 같은 메시지가 나타날 수 있습니다.

```
Virtualization hardware is unavailable.
(No virtualization hardware was detected on this system)
```

이 경고는 가상화 확장 프로그램이 비활성화되었거나 프로세서에 존재하지 않음을 나타냅니다. **CPU**가 나열된 확장을 지원하고 시스템 **BIOS**에서 활성화되어 있는지 확인합니다.

프로세서에 가상화 확장 기능이 있고 활성화되어 있는지 확인하려면 다음을 수행하십시오.

- 호스트 부팅 화면에서 아무 키나 눌러서 직렬 콘솔로 **Boot** 또는 **Boot with serial console** 항목을 선택합니다. **Tab** 을 눌러 선택한 옵션의 커널 매개 변수를 편집합니다. 마지막 커널 매개 변수가 나열된 후 **Space** 가 있는지 확인하고 **rescue** 매개 변수를 추가합니다.
- **Enter** 를 눌러 복구 모드로 부팅합니다.
- 표시되는 프롬프트에서 프로세서에 가상화 확장 기능이 있고 다음 명령을 실행하여 활성화되는지 확인합니다.

```
# grep -E 'svm|vmx' /proc/cpuinfo
```

출력이 표시되면 프로세서는 하드웨어 가상화가 가능합니다. 출력이 표시되지 않는 경우에도 프로세서가 하드웨어 가상화를 지원할 수 있습니다. 일부 상황에서는 제조업체가 **BIOS**에서 가상화 확장을 비활성화합니다. 이 경우 제조업체가 제공하는 시스템의 **BIOS** 및 마더보드 설명서를 참조하십시오.

- 추가 검사로 **kvm** 모듈이 커널에 로드되었는지 확인합니다.

```
# lsmod | grep kvm
```

출력에 **kvm\_intel** 또는 **kvm\_amd** 가 포함된 경우 **kvm** 하드웨어 가상화 모듈이 로드되고 시스템이 요구 사항을 충족합니다.

### 10.7. 게스트 요구 사항 및 지원 제한

RHVH(Red Hat Virtualization Host)에서 실행되는 게스트에 다음과 같은 요구 사항 및 지원 제한이 적용됩니다.

표 10.5. Virtualized Hardware

항목	제한
CPU	Red Hat Enterprise Linux 7에서는 게스트당 최대 4.6.1개의 가상화된 CPU가 지원됩니다.
RAM	게스트마다 다른 RAM 요구 사항이 있습니다. 각 게스트에 필요한 RAM의 양은 게스트 운영 체제의 요구 사항과 게스트가 작동하는 로드 에 따라 다릅니다.  게스트 머신에 대해 지원되는 최대 RAM 및 최소 RAM은 을 참조하십시오 <a href="https://access.redhat.com/articles/rhel-kvm-limits">https://access.redhat.com/articles/rhel-kvm-limits</a> .
PCI 장치	게스트당 최대 31개의 가상화된 PCI 장치가 지원됩니다. 여러 시스템 장치는 이 제한에 대해 개수를 계산하며, 일부는 필수입니다. PCI 장치 제한에 대해 계산하는 필수 장치에는 PCI 호스트 브리지, ISA 브리지, USB 브리지, 보드 브리지, 그래픽 카드 및 IDE 또는 VirtIO 블록 장치가 포함됩니다.
스토리지	게스트당 최대 28개의 가상화 스토리지 장치가 지원되며, 가능한 3개의 IDE와 25개의 Virtio로 구성됩니다.

### 10.8. SPICE 제한

SPICE는 현재 2560x1600의 최대 해상도를 지원합니다.

### 10.9. 참고 자료

이러한 추가 문서 리소스는 Red Hat Virtualization 설명서 제품군의 일부가 아닙니다. 하지만 Red Hat Virtualization 환경을 관리하는 시스템 관리자에게 유용한 정보가 포함되어 있으며, 에서 <https://access.redhat.com/documentation/en/red-hat-enterprise-linux/> 확인할 수 있습니다.

#### Red Hat Enterprise Linux - 시스템 관리자 가이드

Red Hat Enterprise Linux의 배포, 구성 및 관리에 대한 가이드입니다.

#### Red Hat Enterprise Linux - DM-Multipath 가이드

---

**Red Hat Enterprise Linux에서 장치-Mapper Multipathing 사용에 대한 가이드입니다.**

***Red Hat Enterprise Linux - 설치 가이드***

**Red Hat Enterprise Linux 설치에 대한 가이드입니다.**

***Red Hat Enterprise Linux - 스토리지 관리 가이드***

**Red Hat Enterprise Linux에서 스토리지 장치 및 파일 시스템 관리에 대한 가이드입니다.**

***Red Hat Enterprise Linux - 가상화 배포 및 관리 가이드***

**Red Hat Enterprise Linux에서 가상화 기술의 설치, 구성, 관리 및 문제 해결에 대한 가이드입니다.**