



# Red Hat Virtualization 4.1

## 기술 참조

Red Hat Virtualization 환경의 기술 아키텍처



# Red Hat Virtualization 4.1 기술 참조

---

Red Hat Virtualization 환경의 기술 아키텍처

Red Hat Virtualization Documentation Team

Red Hat Customer Content Services

[rhev-docs@redhat.com](mailto:rhev-docs@redhat.com)

## 법적 공지

Copyright © 2016 Red Hat.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

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

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

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

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 초록

이 참조 문서에서는 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. MANAGER에 액세스하기 위한 인터페이스                   | 6         |
| 1.4. MANAGER를 지원하는 구성 요소                       | 7         |
| 1.5. 스토리지                                      | 8         |
| 1.6. 네트워크                                      | 9         |
| 1.7. 데이터 센터                                    | 12        |
| <b>2장. 스토리지</b> .....                          | <b>13</b> |
| 2.1. 스토리지 도메인 개요                               | 13        |
| 2.2. 스토리지가 지원되는 스토리지 도메인 유형                    | 13        |
| 2.3. 스토리지 도메인 유형                               | 14        |
| 2.4. 가상 디스크 이미지의 저장소 형식                        | 14        |
| 2.5. 가상 디스크 이미지 스토리지 할당 정책                     | 14        |
| 2.6. RED HAT VIRTUALIZATION에서의 스토리지 메타데이터 버전   | 15        |
| 2.7. RED HAT VIRTUALIZATION에서 스토리지 도메인 자동 복구   | 16        |
| 2.8. SPM (STORAGE POOL MANAGER)                | 16        |
| 2.9. STORAGE POOL MANAGER 선택 프로세스              | 17        |
| 2.10. RED HAT VIRTUALIZATION의 단독 리소스 및 SANLOCK | 18        |
| 2.11. 씬 프로비저닝 및 스토리지 오버커밋                      | 19        |
| 2.12. 논리 볼륨 확장                                 | 19        |
| <b>3장. 네트워크</b> .....                          | <b>21</b> |
| 3.1. 네트워크 아키텍처                                 | 21        |
| 3.2. 소개: 기본 네트워킹 용어                            | 21        |
| 3.3. NETWORK INTERFACE CONTROLLER              | 21        |
| 3.4. 브리지                                       | 21        |
| 3.5. 본딩  | 22        |
| 3.6. 본딩을 위한 스위치 설정                             | 23        |
| 3.7. 가상 네트워크 인터페이스 카드 (VNIC)                   | 23        |
| 3.8. 가상 LAN (VLAN)                             | 25        |
| 3.9. 네트워크 레이블                                  | 25        |
| 3.10. 클러스터 네트워킹                                | 26        |
| 3.11. 논리 네트워크                                  | 27        |
| 3.12. 필수 네트워크, 선택적 네트워크, 가상 머신 네트워크            | 28        |
| 3.13. 가상 머신 연결                                 | 29        |
| 3.14. 포트 미러링                                   | 29        |
| 3.15. 호스트 네트워크 설정                              | 29        |
| 3.16. 브리지 설정                                   | 30        |
| 3.17. VLAN 설정                                  | 30        |
| 3.18. 브리지 및 본딩 설정                              | 31        |
| 3.19. 다중 브리지, 다중 VLAN, 및 NIC 설정                | 32        |
| 3.20. 다중 브리지, 다중 VLAN, 및 본딩 설정                 | 32        |
| <b>4장. 전원 관리</b> .....                         | <b>34</b> |
| 4.1. 전원 관리 및 펜싱 소개                             | 34        |
| 4.2. RED HAT VIRTUALIZATION에서 프록시를 통해 전원 관리    | 34        |
| 4.3. 전원 관리                                     | 34        |
| 4.4. 펜싱  | 35        |
| 4.5. 호스트 소프트웨어 펜싱                              | 36        |
| 4.6. 여러 전원 관리 펜싱 에이전트 사용                       | 36        |

|  |           |
|--|-----------|
| <b>5장. 로드 밸런싱, 스케줄링, 그리고 마이그레이션</b> .....            | <b>38</b> |
| 5.1. 로드 밸런싱, 스케줄링, 그리고 마이그레이션                        | 38        |
| 5.2. 로드 밸런싱 정책                                       | 38        |
| 5.3. 로드 밸런싱 정책: VM_EVENLY_DISTRIBUTED                | 38        |
| 5.4. 로드 밸런싱 정책: EVENLY_DISTRIBUTED                   | 38        |
| 5.5. 로드 밸런싱 정책: POWER_SAVING                         | 39        |
| 5.6. 로드 밸런싱 정책: NONE                                 | 39        |
| 5.7. 고가용성 가상 머신 예약                                   | 39        |
| 5.8. 스케줄링  | 39        |
| 5.9. 마이그레이션  | 40        |
| <b>6장. 디렉토리 서비스</b> .....                            | <b>41</b> |
| 6.1. 디렉토리 서비스  | 41        |
| 6.2. 로컬 인증: 내부 도메인                                   | 41        |
| 6.3. GSSAPI를 사용한 원격 인증                               | 41        |
| <b>7장. 템플릿 및 풀</b> .....                             | <b>43</b> |
| 7.1. 템플릿 및 풀   | 43        |
| 7.2. 템플릿   | 43        |
| 7.3. 풀   | 44        |
| <b>8장. 가상 머신 스냅샷</b> .....                           | <b>45</b> |
| 8.1. 스냅샷   | 45        |
| 8.2. RED HAT VIRTUALIZATION에서 실시간 스냅샷                | 45        |
| 8.3. 스냅샷 생성  | 46        |
| 8.4. 스냅샷 미리 보기                                       | 47        |
| 8.5. 스냅샷 삭제  | 48        |
| <b>9장. 하드웨어 드라이버 및 장치</b> .....                      | <b>50</b> |
| 9.1. 가상화 하드웨어  | 50        |
| 9.2. RED HAT VIRTUALIZATION에서 변경할 수 없는(STABLE) 장치 주소 | 50        |
| 9.3. CPU (CENTRAL PROCESSING UNIT)                   | 50        |
| 9.4. 시스템 장치  | 51        |
| 9.5. 네트워크 장치   | 51        |
| 9.6. 그래픽 장치  | 51        |
| 9.7. 스토리지 장치   | 51        |
| 9.8. 사운드 장치  | 52        |
| 9.9. 시리얼 드라이버  | 52        |
| 9.10. 벌룬 드라이버  | 52        |
| <b>10장. 최소 요구 사항 및 기술 제한 사항</b> .....                | <b>53</b> |
| 10.1. 최소 요구 사항 및 지원 제한 사항                            | 53        |
| 10.2. 데이터 센터 제한 사항                                   | 53        |
| 10.3. 클러스터 제한 사항                                     | 53        |
| 10.4. 스토리지 도메인 제한 사항                                 | 53        |
| 10.5. RED HAT VIRTUALIZATION MANAGER 제한 사항           | 54        |
| 10.6. 하이퍼바이저 요구 사항                                   | 55        |
| 10.7. 게스트 요구 사항 및 지원 제한 사항                           | 58        |
| 10.8. SPICE 제한 사항                                    | 58        |
| 10.9. 추가 참조  | 58        |



# 1장. 소개

## 1.1. RED HAT VIRTUALIZATION MANAGER

Red Hat Virtualization Manager를 사용하여 가상화된 환경의 관리를 중앙화할 수 있습니다. 여러 다른 인터페이스를 사용해서 Red Hat Virtualization Manager에 액세스할 수 있습니다. 각 인터페이스마다 다른 방식으로 가상화된 환경에 대한 액세스를 가능하게 해줍니다.

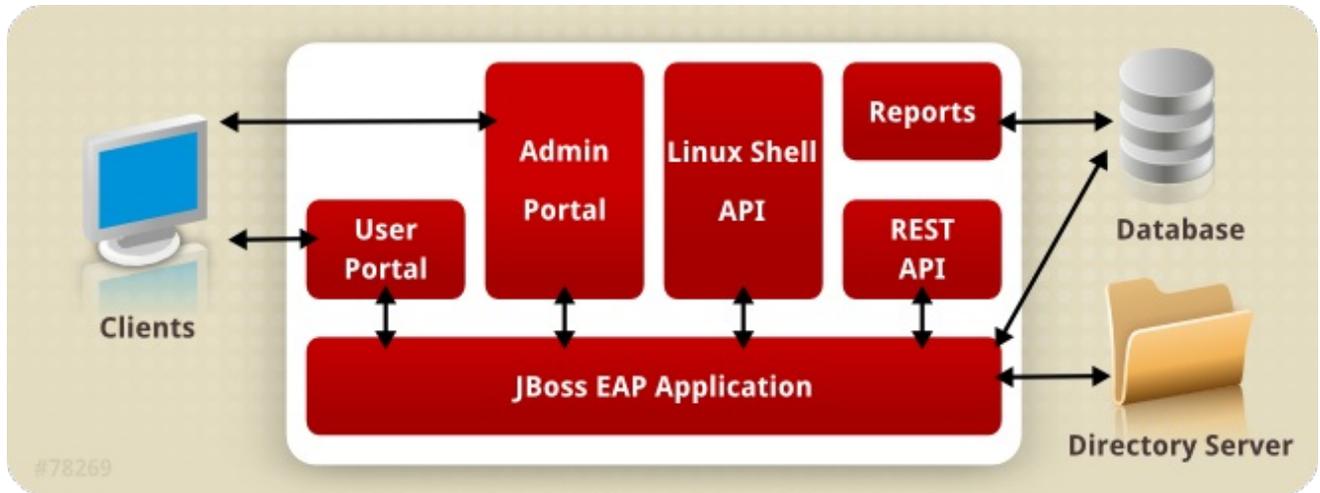


그림 1.1. Red Hat Virtualization Manager 아키텍처

Red Hat Virtualization Manager에서는 그래픽 인터페이스 및 *애플리케이션 프로그래밍 인터페이스(API)*를 제공합니다. 각 인터페이스는 *Red Hat JBoss Enterprise Application Platform*에 내장된 인스턴스에서 제공하는 Manager 애플리케이션에 연결합니다. Red Hat JBoss Enterprise Application Platform에 더해 Red Hat Virtualization Manager를 지원하는 구성 요소가 여러 가지 있습니다.

## 1.2. RED HAT VIRTUALIZATION HOST

Red Hat Virtualization 환경에는 하나 이상의 호스트가 연결되어 있습니다. 호스트란 가상 머신이 활용할 수 있는 물리적 하드웨어를 제공하는 서버를 말합니다.

Red Hat Virtualization Host(RHVM)가 실행하는 최적화된 운영 체제는 가상화 호스트 생성을 위해 만들어진 특수 맞춤형 설치 미디어를 사용해서 설치됩니다.

Red Hat Enterprise Linux 호스트란 일반적인 Red Hat Enterprise Linux 운영 체제의 설치 후 호스트로 사용되도록 설정된 서버를 말합니다.

위의 두 가지 방법으로 설치된 호스트는 모두 결과적으로 같은 방식으로 가상화된 환경과 통신하므로 둘 다 호스트라고 부릅니다.

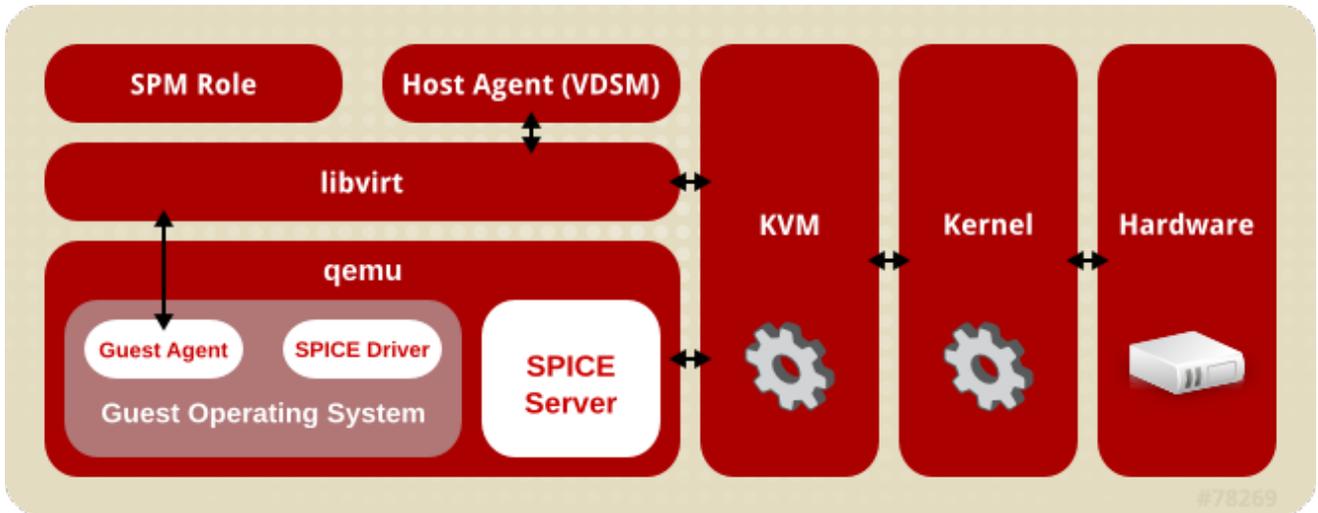


그림 1.2. 호스트 아키텍처

### 커널 기반 가상 머신 (KVM)

커널 기반 가상 머신(KVM)이란 Intel VT 또는 AMD-V 하드웨어 확장을 사용하여 완전한 가상화를 제공하는 로딩 가능한 커널 모델을 말합니다. KVM 자체는 커널 공간에서 실행되지만 KVM 상에서 실행되는 게스트는 사용자 공간에서 개별 QEMU 프로세스로 실행됩니다. 호스트가 KVM을 사용하면 가상 머신이 물리적 하드웨어를 사용할 수 있게 됩니다.

### QEMU

QEMU란 전체 시스템 에뮬레이션을 제공하는 데 사용되는 멀티 플랫폼 에뮬레이터입니다. QEMU를 사용하여 예를 들어 하나 이상의 프로세서와 주변 기기를 포함하는 PC와 같은 전체 시스템을 에뮬레이트합니다. QEMU를 사용하여 다른 운영 체제를 실행하거나 시스템 코드를 디버깅할 수도 있습니다. KVM 및 적절한 가상화 확장을 가진 프로세서와 함께 QEMU를 사용하여 완전한 하드웨어 기반 가상화가 제공됩니다.

### Red Hat Virtualization Manager 호스트 에이전트, VDSM

Red Hat Virtualization에서 VDSM은 가상 머신 및 스토리지에서 작업을 시작합니다. 또한 VDSM은 호스트 간 소통을 원활하게 해줍니다. VDSM은 메모리, 스토리지, 네트워킹 등과 같은 호스트 리소스를 감시합니다. 이에 더해 VDSM은 가상 머신 생성, 통계자료 축적, 로그 수집 등의 작업도 관리합니다. VDSM 인스턴스가 각 호스트에서 실행되며 재설정 가능한 54321 포트를 사용하여 Red Hat Virtualization Manager에서 관리 명령을 받습니다.

### VDSM-REG

VDSM은 VDSM-REG를 사용하여 각 호스트를 Red Hat Virtualization Manager에 등록합니다. VDSM-REG는 80 포트 또는 443 포트를 사용하여 그 자체와 호스트에 대한 정보를 제공합니다.

### libvirt

Libvirt는 가상 머신 및 관련 가상 장치의 관리를 원활하게 해줍니다. Red Hat Virtualization Manager가 가상 머신 생명 주기 명령(시작, 중지, 재부팅 등)을 시작하면 VDSM이 관련 호스트 시스템에서 libvirt를 호출하여 이러한 명령을 실행하도록 합니다.

### Storage Pool Manager, SPM

Storage Pool Manager(SPM)는 데이터 센터의 한 호스트에 할당되는 역할입니다. SPM 호스트는 데이터 센터에서 모든 스토리지 도메인 구조 메타데이터를 변경할 수 있는 유일한 권한이 있습니다. 변경 가능한 내용에는 가상 디스크 이미지, 스냅샷, 그리고 템플릿의 생성, 삭제, 그리고 수정 등이 포함됩니다.

다. 또한 스파스 블록 장치의 스토리지를 **Storage Area Network(SAN)**에 할당하는 것도 포함됩니다. **SPM** 역할은 데이터 센터 내의 아무 호스트에 마이그레이션할 수 있습니다. 그러므로 데이터 센터 내의 모든 호스트는 데이터 센터에서 정의된 모든 스토리지 도메인에 액세스할 수 있어야 합니다.

**Red Hat Virtualization Manager**는 **SPM**이 항상 사용 가능한지 확인합니다. 스토리지 연결 문제 발생 시 **Manager**는 **SPM** 역할을 다른 호스트에 다시 배정합니다.

## 게스트 운영 체제

게스트 운영 체제는 **Red Hat Virtualization** 환경에서 수정하지 않고 가상 머신에 설치할 수 있습니다. 게스트 운영 체제 및 설치된 모든 애플리케이션은 가상화된 환경이라는 것을 인식하지 못하며 정상적으로 작동합니다.

**Red Hat**에서 제공하는 개선된 장치 드라이버를 사용하여 더 빠르고 효율적으로 가상화 장치에 액세스할 수 있습니다. 또한 게스트에 **Red Hat Virtualization** 게스트 에이전트 설치 시 관리 콘솔에 개선된 게스트 정보를 제공합니다.

## 1.3. MANAGER에 액세스하기 위한 인터페이스

### 사용자 포털

사용자는 데스크탑 가상화를 통해 개인 **PC** 데스크탑 환경과 비슷한 데스크탑 환경을 경험할 수 있습니다. 사용자 포털은 사용자에게 *가상 데스크탑 인프라*를 제공하는 데 사용됩니다. 사용자는 웹 브라우저를 통해 사용자 포털에 액세스하며 자신에게 배정된 가상 데스크탑을 표시 및 액세스할 수 있습니다. 사용자 포털에서 사용자가 수행할 수 있는 작업은 시스템 관리자가 설정합니다. 일반 사용자는 시스템 관리자가 배정한 가상 데스크탑을 시작, 중지, 그리고 사용할 수 있습니다. 고급 사용자(**Power User**)는 관리 작업을 일부 수행할 수 있습니다. 두 종류의 사용자 모두 같은 **URL**을 통해 사용자 포털에 액세스하며 로그인 시 자신의 권한에 맞는 옵션을 선택할 수 있습니다.

- **일반 사용자 액세스**

일반 사용자는 사용자 포털을 통해 본인의 가상 데스크탑을 시작/정지/연결할 수 있습니다. 가상 머신에 직접 연결하기 위해 *Simple Protocol for Independent Computing Environments (SPICE)* 또는 *Virtual Network Computing (VNC)* 클라이언트를 사용합니다. 두 프로토콜 모두 사용자에게 로컬로 설치된 데스크탑 환경과 유사한 환경을 제공합니다. 관리자는 가상 머신 생성 시 해당 가상 머신에 연결하는 데 사용될 프로토콜을 지정합니다.

사용자 포털에서 사용 가능한 작업과 지원되는 브라우저 및 클라이언트에 대한 보다 자세한 내용은 [Introduction to the User Portal\(사용자 포털 소개\)](#)에서 참조하십시오.

- **고급 사용자 액세스**

**Red Hat Virtualization** 사용자 포털은 고급 사용자에게 가상 리소스를 생성/사용/감시할 수 있게 해주는 그래픽 사용자 인터페이스(**GUI**)를 제공합니다. 시스템 관리자는 사용자에게 고급 사용자 액세스를 부여해서 일부 관리 작업을 위임할 수 있습니다. 일반 사용자가 수행할 수 있는 작업에 더해서 고급 사용자가 수행할 수 있는 작업은 다음과 같습니다:

- 가상 머신을 생성, 편집, 삭제합니다.
- 가상 디스크 및 네트워크 인터페이스를 관리합니다.
- 가상 머신에 사용자 권한을 부여합니다.
- 템플릿을 생성 및 사용하여 가상 머신을 빠르게 배포할 수 있습니다.
- 리소스 사용량 및 심각성이 높은 이벤트를 감시합니다.
- 스냅샷을 생성 및 사용하여 가상 머신을 이전 상태로 복원합니다.

고급 사용자는 위임된 가상 머신의 관리 작업을 수행할 수 있습니다. 데이터 센터 및 클러스터 수준의 관리 작업은 해당 환경의 관리자가 갖게 됩니다.

## 관리 포털

관리 포털은 **Red Hat Virtualization Manager** 서버의 그래픽 관리 인터페이스입니다. 관리 포털을 사용하여 관리자는 웹 브라우저를 통해 가상화 환경의 모든 요소를 감시, 생성, 그리고 관리할 수 있습니다. 관리 포털에서 수행할 수 있는 작업은 다음과 같습니다:

- 네트워크, 스토리지 도메인 등과 같은 가상 인프라를 생성/관리합니다.
- 호스트를 설치 및 관리할 수 있습니다.
- 데이터 센터, 클러스터 등과 같은 논리적 엔티티를 생성/관리합니다.
- 가상 머신을 생성 및 관리합니다.
- Red Hat Virtualization 사용자 및 권한을 관리합니다.

관리 포털은 JavaScript를 사용하여 표시됩니다.

관리 포털 기능에 대한 보다 자세한 내용은 [Red Hat Virtualization 관리 가이드](#) 에서 참조하십시오. 관리 포털에서 지원되는 브라우저 및 플랫폼에 대한 내용은 [Red Hat Virtualization 설치 가이드](#) 에서 참조하십시오.

## Representational State Transfer (REST) API

Red Hat Virtualization REST API에서는 Red Hat Virtualization 환경을 조사 및 관제할 수 있는 소프트웨어 인터페이스를 제공합니다. REST API는 HTTP 작업을 지원하는 모든 프로그래밍 언어에서 사용할 수 있습니다.

REST API를 사용하여 개발자와 관리자는 다음과 같은 작업을 할 수 있습니다:

- 기업 IT 시스템과 통합합니다.
- 타사 가상화 소프트웨어와 통합합니다.
- 자동화된 유지관리 및 오류 확인 작업을 수행합니다.
- 스크립트를 사용하여 Red Hat Virtualization 환경에서 반복적인 작업을 자동화합니다.

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

## 1.4. MANAGER를 지원하는 구성 요소

### Red Hat JBoss Enterprise Application Platform

Red Hat JBoss Enterprise Application Platform은 Java 애플리케이션 서버입니다. 이를 사용하여 플랫폼 간 Java 애플리케이션의 효율적인 개발 및 전송을 지원하는 프레임워크를 제공합니다. Red Hat Virtualization Manager는 Red Hat JBoss Enterprise Application Platform을 사용하여 제공됩니다.



## 중요

Red Hat Virtualization Manager와 함께 번들로 제공된 Red Hat JBoss Enterprise Application Platform 버전은 다른 애플리케이션에 **사용하지 않아야** 합니다. 이 플랫폼은 Red Hat Virtualization Manager에 사용하도록 특별히 맞춰져 있습니다. Manager에 포함된 Red Hat JBoss Enterprise Application Platform을 다른 목적으로 사용하는 것은 Red Hat Virtualization 환경에서의 플랫폼 기능에 부정적인 영향을 미치게 됩니다.

### 보고서 및 기록 데이터 수집

Red Hat Virtualization Manager에 포함된 data warehouse에서는 호스트, 가상 머신, 그리고 스토리지 감시 데이터를 수집합니다. 사전 정의된 여러 가지 보고서가 제공됩니다. 사용자는 SQL을 지원하는 모든 쿼리 도구를 사용하여 환경을 분석하고 보고서를 생성할 수 있습니다.

Red Hat Virtualization Manager 설치 중 두 개의 데이터베이스가 생성됩니다. 이 데이터베이스는 설치 중 선택된 Postgres 인스턴스에 생성됩니다.

- **engine** 데이터베이스는 Red Hat Virtualization Manager에서 사용하는 주 데이터 저장소입니다. 가상화 환경의 상태, 설정, 성능 등에 관한 정보가 이 데이터베이스에 저장됩니다.
- **ovirt\_engine\_history** 데이터베이스는 **engine** 운영 데이터베이스에서 시간을 두고 수집된 설정 정보 및 통계 지표를 담고 있습니다. **engine** 데이터베이스에 있는 설정 데이터는 매분 검사되어 변경 내용이 **ovirt\_engine\_history** 데이터베이스에 복제됩니다. 데이터베이스 변경 내용을 추적하여 데이터베이스에 있는 대상에 대한 정보를 얻을 수 있습니다. 이를 통해 사용자의 Red Hat Virtualization 환경을 분석하고 개선하여 문제를 해결할 수 있게 해줍니다.

**ovirt\_engine\_history** 데이터베이스를 기반으로 보고서를 생성하는 방법은 *Red Hat Virtualization Data Warehouse 가이드*에 있는 [History Database \(기록 데이터베이스\)](#)에서 참조하시기 바랍니다.



## 중요

**ovirt\_engine\_history** 데이터베이스에서 데이터 복제는 **RHEVM History Service**인 **ovirt-engine-dwhd**가 수행합니다.

### 디렉토리 서비스

디렉토리 서비스는 사용자 및 조직 정보를 저장할 수 있는 중앙화된 네트워크 기반 저장소를 제공합니다. 애플리케이션 설정, 사용자 프로파일, 그룹 데이터, 정책, 액세스 제어 등의 정보가 저장됩니다. Red Hat Virtualization Manager는 Active Directory, Identity Management (IdM), OpenLDAP, 그리고 Red Hat Directory Server 9를 지원합니다. 또한 오직 관리만을 목적으로 하는 로컬 내부 도메인도 있습니다. 이러한 내부 도메인의 유일한 사용자는 관리자입니다.

## 1.5. 스토리지

Red Hat Virtualization은 가상 디스크 이미지, 템플릿, 스냅샷, 그리고 ISO 파일에 대해 중앙화된 스토리지 시스템을 사용합니다. 스토리지는 스토리지 도메인으로 구성된 스토리지 풀에 논리적으로 그룹화합니다. 스토리지 도메인이란 스토리지의 내부 구조를 설명하는 스토리지 용량과 메타데이터의 조합을 말합니다. 스토리지 도메인에는 데이터, 내보내기, 그리고 ISO의 세 가지가 있습니다.

각 데이터 센터에서 유일하게 필요한 스토리지 도메인은 데이터 스토리지 도메인입니다. 데이터 스토리지 도메인은 단 하나의 데이터 센터에서 전용으로 사용합니다. 내보내기 및 ISO 도메인은 선택 사항입니다. 스토리지 도메인은 공유 리소스이므로 데이터 센터 내의 모든 호스트에서 액세스할 수 있어야 합니다.

스토리지 네트워크를 구현하기 위해 **Network File System(NFS)**, **Internet Small Computer System Interface(iSCSI)**, **GlusterFS**, **Fibre Channel Protocol(FCP)**, 또는 모든 네트워킹된 **POSIX** 호환 파일 시스템을 사용할 수 있습니다.

