

**Oracle® VM Server for SPARC 3.4 개발자
설명서**

ORACLE®

부품 번호: E71835
2016년 8월

부품 번호: E71835

Copyright © 2007, 2016, Oracle and/or its affiliates. All rights reserved.

본 소프트웨어와 관련 문서는 사용 제한 및 기밀 유지 규정을 포함하는 라이선스 합의서에 의거해 제공되며, 지적 재산법에 의해 보호됩니다. 라이선스 합의서 상에 명시적으로 허용되어 있는 경우나 법규에 의해 허용된 경우를 제외하고, 어떠한 부분도 복사, 재생, 번역, 방송, 수정, 라이선스, 전송, 배포, 진열, 실행, 발행, 또는 전시될 수 없습니다. 본 소프트웨어를 리버스 엔지니어링, 디스어셈블리 또는 디컴파일하는 것은 상호 운용에 대한 법규에 의해 명시된 경우를 제외하고는 금지되어 있습니다.

이 안의 내용은 사전 공지 없이 변경될 수 있으며 오류가 존재하지 않음을 보증하지 않습니다. 만일 오류를 발견하면 서면으로 통지해 주시기 바랍니다.

만일 본 소프트웨어나 관련 문서를 미국 정부나 또는 미국 정부를 대신하여 라이선스한 개인이나 법인에게 배송하는 경우, 다음 공지사항이 적용됩니다.

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

본 소프트웨어 혹은 하드웨어는 다양한 정보 관리 애플리케이션의 일반적인 사용을 목적으로 개발되었습니다. 본 소프트웨어 혹은 하드웨어는 개인적인 상해를 초래할 수 있는 애플리케이션을 포함한 본질적으로 위험한 애플리케이션에서 사용할 목적으로 개발되거나 그 용도로 사용될 수 없습니다. 만일 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서 사용할 경우, 라이선스 사용자는 해당 애플리케이션의 안전한 사용을 위해 모든 적절한 비상-안전, 백업, 대비 및 기타 조치를 반드시 취해야 합니다. Oracle Corporation과 그 자회사는 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서의 사용으로 인해 발생하는 어떠한 손해에 대해서도 책임지지 않습니다.

Oracle과 Java는 Oracle Corporation 및/또는 그 자회사의 등록 상표입니다. 기타의 명칭들은 각 해당 명칭을 소유한 회사의 상표일 수 있습니다.

Intel 및 Intel Xeon은 Intel Corporation의 상표 내지는 등록 상표입니다. SPARC 상표 일체는 라이선스에 의거하여 사용되며 SPARC International, Inc.의 상표 내지는 등록 상표입니다. AMD, Opteron, AMD 로고, 및 AMD Opteron 로고는 Advanced Micro Devices의 상표 내지는 등록 상표입니다. UNIX는 The Open Group의 등록상표입니다.

본 소프트웨어 혹은 하드웨어와 관련문서(설명서)는 제3자로부터 제공되는 콘텐츠, 제품 및 서비스에 접속할 수 있거나 정보를 제공합니다. 사용자와 오라클 간의 합의서에 별도로 규정되어 있지 않는 한 Oracle Corporation과 그 자회사는 제3자의 콘텐츠, 제품 및 서비스와 관련하여 어떠한 책임도 지지 않으며 명시적으로 모든 보증에 대해서도 책임을 지지 않습니다. Oracle Corporation과 그 자회사는 제3자의 콘텐츠, 제품 및 서비스에 접속하거나 사용으로 인해 초래되는 어떠한 손실, 비용 또는 손해에 대해 어떠한 책임도 지지 않습니다. 단, 사용자와 오라클 간의 합의서에 규정되어 있는 경우는 예외입니다.

설명서 접근성

오라클의 접근성 개선 노력에 대한 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=d0cacc>에서 Oracle Accessibility Program 웹 사이트를 방문하십시오.

오라클 고객센터 액세스

지원 서비스를 구매한 오라클 고객은 My Oracle Support를 통해 온라인 지원에 액세스할 수 있습니다. 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>를 참조하거나, 청각 장애가 있는 경우 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>를 방문하십시오.