**NFS** (및 다른 **POSIX** 호환 파일 시스템) 도메인에서 모든 가상 디스크, 템플릿 및 스냅샷은 단순 파일입니다.

**SAN (iSCSI/FCP)** 도메인에서 블록 장치는 논리 볼륨 관리자( **LVM**)에 의해 볼륨 그룹(**VG**)으로 집계됩니다. 각 가상 디스크, 템플릿 및 스냅샷은 **VG**의 논리 볼륨(**LV**)입니다. **LVM**에 대한 보다 자세한 내용은 [Red Hat Enterprise Linux Logical Volume Manager Administration Guide](#)에서 참조하십시오.

### 데이터 스토리지 도메인

시스템 환경에서 실행 중인 모든 가상 머신의 가상 하드 디스크 이미지는 데이터 도메인에 있습니다. 가상 머신의 템플릿 및 스냅샷도 데이터 도메인에 저장됩니다. 데이터 도메인은 데이터 센터 간에 공유할 수 없습니다.

### 내보내기 스토리지 도메인

내보내기 도메인이란 데이터 센터와 **Red Hat Virtualization** 환경 간에 이미지를 복사 및 이동하는 데 사용되는 임시 스토리지 리포지토리를 말합니다. 내보내기 도메인은 가상 머신과 템플릿을 백업하는 데 사용할 수 있습니다. 내보내기 도메인은 데이터 센터 간에 이동할 수 있지만 한 번에 하나의 데이터 센터에서만 활성화할 수 있습니다.

### ISO 스토리지 도메인

**ISO** 도메인에 저장되는 **ISO** 파일은 가상 머신에 운영 체제 및 애플리케이션을 설치하는 데 사용되는 논리 **CD-ROM**입니다. **ISO** 도메인이라는 논리 엔티티는 물리적 **CD-ROM** 또는 **DVD** 라이브러리를 대체하여 데이터 센터에서 물리적 미디어가 더 이상 필요하지 않게 해줍니다. **ISO** 도메인은 여러 데이터 센터 간에 공유될 수 있습니다.

## 1.6. 네트워크

**Red Hat Virtualization** 네트워크 아키텍처는 **Red Hat Virtualization** 환경의 다양한 요소를 서로 연결해 줍니다. 네트워크 아키텍처는 네트워크 연결을 지원할 뿐만 아니라 네트워크 분리도 가능하게 해줍니다.

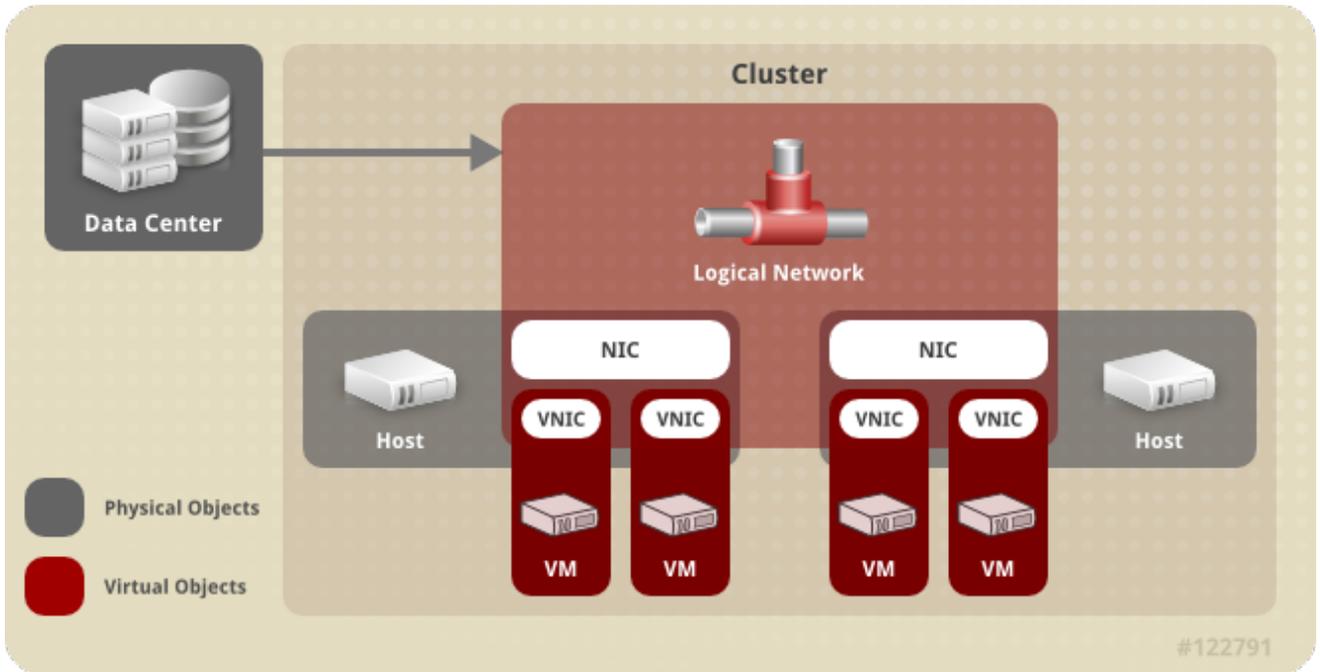


그림 1.3. 네트워크 아키텍처

Red Hat Virtualization에서 네트워크는 여러 층으로 구성됩니다. 기본 물리 네트워크 인프라가 설치 및 설정되어 있어야 Red Hat Virtualization 환경의 하드웨어 요소와 논리 요소 간의 연결이 가능해집니다.

### 네트워크 인프라 계층

Red Hat Virtualization 네트워크 아키텍처는 다음과 같은 일반적인 하드웨어 및 소프트웨어 장치에 의존합니다:

- **Network Interface Controllers(NICs)**는 호스트를 네트워크에 연결하는 물리 네트워크 인터페이스 장치입니다.
- **Virtual NIC(VNIC)**는 호스트의 물리 NIC를 사용하여 작동하는 논리 NIC입니다. VNIC는 가상 머신에 네트워크 연결을 제공합니다.
- **본딩**은 여러 NIC를 하나의 인터페이스로 통합합니다.
- **브리지**란 패킷 교환 네트워크에서 사용하는 패킷 전달 방식을 말합니다. 이는 **가상 머신 논리 네트워크**의 기반이 됩니다.

### 논리 네트워크

논리 네트워크는 환경의 요구 사항에 따라 네트워크 트래픽을 분리할 수 있게 해줍니다. 다음과 같은 종류의 논리 네트워크가 있습니다:

- 가상 머신 네트워크 트래픽을 전송하는 논리 네트워크
- 가상 머신 네트워크 트래픽을 전송하지 않는 논리 네트워크
- 선택적 논리 네트워크
- 필수 네트워크

모든 논리 네트워크는 필수 또는 옵션(선택적)입니다.

가상 머신 네트워크 트래픽을 전송하는 논리 네트워크는 호스트 수준에서 소프트웨어 브리지 장치로 구현됩니다. 기본적으로 **Red Hat Virtualization Manager** 설치 중 **ovirtmgmt** 관리 네트워크라는 하나의 논리 네트워크가 지정됩니다.

그 외에도 관리자는 전용 스토리지 논리 네트워크 및 전용 디스플레이 논리 네트워크 등의 논리 네트워크를 추가할 수 있습니다. 가상 머신 트래픽을 전송하지 않는 논리 네트워크는 호스트에서 관련 브리지 장치가 없습니다. 이러한 논리 네트워크는 호스트 네트워크 인터페이스와 직접적으로 연결되어 있습니다.

**Red Hat Virtualization**은 마이그레이션 관련 네트워크 트래픽으로부터 관리 관련 네트워크 트래픽을 분리합니다. 이를 통해 실시간 마이그레이션 전용 네트워크(라우팅 없음)를 사용할 수 있게 되며 마이그레이션 중에 관리 네트워크(**ovirtmgmt**)가 하이퍼바이저와의 연결을 잃지 않게 해줍니다.

## 계층별 논리 네트워크에 대한 설명

논리 네트워크는 가상화 환경의 각 계층에서 다른 영향을 미칩니다.

### 데이터 센터 계층

논리 네트워크가 데이터 센터 수준에서 정의됩니다. 각 데이터 센터에 기본적으로 **ovirtmgmt** 관리 네트워크가 있습니다. 추가적인 논리 네트워크는 선택 사항이지만 있는 것이 좋습니다. 데이터 센터 수준에서 **가상 머신 네트워크**로 지정하고 사용자 정의 MTU를 설정할 수 있습니다. 데이터 센터에 정의된 논리 네트워크는 그 논리 네트워크를 사용하는 클러스터에도 추가되어야 합니다.

### 클러스터 계층

논리 네트워크는 데이터 센터에서 사용 가능하며 이 네트워크를 사용하게 될 클러스터에 네트워크를 반드시 추가해야 합니다. 기본적으로 각 클러스터는 관리 네트워크에 연결되어 있습니다. 옵션으로 클러스터의 상위 데이터 센터에 정의된 논리 네트워크를 그 클러스터에 추가할 수 있습니다. 필수 논리 네트워크를 클러스터에 추가한 경우 이를 클러스터의 각 호스트에서 구현해야 합니다. 필요시 선택적 논리 네트워크를 호스트에 추가할 수 있습니다.

### 호스트 계층

가상 머신 논리 네트워크는 클러스터 내의 각 호스트에서 주어진 네트워크 인터페이스와 연결된 소프트웨어 브리지로 구현됩니다. 가상 머신 논리 네트워크가 아닌 네트워크에는 연결된 브리지가 없으며 호스트 네트워크 인터페이스와 직접 연결되어 있습니다. 각 호스트를 **Red Hat Virtualization** 환경에 추가하는 경우 관리 네트워크는 호스트의 네트워크 장치를 사용하는 브리지로 구현됩니다. 추가적으로 필요한 논리 네트워크를 클러스터에 추가 시 이러한 논리 네트워크를 각 호스트에서 네트워크 인터페이스에 연결되어야 해당 클러스터에서 작동하게 됩니다.

### 가상 머신 계층

네트워크가 물리적 시스템에 제공되는 것과 같은 방법으로 논리 네트워크가 가상 머신에 제공됩니다. 가상 머신은 이를 실행하는 호스트에서 구현된 아무 가상 머신 논리 네트워크에 가상 NIC를 연결할 수 있습니다. 그 후 가상 머신은 연결된 논리 네트워크에서 사용 가능한 다른 모든 장치 또는 대상에 연결할 수 있게 됩니다.

#### 예 1.1. 관리 네트워크

**ovirtmgmt**라고 부르는 관리 논리 네트워크는 **Red Hat Virtualization Manager** 설치 중 자동으로 생성됩니다. **ovirtmgmt** 네트워크는 **Red Hat Virtualization Manager**와 호스트 간의 관리 트래픽을 전달합니다. 특정한 목적의 다른 브리지가 설정되어 있지 않은 경우 **ovirtmgmt**가 모든 트래픽에 대한 기본 브리지가 됩니다.

## 1.7. 데이터 센터

데이터 센터는 **Red Hat Virtualization**에서 가장 높은 추상화 수준입니다. 데이터 센터란 다음과 같이 세 종류의 하위 컨테이너로 구성된 컨테이너를 말합니다:

- *스토리지 컨테이너*에서는 스토리지 도메인에 대한 연결 정보를 포함한 스토리지 유형 및 스토리지 도메인 정보가 저장됩니다. 데이터 센터에 대하여 스토리지가 정의되며 해당 데이터 센터의 모든 클러스터에서 사용 가능합니다. 데이터 센터 내에 있는 모든 호스트 클러스터는 같은 스토리지 도메인에 액세스할 수 있습니다.
- *네트워크 컨테이너*에서는 데이터 센터의 논리 네트워크 정보가 저장됩니다. 여기에는 네트워크 주소, VLAN 태그 및 STP 지원 등의 상세 정보도 포함됩니다. 데이터 센터에 대하여 논리 네트워크가 정의되며 옵션으로 클러스터 수준에서 구현됩니다.
- *클러스터 컨테이너*에서는 클러스터가 저장됩니다. 클러스터란 **AMD** 또는 **Intel** 프로세서와 같이 호환되는 프로세서 코어를 가진 호스트 그룹을 말합니다. 클러스터는 마이그레이션 도메인이며 가상 머신은 클러스터 내의 모든 호스트에 실시간 마이그레이션할 수 있지만 다른 클러스터에는 할 수 없습니다. 하나의 데이터 센터에 여러 클러스터를 연결할 수 있으며 각 클러스터에 여러 호스트를 저장할 수 있습니다.

## 2장. 스토리지

### 2.1. 스토리지 도메인 개요

스토리지 도메인은 일반적인 스토리지 인터페이스가 있는 이미지 모음입니다. 스토리지 도메인에는 템플릿 이미지와 가상 머신(스냅샷 포함)의 전체 이미지, ISO 파일, 그리고 자체의 메타데이터가 들어 있습니다. 스토리지 도메인은 블록 장치(SAN - iSCSI 또는 FCP) 또는 파일 시스템(NAS - NFS, GlusterFS, 또는 기타 다른 POSIX 호환 파일 시스템)으로 구성됩니다.

NAS에서는 가상 디스크, 템플릿, 스냅샷은 모두 파일입니다.

SAN (iSCSI/FCP)에서는 각 가상 디스크, 템플릿, 스냅샷은 논리 볼륨입니다. 블록 장치는 볼륨 그룹이라는 논리 엔티티로 통합된 후 가상 하드디스크로 사용하기 위해 LVM (Logical Volume Manager)에 의해 논리 볼륨으로 나뉘어 집니다. LVM에 대한 자세한 내용은 [Red Hat Enterprise Linux Logical Volume Manager Administration Guide\(논리 볼륨 관리자 관리 가이드\)](#)에서 참조하십시오.

가상 디스크는 QCOW2 또는 RAW의 두가지 형식 중 하나를 사용할 수 있습니다. 스토리지 유형은 스파스 또는 사전할당될 수 있습니다. 스냅샷은 항상 스파스 지원을 사용하지만 RAW 또는 스파스에서 생성한 디스크를 사용할 수 있습니다.

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

### 2.2. 스토리지가 지원되는 스토리지 도메인 유형

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

#### 파일 기반 스토리지

Red Hat Virtualization에서 지원되는 파일 기반 스토리지 유형에는 NFS, GlusterFS, 기타 POSIX 호환 파일 시스템, 그리고 호스트 로컬 스토리지 등이 있습니다.

파일 기반 스토리지는 Red Hat Virtualization 환경 밖에서 관리됩니다.

NFS 스토리지는 Red Hat Enterprise Linux NFS 서버 또는 타사 네트워크에 연결된 스토리지 서버에 의해 관리됩니다.

호스트는 자체적으로 로컬 스토리지 파일 시스템을 관리할 수 있습니다.

#### 블록 기반 스토리지

블록 스토리지에서는 포맷하지 않은 블록 장치를 사용합니다. 블록 장치는 논리 볼륨 관리자(LVM)에 의해 **볼륨 그룹**으로 통합됩니다. LVM 인스턴스는 다른 호스트에서 실행 중인 인스턴스를 인식하지 못하고 모든 호스트에서 실행됩니다. VDSM은 볼륨 그룹의 변경 내용을 스캔하여 LVM에 클러스터링 로직을 추가합니다. 변경 내용 감지 시 VDSM은 개별 호스트에 볼륨 그룹 정보를 새로고침하라고 지시해서 이러한 호스트를 업데이트합니다. 호스트가 볼륨 그룹을 논리 볼륨으로 나누며 논리 볼륨 메타데이터를 디스크에 씁니다. 기존 스토리지 도메인에 스토리지 용량이 추가되는 경우 Red Hat Virtualization Manager로 인해 각 호스트의 VDSM이 볼륨 그룹 정보를 새로고침하게 됩니다.

논리 단위 번호(LUN)란 개별 블록 장치를 말합니다. 지원되는 블록 스토리지 프로토콜인 iSCSI, FCoE, 또는 SAS 중 하나를 사용하여 LUN에 연결합니다. Red Hat Virtualization Manager에서는 LUN에 대한 소프트웨어 iSCSI 연결을 관리합니다. 다른 모든 블록 장치 연결은 Red Hat Virtualization 환경 밖에서 관리됩니다. 논리 볼륨 생성, 논리 볼륨 확장 또는 삭제, 그리고 새 LUN 추가 등과 같은 블록 기반 스토리지 환경의 모든 변경 사항은 Storage Pool Manager라는 특별히 선택된 호스트에 있는 LVM에서 처리합니다. 그 후 VDSM이 이러한 변경 사항을 동기화하며 스토리지 메타데이터는 이를 클러스터 내의 모든 호스트에서 새로고침합니다.

## 2.3. 스토리지 도메인 유형

Red Hat Virtualization은 각 스토리지 도메인이 지원하는 스토리지 유형을 포함하여 세 가지 유형의 스토리지 도메인을 지원합니다:

- **데이터 스토리지 도메인**에서는 Red Hat Virtualization 환경에 있는 모든 가상 머신의 하드 디스크 이미지가 저장됩니다. 디스크 이미지는 설치된 운영 체제 또는 가상 머신이 저장하거나 생성한 데이터를 포함할 수 있습니다. 데이터 스토리지 도메인은 NFS, iSCSI, FCP, GlusterFS 그리고 POSIX 호환 스토리지를 지원합니다. 데이터 도메인은 여러 데이터 센터 간에 공유될 수 없습니다.
- **내보내기 스토리지 도메인** 데이터 센터 간에 전송되는 하드 디스크 이미지와 가상 머신 템플릿의 임시 저장소입니다. 또한 내보내기 스토리지 도메인은 가상 머신 백업 복사본을 저장합니다. 내보내기 스토리지 도메인은 NFS 스토리지를 지원합니다. 여러 데이터 센터에서 하나의 내보내기 스토리지 도메인에 액세스할 수 있지만 한 번에 하나의 데이터 센터만 스토리지 도메인을 사용할 수 있습니다.
- **ISO 스토리지 도메인**에서는 이미지라고 부르는 ISO 파일이 저장됩니다. ISO 파일은 물리적인 CD 또는 DVD의 표현입니다. Red Hat Virtualization 환경에서 일반적인 ISO 파일에는 운영 체제 설치 디스크, 애플리케이션 설치 디스크, 그리고 게스트 에이전트 설치 디스크 등이 있습니다. 이러한 이미지는 물리적인 디스크를 디스크 드라이브에 삽입하여 부팅하는 것과 같은 방법으로 가상 머신에 연결하여 부팅할 수 있습니다. ISO 스토리지 도메인은 데이터 센터 내의 모든 호스트가 ISO를 공유할 수 있도록 해주로 물리적인 광학 미디어가 필요하지 않게 됩니다.

## 2.4. 가상 디스크 이미지의 저장소 형식

### QCOW2 형식의 가상 머신 스토리지

QCOW2는 가상 디스크 이미지의 저장 형식입니다. QCOW는 *QEMU copy on write*의 약자입니다. QCOW2 형식은 논리 블록과 물리적인 블록 사이에 매핑을 추가하여 가상 계층으로부터 물리적인 스토리지 레이어를 분리합니다. 각 논리 블록은 물리적 오프셋에 매핑되어서 스토리지 초과 할당 및 가상 머신 스냅샷을 사용 가능하게 하며, 여기서 각 QCOW 볼륨은 기반 디스크 이미지의 변경 내용만 보여줍니다.

초기 매핑을 통해 모든 논리 블록이 지원 파일 또는 지원 볼륨에 있는 오프셋과 연결됩니다. 가상 머신이 스냅샷 후 QCOW2 볼륨에 데이터를 쓰면 지원 볼륨에서 관련 블록을 읽고 새 정보로 수정한 후 새 스냅샷 QCOW2 볼륨에 쓰기합니다. 이를 통해 맵은 새 장소로 연결되도록 업데이트됩니다.

### RAW

RAW 스토리지 형식으로 저장된 가상 디스크 이미지에는 형식이 적용되지 않는다는 점에서 QCOW2보다 성능면에서 우월합니다. RAW 형식으로 저장된 디스크 이미지를 사용하여 가상 머신 데이터 작업 시 호스트에서 추가적인 작업을 할 필요가 없습니다. 가상 머신이 가상 디스크에서 주어진 오프셋에 데이터를 쓰면 이러한 I/O는 지원 파일 또는 논리 볼륨의 같은 오프셋에 쓰기됩니다.

Raw 형식의 경우 저장 장치 어레이(storage array)에서 외부에서 관리되는 썬 프로비저닝된 LUN을 사용하지 않는 한 정의된 이미지의 전체 공간이 사전 할당되어야 합니다.

## 2.5. 가상 디스크 이미지 스토리지 할당 정책

### 사전 할당 스토리지

가상 디스크 이미지에서 필요로 하는 모든 스토리지는 가상 머신 생성 전에 할당됩니다. 가상 머신에 대하여 20 GB의 디스크 이미지 생성 시 이 디스크 이미지는 20 GB의 스토리지 도메인 용량을 사용합니다. 사전 할당된 디스크 이미지는 확장될 수 없습니다. 사전 할당된 스토리지 사용 시 유연성이 줄어드는 대신 작동 중에 스토리지 할당 작업을 하지 않아도 되어서 데이터를 쓰는 시간이 절약됩니다. 이

러한 방법으로 스토리지 할당 시 **Red Hat Virtualization Manager**가 스토리지를 오버커밋할 수 있는 능력이 감소하게 됩니다. 스토리지 대기 시간에 대한 허용치가 낮은 집중적인 I/O 작업에 가상 머신을 사용하는 경우 사전 할당 스토리지가 권장됩니다. 일반적으로 서버 가상 머신이 여기에 해당됩니다.

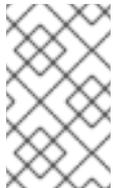


### 참고

스토리지 백엔드에서 제공된 썸 프로비저닝 기능을 사용하는 경우에도 관리 포털에서 가상 머신 스토리지 프로비저닝을 할 때 여전히 사전 할당 스토리지를 선택해야 합니다.

## 스파스 할당 스토리지

가상 디스크 이미지의 최대 용량 제한은 가상 머신 생성 시 설정됩니다. 초기에 디스크 이미지는 스토리지 도메인 용량을 하나도 사용하지 않습니다. 가상 머신이 디스크에 데이터를 쓰에 따라 사용량이 증가하여 최대 용량에 도달합니다. 디스크 이미지에 있는 데이터를 삭제해도 스토리지 도메인 용량이 복원되지 않습니다. 스토리지 대기 시간에 대해 약간의 허용치가 있는 중저급 강도의 I/O 작업에 가상 머신을 사용하는 경우 스파스 할당 스토리지가 권장됩니다. 일반적으로 데스크탑 가상 머신이 여기에 해당됩니다.



### 참고

스토리지 백엔드에서 제공된 썸 프로비저닝 기능을 사용하는 경우 이는 썸 프로비저닝에서 선호하는 구현으로 사용해야 합니다. 스토리지는 GUI에서 사전 할당 프로비저닝하 여 썸 프로비저닝을 백엔드 해결책으로 남겨놓습니다.

## 2.6. RED HAT VIRTUALIZATION에서의 스토리지 메타데이터 버전

Red Hat Virtualization은 스토리지 도메인 정보를 스토리지 도메인 자체에서 메타데이터로 보관합니다. Red Hat Virtualization의 주요 릴리스마다 스토리지 메타데이터의 구현이 개선되었습니다.

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

각 스토리지 도메인에는 자체의 구조를 설명하는 메타데이터가 들어 있으며 가상 디스크 이미지의 백업에 사용되는 모든 실제 볼륨의 이름도 들어 있습니다.

마스터 도메인은 추가로 모든 도메인 및 스토리지 풀의 실제 볼륨 이름에 대한 메타데이터를 담고 있습니다. 이러한 메타데이터의 총 크기는 **2 KB**로 제한되므로 풀 내에 있을 수 있는 스토리지 도메인의 수가 제한됩니다.

템플릿 및 가상 머신 기본 이미지는 읽기 전용입니다.

V1 메타데이터는 NFS, iSCSI, 및 FC 스토리지 도메인에 적용됩니다.

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

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

실제 볼륨 이름은 더 이상 메타데이터에 포함되지 않습니다.

템플릿 및 가상 머신 기본 이미지는 읽기 전용입니다.

V2 메타데이터는 iSCSI 및 FC 스토리지 도메인에 적용됩니다.

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

모든 스토리지 도메인 및 풀 메타데이터는 논리 볼륨에 쓰지 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 변화를 통해 실시간 스냅샷, 실시간 스토리지 마이그레이션, 그리고 스냅샷 복제가 가능해집니다.

영어로 되어 있지 않은 볼륨 이름을 위해 유니코드 메타데이터 지원이 추가됩니다.

V3 메타데이터는 NFS, GlusterFS, POSIX, iSCSI, 및 FC 스토리지 도메인에 적용됩니다.

## 2.7. RED HAT VIRTUALIZATION에서 스토리지 도메인 자동 복구

Red Hat Virtualization 환경에 있는 호스트는 각 도메인의 메타데이터를 읽어서 데이터 센터 내에 있는 스토리지 도메인을 감시합니다. 데이터 센터의 모든 호스트에서 스토리지 도메인에 액세스할 수 없다고 보고하면 스토리지 도메인이 비활성화됩니다.

Manager는 비활성화된 스토리지 도메인의 연결을 끊기보다는 예를 들어 일시적인 네트워크 서비스 중지 등으로 인해 스토리지 도메인이 일시적으로 비활성화되었다고 가정합니다. 5분마다 Manager는 비활성화된 스토리지 도메인을 다시 활성화하려고 시도합니다.

스토리지 연결 장애의 원인을 해결하기 위해 관리자 개입이 필요할 수도 있지만 연결 복구 시 스토리지 도메인을 다시 활성화하는 것은 Manager가 담당합니다.

## 2.8. SPM (STORAGE POOL MANAGER)

Red Hat Virtualization에서는 메타데이터를 사용하여 스토리지 도메인의 내부 구조를 설명합니다. 구조적인 메타데이터는 각 스토리지 도메인의 세그먼트에 씁니다. 호스트는 단일 writer에 여러 reader 구조를 기반으로 스토리지 도메인 메타데이터를 다룹니다. 스토리지 도메인의 구조적인 메타데이터는 이미지 및 스냅샷 생성 및 삭제, 그리고 볼륨 및 도메인 확장을 추적합니다.

데이터 도메인의 구조를 변경할 수 있는 호스트를 Storage Pool Manager(SPM)라고 합니다. SPM은 디스크 이미지 생성 및 삭제, 스냅샷 생성 및 통합, 스토리지 도메인 간 이미지 복사, 템플릿 생성 및 블록 장치를 위한 스토리지 할당 등과 같은 데이터 센터의 모든 메타데이터 변경을 다룹니다. 각 데이터 센터마다 하나의 SPM이 있습니다. 다른 모든 호스트는 스토리지 도메인의 구조적인 메타데이터를 읽기만 할 수 있습니다.