목차

이 설명서 사용	7
1 Oracle VM Server for SPARC 템플릿, XML 파일 및 프로그램 개발	9
Oracle VM Server for SPARC 템플릿, XML 파일 및 프로그램 개발	9
2 Oracle VM Server for SPARC 템플릿 사용	11
Oracle VM Server for SPARC 템플릿 정보	11
Oracle VM Server for SPARC 템플릿 유틸리티 설치	12
Oracle VM Server for SPARC 템플릿 수명 주기	12
Oracle VM Server for SPARC 템플릿 예	15
3 논리적 도메인 Manager에서 XML 인터페이스 사용	21
XML 전송	21
XMPP 서버	22
로컬 연결	22
XML 프로토콜	22
요청 및 응답 메시지	23
이벤트 메시지	27
등록 및 등록 해제	27
<LDM_event> 메시지	28
이벤트 유형	29
논리적 도메인 Manager 작업	32
논리적 도메인 Manager 리소스 및 등록 정보	34
도메인 정보(idom_info) 리소스	34
CPU(cpu) 리소스	35
MAU(mau) 리소스	37
메모리(memory) 리소스	38
가상 SAN(vsan) 리소스	38
가상 디스크 서버(vds) 리소스	39

가상 디스크 서버 볼륨(vds_volume) 리소스	39
디스크(disk) 리소스	40
가상 스위치(vsw) 리소스	40
네트워크(network) 리소스	41
가상 콘솔 집중기(vcc) 리소스	42
변수(var) 리소스	43
물리적 I/O 장치(physio_device) 리소스	43
SP 구성(spconfig) 리소스	46
DRM 정책 구성(policy) 리소스	47
콘솔(console) 리소스	48
도메인 마이그레이션	48
XML 스키마	49
4 논리적 도메인 Manager 검색	51
논리적 도메인 Manager를 실행 중인 시스템 검색	51
멀티캐스트 통신	51
메시지 형식	51
▼ 서브넷에서 실행 중인 논리적 도메인 Manager를 검색하는 방법	52
5 가상 도메인 정보 명령 및 API 사용	55
가상 도메인 정보 명령 사용	55
가상 도메인 정보 API 사용	55
색인	57

이 설명서 사용

- 개요 – Oracle VM Server for SPARC 템플릿, XML 인터페이스, 논리적 도메인 Manager 검색 및 가상 도메인 정보 API를 설명하는 자세한 정보와 절차를 제공합니다.
- 대상 – SPARC 서버에서 가상화를 관리하는 시스템 관리자
- 필요한 지식 - 이러한 서버의 시스템 관리자는 UNIX 시스템 및 Oracle Solaris OS(Oracle Solaris 운영체제)를 사용할 수 있는 실제적인 지식을 보유하고 있어야 합니다.

제품 설명서 라이브러리

이 제품과 관련 제품들에 대한 설명서 및 리소스는 <http://www.oracle.com/technetwork/documentation/vm-sparc-194287.html>에서 사용할 수 있습니다.

피드백

<http://www.oracle.com/goto/docfeedback>에서 이 설명서에 대한 피드백을 보낼 수 있습니다.

Oracle VM Server for SPARC 템플릿, XML 파일 및 프로그램 개발

Oracle VM Server for SPARC 템플릿, XML 파일 및 프로그램 개발

이 설명서에서는 다음 Oracle VM Server for SPARC 기능에 대한 정보를 제공합니다.