호스트를 수동으로 SPM으로 선택할 수도 있고 Red Hat Virtualization Manager가 할당할 수도 있습니다. Manager는 잠재적인 SPM 호스트가 스토리지 중심의 임대를 시도하도록 해서 SPM 역할을 할당합니다. 이러한 임대는 SPM 호스트가 스토리지 메타데이터를 쓰기할 수 있도록 해줍니다. 이것이 스토리지 중심인 이유는 이것이 Manager 또는 호스트에 의해 추적되기보다는 스토리지 도메인에 쓰기가 되기 때문입니다. 스토리지 중심의 임대는 마스터 스토리지 도메인의 특별 논리 볼륨인 임대에 쓰기됩니다. 데이터 도메인의 구조에 대한 메타데이터는 메타데이터라고 부르는 특별 논리 볼륨에 쓰여집니다. 메타데이터 논리 볼륨 변경 사항은 임대 논리 볼륨에 의해 보호됩니다.

Manager는 VDSM을 사용하여 호스트에게 spmStart 명령을 내려서 그 호스트의 VDSM이 스토리지 중심의 임대를 시작하도록 만듭니다. 호스트가 성공하면 SPM이 되며 Red Hat Virtualization Manager가 다른 호스트에게 SPM 역할을 하도록 요청할 때까지 스토리지 중심의 임대를 유지합니다.

Manager는 다음과 같은 경우에 SPM 역할을 다른 호스트로 이동합니다:

- SPM 호스트가 모든 스토리지 도메인에 액세스할 수 없지만 마스터 스토리지 도메인에 액세스할 수 있는 경우
- 스토리지 연결이 끊겼거나 임대 볼륨이 차서 쓰기 작업을 수행할 수 없게 되어서 SPM 호스트가 임대를 갱신하지 못하는 경우

- SPM 호스트가 작동 정지되는 경우

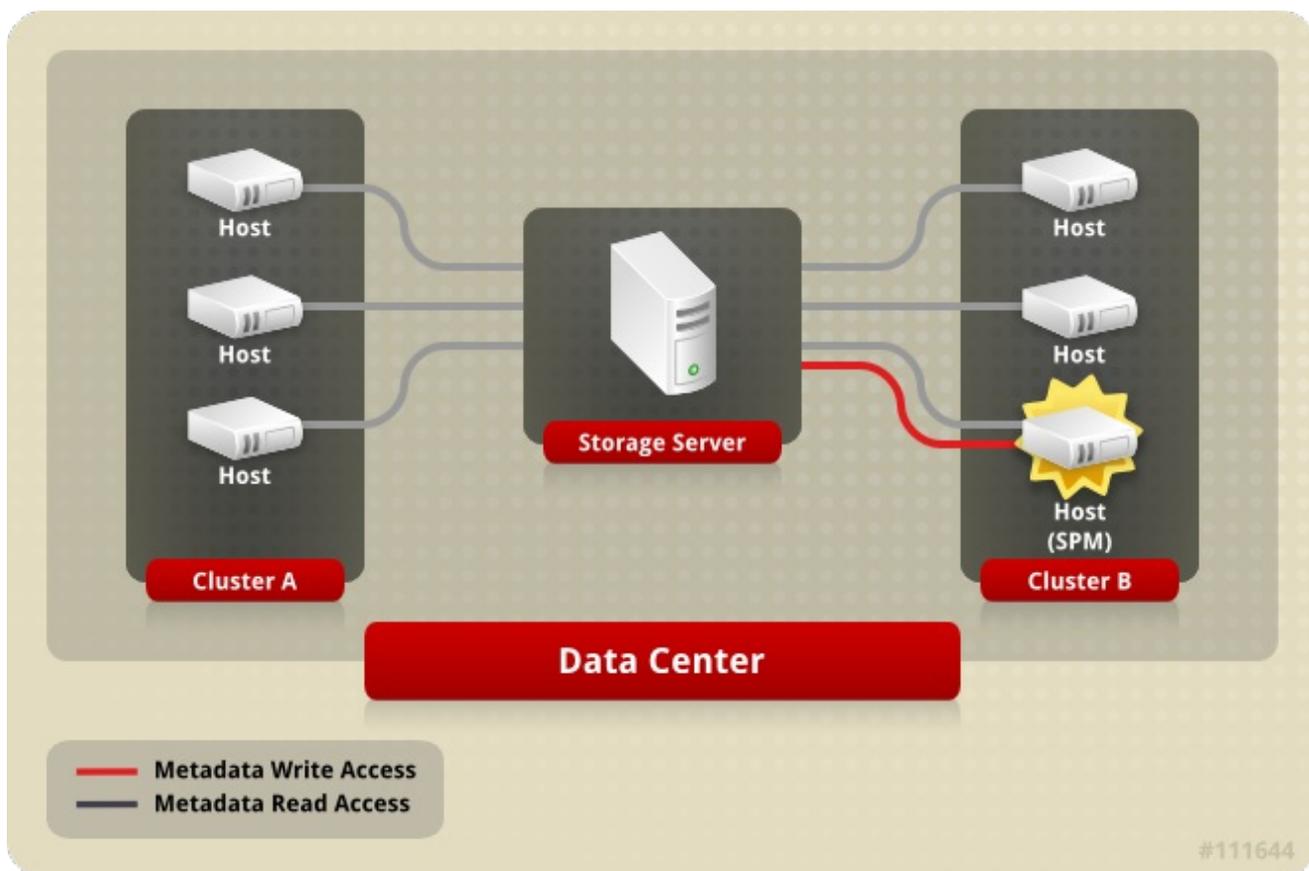


그림 2.1. Storage Pool Manager를 통해 구조적인 메타데이터만 단독적으로 쓰기

## 2.9. STORAGE POOL MANAGER 선택 프로세스

Storage Pool Manager (SPM) 역할에 의해 호스트가 수동 할당되지 않은 경우 Red Hat Virtualization Manager가 SPM 선택 프로세스를 시작 및 관리합니다.

우선 Red Hat Virtualization Manager는 VDSM에 어느 호스트가 스토리지 중심 임대자 있는지 확인하도록 요청합니다.

Red Hat Virtualization Manager는 스토리지 도메인 초기 생성 이후의 SPM 배정 기록을 추적합니다. SPM 역할의 사용 가능 여부를 다음의 세 가지 방법으로 확인합니다:

- "getSPMstatus" 명령: Manager는 VDSM을 사용하여 마지막으로 SPM 상태에 있었던 호스트에 확인 후 "SPM", "Contending", 또는 "Free" 중 하나의 응답을 받습니다.
- 스토리지 도메인의 메타데이터 볼륨에는 마지막으로 SPM 상태에 있었던 호스트가 포함되어 있습니다.
- 스토리지 도메인의 메타데이터 볼륨에는 마지막으로 SPM 상태에 있었던 호스트의 버전이 포함되어 있습니다.

작동 중이며 응답을 하는 호스트가 스토리지 중심의 임대를 유지하는 경우 Red Hat Virtualization Manager는 관리 포털에서 그 호스트 SPM을 표시합니다. 추가적인 작업은 수행되지 않습니다.

SPM 호스트가 응답하지 않는 경우 연결할 수 없는 것으로 간주됩니다. 전원 관리가 호스트에 대해 설정되어 있는 경우 자동으로 펜싱됩니다. 그렇지 않은 경우 수동으로 펜싱해야 합니다. Storage Pool Manager 역할은 이전의 Storage Pool Manager를 펜싱할 때까지는 새 호스트에 배정할 수 없습니다.

SPM 역할 및 스토리지 중심의 임대가 배정되지 않은 경우 Red Hat Virtualization Manager가 데이터 센터 내에서 임의로 선택된 작동 중인 호스트에 이를 배정합니다.

SPM 역할을 새 호스트에 배정하는 것이 실패하는 경우 Red Hat Virtualization Manager는 그 호스트를 이 작업이 실패한 호스트 목록에 추가하여 이러한 호스트가 SPM 역할에 적합하지 않은 것으로 표시합니다. 이 목록은 다음의 SPM 선택 프로세스 시작 시에 해제되어서 모든 호스트가 다시 선택 자격이 되도록 합니다.

Red Hat Virtualization Manager는 SPM이 성공적으로 선택될 때까지 계속해서 임의로 선택된 호스트 중에서 실패한 호스트 목록에 없는 호스트가 Storage Pool Manager 역할 및 스토리지 중심의 임대를 맡도록 요청을 하게 됩니다.

현재 SPM이 응답하지 않거나 제 역할을 완수하지 못할 때마다 Red Hat Virtualization Manager는 Storage Pool Manager 선택 프로세스를 시작합니다.

## 2.10. RED HAT VIRTUALIZATION의 단독 리소스 및 SANLOCK

Red Hat Virtualization 환경에 있는 특정 리소스는 단독으로 액세스해야 합니다.

SPM 역할이 그러한 리소스 중의 하나입니다. 하나 이상의 호스트가 SPM이 되는 경우 동시에 두 곳에서 같은 데이터가 변경될 수 있으므로 데이터 손상의 위험이 있습니다.

Red Hat Enterprise Virtualization 3.1 이전에는 SPM 독점성은 *safelease*라고 부르는 VDSM 기능을 사용하여 관리 및 추적되었습니다. 임대는 데이터 센터의 모든 스토리지 도메인에서 특별 구역에 쪼갬니다. 같은 환경에 있는 모든 호스트는 네트워크와 독립된 방법으로 SPM 상태를 추적할 수 있습니다. VDSM의 세이프 리스(*safe lease*)를 통해 SPM 역할이라는 한 리소스의 독점성만 유지됩니다.

Sanlock은 같은 기능을 제공하지만 SPM 역할을 잠글 수 있는 리소스의 하나로 다룹니다. Sanlock은 추가 리소스를 잠글 수 있게 해주므로 더 신축적입니다.

리소스 잠금을 필요로 하는 애플리케이션은 이를 Sanlock에 등록합니다. 등록된 애플리케이션은 Sanlock이 대신해서 리소스를 잠그도록 요청하여 다른 애플리케이션에서 해당 리소스에 액세스할 수 없도록 할 수 있습니다. 예를 들어 VDSM이 SPM 상태를 잠그는 대신 이제는 VDSM이 Sanlock에게 이러한 작업을 하도록 요청합니다.

잠금은 *lockspace*의 디스크에서 추적합니다. 스토리지 도메인마다 하나의 *lockspace*가 있습니다. SPM 리소스의 잠금의 경우 각 호스트의 활성화 여부를 *lockspace*에서 추적하기 위해 해당 호스트가 스토리지 연결 시 Manager에게서 받은 *hostid*를 갱신할 수 있는지 확인하고 정기적인 간격으로 *lockspace*에 *timestamp*를 쓸 수 있는지 확인합니다. *ids* 논리 볼륨은 각 호스트의 고유 식별자를 추적하며 호스트가 *hostid*를 갱신할 때마다 업데이트됩니다. SPM 리소스는 활성화된 호스트에서만 가질 수 있습니다.

리소스는 *임대* 논리 볼륨에 있는 디스크에서 추적합니다. 디스크 상의 리소스 표현이 이를 사용하는 프로세스의 고유 식별자를 사용하여 업데이트된 경우 이러한 리소스가 *취득*되었다고 합니다. SPM 역할의 경우 SPM 리소스는 해당 리소스를 취득한 *hostid*로 업데이트됩니다.

각 호스트의 Sanlock 프로세스에서는 리소스 취득 시 이러한 취득된 리소스만 확인하면 됩니다. 초기 확인 후 Sanlock은 잠금된 리소스가 있는 호스트의 *timestamp*가 오래될 때까지 *lockspace*를 감시할 수 있습니다.

Sanlock은 리소스를 사용하는 애플리케이션을 감시합니다. 예를 들어, VDSM을 사용하여 SPM 상태 및 *hostid*를 감시합니다. 호스트가 Manager에서 *hostid*를 갱신하지 못하는 경우 *lockspace*에 있는 모든 리소스에 대한 독점성을 잃게 됩니다. Sanlock은 리소스를 업데이트하여 그것이 더 이상 취득된 상태가 아님을 보여줍니다.

SPM 호스트가 일정 시간동안 스토리지 도메인에서 *lockspace*에 *timestamp*를 쓰지 못하는 경우 호스트의 Sanlock 인스턴스에서 VDSM 프로세스에 리소스를 해제하도록 요청하게 됩니다. VDSM 프로세스에서 응답하면 리소스가 해제되고 *lockspace*에 있는 SPM 리소스는 다른 호스트가 차지할 수 있게 됩니다.

SPM 호스트에 있는 VDSM에서 리소스 해제 요청에 응답하지 않는 경우 호스트의 Sanlock이 VDSM 프로세스를 중지시킵니다. 중지 명령이 성공적이지 못한 경우 Sanlock은 이를 에스컬레이션하여 sigkill을 사용해서 VDSM 중지를 시도합니다. sigkill이 성공적이지 못한 경우 Sanlock은 호스트를 재부팅하도록 위치독 데몬에 의존해야 합니다.

호스트에서 VDSM이 hostid를 갱신하고 lockspace에서 timestamp를 쓸 때마다 위치독 데몬이 pet를 수신합니다. VDSM이 이것을 하지 못하게 되면 위치독 데몬은 더 이상 pet를 받지 않습니다. 정해진 시간동안 위치독 데몬이 pet를 받지 못하게 되면 호스트를 재부팅합니다. 이러한 에스컬레이션의 마지막 단계에 도달하게 되면 SPM 리소스가 확실하게 해제되며 다른 호스트에서 차지할 수 있게 됩니다.

## 2.11. 씬 프로비저닝 및 스토리지 오버커밋

Red Hat Virtualization Manager는 가상화 환경 내에서 스토리지 사용을 최적화하기 위한 프로비저닝 정책을 제공합니다. 씬 프로비저닝 정책을 사용하여 스토리지 리소스를 오버커밋해서 사용자의 가상화 환경에서 실제 스토리지 사용량에 따라 스토리지를 프로비저닝합니다.

스토리지 오버커밋이란 스토리지 풀에 실제로 사용 가능한 것보다 더 많은 스토리지를 가상 머신에 할당하는 것을 말합니다. 일반적으로 가상 머신은 할당된 것보다 적은 양의 스토리지를 사용합니다. 씬 프로비저닝을 통해 가상 머신은 실제로 스토리지의 아주 일부만 할당되었음에도 마치 해당 가상 머신에 정의된 스토리지가 완전히 할당된 것처럼 작동하게 됩니다.



### 참고

Red Hat Virtualization Manager가 자체의 씬 프로비저닝 기능을 제공하지만 사용자의 스토리지 백엔드에서 씬 프로비저닝 기능을 제공하면 이를 사용하는 것이 좋습니다.

스토리지 오버커밋을 지원하기 위해 로컬 스토리지 할당과 실제 스토리지 사용량을 비교하는 임계값이 VDSM에 정의됩니다. 이러한 임계값을 사용하여 디스크 이미지에 쓰인 데이터가 이것을 백업해주는 논리 볼륨보다 작은지 확인합니다. QEMU에서는 가장 스토리지 사용량이 높은 지점을 표시하는 논리 볼륨에 쓰인 최대 오프셋을 확인합니다. VDSM은 QEMU에서 표시한 최대 오프셋을 감시해서 사용량이 정의된 임계값을 넘지 않는지 확인합니다. VDSM이 계속해서 최대 오프셋이 임계값을 넘지 않는다고 표시하는 한 Red Hat Virtualization Manager는 문제의 논리 볼륨에 작업을 계속할 수 있을 정도로 충분한 스토리지가 있다는 것을 알 수 있습니다.

QEMU에서 사용량이 증가하여 임계값을 초과했다고 표시하는 경우 VDSM은 디스크 이미지가 곧 논리 볼륨의 크기에 도달한다고 Manager에 알립니다. Red Hat Virtualization Manager는 SPM 호스트에서 논리 볼륨을 확대하도록 요청합니다. 데이터 센터의 데이터 스토리지 도메인에 사용 가능한 공간이 있는 한 이러한 프로세스를 반복할 수 있습니다. 데이터 스토리지 도메인에 사용 가능한 빈 공간이 없어지게 되면 사용자가 스토리지 용량을 수동으로 추가하여 확대할 수 있습니다.

## 2.12. 논리 볼륨 확장

Red Hat Virtualization Manager는 씬 프로비저닝을 사용하여 스토리지 풀에 사용 가능한 스토리지를 오버커밋하고 실제 사용 가능한 것보다 많은 스토리지를 할당합니다. 가상 머신은 작동하면서 데이터를 씁니다. 씬 프로비저닝된 디스크 이미지를 가진 가상 머신은 결과적으로 그 디스크 이미지를 백업하는 논리 볼륨에서 감당할 수 있는 것보다 많은 데이터를 쓰게 됩니다. 이렇게 되는 경우 논리 볼륨 확장을 사용하여 스토리지가 추가되고 가상 머신이 계속 작동할 수 있게 됩니다.

Red Hat Virtualization은 LVM에서 씬 프로비저닝 메커니즘을 제공합니다. QCOW2 포맷 스토리지 사용 시 Red Hat Virtualization은 호스트 시스템 프로세스인 qemu-kvm에 의존하여 순차적으로 논리 볼륨에 스토리지 블록을 매핑하도록 합니다. 이를 통해 예를 들어 1 GB 논리 볼륨이 백업하는 100 GB 논리 디스크가 정의될 수 있습니다. qemu-kvm이 VDSM에서 설정된 사용량 임계값을 넘는 경우 로컬 VDSM 인스턴스에서 논리 볼륨을 1 GB 확장하도록 SPM에 요청합니다. 볼륨 확장이 필요한 가상 머신을 실행하는 호스트의

VDSM은 더 많은 공간이 필요하다는 것을 SPM VDSM에 알립니다. SPM은 논리 볼륨을 확장하고 SPM VDSM 인스턴스는 호스트 VDSM이 볼륨 그룹 정보를 새로고침하고 확장 작업이 완료되었다는 것을 인지하도록 합니다. 호스트는 작업을 계속할 수 있습니다.

논리 볼륨을 확장하려면 호스트는 다른 어느 호스트가 SPM인지 알아야 할 필요가 없습니다. 이 호스트가 바로 SPM일 수도 있습니다. 스토리지 확장 통신은 스토리지 메일박스를 통해서 진행됩니다. 스토리지 메일박스는 데이터 스토리지 도메인에 있는 전용 논리 볼륨을 말합니다. 논리 볼륨 확장에 SPM이 필요한 호스트는 스토리지 메일박스에서 해당 호스트에 대해 지정된 구역에 메시지를 씁니다. SPM은 수신 메일을 정기적으로 읽고 요청된 논리 볼륨 확장을 수행하며 송신 메일에 답메일을 작성합니다. 호스트는 요청을 보낸 후 2초마다 이에 대한 답변을 수신 메일함에서 확인합니다. 호스트가 요청한 논리 볼륨 확장에 대한 성공적인 답변을 받으면 새로 할당된 스토리지를 인식하도록 **device mapper**에 있는 논리 볼륨 맵을 새로고침합니다.

스토리지 풀에 사용 가능한 실제 스토리지가 거의 소진 시 여러 이미지가 사용할 수 있는 스토리지가 떨어져 리소스를 보충할 방도가 없게 됩니다. 스토리지 풀이 스토리지 소진 시 QEMU에서 **enospc error**를 보내며, 이는 장치에 더 이상 사용 가능한 스토리지가 없다는 것을 의미합니다. 이 시점에서 가상 머신 실행이 자동으로 일시중지되며 볼륨 그룹에 새 LUN을 수동으로 추가하는 작업을 해야 합니다.

볼륨 그룹에 새 LUN 추가 시 **Storage Pool Manager**는 추가 스토리지를 필요로 하는 논리 볼륨에 스토리지를 배포합니다. 추가 리소스의 자동 할당을 통해 관련 가상 머신은 서비스 중단 없이 작업을 계속하거나 서비스가 중단된 경우 작업을 복귀할 수 있습니다.

## 3장. 네트워크

### 3.1. 네트워크 아키텍처

Red Hat Virtualization 네트워크에는 기본 네트워킹, 클러스터 내에서의 네트워킹, 그리고 호스트 네트워킹 설정 등이 있습니다. 기본 네트워킹은 네트워킹을 가능하게 해주는 기본적인 하드웨어 및 소프트웨어 요소를 포함합니다. 클러스터 내에서의 네트워킹은 호스트, 논리 네트워크, 그리고 가상머신 등과 같은 클러스터 수준의 개체 간의 네트워크 소통이 포함됩니다. 호스트 네트워킹 설정은 호스트 내에서의 네트워킹을 지원하는 설정을 다룹니다.

네트워크를 잘 설계해서 만드는 경우 예를 들어 높은 광대역을 사용하는 작업에 적절한 광대역이 주어지고 사용자가 사용 시 속도가 느리지 않고 가상 머신이 마이그레이션 도메인 내에서 성공적으로 마이그레이션될 수 있게 됩니다. 네트워크가 제대로 설계되지 않은 경우 예를 들어 대기 시간이 너무 길어지고 네트워크 플러딩(flooding)으로 인해 마이그레이션 및 복제가 실패합니다.

### 3.2. 소개: 기본 네트워킹 용어

Red Hat Virtualization은 가상 머신, 가상화 호스트, 그리고 다음과 같은 것을 사용하는 폭넓은 네트워크 간의 네트워킹 기능을 제공합니다:

- Network Interface Controller (NIC)
- 브리지
- 본딩
- 가상 NIC
- 가상 LAN (VLAN)

NIC, 브리지, 그리고 vNIC는 호스트, 가상 머신, 로컬 네트워크, 그리고 인터넷 간의 네트워크 통신을 가능하게 해줍니다. 옵션으로 본딩 및 VLAN을 구현하여 보안, 내결함성, 및 네트워크 용량을 향상합니다.

### 3.3. NETWORK INTERFACE CONTROLLER

NIC(Network Interface Controller)란 컴퓨터를 컴퓨터 네트워크에 연결하는 네트워크 아답터 또는 LAN 아답터를 말합니다. NIC는 시스템의 물리적 및 데이터 링크 계층에서 작동하여 네트워크 연결을 가능하게 해줍니다. Red Hat Virtualization 환경의 모든 가상화 호스트는 최소한 하나의 NIC가 있으며, 호스트에 두 개 이상의 NIC가 있는 경우가 더 많습니다.

하나의 물리적 NIC에 여러 개의 Virtual NIC(vNIC)가 논리적으로 연결될 수 있습니다. 가상 NIC는 가상 머신의 물리적인 네트워크 인터페이스 역할을 합니다. vNIC와 이를 지원하는 NIC에서 구분하기 위해 Red Hat Virtualization Manager는 각 vNIC에 고유의 MAC 주소를 배정합니다.

### 3.4. 브리지

브리지란 패킷 교환 네트워크에서 패킷 포워딩을 사용하는 소프트웨어 장치를 말합니다. 브리징을 사용하여 여러 네트워크 인터페이스 장치가 하나의 NIC 연결을 공유하고 네트워크에서 개별적인 물리 장치로 표시됩니다. 브리지는 패킷의 소스 주소를 확인하여 관련 대상 주소를 파악합니다. 대상 주소가 파악되면 브리지는 해당 위치를 나중에 참조하기 위해 표에 추가합니다. 이를 통해 호스트는 브리지의 멤버인 가상 머신에 관련된 vNIC로 네트워크 통신을 리디렉션할 수 있게 됩니다.

Red Hat Virtualization에서 브리지를 사용하여 논리 네트워크를 구현합니다. 호스트의 물리적 인터페이스가 아닌 브리지가 IP 주소를 받게 됩니다. 브리지와 관련된 IP 주소는 연결을 위해 브리지를 사용하는 가

상 머신과 같은 서브넷에 있지 않아도 됩니다. 브리지가 해당 브리지를 사용하는 가상 머신과 같은 서브넷에 있는 IP 주소를 배정받는 경우 논리 네트워크 내에서 가상 머신이 그 호스트에 대해 주소를 지정할 수 있게 됩니다. 원칙적으로는 가상화 호스트에서 네트워크에 노출된 서비스를 실행하지 않는 것이 좋습니다. 게스트는 VNIC를 사용하여 논리 네트워크에 연결되며 호스트는 NIC를 사용하여 논리 네트워크의 원격 요소에 연결됩니다. 각 게스트는 게스트의 VNIC의 IP 주소를 독립적으로, DHCP에 의해서 또는 정적으로 설정할 수 있습니다. 브리지는 호스트 외부에 있는 개체에 연결할 수 있지만 이러한 연결이 필수적인 것은 아닙니다.

브리지 및 Ethernet 연결 모두에 대해 사용자 지정 설정을 정의할 수 있습니다. VDSM은 네트워크 정의 및 사용자 지정 속성을 네트워크 후크 설정 스크립트에 보냅니다.

### 3.5. 본딩

본딩은 여러 네트워크 인터페이스 카드를 단일 소프트웨어 장치로 정의하는 것입니다. 본딩된 네트워크 인터페이스는 본딩에 포함된 네트워크 인터페이스 카드의 전송 기능을 결합하여 단일 네트워크 인터페이스로 동작하기 때문에 단일 네트워크 인터페이스 카드 보다 더 신속한 전송 속도 기능을 제공합니다. 본딩에 있는 모든 네트워크 인터페이스 카드에 오류가 발생하지 않으면 본딩 자체에 오류가 발생하지 않기 때문에 본딩의 내결함성을 증가시킬 수 있습니다. 하지만 본딩된 네트워크 인터페이스를 구성하는 네트워크 인터페이스 카드는 본딩에서 동일한 옵션 및 모드를 지원하도록 동일한 제조업체 및 모델이어야 한다는 제한이 있습니다.

본딩의 패킷 분산 알고리즘은 사용하는 본딩 모드에 의해 결정됩니다.



#### 중요

모드 1, 2, 3, 4는 가상 머신 (브리지됨) 및 비 가상 머신 (브리지되지 않음) 네트워크 유형 모드를 지원합니다. 모드 0, 5, 6은 비 가상 머신 (브리지되지 않음) 네트워크만 지원합니다.

#### 본딩 모드

Red Hat Virtualization은 기본적으로 모드 4를 사용하지만 다음과 같은 일반적인 본딩 모드를 지원합니다:

##### 모드 0 (round-robin 정책)

네트워크 인터페이스 카드를 통해 순차적으로 패킷을 전송합니다. 패킷은 본딩에서 처음 사용 가능한 네트워크 인터페이스 카드로 시작하여 마지막으로 사용 가능한 네트워크 인터페이스 카드로 종료하는 루프에 전송됩니다. 이후의 모든 루프는 처음 사용 가능한 네트워크 인터페이스 카드에서 시작됩니다. 모드 0는 내결함성을 제공하고 본딩의 모든 네트워크 인터페이스 카드에서 부하 균형을 조정합니다. 하지만 모드 0는 브리지와 함께 사용할 수 없으므로 가상 머신 논리 네트워크와 호환되지 않습니다.