- **Oracle VM Server for SPARC** 템플릿 유틸리티. 해당 유틸리티를 사용하여 도메인 구성 배치에 사용 가능한 템플릿을 만들 수 있습니다. [2장. Oracle VM Server for SPARC 템플릿 사용](#)을 참조하십시오.
- **Oracle VM Server for SPARC XML** 인터페이스. 이 XML 인터페이스를 사용하여 논리적 도메인의 구성을 기술할 수 있습니다. [3장. 논리적 도메인 Manager에서 XML 인터페이스 사용](#)을 참조하십시오.
- 논리적 도메인 **Manager** 검색 API. 이 API를 사용하여 논리적 도메인 Manager를 실행하는 도메인을 검색할 수 있는 프로그램을 작성할 수 있습니다. [4장. 논리적 도메인 Manager 검색](#)을 참조하십시오.
- 가상 도메인 **정보 API**. 이 API를 사용하여 가상 도메인에 대한 정보를 제공하는 프로그램을 작성할 수 있습니다. [5장. 가상 도메인 정보 명령 및 API 사용](#)을 참조하십시오.

주 - 이 설명서에 설명된 기능은 [Oracle VM Server for SPARC 3.4 설치 설명서](#)에 나와 있는 지원되는 모든 시스템 소프트웨어와 하드웨어 플랫폼에서 사용할 수 있습니다. 하지만 일부 기능은 지원되는 시스템 소프트웨어 및 하드웨어 플랫폼의 하위 세트에서만 사용할 수 있습니다. 이러한 예외사항에 대한 자세한 내용은 [Oracle VM Server for SPARC 3.4 릴리스 노트](#)의 “이 릴리스의 새로운 기능” 및 [What's New in Oracle VM Server for SPARC Software](#) (<http://www.oracle.com/technetwork/server-storage/vm/documentation/sparc-whatsnew-330281.html>)를 참조하십시오.

Oracle VM Server for SPARC 템플릿 사용

이 장에서는 다음 주제를 다룹니다.

- “Oracle VM Server for SPARC 템플릿 정보” [11]
- “Oracle VM Server for SPARC 템플릿 유틸리티 설치” [12]
- “Oracle VM Server for SPARC 템플릿 수명 주기” [12]
- “Oracle VM Server for SPARC 템플릿 예” [15]

Oracle VM Server for SPARC 템플릿 정보

Oracle VM Server for SPARC 템플릿 명령으로 SPARC 시스템에서 Oracle VM Server for SPARC 템플릿을 만들고 배치 및 구성할 수 있습니다. 이 명령은 아카이브(.ova)에 포함된 등록 정보의 XML 설명자와 디스크 이미지 파일을 포함하는 OVF 템플릿 사양을 기반으로 합니다.

이 명령은 Oracle VM Server for SPARC 시스템의 컨트롤 도메인에서 실행합니다. 명령줄에서 실행하는 것 외에도, 완전 자동화된 방식으로 명령을 실행하거나 다른 프로그램에서 대규모 워크플로우의 일부로 실행할 수 있습니다.

이러한 템플릿을 사용하여 게스트 도메인, I/O 도메인 및 루트 도메인만 배치하고 구성할 수 있습니다. 그러나 템플릿을 사용하여 `primary` 도메인으로 지정된 I/O 도메인 또는 루트 도메인을 배치하고 구성하거나, 서비스 도메인을 배치하고 구성할 수는 없습니다.

Oracle VM Server for SPARC 템플릿 명령은 다음과 같습니다.

- `ovmtadm - ovmtcreate, ovmtconfig` 및 `ovmtdeploy` 명령을 시작하여 템플릿을 만들고 구성, 배치 및 제거할 수 있도록 해줍니다. `ovmtadm(1M)` 매뉴얼 페이지를 참조하십시오.
- `ovmtconfig` - 등록 정보 이름-값 쌍을 대상 도메인의 응용 프로그램 및 프로세스로 전송하여 도메인에 대해 구성 작업(예: Oracle Solaris OS 구성 방식 및 처음 부트 스크립트)을 수행합니다. 등록 정보 이름-값 쌍은 `ovmtprop` 명령을 사용하여 설정됩니다.

또한 이 명령을 사용하여 도메인의 ZFS 파일 시스템을 백마운트할 수 있습니다. 그러면 컨트롤 도메인이 해당 파일 시스템에서 직접 명령을 실행합니다. 이 방법에서는 파일을 복사하고, Oracle Solaris OS 패키지를 설치 및 업그레이드하고, 구성 작업을 수행할 수 있습니다. `ovmtconfig(1M)` 매뉴얼 페이지를 참조하십시오.

- `ovmcreate` - 기존 Oracle VM Server for SPARC 도메인에서 템플릿을 만듭니다. [ovmcreate\(1M\)](#) 매뉴얼 페이지를 참조하십시오.
- `ovmdeploy` - Oracle VM Server for SPARC 템플릿에서 도메인을 만듭니다. [ovmdeploy\(1M\)](#) 매뉴얼 페이지를 참조하십시오.
- `ovmtlibrary` - 데이터베이스 및 파일 시스템 기반의 Oracle VM Server for SPARC 템플릿 저장소에서 파일을 구성하고 데이터베이스 정보를 저장, 검색, 편집하여 저장소를 관리합니다. [ovmtlibrary\(1M\)](#) 매뉴얼 페이지를 참조하십시오.
- `ovmprop` - 템플릿에서 배치된 도메인에서 Oracle Solaris OS 등록 정보를 확인 및 설정할 수 있도록 해줍니다. 등록 정보는 이름-값 쌍으로 지정됩니다. 이 명령은 구성 작업 및 등록 정보를 기반으로 할 수 있는 작업을 수행할 다른 스크립트 및 프로그램에서 호출됩니다. [ovmprop\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

Oracle VM Server for SPARC 템플릿 시작하기에 대한 자세한 내용은 다음 블로그를 참조하십시오.

- [Deploying a Template \(https://blogs.oracle.com/cmt/en/entry/getting_started_with_ovm_templates\)](https://blogs.oracle.com/cmt/en/entry/getting_started_with_ovm_templates)
- [Creating a Template \(https://blogs.oracle.com/cmt/en/entry/getting_started_with_ovm_templates1\)](https://blogs.oracle.com/cmt/en/entry/getting_started_with_ovm_templates1)
- [Using Templates on Oracle SuperCluster \(https://blogs.oracle.com/cmt/en/entry/getting_started_with_ovm_templates2\)](https://blogs.oracle.com/cmt/en/entry/getting_started_with_ovm_templates2)

Oracle VM Server for SPARC 템플릿 유틸리티 설치

Oracle VM Server for SPARC 템플릿 유틸리티 소프트웨어 패키지를 컨트롤 도메인에 설치하려면 먼저 Oracle VM Server for SPARC 3.4 소프트웨어를 컨트롤 도메인에 다운로드하여 설치해야 합니다. [Oracle VM Server for SPARC 3.4 설치 설명서 의 2 장, “소프트웨어 설치”](#)를 참조하십시오.

그 다음, Oracle VM Server for SPARC 템플릿 유틸리티 소프트웨어 패키지 `ovmtutils`를 설치합니다.

```
# pkg install -v pkg:/system/ldoms/ovmtutils
```

Oracle VM Server for SPARC 템플릿 수명 주기

이 절에서는 템플릿 만들기 프로세스의 각 단계와 수행할 작업, 그리고 프로세스를 보조할 Oracle VM Server for SPARC 템플릿 유틸리티 사용 방법을 설명합니다.

주 - 응용 프로그램과 처음 부트 스크립트로 템플릿을 만들고 개발하는 과정은 반복적 프로세스입니다. 소스 코드 관리 시스템으로 스크립트 및 등록 정보를 관리하여 모든 구성 측면을 동기화하십시오.

다음은 템플릿 만들기 프로세스의 단계와 수행할 작업, 그리고 프로세스를 보조할 Oracle VM Server for SPARC 템플릿 유틸리티 사용 방법을 설명합니다.

1. **템플릿 저작.** 사전 구축된 일반 템플릿을 사용할 수 있지만, 기존 도메인에서 사용자정의 템플릿을 만들 수 있습니다. 이 도메인에는 운영체제 구성 요소, 응용 프로그램 소프트웨어 및 기타 유틸리티가 완전히 설치되어야 합니다.