##### 모드 1 (active-backup 정책)

하나의 네트워크 인터페이스 카드는 활성 상태로 두고 다른 모든 네트워크 인터페이스 카드를 백업 상태로 설정합니다. 활성 네트워크 인터페이스 카드에 오류가 발생하는 경우 백업 네트워크 인터페이스 카드 중 하나가 본딩에서 활성 네트워크 인터페이스 카드로 네트워크 인터페이스 카드를 대체합니다. 모드 1에서 본딩의 MAC 주소가 활성 네트워크 인터페이스 카드를 반영하도록 변경될 경우 발생할 수 있는 혼란을 방지하기 위해 MAC 주소는 하나의 포트에만 표시됩니다. 모드 1은 내결함성을 제공하고 Red Hat Virtualization에서 지원됩니다.

##### 모드 2 (XOR 정책)

소스에서 XOR 연산 결과 및 대상 MAC 주소 모듈 네트워크 인터페이스 카드 슬레이브 수에 따라 패킷을 전송할 네트워크 인터페이스 카드를 선택합니다. 이러한 계산을 통해 각각의 대상 MAC 주소에 동일한 네트워크 인터페이스 카드가 선택됩니다. 모드 2는 내결함성 및 부하 분산을 제공하고 Red Hat Virtualization에서 지원됩니다.

**모드 3 (broadcast 정책)**

모든 네트워크 인터페이스 카드에 모든 패킷을 전송합니다. 모드 3은 내결합성을 제공하고 Red Hat Virtualization에서 지원됩니다.

**모드 4 (IEEE 802.3ad 정책)**

인터페이스가 동일한 속도 및 이중 설정을 공유하는 집계 그룹을 생성합니다. 모드 4는 IEEE 802.3ad 사양에 따라 활성 집계 그룹에 있는 모든 네트워크 인터페이스 카드를 사용하며 Red Hat Virtualization에서 지원됩니다.

**모드 5 (adaptive transmit load balancing 정책)**

본딩의 각 네트워크 인터페이스 카드에서 부하에 따라 발신 트래픽 계정이 분산되어 현재 네트워크 인터페이스 카드가 모든 들어오는 트래픽을 수신할 수 있게 합니다. 수신 트래픽에 할당된 네트워크 인터페이스 카드에 오류가 발생할 경우 다른 네트워크 인터페이스 카드가 수신 트래픽의 역할을 하도록 지정됩니다. 모드 5는 브리지와 함께 사용할 수 없으므로 가상 머신 논리 네트워크와 호환되지 않습니다.

**모드 6 (adaptive load balancing 정책)**

특별한 전환 요구 사항없이 모드 5 (adaptive transmit load balancing 정책)는 IPv4 트래픽의 수신 부하 분산과 결합되어 있습니다. ARP 협상은 수신 부하 분산에 사용됩니다. 모드 6는 브리지와 함께 사용할 수 없으므로 가상 머신 논리 네트워크와 호환되지 않습니다.

### 3.6. 본딩을 위한 스위치 설정

스위치 설정을 사용자의 하드웨어 요구 사항에 따라 다릅니다. 사용자 운영 체제에 해당하는 배포 및 네트워킹 설정 가이드를 참조하시기 바랍니다.

**중요**

모든 스위치 종류에 대하여 Cisco Port Aggregation Protocol (PAgP) 프로토콜이 아닌 Link Aggregation Control Protocol (LACP) 프로토콜과 스위치 본딩을 설정해야 합니다.

### 3.7. 가상 네트워크 인터페이스 카드 (vNIC)

vNIC란 호스트의 물리적인 네트워크 인터페이스 카드를 기반으로 하는 가상 네트워크 인터페이스를 말합니다. 각 호스트는 여러 NIC가 있을 수 있으며 각 네트워크 인터페이스 마드는 여러 가상 NIC(vNIC)의 기반이 될 수 있습니다.

가상 머신에 vNIC 연결 시 Red Hat Virtualization Manager는 vNIC를 연결하는 가상 머신, vNIC 자체, 그리고 vNIC가 기반으로 하는 물리적인 호스트 NIC 간에 여러 연결을 생성합니다. 특히 vNIC가 가상 머신에 연결되는 경우 vNIC가 기반으로 하는 물리적인 호스트 NIC 상에 새로운 vNIC 및 MAC 주소가 생성됩니다. 이렇게 vNIC 연결 후 처음 가상 머신 시작 시 libvirt는 이러한 vNIC에 PCI 주소를 배정합니다. 그 후 MAC 주소 및 PCI 주소를 사용하여 가상 머신에 있는 vNIC의 이름(예: eth0)을 가져옵니다.

MAC 주소를 할당하고 해당 MAC 주소를 PCI 주소와 연결하는 과정은 템플릿 또는 스냅샷을 기반으로 가상 머신을 생성하는 경우에 약간 다릅니다. 템플릿 또는 스냅샷에 대하여 PCI 주소가 이미 생성된 경우 해당 템플릿이나 스냅샷을 기반으로 생성된 가상 머신에 있는 vNIC는 이러한 PCI 주소에 따라 순서가 정해지며 이러한 순서대로 MAC 주소가 할당됩니다. PCI 주소가 템플릿에 대하여 아직 생성되지 않은 경우 해당 템플릿을 기반으로 생성된 가상 머신의 vNIC는 vNIC의 이름 순서대로 할당됩니다. PCI 주소가 스냅샷에 대하여 아직 생성되지 않은 경우 Red Hat Virtualization Manager는 해당 스냅샷을 기반으로 하는 가상 머신의 vNIC에 새 MAC 주소를 할당합니다.

VNIC는 생성된 후 네트워크 브리지 장치에 추가됩니다. 네트워크 브리지 장치는 가상 머신이 가상 머신 논리 네트워크에 연결할 수 있게 해줍니다.