일반적으로 가능한 완전히 환경이 구성되어 있으며, 환경을 마무리하기 위해 몇 가지 작업만 수행하면 됩니다. 메모리, 가상 CPU, 가상 네트워킹, 디스크와 같은 도메인 설정이 원하는 배치 환경을 반영해야 합니다.

이 단계에서 하나 이상의 “처음 부트” 스크립트를 만듭니다. 이 스크립트는 사용자가 제공한 등록 정보에 준하여 최종 구성을 수행하는 환경에 포함됩니다. 각 템플릿에 대해 이 등록 정보를 README 파일에 기록하고 설명해야 합니다.

주 - 처음 부트 스크립트가 도메인 변수에 액세스하는 경우 `ovmtprop` 유틸리티가 게스트 도메인에 설치되어 있는지 확인합니다.

2. **템플릿 만들기.** 템플릿을 만들기 전에 소스 도메인 환경이 구성되지 않았는지 확인합니다. 그래야 처음 부트 스크립트에 속하는 규정된 작업으로 나중에 구성할 수 있습니다.

예를 들어, 다음 단계를 수행합니다.

- 응용 프로그램 특정 구성을 제거합니다(나중에 다시 만들 예정).
- 구성 파일에 기본값을 사용합니다.
- 루트 파일 시스템을 제외한 모든 `zpool`을 내보내어 새 도메인에서 인식할 수 있도록 합니다.
- 템플릿에서 배치한 후 처음 부트 시 구성 등록 정보를 수락할 준비가 되도록 운영체제를 구성되지 않은 상태로 되돌립니다. 다음 명령을 실행하여 사이트 레벨 사용자 정의를 제거하고 구성을 해제한 다음 운영체제를 중단합니다.

```
# sysconfig unconfigure -g system -s --include-site-profile --destructive
```

해당 단계를 수행한 후 `ovmtcreate` 명령을 실행하여 도메인에서 템플릿을 만듭니다.

주 - 이 명령을 실행한 후 배치 시 등록 정보가 제공되지 않을 경우 시스템에서 다음 번 부트 시 Oracle Solaris 대화식 설치 프로그램을 실행합니다.

3. **템플릿 이름 지정.** 다음 형식과 같이 일관된 규칙을 사용하여 템플릿을 식별할 수 있습니다.

```
technology.OS.application.architecture.build.ova
```

예를 들어, 다음 템플릿 이름은 Oracle Solaris 11.2 OS 빌드 2를 SPARC 플랫폼에서 실행하고 WebLogic Server 버전 12.1.2를 실행하는 도메인을 위한 것입니다. `ovm_s11.2_wls12.1.2_sparc_b2.ova`

4. **템플릿 배포.** 템플릿은 `.ova` 확장자를 가진 단일 파일입니다. 파일에는 압축된 디스크 이미지와 배치에 필요한 메타데이터가 들어 있습니다. 또한 템플릿은 페이로드 파일 체크섬의 매니페스트 파일을 포함하며, 이것을 전체 아카이브 체크섬과 조합하여 배포 이후 내용이 변경되지 않았는지 검증할 수 있습니다.

템플릿을 복제하기보다는 중앙 저장소를 운영하거나 웹 기반 서비스를 사용하여 템플릿을 배포할 수 있습니다.

5. **템플릿 배치.** 템플릿은 소스 도메인에 보이는 시스템 측면만 캡처하기 때문에 템플릿 배치를 지원하려면 어떤 서비스를 제공해야 하는지 이해해야 합니다.

필요한 서비스는 다음과 같습니다.

- 템플릿에서 가상 네트워크로 연결할 수 있는 적절한 인터페이스에 필요한 하나 이상의 가상 스위치
- 가상 디스크 서비스
- 콘솔 서비스
- 템플릿 요구 사항을 수용하기에 충분한 가상 CPU 및 메모리

`ovmdeploy` 유틸리티가 이러한 설정의 대부분을 대체할 수 있지만, 템플릿과 함께 제공된 최소값은 기준 요구 사항을 나타냅니다.