가상화 호스트에서 **ip addr show** 명령 실행 시 해당 호스트에 있는 가상 머신과 연결된 모든 VNIC가 표시됩니다. 또한 논리 네트워크를 백업하기 위해 생성된 모든 네트워크 브리지도 표시되며 해당 호스트가 사용하는 모든 네트워크 인터페이스 카드(NIC)가 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
```

위의 명령에 대하여 콘솔에 다음과 같이 여러 장치가 표시됩니다: 한 개의 루프 백 장치 (**lo**), 한 개의 Ethernet 장치(**eth0**), 한 개의 무선 장치 (**wlan0**), 한 개의 VDSM 더미(**dummy**) 장치( **;vdsmdummy;**), 다섯 개의 본딩 장치(**bond0**, **bond4**, **bond1**, **bond2**, **bond3**), 그리고 한 개의 네트워크 브리지(**ovirtmgmt**).

VNIC는 모두 네트워크 브리지 장치 및 논리 네트워크의 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** 명령에 대한 콘솔 출력 내용을 보면 virtio VNIC가 **ovirtmgmt** 브리지의 멤버인 것으로 표

시됩니다. VNIC가 연결된 모든 가상 머신은 `ovirtmgmt` 논리 네트워크에 연결되어 있습니다. `eth0` NIC도 `ovirtmgmt` 브리지의 멤버입니다. `eth0` 장치는 호스트 범위 밖에서 연결을 제공하는 스위치에 케이블로 연결되어 있습니다.

### 3.8. 가상 LAN (VLAN)

VLAN(Virtual LAN)이란 네트워크 패킷에 적용할 수 있는 속성을 말합니다. 네트워크 패킷은 번호가 부여된 VLAN으로 "태그"될 수 있습니다. VLAN이란 스위치 수준에서 네트워크 트래픽을 완전히 분리하는 데 사용되는 보안 기능입니다. VLAN은 완전히 구분되어 있으며 상호 배타적입니다. Red Hat Virtualization Manager는 VLAN을 인지하고 있으며 VLAN 트래픽을 태그하고 리디렉션할 수 있지만 VLAN 구현을 위해서는 VLAN을 지원하는 스위치가 필요합니다.

스위치 수준에서 포트에 VLAN 대상이 할당됩니다. 스위치는 특정 포트에서 발생하는 트래픽에 VLAN 태그를 적용하여 이러한 트래픽을 VLAN의 일부로 표시하고 이에 대한 응답이 같은 VLAN 태그를 갖도록 합니다. VLAN은 여러 스위치에 걸쳐서 확장될 수 있습니다. 스위치에서 VLAN 태그가된 네트워크 트래픽은 올바른 VLAN이 지정된 포트에 연결된 가상 머신 이외에는 전혀 감지할 수 없습니다. 하나의 포트가 여러 VLAN에 태그될 수 있으며 이를 통해 여러 VLAN의 트래픽이 하나의 포트에 전송될 수 있고 이러한 트래픽을 수신하는 시스템의 소프트웨어를 사용하여 해독할 수 있습니다.

### 3.9. 네트워크 레이블

네트워크 레이블을 사용하여 논리 네트워크 생성 및 관리와 이러한 논리 네트워크를 물리적인 호스트 네트워크 인터페이스 및 본딩과 연결하는 것과 관련된 여러 관리 작업을 간소화할 수 있습니다.

네트워크 레이블이란 논리 네트워크 또는 물리적인 호스트 네트워크 인터페이스에 연결할 수 있으며 사람이 읽을 수 있는 일반 텍스트 레이블을 말합니다. 엄격하게 정해진 레이블 길이는 없지만 알파벳 대문자, 소문자, 밑줄, 그리고 하이픈의 조합을 사용해야 하며 공백이나 특수 문자는 사용할 수 없습니다.

논리 네트워크 또는 물리적인 호스트 네트워크 인터페이스에 레이블 연결 시 같은 레이블이 이미 연결되어 있는 다른 논리 네트워크 또는 물리적인 호스트 네트워크 인터페이스와 다음과 같은 연결이 생기게 됩니다:

#### 네트워크 레이블 연결

- 논리 네트워크에 레이블 연결 시 그 논리 네트워크는 해당 레이블이 있는 아무 물리적인 호스트 네트워크 인터페이스에 자동으로 연결됩니다.
- 물리적인 호스트 네트워크 인터페이스에 레이블 연결 시 해당 레이블이 있는 모든 논리 네트워크는 그 물리적인 호스트 네트워크 인터페이스에 자동으로 연결됩니다.
- 논리 네트워크 또는 물리적인 호스트 네트워크 인터페이스에 연결된 레이블을 변경하는 것은 레이블을 제거하거나 추가하는 것과 같은 작동을 하게 됩니다. 관련된 논리 네트워크 또는 물리적인 호스트 네트워크 인터페이스 간의 연결이 업데이트됩니다.

#### 네트워크 레이블 및 클러스터

- 레이블이 있는 논리 네트워크가 클러스터에 추가되고 그 클러스터에 같은 레이블이 있는 물리적인 호스트 네트워크 인터페이스가 있는 경우 그 논리 네트워크는 이러한 물리적인 호스트 네트워크 인터페이스에 자동으로 추가됩니다.
- 레이블이 있는 논리 네트워크가 클러스터에서 분리되고 그 클러스터에 같은 레이블이 있는 물리적인 호스트 네트워크 인터페이스가 있는 경우 그 논리 네트워크는 이러한 물리적인 호스트 네트워크 인터페이스에서 자동으로 분리됩니다.

#### 네트워크 레이블 및 역할

- 레이블이 있는 논리 네트워크가 디스플레이 네트워크 또는 마이그레이션 네트워크 역할을 할당 받는 경우 그 논리 네트워크는 IP 주소를 할당 받을 수 있도록 DHCP를 사용하여 물리적인 호스트 네트워크 인터페이스에서 설정되게 됩니다.

역할 네트워크 (예: "마이그레이션 네트워크" 또는 "디스플레이 네트워크")에 레이블을 설정하면 네트워크 전체 호스트에 일괄적으로 대량 배포됩니다. 이러한 네트워크 일괄 추가는 DHCP 사용을 통해 처리됩니다. 이러한 일괄적인 대량 배포 방식은 다수의 고정 IP 주소를 입력하는 작업이 쉽지 않기 때문에 선택되었습니다.

### 3.10. 클러스터 네트워킹

클러스터 수준의 네트워킹 개체는 다음과 같습니다:

- 클러스터
- 논리 네트워크

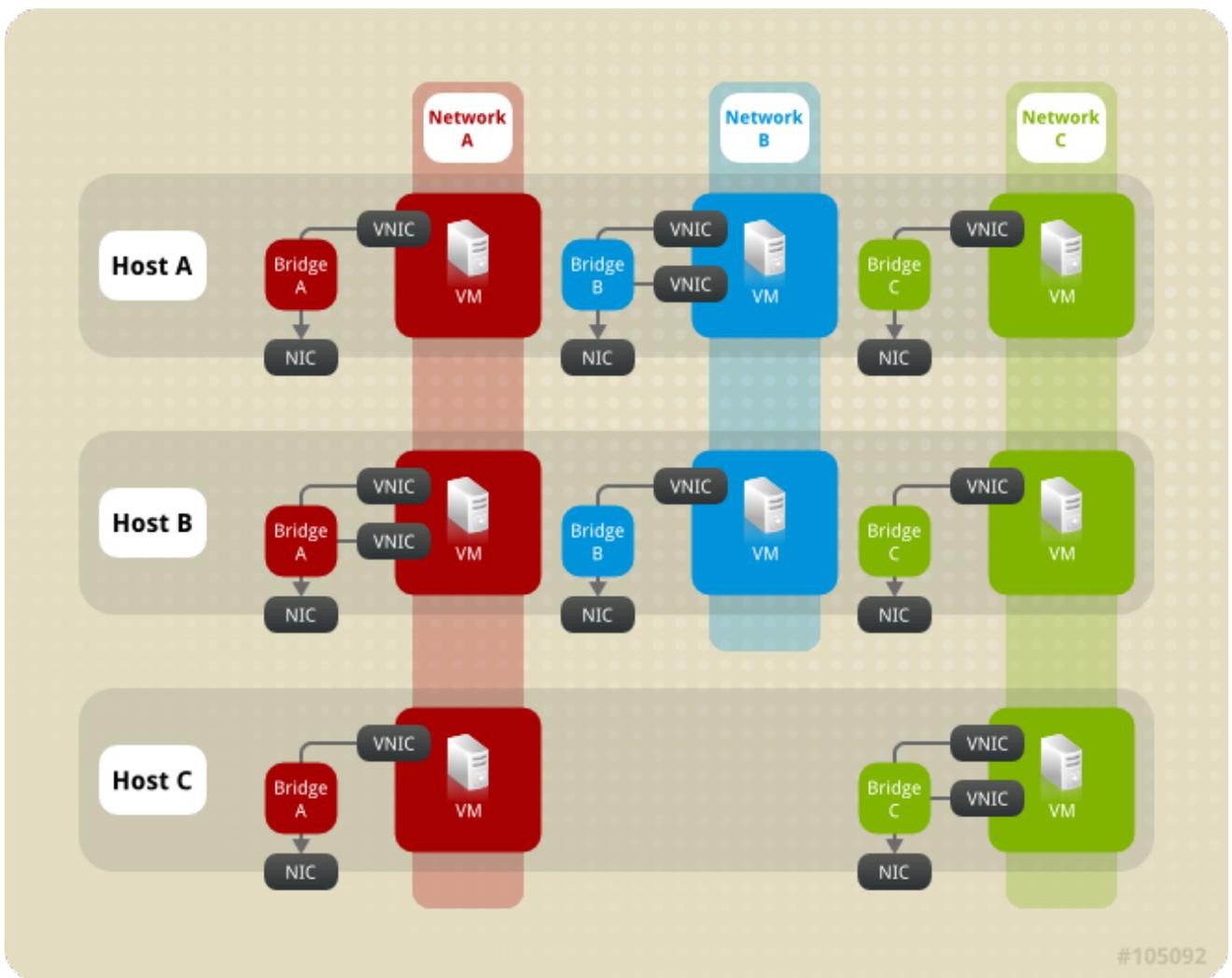


그림 3.1. 클러스터 내에서의 네트워킹

데이터 센터란 여러 호스트의 논리적인 그룹인 각 클러스터를 논리적인 여러 개 모아놓은 그룹을 말합니다. 그림 3.1. “클러스터 내에서의 네트워킹”에서는 하나의 클러스터의 콘텐츠를 묘사합니다.

클러스터 내의 모든 호스트는 같은 스토리지 도메인에 액세스할 수 있습니다. 또한 클러스터 내의 호스트는 클러스터 수준에서 논리 네트워크가 적용됩니다. 가상 머신 논리 네트워크가 가상 머신과 작동할 수 있게 되려면 Red Hat Virtualization Manager를 사용하여 클러스터 내의 각 호스트에 대하여 네트워킹을 정

의하고 구현해야 합니다. 다른 종류의 논리 네트워크는 이러한 네트워크를 사용하는 호스트에서만 구현될 수 있습니다.

멀티 호스트 네트워크 설정은 네트워크가 할당된 데이터 센터 내에 있는 모든 호스트에 업데이트된 네트워크 설정을 모두 자동으로 적용합니다.

### 3.11. 논리 네트워크

논리 네트워크를 통해 **Red Hat Virtualization** 환경에서 네트워크 트래픽을 종류별로 분리할 수 있습니다. 예를 들어, **Red Hat Virtualization** 설치 시 **Manager**와 호스트 간의 관리 통신에 사용될 **ovirtmgmt** 네트워크가 기본으로 생성됩니다. 논리 네트워크는 요구 사항이나 사용 용도가 비슷한 네트워크 트래픽을 그룹으로 묶는데 가장 흔히 사용됩니다. 대부분의 경우 최적화 및 문제 해결을 위해 트래픽을 종류별로 분리하기 위해 관리자가 스토리지 네트워크와 디스플레이 네트워크를 생성하게 됩니다.

논리 네트워크에는 다음과 같은 종류가 있습니다:

- 가상 머신 네트워크 트래픽을 전송하는 논리 네트워크
- 가상 머신 네트워크 트래픽을 전송하지 않는 논리 네트워크
- 선택적 논리 네트워크
- 필수 네트워크

모든 논리 네트워크는 필수 또는 옵션(선택적)입니다.

논리 네트워크는 데이터 센터 수준에서 정의되어서 호스트에 추가됩니다. 필수 논리 네트워크가 작동하려면 해당 클러스터 내의 모든 호스트에 대해 구현해야 합니다.

**Red Hat Virtualization** 환경의 각 가상 머신은 호스트의 네트워크 브리지 장치에 의해 백업됩니다. 즉 어떤 클러스터에 대해 새로운 가상 머신 논리 네트워크 정의 시 클러스터 내의 각 호스트에서 일치하는 브리지 장치를 우선 생성해야 논리 네트워크가 작동을 하여 가상 머신에서 사용할 수 있게 됩니다. **Red Hat Virtualization Manager**는 가상 머신 논리 네트워크에 대하여 필요한 브리지를 자동으로 생성합니다.

가상 머신 논리 네트워크를 백업하기 위해 **Red Hat Virtualization Manager**에서 생성한 브리지 장치는 호스트 네트워크 인터페이스와 연결됩니다. 브리지의 일부인 호스트 네트워크 인터페이스가 네트워크 연결이 되는 경우 이후에 브리지에 추가되는 모든 네트워크 인터페이스는 브리지의 이러한 네트워크 연결을 공유하게 됩니다. 가상 머신 생성 후 특정 논리 네트워크에 배치 시 이러한 가상 머신의 가상 네트워크 카드는 해당 논리 네트워크의 브리지에 포함됩니다. 이를 통해 가상 머신은 브리지에 연결된 다른 가상 머신이나 개체와 통신할 수 있게 됩니다.

가상 머신 네트워크 트래픽에 사용되지 않는 논리 네트워크는 호스트 네트워크 인터페이스에 직접 연결됩니다.

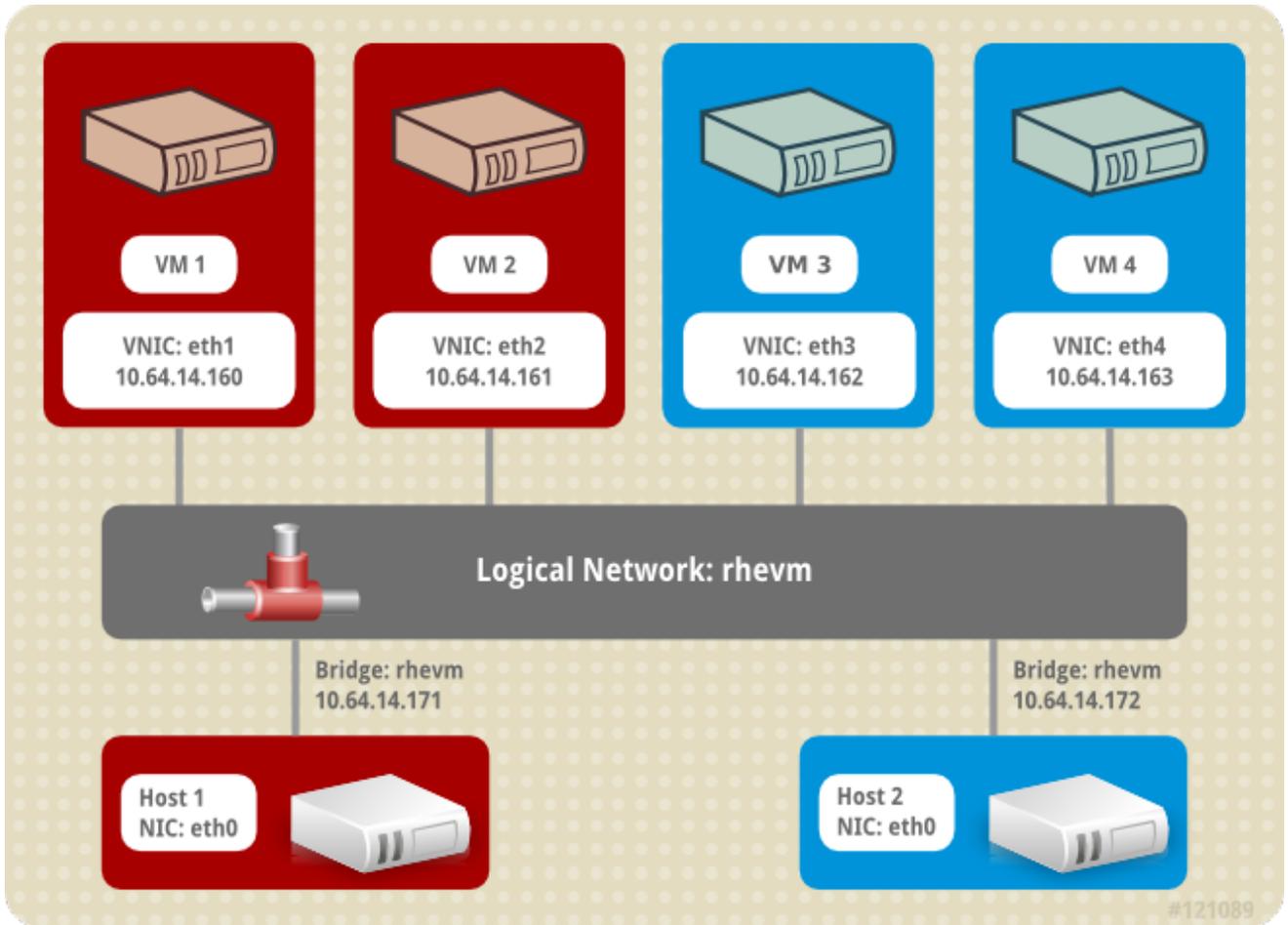


그림 3.2. ovirtmgmt 논리 네트워크

예 3.1. 논리 네트워크 사용 예

Purple이라는 데이터 센터 내에 있는 Pink라는 클러스터에 Red 및 White라는 두 호스트가 있습니다. Red와 White는 모두 기본 논리 네트워크인 **ovirtmgmt**를 사용하여 모든 네트워크 기능을 수행했습니다. Pink를 담당하는 시스템 관리자는 웹 서버의 네트워크 테스트를 분리하기 위해 웹 서버와 일부 클라이언트 가상 머신을 개별 논리 네트워크에 놓기로 합니다. 해당 관리자는 이러한 새 논리 네트워크를 **network\_testing**이라고 부르기로 합니다.

먼저 이 관리자는 Purple 데이터 센터의 논리 네트워크를 정의합니다. 그리고 나서 이를 Pink 클러스터에 적용합니다. 논리 네트워크는 반드시 유지보수 모드에 있는 호스트에서 구현해야 합니다. 즉, 관리자는 실행 중인 모든 가상 머신을 Red로 마이그레이션한 후 White를 유지보수 모드로 전환합니다. 그 후 브리지에 포함될 물리적인 네트워크 인터페이스와 연결된 **네트워크**를 수정합니다. 선택된 네트워크 인터페이스의 **연결상태**가 Down에서 **Non-Operational**로 바뀝니다. 작동하지 않는 (**non-operational**) 상태가 되는 이유는 Pink 클러스터에 있는 각 호스트의 물리적인 네트워크 인터페이스를 **network\_testing** 네트워크에 추가하여 클러스터 내의 모든 호스트에서 관련 브리지가 설정되어야 하기 때문입니다. 그 다음으로 관리자는 White를 활성화하고 Red에서 실행 중인 모든 가상 머신을 마이그레이션한 후 위와 같은 절차를 Red에서 반복합니다.

White와 Red 모두 **network\_testing** 논리 네트워크가 물리적인 네트워크 인터페이스에 브리징된 경우 **network\_testing** 논리네트워크는 **Operational** 상태가 되어 가상 머신에서 사용할 수 있게 됩니다.

3.12. 필수 네트워크, 선택적 네트워크, 가상 머신 네트워크

필수네트워크란 클러스터의 모든 호스트에서 사용할 수 있어야 하는 논리 네트워크를 말합니다. 호스트의 필수 네트워크가 **non-operational** 상태가 된 경우 호스트에서 실행되는 가상 머신은 다른 호스트로 마이그레이션됩니다. 마이그레이션 범위는 선택한 스케줄링 정책에 따라 다릅니다. 이는 중요 작업을 실행하는 가상 머신의 경우 유용합니다.

선택적(옵션) 네트워크란 명시적으로 **필수** 네트워크로 선언하지 않은 논리 네트워크를 말합니다. 선택적 네트워크는 이를 사용하는 호스트에서만 구현될 수 있습니다. 선택적 네트워크의 존재 유무가 호스트의 **Operational** 상태에 영향을 미치지 않습니다. 필수가 아닌 네트워크가 작동하지 않는 (**non-operational**) 상태가 된 경우 네트워크에서 실행되는 가상 머신은 다른 호스트로 마이그레이션되지 않습니다. 이는 대용량 마이그레이션에 의한 불필요한 I/O 과부하가 발생하지 않게 합니다. 논리네트워크 생성 후 클러스터에 추가 시 **Required** 체크 상자가 기본으로 선택되어 있음에 유의합니다.

네트워크의 **필수** 대상을 변경하려면 관리 포털에서 네트워크 선택 후 **클러스터** 탭을 클릭하고 **네트워크 관리** 버튼을 클릭합니다.

가상 머신 네트워크 (사용자 인터페이스에서 **가상 머신 네트워크**라고 함)는 논리 네트워크로 가상 머신의 네트워크 트래픽만을 전송하도록 지정되어 있습니다. 가상 머신 네트워크는 필수 또는 옵션으로 설정할 수 있습니다. 선택적인 가상 머신 네트워크를 사용하는 가상 머신은 그 네트워크를 사용하는 호스트에서만 시작할 수 있습니다.

### 3.13. 가상 머신 연결

Red Hat Virtualization에서 가상 머신 생성 시 그 가상 머신의 NIC가 논리 네트워크에 놓이게 됩니다. 그 시점부터 가상 머신은 같은 네트워크에 있는 모든 대상과 통신을 할 수 있게 됩니다.

호스트 관점에서 볼 때 가상 머신을 논리 네트워크에 두면 그 가상 머신의 NIC를 백업하는 vNIC는 해당 논리 네트워크의 브리지 장치의 멤버로 추가됩니다. 예를 들어, 가상 머신이 **ovirtmgmt** 네트워크에 있는 경우 그 가상 머신의 vNIC는 가상 머신이 실행되는 호스트의 **ovirtmgmt** 브리지의 멤버로 추가됩니다.

### 3.14. 포트 미러링

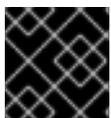
포트 미러링은 주어진 논리 네트워크 및 호스트에 있는 레이어 3 네트워크 트래픽을 가상 머신의 가상 인터페이스에 복사합니다. 이 가상 머신을 사용하여 네트워크 디버깅 및 튜닝, 침입 감지, 그리고 같은 호스트 및 논리 네트워크에 있는 다른 가상 머신의 동작 등의 작업을 할 수 있습니다.

유일하게 복사되는 트래픽은 하나의 호스트에 있는 논리 네트워크의 내부 트리플렉스입니다. 호스트 외부 네트워크에서 트래픽이 증가하지 않지만 포트 미러링이 활성화된 가상 머신은 다른 가상 머신보다 호스트 CPU 및 RAM을 더 많이 사용합니다.

포트 미러링은 논리 네트워크의 vNIC 프로파일에서 활성화/비활성화되며 다음과 같은 제한 사항이 있습니다:

- 포트 미러링이 활성화된 포트가 있는 프로파일의 vNIC를 핫 플래깅하는 것은 지원되지 않습니다.
- vNIC 프로파일이 가상 머신에 연결되어 있는 경우 포트 미러링을 변경할 수 없습니다.

이러한 제한 사항으로 인해 추가적인 전용 vNIC 프로파일에서 포트 미러링을 활성화하는 것이 좋습니다.



#### 중요

포트 미러링 활성화 시 다른 네트워크 사용자의 개인정보 보호 수준이 낮아집니다.

### 3.15. 호스트 네트워크 설정

가상화 호스트에서 일반적인 네트워크 설정은 다음과 같습니다:

- 브리지 및 NIC 설정
- 브리지, VLAN, 및 NIC 설정
- 브리지, 본딩, 및 VLAN 설정
- 다중 브리지, 다중 VLAN, 및 NIC 설정

### 3.16. 브리지 설정

Red Hat Virtualization에서 가장 간단한 호스트 설정은 브리지 및 NIC 설정입니다. [그림 3.3. “브리지 및 NIC 설정”](#)에서 설명된 것처럼 이러한 설정 시 브리지를 사용하여 하나 이상의 가상 머신(또는 게스트)을 호스트의 NIC에 연결하게 됩니다.

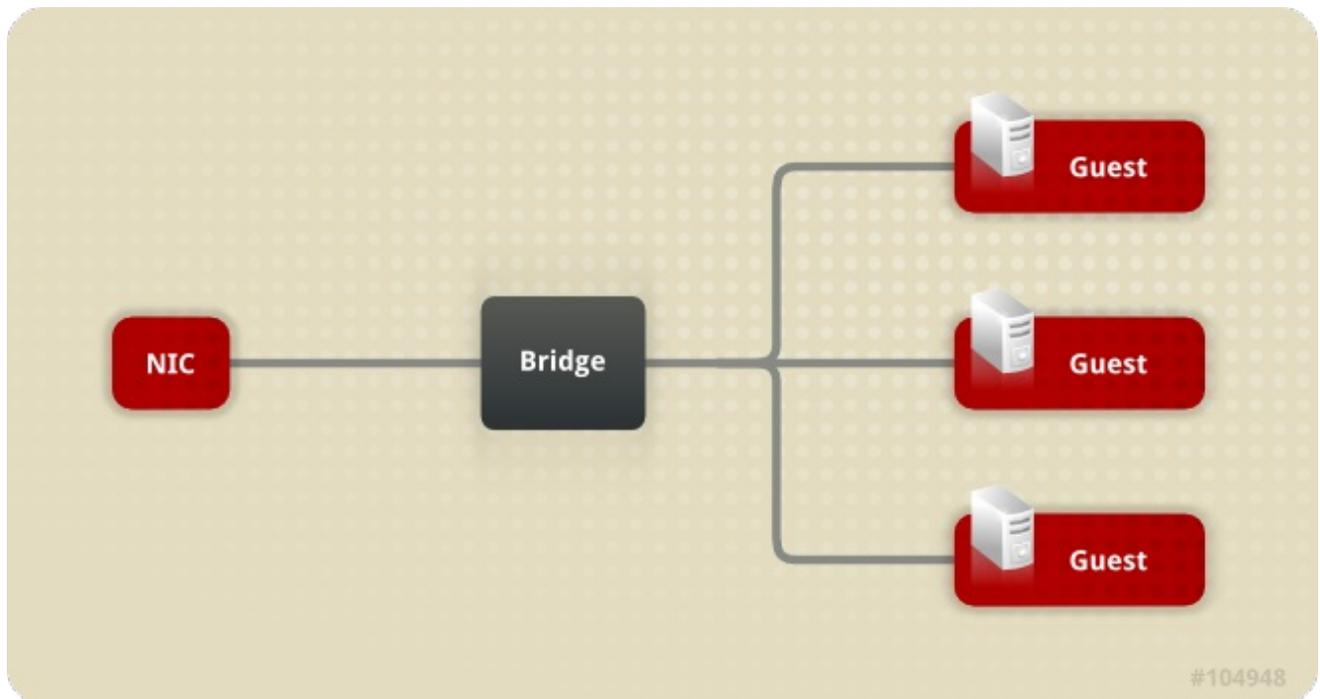


그림 3.3. 브리지 및 NIC 설정

이러한 설정의 예는 Red Hat Virtualization Manager 설치 시 브리지 `ovirtmgmt`가 자동 생성되는 것에서 볼 수 있습니다. Red Hat Virtualization Manager 설치 시 호스트에 `VDSM`을 설치합니다. `VDSM` 설치 프로세스 중 브리지 `ovirtmgmt`가 생성됩니다. 그 후 `ovirtmgmt` 브리지가 호스트의 IP 주소를 획득하여 호스트의 관리 통신을 활성화합니다.

### 3.17. VLAN 설정

[그림 3.4. “브리지, VLAN, 및 NIC 설정”](#)에서 묘사된 대체 설정을 보면 가상 LAN(VLAN)이 호스트 NIC 및 브리지에 연결됩니다.

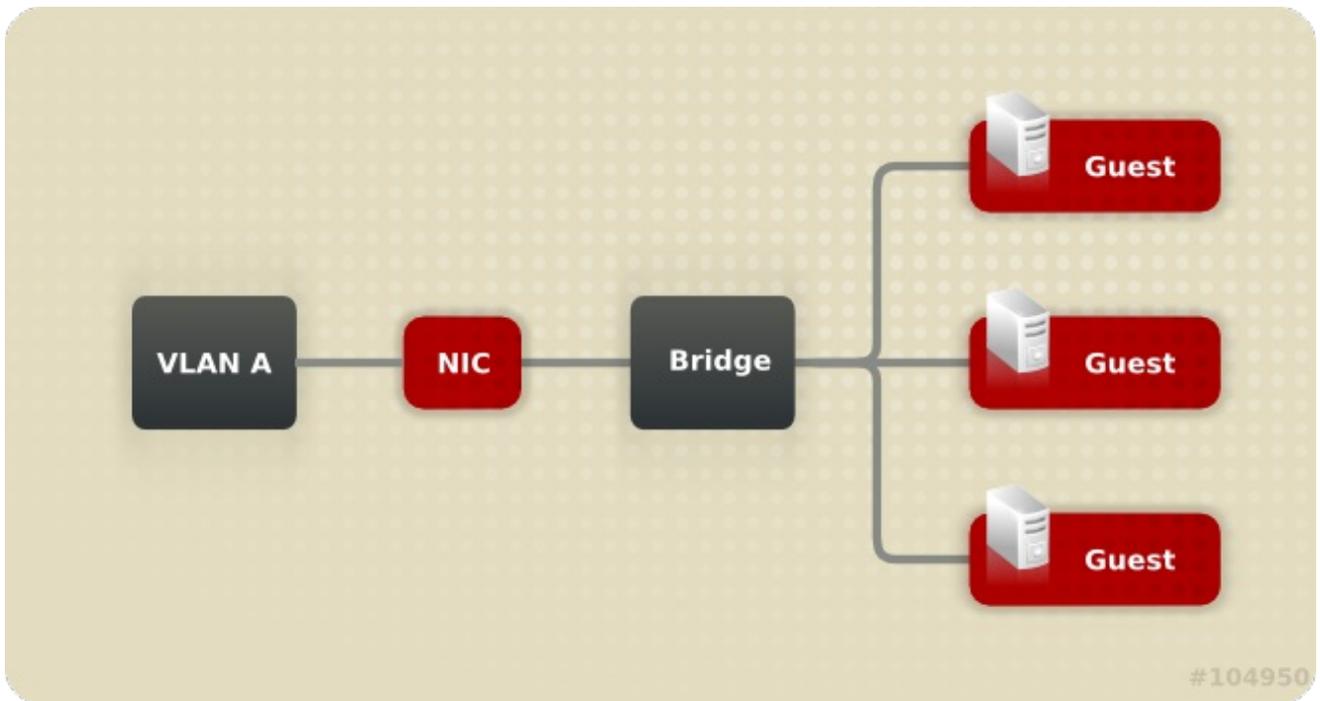


그림 3.4. 브리지, VLAN, 및 NIC 설정

여기에 포함되는 VLAN은 이 네트워크를 통해 데이터가 안전하게 전송될 수 있게 해주며 또한 여러 VLAN를 사용하는 하나의 NIC에 여러 브리지를 연결할 수 있는 옵션도 지원합니다.

### 3.18. 브리지 및 본딩 설정

그림 3.5. “브리지, 본딩, 그리고 NIC 설정”에서 묘사된 설정을 보면 본딩을 사용하여 여러 호스트 NIC를 같은 브리지 및 네트워크에 연결할 수 있습니다.

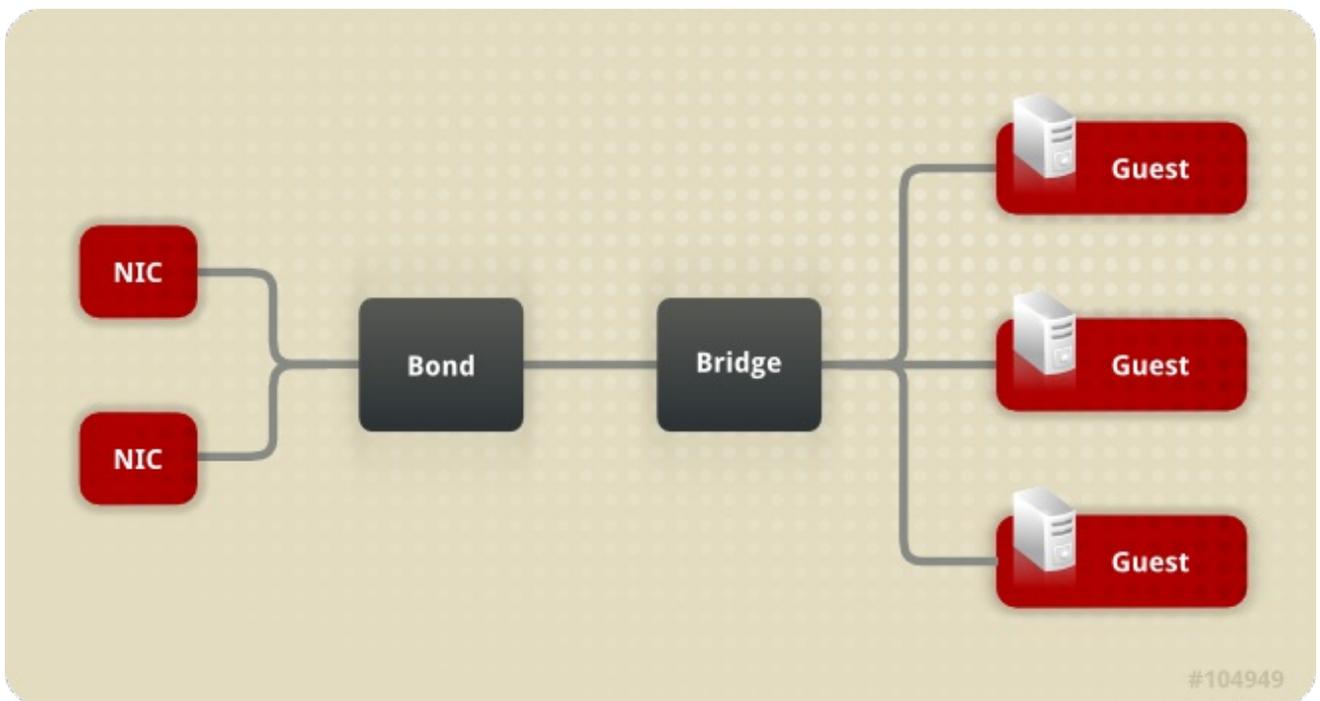


그림 3.5. 브리지, 본딩, 그리고 NIC 설정

여기에 포함된 본딩은 두 개 이상의 물리적인 Ethernet 링크를 통합하는 논리 링크를 생성합니다. 그 결과 본딩 모드에 따라 NIC 내결함성 및 대역폭 확장 가능성 등과 같은 이점이 생깁니다.

### 3.19. 다중 브리지, 다중 VLAN, 및 NIC 설정

그림 3.6. “다중 브리지, 다중 VLAN, 및 NIC 설정” 에서 묘사된 설정에서는 하나의 NIC를 두 개의 VLAN에 연결합니다. 이러한 작업은 네트워크 스위치가 두 개의 VLAN 중 하나에 태그된 네트워크 트래픽을 호스트에 있는 하나의 NIC로 전송하도록 설정된 것을 전제로 합니다. 호스트는 두 VNIC를 각 VLAN에 하나씩 지정하여 VLAN 트래픽을 분리합니다. 두 VLAN 중 하나에 태그된 후 트래픽은 적절한 VNIC을 브리지 멤버로 뒤서 개별 브리지에 연결합니다. 그리고 나서 여러 가상 머신에서 각 브리지에 연결합니다.

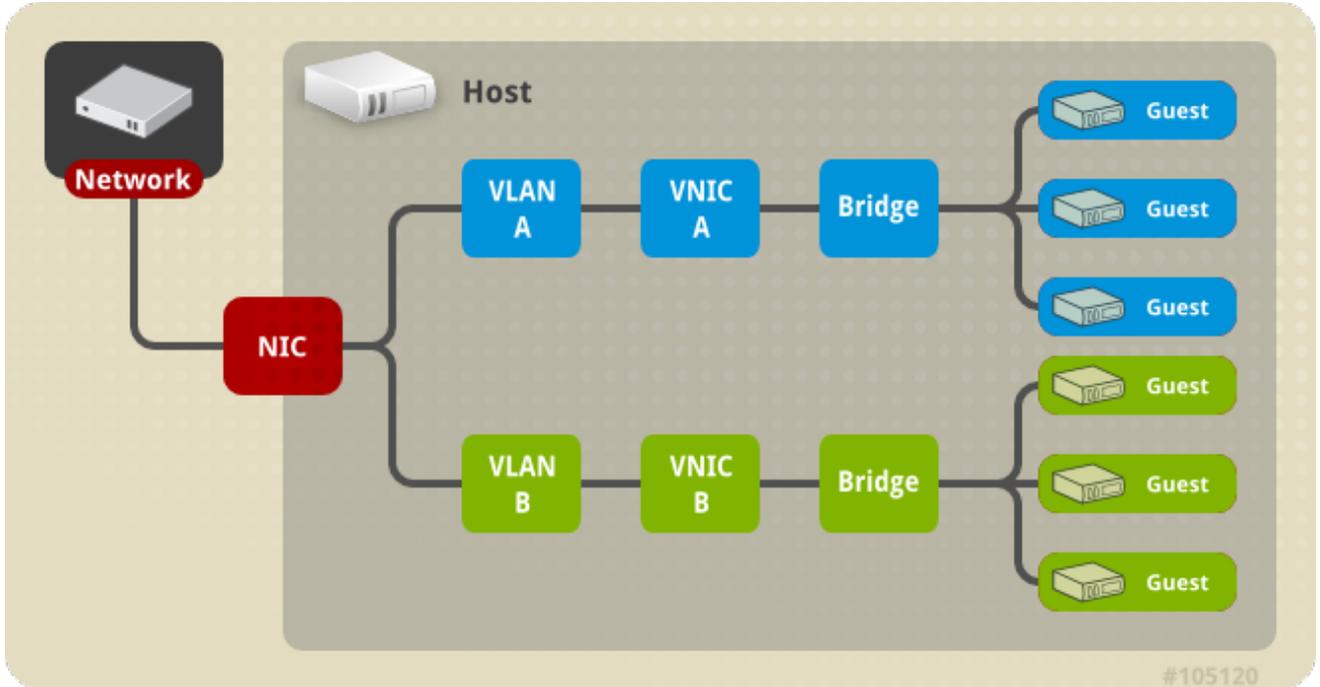


그림 3.6. 다중 브리지, 다중 VLAN, 및 NIC 설정

### 3.20. 다중 브리지, 다중 VLAN, 및 본딩 설정

그림 3.7. “본딩 연결을 사용하는 다중 브리지, 다중 VLAN, 및 다중 NIC” 에서 묘사된 설정을 보면 여러 NIC를 본딩하여 여러 VLAN에 연결할 수 있습니다.

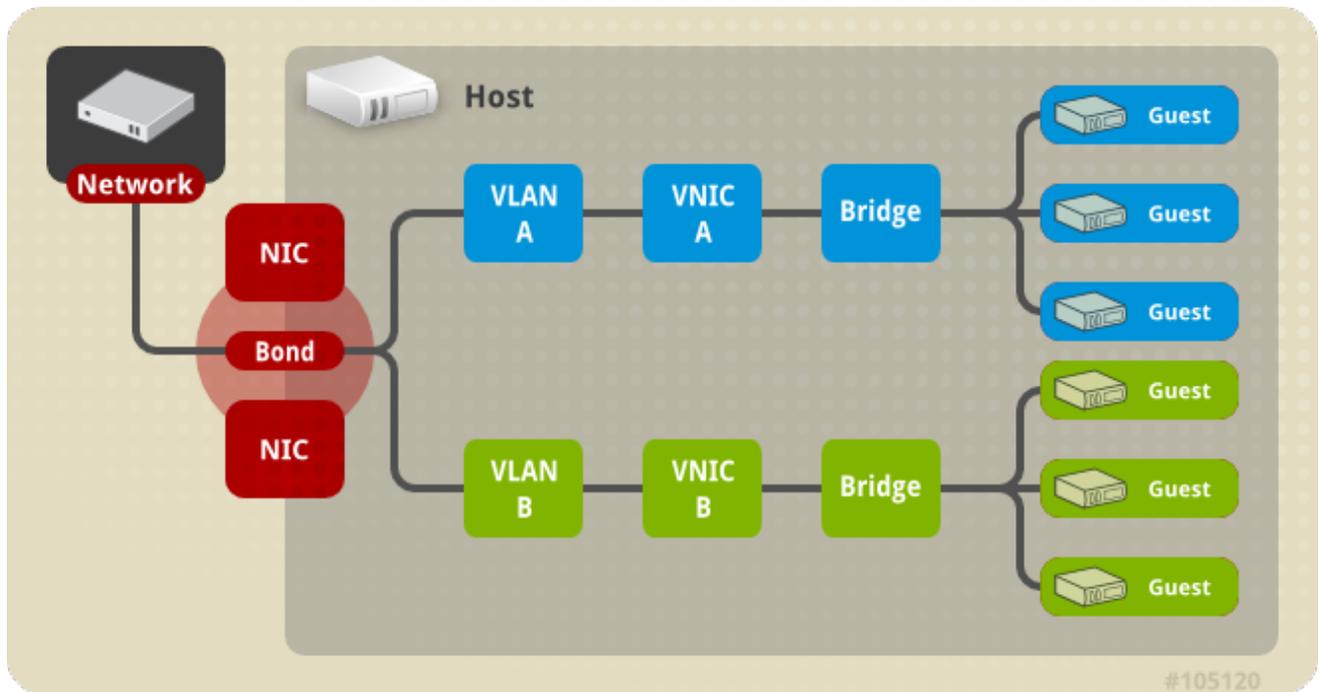


그림 3.7. 본딩 연결을 사용하는 다중 브리지, 다중 VLAN, 및 다중 NIC

이러한 설정에서 각 VLAN은 NIC를 연결하는 본딩에 정의됩니다. 각 VLAN은 개별 브리지에 연결하며 각 브리지는 하나 이상의 게스트에 연결합니다.

## 4장. 전원 관리

### 4.1. 전원 관리 및 펜싱 소개

Red Hat Virtualization 환경은 전원 관리 및 펜싱이 설정된 경우 가장 유연하며 복원력이 높습니다. Red Hat Virtualization Manager는 전원 관리를 사용하여 호스트 전원 주기 작업을 제어해서 문제가 감지된 호스트를 재부팅하는 가장 중요한 작업을 할 수 있습니다. 펜싱 사용 시 문제가 되는 호스트를 재부팅하여 이 호스트를 작동 중인 Red Hat Virtualization 환경에서 분리해서 성능 저하를 방지합니다. 그 후 관리자는 펜싱된 호스트를 응답 가능한 상태로 복원하고 환경에 다시 놓습니다.

전원 관리 및 펜싱은 특별 전용 하드웨어를 사용하여 호스트 운영 체제와 독립적으로 호스트를 다시 시작합니다. Red Hat Virtualization Manager는 네트워크 IP 주소 또는 호스트 이름을 사용하여 전원 관리 장치에 연결합니다. Red Hat Virtualization의 경우 전원 관리 장치와 펜싱 장치는 같습니다.

### 4.2. RED HAT VIRTUALIZATION에서 프록시를 통해 전원 관리

Red Hat Virtualization Manager는 펜스 에이전트와 직접 통신하지 않습니다. 대신 Manager는 프록시를 사용하여 호스트 전원 관리 장치에 전원 관리 명령을 전송합니다. Manager는 VDSM을 사용하여 전원 관리 장치 작업을 실행하고 환경에 있는 다른 호스트는 펜싱 프록시로 사용됩니다.

다음 중 하나를 선택할 수 있습니다:

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

사용 가능한 펜싱 프록시 호스트는 *UP* 또는 *유지 관리* 상태입니다.

### 4.3. 전원 관리

Red Hat Virtualization Manager는 작동하지 않거나 응답하지 않는 상태가 된 호스트를 재부팅할 수 있으며 절전을 위해 사용량이 적은 호스트를 종료할 준비를 할 수 있습니다. 이러한 기능은 올바르게 설정된 전원 관리 장치를 필요로 합니다. Red Hat Virtualization 환경은 다음과 같은 전원 관리 장치를 지원합니다:

- *American Power Conversion (apc).*
- *Bladecenter.*
- *Cisco Unified Computing System(cisco\_ucs).*
- *Dell Remote Access Card 5(drac5).*
- *Dell Remote Access Card 7(drac7).*
- *Electronic Power Switch (eps).*
- *HP BladeSystem (hpblade).*
- *Integrated Lights Out(ilo, ilo2, ilo3, ilo4).*
- *Intelligent Platform Management Interface(ipmilan).*
- *Remote Supervisor Adapter (rsa).*
- *rsb.*

- *Western Telematic, Inc (wti).*



## 참고

APC 5.x 전원 관리장치는 **apc** 펜스 에이전트에서 지원하지 않습니다. 대신 **apc\_snmp** 펜스 에이전트를 사용합니다.

위에 나열된 전원 관리 장치와 통신하기 위해 **Red Hat Virtualization Manager**는 *펜스 에이전트*를 사용합니다. **Red Hat Virtualization Manager**는 환경에서 관리자가 전원 관리 장치가 허용하고 응답하는 매개 변수를 사용하여 전원 관리 장치에 대한 펜스 에이전트를 설정할 수 있게 해줍니다. **GUI**를 사용하여 기본 설정 옵션을 설정할 수 있습니다. 또한 특별 설정 옵션도 입력할 수 있으며 이는 구문 분석(파싱)되지 않고 펜스 장치로 전달됩니다. 특별 설정 옵션은 특정 펜스 장치에 해당되지만 기본 설정 옵션은 지원되는 모든 전원 관리 장치에서 제공하는 기능에 해당됩니다. 모든 전원 관리 장치에서 제공하는 기본 기능은 다음과 같습니다:

- **Status:** 호스트의 상태를 확인합니다.
- **Start:** 호스트의 전원을 켭니다.
- **Stop:** 호스트의 전원을 끕니다.
- **Restart:** 호스트를 다시 시작합니다. 실제로는 중지, 대기, 상태, 시작, 대기, 상태로 실행됩니다.

가장 모범적인 방법은 초기 전원 관리 설정 시 이러한 설정을 테스트하고 그 후에 계속 제 기능을 할 수 있도록 종종 테스트하는 것입니다.

환경 내의 모든 호스트에서 전원 관리 장치를 올바르게 설정하여 회복력을 제공합니다. 펜싱 에이전트는 **Red Hat Virtualization Manager**가 호스트 전원 관리 장치와 통신하여 문제가 있는 호스트의 운영 체제를 우회하고 해당 호스트를 다시 부팅하여 환경에서 분리할 수 있게 해줍니다. 그 후 **Manager**는 문제가 있는 호스트가 **SPM** 역할을 했던 경우 이 역할을 다른 호스트에 배정하며 모든 고가용성 가상 머신을 다른 호스트에서 안전하게 다시 시작합니다.

## 4.4. 펜싱

**Red Hat Virtualization** 환경에서 펜싱이란 **Manager**가 펜스 에이전트를 사용하여 호스트 재부팅을 시작하고 전원 관리 장치가 해당 재부팅을 수행하는 것을 말합니다. 펜싱을 통해 클러스터는 예기치 않은 호스트 오류, 절전, 부하 분산, 가상 머신 가용성 정책에 대응할 수 있습니다.

펜싱을 통해 **Storage Pool Manager (SPM)** 역할이 항상 작동 중인 호스트에 배정되도록 합니다. 펜싱된 호스트가 **SPM**인 경우 해당 **SPM** 역할을 포기하고 이 역할은 응답을 하는 호스트에 다시 배정됩니다. **SPM** 역할을 갖고 있는 호스트만이 데이터 도메인 구조 메타데이터를 쓸 수 있기 때문에 응답을 하지 않고 펜싱되지 않은 **SPM** 호스트로 인해 해당 환경에서는 데이터 도메인 구조 메타데이터를 변경해야 하는 가상 디스크 생성 및 파기, 스냅샷 생성, 논리 볼륨 확장 등의 모든 작업을 할 수 없게 됩니다.

호스트가 응답하지 않는 상태가 되면 현재 해당 호스트에서 실행 중인 모든 가상 머신도 응답하지 않는 상태가 될 수 있습니다. 하지만 응답하지 않는 호스트는 해당 호스트에서 실행 중인 가상 머신의 가상 머신 하드 디스크 이미지에 대한 잠금을 유지합니다. 가상 머신을 제 2의 호스트에서 시작하고 가상 머신 하드 디스크 이미지 쓰기 권한을 이러한 호스트에 할당하는 것을 시도하는 경우 데이터 손상을 초래할 수 있습니다.

펜싱을 통해 **Red Hat Virtualization Manager**은 가상 머신 하드 디스크 이미지에 대한 잠금이 해제된 것으로 간주할 수 있으며 **Manager**는 펜스 에이전트를 사용하여 문제가 되는 호스트가 재부팅되었는지 확인할 수 있습니다. 이에 대한 확인을 수신하는 경우 **Red Hat Virtualization Manager**는 데이터 손상의 위험 없이

문제의 호스트에 있는 가상 머신을 다른 호스트에서 시작할 수 있습니다. 펜싱은 고가용성 가상 머신의 기반이 됩니다. 고가용성으로 표시된 가상 머신은 다른 호스트에서 시작 시 데이터를 손상하지 않을 것이라는 확신이 없으면 다른 호스트에서 안전하게 시작할 수 없습니다.

호스트가 응답 없음 (non-responsive) 상태가 되는 경우 Red Hat Virtualization Manager는 어떠한 작업을 하기 전에 호스트가 일시적인 오류에서 회복할 수 있도록 30초의 유예 기간을 둡니다. 유예 기간이 지날 때까지 호스트가 응답을 하지 않는 경우 Manager는 자동으로 응답 없음 상태의 호스트로 인한 모든 부정적인 영향을 최소화하기 시작합니다. Manager는 호스트의 전원 관리 카드에 대한 펜싱 에이전트를 사용하여 호스트 중지, 중지 여부 확인, 호스트 시작, 그리고 호스트 시작 여부 확인 등의 작업을 합니다. 호스트 부팅 완료 후 호스트는 펜싱되기 전에 소속되어 있던 클러스터에 다시 참여하려고 시도하게 됩니다. 호스트가 응답 없음 상태가 되도록 만든 문제가 재부팅으로 인해 해결된 경우 호스트가 자동으로 Up 상태가 되며 다시 한번 가상 머신을 시작 및 호스팅할 수 있게 됩니다.

### 4.5. 호스트 소프트웨어 펜싱

경우에 따라 호스트는 예상치 못한 문제가 발생하여 응답 없음 (non-responsive) 상태가 될 수 있으며 VDSM은 요청에 응답할 수 없지만 VDSM에 의존하는 가상 머신은 가동 및 액세스 가능한 상태가 됩니다. 이러한 경우 VDSM을 다시 시작하면 VDSM은 응답 가능한 상태가 되어 이러한 문제가 해결됩니다.

"SSH 소프트웨어 펜싱"은 Manager가 응답 없음 상태의 호스트에서 SSH를 통해 VDSM을 다시 시작하는 절차입니다. Manager가 SSH를 통해 VDSM을 다시 시작할 수 없는 경우 외부 펜싱 에이전트가 설정되어 있을 때 펜싱은 외부 펜싱 에이전트의 책임이 됩니다.

SSH를 통한 소프트웨어 펜싱은 다음과 같은 경우에 작동합니다. 호스트에서 펜싱이 설정되어 활성화되어 있어야 하며 유효한 프록시 호스트 (데이터 센터에서 UP 상태의 두 번째 호스트)가 존재해야 합니다. Manager와 호스트 간의 연결 시간 초과 시 다음과 같은 상태가 발생합니다:

1. 첫 번째 네트워크 장애 발생 시 호스트 상태는 "connecting"으로 변경됩니다.
2. Manager는 VDSM에게 상태를 3번 확인하거나 호스트에서의 로딩에 의해 결정되는 대기 시간 동안 기다립니다. 이러한 대기 시간을 결정하는 수식은  $TimeoutToResetVdsInSeconds$  (기본값 60 초) +  $[DelayResetPerVmlnSeconds$  (기본값 0.5 초)]\*(호스트에서 실행 중인 가상 머신 수) +  $[DelayResetForSpmlnSeconds$  (기본값 20 초)] \* 1 (호스트가 SPM으로 실행 중인 경우) 또는 0 (호스트가 SPM으로 실행하지 않는 경우)의 설정 값에 의해 지정됩니다. VDSM에 최대 응답 시간을 부여하려면 Manager는 위의 옵션 (VDSM의 상태를 3번 확인하거나 위의 수식에 의해 지정되는 대기 시간 동안 대기) 중 두 개의 긴 옵션을 선택합니다.
3. 시간이 경과해도 호스트가 응답하지 않을 경우 SSH를 통해 `vdsmd restart`가 실행됩니다.
4. `vdsmd restart`가 호스트와 Manager 간의 연결을 다시 설정하지 않을 경우 호스트 상태는 **Non Responsive**로 변경되며 전원 관리가 설정되어 있을 경우 펜싱은 외부 펜싱 에이전트로 전달됩니다.



#### 참고

SSH를 통한 소프트웨어 펜싱은 전원 관리가 설정되어 있지 않은 호스트에서 실행될 수 있습니다. 이는 "펜싱 (fencing)"과는 다릅니다. 펜싱은 전원 관리가 설정되어 있는 호스트에서만 수행할 수 있습니다.

### 4.6. 여러 전원 관리 펜싱 에이전트 사용

단일 에이전트는 주요 에이전트로 취급됩니다. 보조 에이전트는 예를 들어 각 전원 스위치에서 같은 전원 스위치에 연결된 두 개의 에이전트를 갖는 이중 전원 호스트의 경우와 같이 펜싱 에이전트가 두 개 있는 경우 유효합니다. 에이전트 유형은 같거나 다를 수 있습니다.

호스트에 여러 펜싱 에이전트가 있는 경우 펜싱 절차의 신뢰성이 증가합니다. 예를 들어 호스트의 단일 펜싱 에이전트가 실패하는 경우 호스트는 수동으로 재부팅되기 전까지 작동하지 않는 상태로 남아 있습니다. 그 호스트에서 실행 중이던 가상 머신은 일시 중단되며 기존 호스트가 수동으로 펜싱된 후에만 클러스터 내의 다른 호스트로 페일오버합니다. 여러 에이전트가 있는 경우 첫 에이전트가 실패 시 다음 에이전트를 호출합니다.

호스트에 두 펜싱 에이전트가 정의된 경우 이러한 펜싱 에이전트는 **동시** 또는 **순차적** 흐름을 사용하도록 설정할 수 있습니다:

- **동시:** 주요 및 보조 에이전트 모두 중지 명령에 응답해야 호스트가 중지될 수 있습니다. 한 에이전트가 시작 명령에 응답 시 호스트가 시작하게 됩니다.
- **순차적:** 호스트를 중지 또는 시작하기 위해 먼저 주요 에이전트를 사용하며 이것이 실패하면 보조 에이전트를 사용합니다.

## 5장. 로드 밸런싱, 스케줄링, 그리고 마이그레이션

### 5.1. 로드 밸런싱, 스케줄링, 그리고 마이그레이션

개별 호스트는 하드웨어 리소스가 한정되어 있으며 오류가 생길 수 있습니다. 오류와 리소스 소모를 완화하기 위해 호스트는 클러스터로 묶이며 이는 기본적으로 공유 리소스를 묶는 것입니다. **Red Hat Virtualization** 환경은 로드 밸런싱 정책, 스케줄링, 그리고 마이그레이션을 사용하여 호스트 리소스에 대한 수요의 변화에 대응합니다. **Manager**는 클러스터 내에서 단 하나의 호스트가 해당 클러스터의 모든 가상 머신을 담당하지 않도록 합니다. 반대로 말하면 **Manager**는 사용률이 낮은 호스트를 감지하고 해당 호스트에서 모든 가상 머신을 다른 호스트로 마이그레이션하여 관리자가 해당 호스트를 종료하고 전원을 절약할 수 있게 해줍니다.

다음과 같은 세 가지 이벤트의 결과로 사용 가능한 리소스를 확인합니다:

- 가상 머신 시작: 가상 머신을 어느 호스트에서 시작할지 결정하기 위해 리소스를 확인합니다.
- 가상 머신 마이그레이션: 적절한 대상 호스트를 결정하기 위해 리소스를 확인합니다.
- 시간 경과: 개별 호스트의 부하(load)가 클러스터 로드 밸런싱 정책을 준수하고 있는지 확인하기 위해 리소스를 정기적으로 확인합니다.

**Manager**는 사용 가능한 리소스의 변화에 대응하기 위해 클러스터의 로드 밸런싱 정책을 사용하여 클러스터 내의 한 호스트에서 다른 호스트로 가상 머신의 마이그레이션을 스케줄링합니다. 로드 밸런싱 정책, 스케줄링, 그리고 가상 머신 마이그레이션 간의 관계는 다음 부분에서 설명됩니다.

### 5.2. 로드 밸런싱 정책

클러스터에 대해 로드 밸런싱 정책이 설정되며 이때 하나 이상의 호스트에서 각각 다른 하드웨어 매개 변수와 가용 메모리를 갖게 될 수 있습니다. **Red Hat Virtualization Manager**는 로드 밸런싱 정책을 사용하여 클러스터 내의 어느 호스트에서 가상 머신을 시작할지 결정합니다. 또한 로드 밸런싱 정책은 **Manager**가 언제 사용률이 높은 호스트에서 사용률이 낮은 호스트로 가상 머신을 이동할지 결정할 수 있게 해줍니다.

로드 밸런싱 절차는 데이터 센터 내의 각 클러스터에서 매분 한 번씩 실행됩니다. 이를 통해 사용률이 높거나 낮은 호스트를 확인하고 어느 호스트가 유효한 가상 머신 마이그레이션 대상인지 결정할 수 있습니다. 이러한 결정은 특정 클러스터에 대해 관리자가 설정한 로드 밸런싱 정책을 바탕으로 이루어집니다. 로드 밸런싱 정책 옵션은 **VM\_Evenly\_Distributed**, **Evenly\_Distributed**, **Power\_Saving**, 그리고 **None**입니다.

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

가상 머신에서 균등 배분 (**evenly distributed**) 로드 밸런싱 정책은 가상 머신 수에 따라 가상 머신을 호스트 간에 균등하게 배분합니다. 최대 가상 머신 수는 각 호스트에서 실행 가능한 최대 가상 머신 수를 말하며 이를 초과 시 호스트는 과부하 상태가 됩니다. **VM\_Evenly\_Distributed** 정책은 관리자가 호스트에 대하여 최대 가상 머신 수를 설정할 수 있게 해줍니다. 관리자는 또한 사용률이 가장 높은 호스트와 사용률이 가장 낮은 호스트 간의 최대 가상 머신 수 차이 값도 설정합니다. 클러스터 내의 모든 호스트에 있는 가상 머신 수가 이러한 마이그레이션 임계값에 내에 있는 경우 클러스터는 균형 상태가 됩니다. 또한 관리자는 **SPM** 호스트에 예약된 가상 머신용 슬롯 수를 정의합니다. **SPM** 호스트는 다른 호스트보다 부하가 낮기 때문에 이러한 변수는 호스트 이외에 얼마나 적은 수의 가상 머신을 실행할 수 있는지 정의합니다. 호스트가 최대 가상 머신 수보다 많은 가상 머신을 실행하고 있고 마이그레이션 임계값 범위를 벗어나는 수의 가상 머신을 가진 호스트가 최소 하나 있는 경우 가상 머신은 차례대로 하나씩 **CPU** 사용량이 가장 낮은 클러스터의 호스트로 마이그레이션됩니다. 클러스터의 모든 호스트의 가상 머신 수가 마이그레이션 임계값 범위에 들게 될 때까지 가상 머신을 한 번에 하나씩 마이그레이션합니다.

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

균등 배분 (**evenly distributed**) 로드 밸런싱 정책에서는 새 가상 머신을 실행하기 위해 CPU 또는 메모리 사용량이 가장 낮은 호스트를 선택합니다. 최대 서비스 수준은 클러스터의 호스트에 허용된 최대 CPU 및 메모리 사용량이며 이를 초과 시 환경의 성능이 저하됩니다. **Evenly\_Distributed** 정책은 관리자가 가상 머신 실행에 대한 최대 서비스 수준을 설정할 수 있게 해줍니다. 또한 관리자는 **Red Hat Virtualization Manager**가 개입하기 전까지 호스트가 이러한 최대 서비스 수준을 지속할 수 있는 시간도 설정합니다. 호스트가 최대 서비스 수준에 도달하고 정해진 시간을 초과하여 이 수준을 유지하는 경우 해당 호스트의 가상 머신은 클러스터에서 CPU 또는 메모리 사용량이 가장 낮은 호스트로 하나씩 마이그레이션됩니다. 매 분 한 번씩 호스트 리소스가 확인되며 CPU 및 메모리 사용량이 최대 서비스 수준 임계값 미만으로 떨어질 때까지 가상머신 이 한 번에 하나씩 마이그레이션됩니다.

## 5.5. 로드 밸런싱 정책: **POWER\_SAVING**

절전 로드 밸런싱 정책에서는 새 가상 머신을 실행하기 위해 CPU 및 메모리 사용량이 가장 낮은 호스트를 선택합니다. 최대 서비스 수준은 클러스터의 호스트에 허용된 최대 CPU 및 메모리 사용량이며 이를 초과 시 환경의 성능이 저하됩니다. 최소 서비스 수준은 호스트를 계속해서 작동하는 것이 전기를 비효율적으로 사용하는 것으로 간주될 때까지 호스트의 작동이 허용되는 최소 CPU 및 메모리 사용량을 말합니다. **Power\_Saving** 정책은 관리자가 가상 머신 실행에 대한 최대 및 최소 서비스 수준을 설정할 수 있게 해줍니다. 또한 관리자는 **Red Hat Virtualization Manager**가 개입하기 전까지 호스트가 이러한 최대 및 최소 서비스 수준을 지속할 수 있는 시간도 설정합니다. 호스트가 최대 서비스 수준에 도달하고 정해진 시간을 초과하여 이 수준을 유지하는 경우 해당 호스트의 가상 머신은 CPU 및 메모리 사용량이 가장 낮은 호스트로 하나씩 마이그레이션됩니다. 이러한 작업은 CPU 및 메모리 사용량이 최대 서비스 수준 밑으로 떨어질 때까지 계속됩니다. 호스트 CPU 또는 메모리 사용량이 최대서비스 수준 미만으로 떨어지는 경우 클러스터의 다른 호스트의 최대 서비스 수준이 허용하는 데 한해 해당 호스트로 가상 머신이 마이그레이션됩니다. 사용률이 낮은 호스트에 남아 있던 가상 머신이 모두 해제되는 경우 **Manager**는 해당 호스트 시스템의 전원을 자동으로 종료하며 로드 밸런싱이 필요하거나 클러스터에 사용 가능한 호스트가 충분하지 않을 때 이 호스트를 다시 시작하게 됩니다.

## 5.6. 로드 밸런싱 정책: **NONE**

로드 밸런싱 정책을 선택 하지 않는 경우 CPU 사용량 및 가용 메모리가 가장 낮은 클러스터의 호스트에서 가상 머신이 시작됩니다. CPU 사용량을 확인하기 위해 가상 CPU 수 및 사용량의 퍼센트 값의 혼합 메트릭이 사용됩니다. 이 방법에서 유일하게 호스트를 선택하는 지점은 새 가상 머신이 시작할 때이므로 이것이 가장 덜 동적(**dynamic**)인 방법입니다. 호스트에서 수요의 증가를 반영하기 위해 가상 머신이 자동으로 마이그레이션되지는 않습니다.

관리자는 어떤 가상 머신에 대하여 어느 호스트가 가장 적절한 마이그레이션 대상인지 결정해야 합니다. 또한 가상 머신은 **고정(pinning)**을 사용하여 특정 호스트에 연결될 수도 있습니다. 가상 머신 고정 시 다른 호스트에 자동으로 마이그레이션되는 것을 방지합니다. 리소스 소비량이 높은 환경에서는 수동 마이그레이션을 사용하는 것이 가장 좋습니다.

## 5.7. 고가용성 가상 머신 예약

고가용성 (**HA**) 가상 머신 예약 정책을 통해 **Red Hat Virtualization Manager**는 고가용성 가상 머신에 대한 클러스터 용량을 모니터링할 수 있습니다. **Manager**는 개별 가상 머신을 고가용성(**HA**)으로 표시할 수 있으며, 이는 곧 호스트 실패 시 이렇게 표시된 가상 머신이 다른 호스트에서 재부팅된다는 것을 의미합니다. 이러한 정책을 통해 클러스터의 호스트에 걸쳐 고가용성 가상 머신의 균형을 유지할 수 있습니다. 클러스터의 호스트가 실패하는 경우 남은 호스트는 클러스터 성능에 영향을 주지 않고 고가용성 가상 머신의 마이그레이션 로드를 뒷받침할 수 있습니다. 고가용성 가상 머신 예약 활성화 시 **Manager**는 기존 호스트가 예기치 않게 실패하는 경우 고가용성 가상 머신을 마이그레이션할 수 있는 적절한 용량이 클러스터 내에 있는지 확인합니다.

## 5.8. 스케줄링

Red Hat Virtualization에서 스케줄링이란 Red Hat Virtualization Manager가 새로운 또는 마이그레이션된 가상 머신의 대상 호스트를 클러스터에서 선택하는 방식을 의미합니다.

호스트가 가상 머신을 시작하거나 다른 호스트에서 마이그레이션된 가상 머신을 수용할 수 있으려면 이러한 가상 머신의 요구 사항을 뒷받침할 수 있을 만큼 충분한 가용 메모리 및 CPU가 있어야 합니다. 이러한 자격이 되는 대상 호스트가 여러 개 있는 경우 해당 클러스터의 로드 밸런싱 정책에 따라 하나의 호스트가 선택됩니다. 예를 들어 **Evenly\_Distributed** 정책이 설정된 경우 **Manager**는 CPU 사용량이 가장 낮은 호스트를 선택합니다. **Power\_Saving** 정책이 설정된 경우 최대 및 최소 서비스 수준 사이에 있으며 CPU 사용량이 가장 낮은 호스트가 선택됩니다. 또한 어떤 호스트가 **Storage Pool Manager (SPM)** 상태인 경우에도 가상 머신 시작 또는 마이그레이션 대상으로서의 자격에 영향을 미치게 됩니다. **SPM**이 아닌 호스트가 선호 대상 호스트입니다. 예를 들어 클러스터에서 첫 가상 머신 시작 시 해당 클러스터의 호스트가 **SPM** 역할을 하는 경우 가상 머신은 이러한 **SPM** 호스트에서 실행되지 않습니다.

## 5.9. 마이그레이션

Red Hat Virtualization Manager는 마이그레이션을 사용하여 클러스터의 로드 밸런싱 정책을 강적용합니다. 클러스터의 로드 밸런싱 정책 및 클러스터 내의 호스트에 대한 현재 수요에 따라 가상 머신 마이그레이션이 진행됩니다. 또한 호스트가 펜싱되거나 유지관리 모드로 변경 시 마이그레이션이 자동으로 발생하도록 설정될 수도 있습니다. Red Hat Virtualization Manager는 CPU 사용량이 가장 낮은 가상 머신을 먼저 마이그레이션합니다. 이러한 사용량은 퍼센트 값으로 계산되며 CPU 사용량에 영향을 미치는 I/O 작업 외에는 RAM 사용량 또는 I/O 작업을 고려하지 않습니다. CPU 사용량이 같은 가상 머신이 두 개 이상 있는 경우 가상 머신 CPU 사용량을 확인하기 위해 Red Hat Virtualization Manager가 실행한 데이터베이스 쿼리에 대하여 수신한 첫 가상 머신이 먼저 마이그레이션됩니다.

가상 머신 마이그레이션은 기본적으로 다음과 같은 제한 사항이 있습니다:

- 52 MiBps(초당 메가바이트)의 대역폭 제한이 각 가상 머신 마이그레이션에 적용됩니다.
- 가상 머신 메모리 1 GB당 64초 초과 시 마이그레이션이 종료됩니다.
- 프로세스가 240초 이상 중지된 경우 마이그레이션 작업이 중지됩니다.
- 동시에 마이그레이션을 보내는 경우 각 호스트의 CPU 코어마다 한 개 또는 2 중에서 더 적은 숫자로 개수가 제한됩니다.

마이그레이션 설정을 튜닝하는 방법은 <https://access.redhat.com/solutions/744423>에서 참조하시기 바랍니다.

## 6장. 디렉토리 서비스

### 6.1. 디렉토리 서비스

Red Hat Virtualization 플랫폼은 디렉토리 서비스를 사용하여 사용자를 인증 및 권한 부여합니다. 사용자 포털, 관리 포털, 그리고 REST API 등과 같은 모든 Manager 인터페이스와의 통신은 인증 및 권한 부여된 사용자에게만 허용됩니다. Red Hat Virtualization 환경 내의 가상 머신은 같은 디렉토리 서비스를 사용하여 인증 및 권한 부여를 할 수 있지만 그렇게 하도록 설정된 경우에만 가능합니다. 현재 Red Hat Virtualization Manager와 함께 사용하도록 지원되는 디렉토리 서비스 공급자는 Identity Management (IdM), Red Hat Directory Server 9(RHDS), Active Directory (AD), 그리고 OpenLDAP입니다. Red Hat Virtualization Manager는 다음과 같은 작업을 위해 디렉토리 서버와 통신합니다:

- 포털 로그인 (사용자, 고급 사용자, 관리자, REST API)
- 사용자 정보 표시 쿼리
- 도메인에 Manager 추가

인증이란 데이터를 생성한 파티(party)와 해당 파티가 생성한 데이터의 무결성을 검증 및 확인하는 것을 말합니다. 주체(principal)란 신원이 검증된 파티를 말합니다. 검증자(verifier)란 주체의 신원에 대한 보증을 요구하는 파티를 말합니다. Red Hat Virtualization의 경우 Manager가 검증자이며 사용자가 주체입니다. 데이터의 무결성이란 수신된 데이터가 주체에 의해 생성된 데이터와 일치한다는 보증을 말합니다.

인증에는 기밀성과 권한 부여가 긴밀히 연관되어 있습니다. 기밀성은 데이터가 이를 수신하도록 의도되지 않은 사람에게 공개되는 것으로부터 보호합니다. 옵션으로 강력한 인증 방법을 통해 기밀성이 제공됩니다. 권한 부여는 주체가 어떤 작업을 수행하도록 허용되었는지 확인합니다. Red Hat Virtualization에서는 디렉토리 서비스를 사용하여 사용자와 역할을 연결하고 이에 따라 권한을 부여합니다. 권한 부여는 일반적으로 주체 인증 후 수행되며 검증자의 로컬 또는 원격 정보를 바탕으로 할 수 있습니다.

설치 도중 Red Hat Virtualization 환경 관리를 위해 로컬 내부 도메인이 자동으로 설정됩니다. 설치 완료 후 도메인을 더 추가할 수 있습니다.

### 6.2. 로컬 인증: 내부 도메인

Red Hat Virtualization Manager는 설치 중 제한된 내부 관리 도메인을 생성합니다. 이러한 도메인은 디렉토리 서버의 디렉토리 서비스 사용자로서가 아니라 Red Hat Virtualization PostgreSQL 데이터베이스에 있는 키(key)를 바탕으로 하므로 AD 또는 IdM 도메인과 다릅니다. 또한 내부 도메인은 단 하나의 사용자(admin@internal)만 있기 때문에 외부 도메인과는 차이가 있습니다. 초기 인증 시 이러한 방식을 사용하면 완료되었으며 작동 중인 디렉토리 서버가 필요 없이 Red Hat Virtualization을 확인할 수 있으며 외부 디렉토리 서비스에 발생하는 모든 문제를 해결할 수 있는 관리 계정이 사용 가능하도록 할 수 있습니다.

admin@internal 사용자는 환경의 초기 설정을 위한 것입니다. 여기에 포함된 작업은 호스트 설치 및 허용, 외부 AD 또는 IdM 인증 도메인 추가, 그리고 외부 도메인의 사용자에게 권한 위임 등이 있습니다.

### 6.3. GSSAPI를 사용한 원격 인증

Red Hat Virtualization의 컨텍스트에서 원격 인증이란 Red Hat Virtualization Manager에서 원격으로 처리하는 인증 작업을 말합니다. 원격 인증은 AD, IdM, 또는 RHDS 도메인 내에서 오는 사용자 또는 API 연결에 대하여 사용됩니다. Red Hat Virtualization Manager는 반드시 관리자가 engine-manage-domains 도구를 사용하여 RHDS, AD, 또는 IdM 도메인에 소속되도록 설정해야 합니다. 이를 위해 Manager는 RHDS, AD, 또는 IdM 디렉토리 서버로부터 도메인에 시스템을 연결할 수 있을 만큼 충분한 권한을 가진 계정에 대한 인증 정보를 받아야 합니다. 도메인 추가 후 도메인 사용자는 암호를 사용하여 Red Hat Virtualization Manager에 의해 디렉토리 서버에 대한 인증을 받습니다. Manager는 Simple Authentication

and Security Layer(SASL)라는 프레임워크를 사용하며 이러한 프레임워크는 또한 *Generic Security Services Application Program Interface(GSSAPI)*를 사용하여 사용자의 신원을 안전하게 검증하고 해당 사용자의 권한 부여 수준을 파악할 수 있습니다.

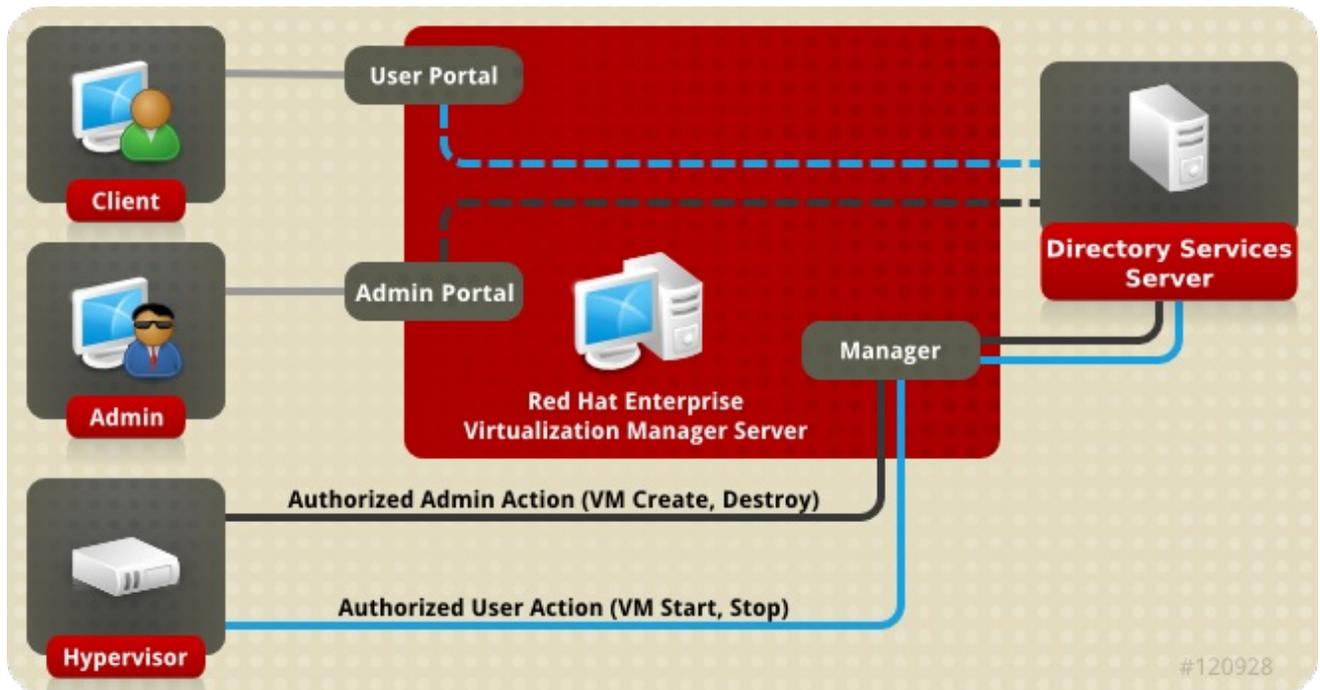


그림 6.1. GSSAPI 인증

## 7장. 템플릿 및 풀

### 7.1. 템플릿 및 풀

**Red Hat Virtualization** 환경에서 관리자에게 제공하는 도구인 **템플릿 및 풀**을 사용하여 사용자에게 가상 머신을 간단하게 제공할 수 있습니다. 템플릿은 관리자가 운영 체제 설치 및 설정을 우회하여 기존 사전 설정 가상 머신을 기반으로 새 가상 머신을 생성할 수 있게 해줍니다. 이는 예를 들어 웹 서버 가상 머신과 같이 어플라이언스처럼 사용될 가상 머신의 경우에 특히 유용합니다. 어떤 조직에서 특정 웹 서버의 여러 인터페이스를 사용하는 경우 관리자는 템플릿으로 사용될 가상 머신을 생성하여 운영 체제, 웹 서버, 및 지원 패키지를 설치하고 고유의 설정 변경 내용을 적용할 수 있습니다. 그 후 관리자는 작동 중인 가상 머신을 바탕으로 템플릿을 생성하여 필요에 따라 동일한 새 가상 머신을 생성하는 데 사용할 수 있습니다.

가상 머신 풀이란 특정 템플릿을 기반으로 하는 가상 머신 그룹이며 이를 사용자에게 빠르게 제공할 수 있게 해줍니다. 풀에 있는 가상 머신에 대한 사용 권한은 풀 수준에서 부여되며 풀을 사용하도록 권한이 부여된 사용자는 해당 풀의 아무 가상 머신이나 배정받게 됩니다. 가상 머신 풀에는 해당 풀에 있는 가상 머신의 일시적인 성격이 내재되어 있습니다. 사용자가 과거에 해당 풀의 어느 가상 머신을 사용했는지와 상관 없이 가상 머신을 배정받기 때문에 가상 머신 풀은 데이터 지속성을 필요로 하는 목적에는 적합하지 않습니다. 가상 머신 풀은 사용자 데이터가 한 곳에 저장되어 있고 가상 머신이 그러한 데이터에 액세스하여 사용하는 수단인 경우 또는 데이터 지속성이 중요하지 않은 두 가지 경우에 가장 적합합니다. 풀 생성 시 가상 머신이 생성되어 중지 상태에서 해당 풀에 배치됩니다. 그 후 이러한 가상 머신은 사용자 요청 시 시작됩니다.

### 7.2. 템플릿

템플릿을 생성하기 위해 관리자는 가상 머신을 생성 및 사용자 정의합니다. 원하는 패키지 설치, 사용자 정의 설정 적용, 그리고 가상 머신을 용도에 맞게 준비하여 배포 후 변경의 필요성을 최소화합니다. 옵션으로 가상 머신에서 템플릿을 생성하기 전에 **일반화**를 하는 것이 좋습니다. 일반화를 사용하여 배포 후 변경되는 시스템 사용자 이름, 암호, 그리고 시간대 정보 등의 상세 정보를 삭제합니다. 일반화는 사용자 정의 설정에는 영향을 미치지 않습니다. **Red Hat Virtualization** 환경에서 **Windows** 및 **Linux** 게스트의 일반화에 대한 보다 자세한 내용은 **가상 머신 관리 가이드**의 **템플릿**에서 참조하시기 바랍니다. **Red Hat Enterprise Linux** 게스트는 **sys-unconfig**를 사용하여 일반화합니다. **Windows** 게스트는 **sys-prep**를 사용하여 일반화합니다.

템플릿의 기반을 제공하는 가상 머신이 성공적으로 설정되고 필요시 일반화된 후 중지된 경우 관리자는 해당 가상 머신에서 템플릿을 생성할 수 있습니다. 가상 머신에서 템플릿 생성 시 특별히 설정된 가상 디스크 이미지의 읽기 전용 사본이 생성됩니다. 이러한 읽기 전용 이미지는 해당 템플릿을 기반으로 생성되는 모든 가상 머신에 대한 백업 이미지가 됩니다. 다시 말해 템플릿은 기본적으로 가상 하드웨어 설정이 연결된 사용자 정의 읽기 전용 디스크 이미지와도 같습니다. 예를 들어 1기가 **RAM**의 템플릿에서 생성된 가상 머신에 2기가 **RAM**을 제공하는 것과 같이 템플릿에서 생성된 가상 머신의 하드웨어를 변경할 수 있습니다. 하지만 템플릿 디스크 이미지 변경 시 해당 템플릿을 기반으로 하는 모든 가상 머신이 변경되므로 템플릿 디스크 이미지는 변경할 수 없습니다.

템플릿 생성 시 이러한 템플릿을 여러 가상 머신의 기반으로 사용할 수 있습니다. 가상 머신은 **썸(Thin)** 프로비저닝 방식 또는 **복제(Clone)** 프로비저닝 방식을 사용하여 특정 템플릿을 기반으로 생성됩니다. 템플릿에서 복제된 가상 머신은 템플릿 기본 이미지의 쓰기 가능한 완전한 사본을 가져가며, 그 결과 썸 프로비저닝 방식으로 절약된 공간을 희생하는 대신 더 이상 템플릿에 의존하지 않게 됩니다. 썸 프로비저닝을 사용하여 템플릿에서 생성된 가상 머신은 템플릿의 읽기 전용 이미지를 기본 이미지로 사용하며, 그 결과 이러한 이미지를 기반으로 생성된 템플릿과 모든 가상 머신이 반드시 같은 스토리지 도메인에 저장되어야 합니다. 데이터 변경 내용 및 새로 생성된 데이터는 **Copy On Write** 이미지에 저장됩니다. 템플릿 기반의 가상 머신은 각각 같은 기본 읽기 전용 이미지를 사용하며 해당 가상 머신 고유의 **Copy On Write** 이미지도 사용합니다. 이를 통해 동일한 데이터가 스토리지에 저장되는 횟수를 제한하여 스토리지 공간을 절약할 수 있습니다. 또한 읽기 전용 백업 이미지의 잦은 사용은 액세스되는 데이터가 캐시(cached)되게 할 수 있으며 그 결과 순 성능이 증가하게 됩니다.

## 7.3. 풀

가상 머신 풀은 다수의 동일한 가상 머신을 사용자에게 데스크탑으로 빠르게 제공할 수 있게 해줍니다. 풀에서 가상 머신에 대한 액세스 및 사용 권한을 부여받은 사용자는 요청 큐(queue)에서의 위치에 따라 사용할 가능한 가상 머신을 받게 됩니다. 풀의 가상 머신은 데이터 지속성을 허용하지 않으므로 풀에서 가상 머신이 배정될 때마다 기본 상태로 할당됩니다. 이는 사용자 데이터가 중앙 저장소에 있는 경우에 가장 사용하기 적합합니다.

가상 머신 풀이 템플릿에서 생성됩니다. 풀의 각 가상 머신은 같은 읽기 전용 백업 이미지를 사용하며 임시 **Copy On Write** 이미지를 사용하여 변경 및 새로 생성된 데이터를 저장합니다. 풀의 가상 머신은 사용자가 생성하고 변경된 데이터를 저장하는 **Copy On Write** 데이터가 종료 시 상실된다는 점에서 다른 가상 머신과 다릅니다. 이것은 곧 가상 머신 풀은 해당 풀을 백업하는 템플릿보다 큰 스토리지를 필요로 하지 않으며 이에 더해 사용 중 생성 및 변경된 데이터를 위해 약간의 공간만 더 필요로 한다는 것을 의미합니다. 가상 머신 풀은 각 사용자에게 전용 가상 데스크탑 제공 시 필요한 스토리지가 필요 없이 일부 작업을 위해 사용자에게 컴퓨팅 능력을 효율적으로 제공합니다.

### 예 7.1. 풀 사용 예

기술 지원 회사가 10명의 헬프 데스크 직원을 고용합니다. 하지만 5명만 동시에 일을 합니다. 각 헬프 데스크 직원에게 가상 머신을 하나씩 주기 위해 총 10개의 가상 머신을 생성하지 않고 5개의 가상 머신으로 이루어진 풀이 생성됩니다. 헬프 데스크 직원은 근무 시작 시 가상 머신을 할당받고 근무 종료 시 가상 머신을 풀에 반환합니다.

## 8장. 가상 머신 스냅샷

### 8.1. 스냅샷

스냅샷은 관리자가 특정 시점에서 가상 머신의 운영 체제, 애플리케이션, 그리고 데이터의 복원 지점을 생성할 수 있게 해주는 스토리지 기능입니다. 스냅샷은 현재 가상 머신 하드 디스크 이미지에 있는 데이터를 COW 볼륨으로 저장하여 스냅샷을 찍을 당시 존재했던 데이터를 복구할 수 있게 해줍니다. 스냅샷은 현재 COW층 위에 새 COW층이 생성되게 합니다. 스냅샷을 찍은 후에 수행된 모든 쓰기 작업은 새 COW층에 쓰기됩니다.

가상 머신 하드 디스크 이미지는 하나 이상의 볼륨의 연쇄라는 것을 이해해야 합니다. 가상 머신 측면에서 이러한 볼륨은 단일 디스크 이미지로 보입니다. 가상 머신은 이러한 디스크가 여러 볼륨으로 이루어져 있다는 사실을 알지 못합니다.

COW 볼륨 및 COW층이라는 표현은 교체 사용되지만 "층"이라는 표현이 스냅샷의 일시적인 속성을 더 잘 나타냅니다. 스냅샷을 찍은 후에 만족스럽게 변경되지 못한 데이터를 관리자가 삭제할 수 있습니다. 스냅샷은 각종 워드 프로세서에 있는 **실행 취소(Undo)** 기능과 비슷한 기능을 제공합니다.



#### 참고

**공유 가능**으로 표시된 가상 머신 하드 디스크 스냅샷과 **직접 LUN** 연결을 기반으로 하는 가상 머신 하드 디스크 스냅샷은 실시간 및 다른 모든 경우에 지원되지 않습니다.

세 가지 주요 스냅샷 작업은 다음과 같습니다:

- **생성:** 가상 머신에 대하여 생성된 첫 스냅샷과 관련됩니다.
- **미리 보기:** 스냅샷을 찍은 당시의 시스템 데이터를 복원하는지 여부를 확인하기 위해 스냅샷을 미리 보는 것입니다.
- **삭제:** 더 이상 필요하지 않은 복원 지점을 삭제하는 것입니다.

스냅샷 작업에 대한 보다 자세한 내용은 *Red Hat Virtualization 가상 머신 관리 가이드*에 있는 [스냅샷](#)에서 참조하십시오.

### 8.2. RED HAT VIRTUALIZATION에서 실시간 스냅샷

**공유 가능**으로 표시된 가상 머신 하드 디스크 스냅샷과 **직접 LUN** 연결을 기반으로 하는 가상 머신 하드 디스크 스냅샷은 라이브 및 다른 모든 경우에 지원되지 않습니다.

복제 또는 마이그레이션되고 있지 않은 다른 모든 가상 머신은 실행, 일시 정지, 또는 중지 중 스냅샷을 찍을 수 있습니다.

가상 머신의 라이브 스냅샷 시작 시 **Manager**는 **SPM** 호스트에 가상 머신이 사용할 새 볼륨을 생성하도록 요청합니다. 새 볼륨이 준비되면 **Manager**가 **VDSM**을 사용하여 가상 머신을 실행하는 호스트의 **libvirt** 및 **qemu**와 통신해서 가상 머신 쓰기 작업에 새 볼륨을 사용하도록 해당 호스트에 전달합니다. 가상 머신이 새 볼륨에 쓰기 작업을 할 수 있는 경우 스냅샷 동작이 성공한 것이며 가상 머신은 기존 볼륨에 대한 쓰기 작업을 중지합니다. 가상 머신이 새 볼륨에 쓰기 작업을 할 수 없는 경우 스냅샷 동작이 실패한 것이며 새 볼륨이 삭제됩니다.

가상 머신은 라이브 스냅샷 시작부터 새 볼륨이 준비될 때까지 기존 볼륨과 새 볼륨에 액세스할 수 있어야 하므로 두 볼륨은 모두 읽기-쓰기가 가능하도록 열립니다.

가상 머신에 정지(quieting)를 지원하는 게스트 에이전트가 설치된 경우 모든 스냅샷에 걸쳐 filesystem 지속성이 유지될 수 있습니다. 등록된 Red Hat Enterprise Linux 게스트는 **qemu-guest-agent**를 설치하여 스냅샷 전에 정지(quieting)를 활성화할 수 있습니다.

스냅샷을 찍을 당시 가상 머신에 정지(quieting) 호환 게스트 에이전트가 존재하는 경우 VDSM은 libvirt를 사용하여 에이전트와 통신해서 스냅샷을 준비하도록 합니다. 진행 중인 쓰기 작업 완료 후 스냅샷을 찍기 전에 filesystems이 고정됩니다. 스냅샷이 완료되고 libvirt가 가상 머신의 디스크 쓰기 작업을 새 볼륨으로 변경한 후 filesystem 고정이 해제되며 디스크 쓰기 작업이 재개됩니다.

모든 라이브 스냅샷은 정지(quieting)를 활성화한 상태에서 시도하게 됩니다. 호환 게스트 에이전트가 없어서 스냅샷 명령이 실패하는 경우 정지 플래그 없이 라이브 스냅샷을 다시 시작합니다. 가상 머신을 파일 시스템이 정지된 스냅샷 전 상태로 되돌리는 경우 파일 시스템을 확인할 필요 없이 올바르게 부팅합니다. 파일 시스템이 정지되지 않은 이전 스냅샷으로 되돌리는 경우 부팅 시 파일 시스템 확인을 해야 합니다.

### 8.3. 스냅샷 생성

Red Hat Virtualization에서 가상 머신의 최초 스냅샷은 QCOW2 또는 RAW 포맷을 유지한다는 점에서 이후의 스냅샷과 다릅니다. 가상 머신의 최초 스냅샷은 기존 볼륨을 기본 이미지로 지정합니다. 추가 스냅샷은 이전 스냅샷 이후에 이미지에 저장된 데이터의 변경 내용을 추적하는 추가적인 COW층입니다.

Red Hat Virtualization에서 게스트 가상 머신은 디스크 이미지가 썸 프로비저닝되었거나 사용자가 QCOW2 이미지를 특별히 요청한 경우를 제외하면 일반적으로 RAW 디스크 이미지와 통신합니다. [그림 8.1. “최초 스냅샷 생성”](#)에 묘사된 것과 같이 스냅샷 생성 시 가상 디스크 이미지를 구성하는 볼륨이 이후의 모든 스냅샷에 대한 기본 이미지로 사용되게 됩니다.

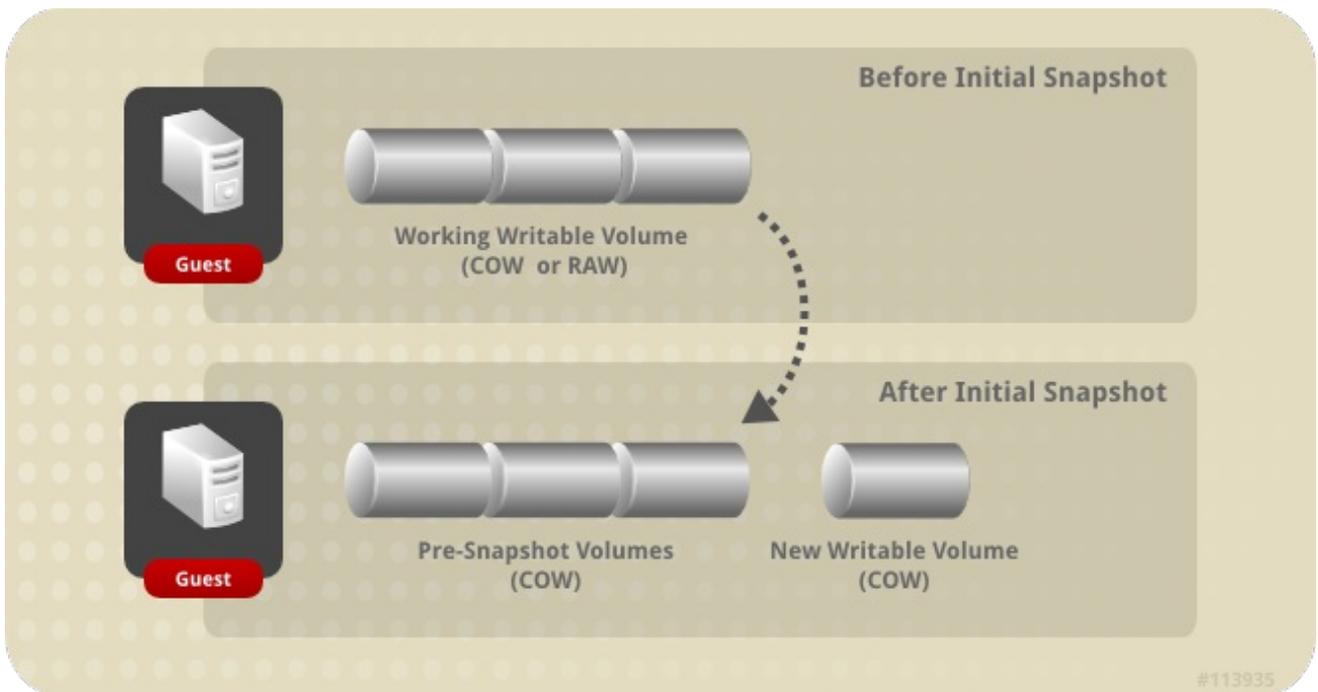


그림 8.1. 최초 스냅샷 생성

최초 스냅샷 이후에 스냅샷을 찍는 경우 새 COW 볼륨이 생성되며 스냅샷을 찍은 후 생성 또는 변경된 데이터가 이러한 볼륨에 저장됩니다. 새 COW층은 각각 COW 메타데이터만 저장하게 됩니다. 스냅샷을 찍은 후 가상 머신 사용 및 작업을 통해 생성된 데이터는 새 COW층에 쓰기됩니다. 이전 COW층에 있는 데이터를 수정하는 데 가상 머신을 사용하는 경우 이전 층에서 데이터를 읽고 새로운 층에 쓰기합니다. 가상 머신은 각 COW를 최신 항목순으로 가상 머신에 대하여 투명하게 확인하여 데이터를 찾습니다.

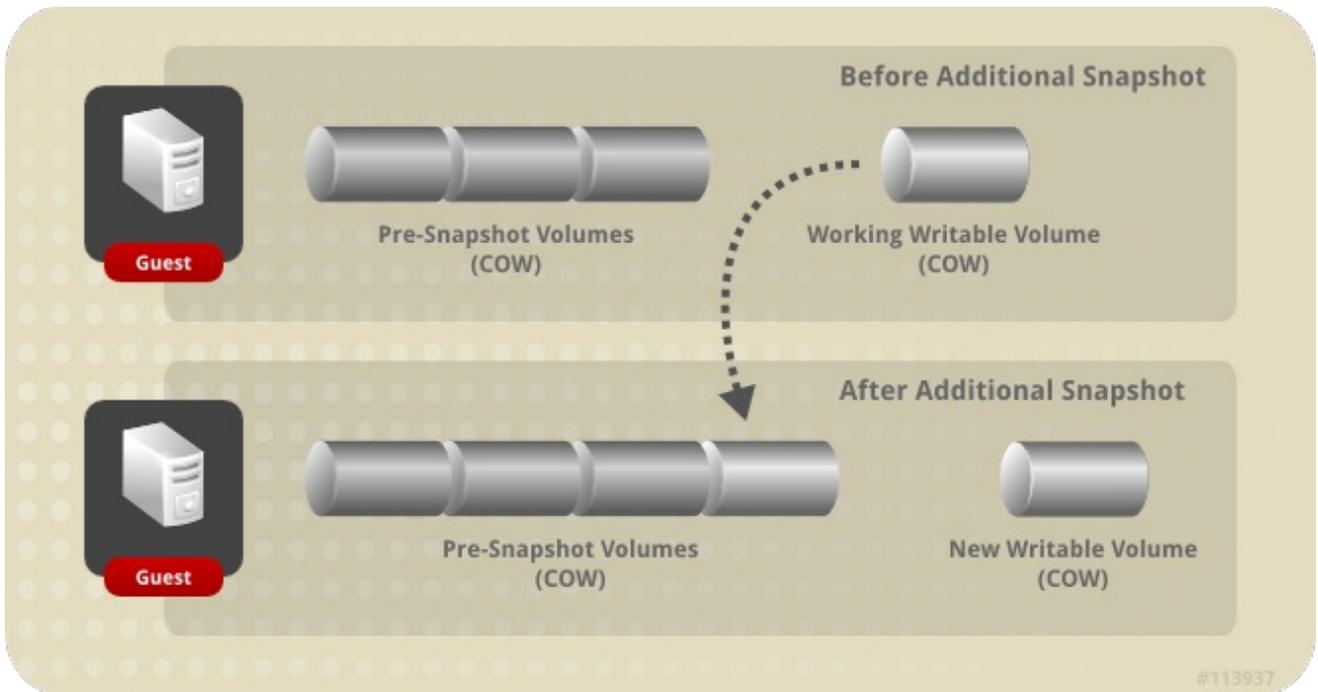


그림 8.2. 추가 스냅샷 생성

## 8.4. 스냅샷 미리 보기

가상 디스크 이미지를 어느 스냅샷으로 되돌리는지 선택하기 위해 관리자는 이전에 생성된 스냅샷을 모두 미리 볼 수 있습니다.

관리자는 게스트별로 사용 가능한 스냅샷 중에서 하나의 스냅샷 볼륨을 선택하여 콘텐츠를 미리 볼 수 있습니다. [그림 8.3. “미리 보기 스냅샷”](#)에 묘사된 것과 같이 각 스냅샷은 COW 볼륨으로 저장되며 미리 보기 시 이렇게 미리 보기되는 스냅샷에서 새 미리 보기층이 복사됩니다. 게스트는 실제 스냅샷 볼륨 대신 미리 보기층과 통신합니다.

관리자가 선택된 스냅샷을 미리 본 후 해당 미리 보기는 스냅샷에 캡처된 상태로 게스트 데이터를 복원하도록 커밋될 수 있습니다. 관리자가 해당 미리 보기를 커밋하는 경우 게스트가 이전 층에 연결됩니다.

스냅샷 미리 보기 후 관리자는 **실행 취소(Undo)**를 선택하여 미리 보기된 스냅샷의 이전 층을 삭제할 수 있습니다. 이전 층이 삭제되더라도 스냅샷 자체를 포함하는 층은 남아 있습니다.

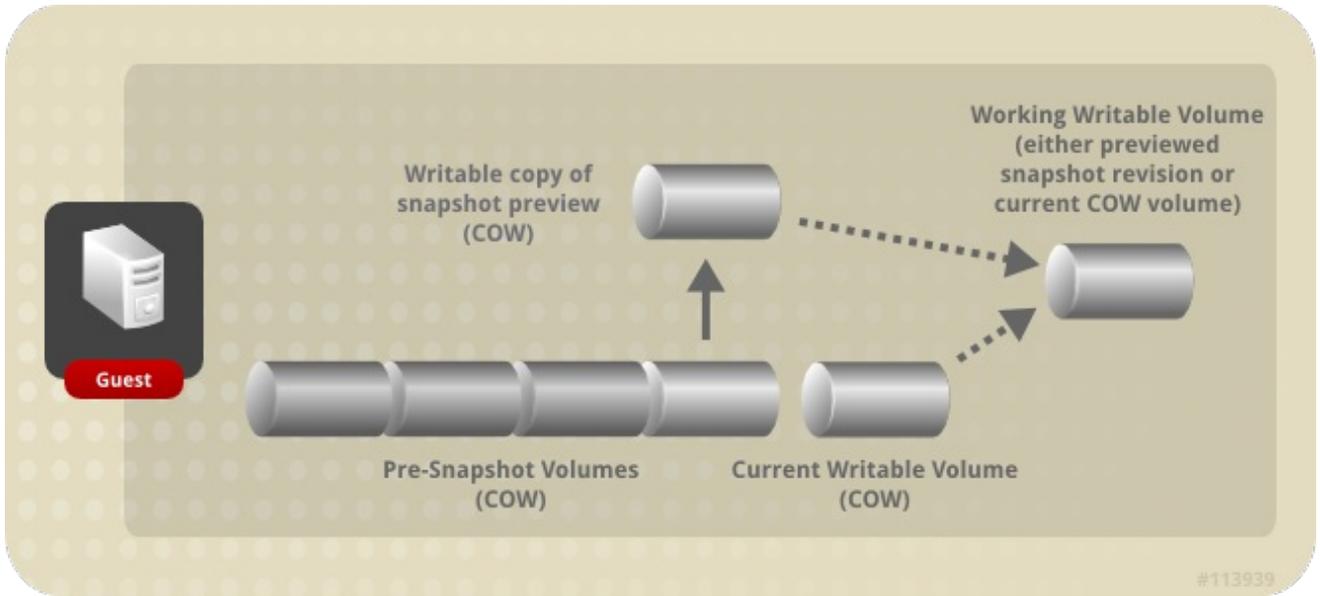


그림 8.3. 미리 보기 스냅샷

### 8.5. 스냅샷 삭제

더 이상 필요하지 않은 개별 스냅샷을 삭제할 수 있습니다. 스냅샷 삭제 시 해당 특정 복원 지점으로 가상 디스크 이미지를 복원할 수 있는 능력이 제거됩니다. 이러한 작업이 스냅샷이 사용한 디스크 공간을 다시 확보하거나 데이터를 삭제하는 것은 아닙니다. 이후의 스냅샷이 삭제된 스냅샷의 데이터에 덮어쓰기를 한 경우에만 디스크 공간이 다시 확보됩니다. 예를 들어 다섯 개의 스냅샷 중 세 번째 스냅샷이 삭제되는 경우 이러한 세 번째 스냅샷에서 변경되지 않은 데이터가 디스크에 남아 있어야 네 번째 및 다섯 번째 스냅샷을 사용할 수 있습니다. 하지만 네 번째 또는 다섯 번째 스냅샷이 세 번째 스냅샷의 데이터에 덮어쓰기를 한 경우 세 번째 스냅샷은 중복되게 되며 해당 디스크 공간이 다시 확보됩니다. 스냅샷을 삭제하는 것은 디스크 공간 확보 외에도 가상 머신 성능을 개선할 수 있습니다.

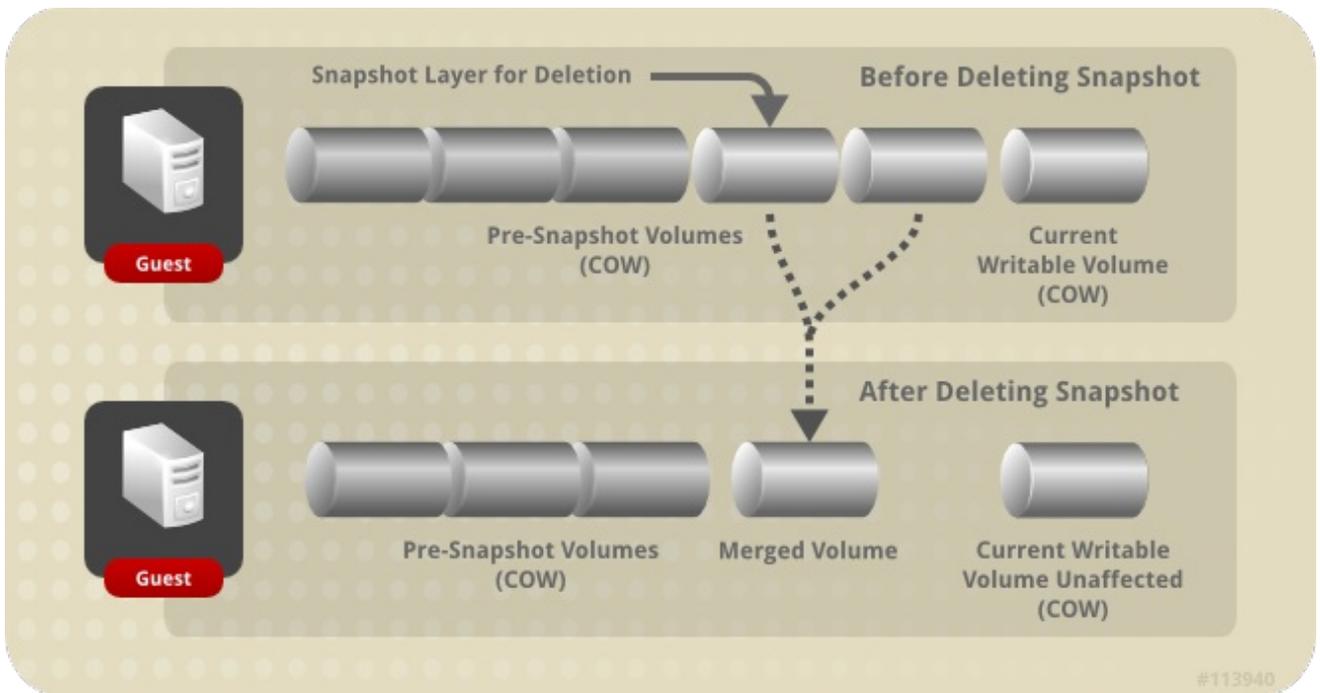


그림 8.4. 스냅샷 삭제

스냅샷 삭제는 비동기 블록 작업으로 다루게 됩니다. 이러한 경우 VDSM이 다시 시작하거나 가상 머신이 작업 중 종료되더라도 작업을 추적할 수 있도록 VDSM은 가상 머신의 복원 파일에 해당 작업 기록을 보관합니다. 작업 시작 후에는 이러한 작업이 실패하거나 중단되더라도 삭제되는 스냅샷을 미리 보거나 복원

지점으로 사용할 수 없습니다. 활성 레이어(layer)를 부모와 통합하는 작업은 활성 레이어에서 부모 레이어로 데이터를 복사하고 디스크 쓰기 내용이 활성 및 부모 레이어에 동일하게 작용되는 두 단계로 진행됩니다. 마지막으로 삭제되는 스냅샷의 데이터가 부모 스냅샷과 통합되고 VDSM이 이러한 변경 내용을 이미지 체인에 걸쳐 동기화하면 작업이 완료됩니다.

## 9장. 하드웨어 드라이버 및 장치

### 9.1. 가상화 하드웨어

Red Hat Virtualization에서는 가상화 게스트에게 세 가지 유형의 시스템 장치를 제공합니다. 이러한 하드웨어 장치는 모두 가상화 게스트에게는 물리적으로 연결된 하드웨어 장치로 보이지만 장치 드라이버는 작동 방식이 다릅니다.

#### 에뮬레이트된 장치

가상 장치라고도 부르는 에뮬레이트된 장치는 온전히 소프트웨어에서만 존재합니다. *에뮬레이트된 장치 드라이버*란 (소스 장치를 관리하는) 호스트에서 실행 중인 운영 체제와 게스트에서 실행 중인 운영 체제 간의 변환 레이어를 말합니다. 장치 수준에서 에뮬레이트된 장치와의 통신은 하이퍼바이저에서 가로채기 및 변환하게 됩니다. Linux 커널에서 인식하는 에뮬레이트 장치와 같은 유형의 모든 장치는 에뮬레이트된 드라이버의 백업 소스 장치로 사용 가능합니다.

#### 반가상화 장치

반가상화 장치를 사용하려면 게스트 운영 체제에 장치 드라이버를 설치해서 호스트 시스템의 하이퍼바이저와 통신할 수 있는 인터페이스를 제공해 줘야 합니다. 이러한 인터페이스를 사용하여 디스크 I/O 등 일반적으로 집약적인 작업을 가상 환경 밖에서 수행할 수 있게 해줍니다. 이러한 방법으로 가상화 환경 고유의 오버헤드를 감소하여 게스트 운영 체제를 물리적인 하드웨어에 직접 실행했을 때와 비슷한 성능을 보일 수 있습니다.

#### 물리적으로 공유된 장치

특정 하드웨어 플랫폼은 가상화 게스트가 각종 하드웨어 장치 및 구성 요소에 직접 액세스할 수 있게 해줍니다. 가상화에서 이러한 과정을 *통과(passthrough)* 또는 *장치 지정*이라고 합니다. 통과를 사용하면 장치가 게스트 운영 체제에 물리적으로 연결된 것처럼 보이고 작동하게 됩니다.

### 9.2. RED HAT VIRTUALIZATION에서 변경할 수 없는(STABLE) 장치 주소

가상 하드웨어 PCI 주소 지정은 `ovirt-engine` 데이터베이스에서 지속됩니다.

PCI 주소는 가상 머신 생성 중 `QEMU`에 의해 지정되며 `libvirt`에 의해 `VDSM`에 보고됩니다. `VDSM`가 다시 이를 `Manager`에 보고하면 여기에 있는 `ovirt-engine` 데이터베이스에 저장됩니다.

가상 머신 시작 시 `Manager`는 데이터베이스에서 장치 주소를 `VDSM`으로 보냅니다. `VDSM`가 이를 `libvirt`에 보내면 가상 머신 최초 실행 시 지정된 PCI 장치 주소를 사용하여 가상 머신이 시작됩니다.

가상 머신에서 장치를 제거하는 경우 변경할 수 없는 (stable) PCI 주소를 포함하여 해당 장치와 관련된 모든 참조도 삭제됩니다. 제거된 장치 대신에 다른 장치 추가 시 `QEMU`에서 이러한 장치에 PCI 주소를 배정하며 이 주소는 교체된 장치의 PCI 주소와 다를 가능성이 높습니다.

### 9.3. CPU (CENTRAL PROCESSING UNIT)

클러스터 내의 각 호스트에 여러 가상 CPU(vCPU)가 있습니다. 가상 CPU는 호스트에 실행 중인 게스트에 표시됩니다. 클러스터 내에서 호스트에 의해 표시된 모든 가상 CPU 유형은 Red Hat Virtualization `Manager`에서 클러스터 최초 생성 당시 선택됩니다. 클러스터 내에 여러 가상 CPU 유형이 존재할 수 없습니다.

사용 가능한 가상 CPU 유형은 같은 이름을 가진 물리적 CPU의 속성을 기반으로 합니다. 가상 CPU는 게스트 운영 체제의 물리적 CPU와 구분할 수 없습니다.



## 참고

x2APIC에 대한 지원

Red Hat Enterprise Linux 7 호스트에서 제공하는 모든 가상 CPU 모델은 x2APIC에 대한 지원을 포함합니다. 이를 통해 하드웨어 인터럽트(interrupt)를 더 잘 처리할 수 있는 *Advanced Programmable Interrupt Controller (APIC)*가 제공됩니다.

## 9.4. 시스템 장치

시스템 장치는 게스트를 실행하는 데 중요하며 삭제할 수 없습니다. 또한 게스트에 연결된 각 시스템 장치는 사용 가능한 PCI 슬롯을 차지하게 됩니다. 기본 시스템 장치는 다음과 같습니다:

- 호스트 브리지
- ISA 브리지 및 USB 브리지 (USB 브리지 및 ISA 브리지는 같은 장치입니다)
- (Cirrus 또는 qxl 드라이버를 사용하는) 그래픽 카드
- 메모리 별론 장치

## 9.5. 네트워크 장치

Red Hat Virtualization은 세 가지 유형의 NIC를 게스트에 표시할 수 있습니다. 게스트에 표시하는 NIC 유형은 게스트 생성 시 선택하지만 Red Hat Virtualization Manager에서 변경할 수 있습니다.

- **e1000** NIC는 가상화 Intel PRO/1000 (e1000)을 게스트에 표시합니다.
- **virtio** NIC는 반가상화 네트워크 장치를 게스트에 표시합니다.
- **rtl8139** NIC는 가상화 Realtek Semiconductor Corp RTL8139를 게스트에 표시합니다.

한 게스트에 여러 NIC가 허용됩니다. 추가되는 각 NIC는 게스트에서 사용 가능한 PCI 슬롯을 차지합니다.

## 9.6. 그래픽 장치

두 개의 에뮬레이트된 그래픽 장치가 제공됩니다. SPICE 프로토콜 또는 VNC를 사용하여 이러한 장치에 연결할 수 있습니다.

- **ac97**은 Cirrus CLGD 5446 PCI VGA 카드를 에뮬레이트합니다.
- **vga** 는 (모든 비표준 모드를 포함하여 하드웨어 수준에서) BochsVESA 확장으로 더미(dummy) VGA 카드를 에뮬레이트합니다.

## 9.7. 스토리지 장치

스토리지 장치 및 스토리지 풀에서는 블록 장치 드라이버를 사용하여 스토리지 장치를 가상화 게스트에 연결할 수 있습니다. 스토리지 드라이버는 스토리지 장치가 아님에 유의합니다. 드라이버를 사용하여 백업 스토리지 장치, 파일 또는 스토리지 풀 볼륨을 가상화 게스트에 연결할 수 있습니다. 지원되는 모든 스토리지 장치, 파일, 또는 스토리지 풀 볼륨이 백업 스토리지 장치가 될 수 있습니다.

- **IDE** 드라이버는 에뮬레이트된 블록 장치를 게스트에 표시합니다. 에뮬레이트된 **IDE** 드라이버를 사용하여 각 가상화 게스트에 가상화 **IDE** 하드 디스크 또는 가상화 **IDE CD-ROM** 장치를 조합하여 최대 4대 연결할 수 있습니다. 또한 에뮬레이트된 **IDE** 드라이버를 사용하여 가상화된 DVD-

ROM 드라이버를 제공할 수 있습니다.

- **VirtIO** 드라이버는 반가상화 블록 장치를 게스트에 표시합니다. 반가상화 블록 드라이버는 가상화 게스트에 연결된 하이퍼바이저가 지원하는 모든 스토리지 장치에 대한 드라이버입니다. (에뮬레이션이 필요한 플로피 디스크 장치 제외)

## 9.8. 사운드 장치

두 개의 에뮬레이트된 사운드 장치가 있습니다:

- **ac97** 은 **Intel 82801AA AC97 Audio** 호환 사운드 카드를 에뮬레이트합니다.
- **es1370** 은 **ENSONIQ AudioPCI ES1370** 사운드 카드를 에뮬레이트합니다.

## 9.9. 시리얼 드라이버

반가상화 시리얼 드라이버(**virtio-serial**)는 바이트 스트림 지향의 문자 스트림 드라이버입니다. 반가상화 시리얼 드라이버는 네트워킹을 사용할 수 없는 경우 호스트의 사용자 공간과 게스트의 사용자 공간 사이의 간단한 통신 인터페이스를 제공합니다.

## 9.10. 벌룬 드라이버

벌룬 드라이버는 게스트가 필요한 메모리 용량을 하이퍼바이저에 알릴 수 있게 해줍니다. 호스트는 벌룬 드라이버를 사용하여 게스트에 메모리를 효율적으로 할당할 수 있으며 여유 메모리를 다른 게스트 및 프로세스에 할당할 수 있습니다.

게스트는 벌룬 드라이버를 사용하여 게스트에서 사용 중이지 않은 **RAM** 섹션을 표시할 수 있습니다 (**balloon inflation**). 하이퍼바이저는 이러한 가용 메모리를 다른 호스트 프로세스 또는 호스트의 다른 게스트에 사용할 수 있습니다. 게스트가 이러한 메모리를 다시 필요로 하는 경우 하이퍼바이저는 게스트에 **RAM**을 다시 할당하게 됩니다(**balloon deflation**).

## 10장. 최소 요구 사항 및 기술 제한 사항

### 10.1. 최소 요구 사항 및 지원 제한 사항

Red Hat Virtualization에 다양한 물리적 및 논리적 제한 사항이 있습니다. 이러한 제한 사항을 벗어나는 설정을 가진 환경은 현재 지원되지 않습니다.

### 10.2. 데이터 센터 제한 사항

관리되는 가상 환경에서 데이터 센터는 모든 리소스에 대한 가장 높은 수준의 컨테이너입니다. 각 데이터 센터에 저장될 수 있는 리소스에는 제한 사항이 있습니다.

표 10.1. 데이터 센터 제한 사항

| 항목         | 제한 사항   |
|------------|---|
| 스토리지 도메인 수 | <ul style="list-style-type: none"> <li>각 데이터 센터에 최소 2개의 스토리지 도메인이 권장됩니다. 하나의 데이터 스토리지 도메인이 필요하며 각 데이터 센터에 하나의 ISO 스토리지 도메인이 권장됩니다.</li> </ul> |
| 호스트 수      | <ul style="list-style-type: none"> <li>각 데이터 센터에서 최대 200개 호스트가 지원됩니다.</li> </ul>  |

### 10.3. 클러스터 제한 사항

클러스터는 일련의 가상 머신의 리소스 풀로 취급하는 물리적인 호스트의 모음입니다. 클러스터의 호스트는 같은 네트워크 인프라 및 스토리지를 공유합니다. 클러스터는 가상 머신이 한 호스트에서 다른 호스트로 이동할 수 있는 마이그레이션 도메인입니다. 안정성을 위해 각 클러스터에 여러 제한 사항이 있습니다.

- 관리되는 모든 하이퍼바이저는 클러스터에 있어야 합니다.
- 클러스터 내에서 관리되는 모든 하이퍼바이저는 같은 CPU 유형이어야 합니다. Intel 및 AMD CPU는 같은 클러스터 내에서 공존할 수 없습니다.



#### 참고

클러스터에 대한 보다 자세한 내용은 *관리 가이드*에 있는 [클러스터](#)에서 참조하시기 바랍니다.

### 10.4. 스토리지 도메인 제한 사항

스토리지 도메인은 가상 디스크 이미지 및 ISO 이미지를 보관할 수 있는 공간을 제공하며 가상 머신 가져오기 및 내보내기를 위한 공간도 제공합니다. 하나의 데이터 센터 내에 여러 스토리지 도메인을 생성할 수 있지만 각 스토리지 도메인에는 여러 제한 사항 및 권장 사항이 있습니다.

표 10.2. 스토리지 도메인 제한 사항

| 항목                        | 제한 사항   |
|---------------------------|---|
| 스토리지 유형                   | 지원되는 스토리지 유형은 다음과 같습니다: <ul style="list-style-type: none"> <li>● FCP (Fibre Channel Protocol)</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> Red Hat Virtualization 4.1에서 새 ISO 및 내보내기 스토리지 도메인은 모든 파일 기반 스토리지(NFS, Posix 또는 GlusterFS)에서 제공될 수 있습니다.  |
| Logical Unit Number (LUN) | iSCSI 또는 FCP에서 제공되는 각 스토리지 도메인에 허용되는 LUN은 최대 300 LUN입니다.  |
| 논리 볼륨 (LV)                | Red Hat Virtualization에서 논리 볼륨은 가상 머신, 템플릿, 그리고 가상 머신 스냅샷에 대한 가상 디스크를 표현합니다.<br><br>iSCSI 또는 FCP에서 제공하는 각 스토리지 도메인의 논리 볼륨 수는 350을 넘지 않는 것이 좋습니다. 어떤 스토리지 도메인의 논리 볼륨 수가 이 숫자를 넘는 경우 논리 볼륨 수가 350을 넘지 않는 다른 스토리지 도메인에 가용 스토리지를 나누는 것이 좋습니다.<br><br>이러한 제한 사항이 있는 근본적인 이유는 LVM 메타데이터 크기 때문입니다. 논리 볼륨의 수가 증가함에 따라 이러한 논리 볼륨과 연결된 LVM 메타데이터 역시 증가합니다. 이러한 메타데이터 크기가 1 MB를 초과 시 새 디스크 또는 스냅샷 생성과 같은 프로비저닝 작업과 QCOW 디스크 실행 시 논리 볼륨을 씬(thin) 프로비저닝하는 lvextend 작업 등이 더 오래 걸립니다.<br><br>논리 볼륨에 대한 보다 자세한 내용은 <a href="https://access.redhat.com/solutions/441203">https://access.redhat.com/solutions/441203</a> 에서 참조하시기 바랍니다. |



**참고**

스토리지 도메인에 대한 보다 자세한 내용은 *관리 가이드*에 있는 [스토리지](#)에서 참조하십시오.

**10.5. RED HAT VIRTUALIZATION MANAGER 제한 사항**

Red Hat Virtualization Manager 서버는 반드시 Red Hat Enterprise Linux 7을 실행해야 합니다. 이외의 추가적인 하드웨어 요구 사항이 반드시 충족되어야 합니다.

**표 10.3. Red Hat Virtualization Manager 제한 사항**

| 항목     | 제한 사항  |
|--------|--|
| RAM    | <ul style="list-style-type: none"> <li>• 최소 4 GB RAM이 있어야 합니다.</li> </ul>                    |
| PCI 장치 | <ul style="list-style-type: none"> <li>• 최소 1 Gbps 대역폭의 네트워크 컨트롤러가 적어도 한 개 필요합니다.</li> </ul> |
| 스토리지   | <ul style="list-style-type: none"> <li>• 사용 가능한 로컬 디스크 공간이 최소 25 GB 있어야 합니다.</li> </ul>      |



### 참고

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

## 10.6. 하이퍼바이저 요구 사항

Red Hat Virtualization Host(RHVM)에서 여러 가지 요구 사항 및 지원 제한 사항이 있습니다. Red Hat Enterprise Linux 호스트의 스토리지 요구 사항은 기존 설정으로 사용되고 있는 디스크 공간에 따라 다르지만 RHVM의 스토리지 요구 사항보다는 클 것으로 예상됩니다.

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

| 항목 | 지원 제한 사항 |
|----|----------|
|----|----------|

| 항목  | 지원 제한 사항   |
|-----|--|
| CPU | <p>최소한 하나의 물리적 CPU가 필요합니다. Red Hat Virtualization에서는 호스트에서 지원하는 CPU 모델은 다음과 같습니다:</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 Family</li> <li>• IBM POWER 8</li> </ul> <p>모든 CPUs에서는 Intel® 64 또는 AMD64 CPU 확장을 지원해야 하며 AMD-V™ 또는 Intel VT® 하드웨어 가상화 확장이 활성화되어 있어야 합니다. 또한 <b>No eXecute</b> 플래그(NX)도 지원해야 합니다.</p> |
| RAM | <p>각 가상 머신에서 필요한 RAM 크기는 다음과 같은 것에 따라 다릅니다:</p> <ul style="list-style-type: none"> <li>• 게스트 운영 체제 요구 사항</li> <li>• 게스트 애플리케이션 요구 사항</li> <li>• 가상 머신의 메모리 작업 및 사용</li> </ul> <p>또한 KVM은 물리적 RAM을 가상 머신에 대해 오버커밋할 수 있습니다. 이를 위해 필요 시에만 가상 머신에 RAM을 할당하고 사용률이 낮은 가상 머신을 스왑으로 이동합니다.</p> <p>지원되는 최대 및 최소 RAM에 대한 보다 자세한 내용은 <a href="https://access.redhat.com/articles/rhel-limits">https://access.redhat.com/articles/rhel-limits</a>에서 참조하시기 바랍니다.</p>  |

| 항목     | 지원 제한 사항   |
|--------|--|
| 스토리지   | <p>호스트에 대하여 지원되는 최소 내부 스토리지는 다음 목록의 항목을 모두 합친 값입니다:</p> <ul style="list-style-type: none"> <li>• 루트 (/) 파티션에 최소 6 GB의 스토리지가 필요합니다.</li> <li>• /boot 파티션에 최소 1 GB의 스토리지가 필요합니다.</li> <li>• /var 파티션에 최소 15 GB의 스토리지가 필요합니다. 셀프 호스트 엔진 배포 시 최소 60 GB의 스토리지가 필요합니다.</li> <li>• 스왑 파티션에 최소 8 MB의 스토리지가 필요합니다. 스왑 파티션 권장 용량은 호스트를 설치하는 시스템과 해당 환경에서 예상되는 오버커밋 수준에 따라 다릅니다. 보다 자세한 내용은 <a href="https://access.redhat.com/ko/solutions/744483">https://access.redhat.com/ko/solutions/744483</a>에서 참조하십시오.</li> </ul> <p>이것은 호스트 설치를 위한 최소 스토리지 요구 사항임에 유의합니다. 더 많은 스토리지 공간을 사용하는 기본 할당을 사용하는 것이 좋습니다.</p> |
| PCI 장치 | <p>최소 1 Gbps 권장 대역폭의 네트워크 컨트롤러가 적어도 한 개 필요합니다.</p>   |

## 중요

Red Hat Virtualization Host 부팅 시 다음과 같은 메시지가 표시될 수 있습니다:

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

이러한 경고 메시지는 가상화 확장이 사용자의 프로세서에서 비활성화되었거나 존재하지 않는다는 것을 의미합니다. CPU에서 목록에 있는 확장을 지원하며 시스템 BIOS에 활성화 되어 있는지 확인합니다.

다음 작업을 통해 사용자의 프로세서에 가상화 확장이 활성화되어 있는지 확인합니다:

- 호스트 부팅 화면에서 아무 키나 누른 후 목록에서 **Boot** 또는 **Boot with serial console** 항목을 선택합니다. **Tab**을 눌러서 선택된 옵션에 대한 커널 매개변수를 수정합니다. 목록의 마지막 커널 매개변수 뒤에 **Space**가 있어야 하므로 이를 확인하고 **rescue** 매개변수를 추가합니다.
- **Enter**를 눌러서 복구 모드로 부팅합니다.
- 표시되는 프롬프트에서 다음의 명령을 실행해서 사용자의 프로세서에 가상화 확장 기능이 있으며 활성화되어 있는지 확인합니다:

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

표시되는 내용이 있다면 프로세서에 하드웨어 가상화가 가능한 것입니다. 표시되는 내용이 없더라도 사용자의 프로세서가 하드웨어 가상화를 지원할 수도 있습니다. 일부의 경우 제조업체가 BIOS에서 가상화 확장을 비활성화하기도 합니다. 사용자가 이 경우에 해당된다고 생각된다면 해당 제조업체가 제공한 시스템 BIOS 및 마더보드 사용 설명서를 확인합니다.

- 추가 확인을 위해 **kvm** 모듈이 커널에 로딩되어 있는지 확인합니다:

```
# lsmod | grep kvm
```

표시되는 내용에 **kvm\_intel** 또는 **kvm\_amd**가 있다면 **kvm** 하드웨어 가상화 모듈이 로딩되어 있으며 사용자 시스템이 요구 사항을 충족한다는 것을 의미합니다.

## 10.7. 게스트 요구 사항 및 지원 제한 사항

게스트에 적용되는 요구 사항 및 제한 사항에 대한 보다 자세한 내용은 [Red Hat Enterprise Linux 기술 기능 및 제한](#)과 [Red Hat Enterprise Virtualization의 가상화 제한](#)에서 참조하십시오.

## 10.8. SPICE 제한 사항

SPICE에서 현재 지원하는 최대 해상도는 2560x1600 픽셀입니다.

## 10.9. 추가 참조

다음의 추가 문서 리소스는 Red Hat Virtualization 문서 모음의 일부가 아닙니다. 하지만 다음 문서는 Red Hat Virtualization 환경을 관리하는 시스템 관리자에게 유용한 정보를 담고 있으며 <https://access.redhat.com/documentation/ko/red-hat-enterprise-linux/>에서 볼 수 있습니다.

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

Red Hat Enterprise Linux 배포, 설정 및 관리 가이드입니다.

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

Red Hat Enterprise Linux에서 Device-Mapper Multipathing 사용에 대한 가이드입니다.

**Red Hat Enterprise Linux - 설치 가이드**

Red Hat Enterprise Linux 설치 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, 설정, 관리 및 문제 해결 가이드입니다.