`ovmdeploy` 유틸리티를 사용하여 자동으로 가상 디스크를 배치 디렉토리로 추출, 압축 해제, 복사할 수 있고 템플릿이 설명하는 다양한 가상 장치를 구축할 수 있습니다.

이 시점에서 도메인을 시작할 수 있지만, 도메인이 완벽히 작동하려면 도메인 콘솔을 사용하여 몇 가지 수동 구성 단계를 수행해야 합니다.

6. **자동으로 도메인 구성.** 템플릿으로 만들어진 도메인의 구성은 여러 유형의 작업으로 이루어집니다. 예를 들어, 등록 정보 이름-값 쌍을 지정하여 구성할 정보와 함께 처음 부트 스크립트를 제공할 수 있습니다. 또한 가상 디스크를 컨트롤 도메인으로 백마운트하여 도메인 파일 시스템에 구성 파일 복사 등의 작업을 수행할 수 있습니다.

`ovmconfig` 유틸리티는 이 도메인 구성 작업을 자동화하므로 사용자는 하나 이상의 명령 스크립트 및 등록 정보 파일을 지정하여 도메인 구성에 사용할 등록 정보와 수행할 작업을 지정할 수 있습니다.

Oracle Solaris OS를 구성하기 위해 `ovmconfig` 유틸리티는 도메인의 루트 파일 시스템을 백마운트하고 제공된 구성 스크립트 및 등록 정보로부터 `sc_profile.xml` 파일을 만듭니다. 이 프로파일을 사용하면 처음 부트 시 Oracle Solaris OS가 자체 구성됩니다.

7. **첫번째 구성.** 성공적인 Oracle Solaris OS 구성과 처음 부트를 수행하려면 설치된 응용 프로그램을 구성해야 합니다. 구성 단계 동안 `ovmconfig` 유틸리티는 다음 방법 중 하나를 사용하여 배치된 도메인에 구성 정보를 전달합니다.

- **도메인 변수 - 로컬 등록 정보 파일 외에도, 컨트롤 도메인에서 `ovmconfig` 유틸리티를 실행하여 도메인 변수를 설정할 수 있습니다. 그러면 게스트 도메인의 `ovmprop` 유틸리티에서 사용할 수 있습니다. 이 방법으로 처음 부트 스크립트를 통해 등록 정보에 직접 액세스하고, 구성이 완료된 후 게스트 도메인에 직접 구성 정보를 제공할 수 있습니다.**

예를 들어, 게스트 도메인에서 `ovmtprop`를 실행하는 슈퍼바이저 스크립트를 사용하고 컨트롤 도메인에서 `ovmtconfig -v`를 실행하면 네트워크 액세스가 관여하지 않는 구성 측면을 자동 변경할 수 있습니다.

- **직접 작업** - `ovmtconfig` 유틸리티는 게스트 도메인 파일 시스템을 컨트롤 도메인으로 백 마운트하고 파일 및 파일 시스템에 직접 작업을 수행합니다. 구성 파일 만들기, 시스템 바이너리 복사 등의 작업이 포함될 수 있습니다. 이 작업은 `ovmtconfig` 유틸리티에 제공한 스크립트에 설명되어 있습니다. `/opt/ovmtutils/share/scripts` 디렉토리에서 명령 작업을 수행하는 스크립트를 찾을 수 있습니다.

주 - 이 작업은 컨트롤 도메인에 영향을 미칠 수 있기 때문에 일반적으로 게스트 도메인에서 실행하도록 설계된 처음 부트 프로세스를 포함하지 않습니다.

`ovmtconfig -c` 명령을 사용하여 실행할 명령을 지정합니다.



주의 - 암호화되지 않은 등록 정보를 사용하여 암호와 같은 중요한 정보를 도메인에 전달하지 마십시오. Oracle Solaris OS를 구성하는 데 사용되는 것 이외의 등록 정보는 일반 텍스트로 도메인에 `ldm` 변수로 전달됩니다. 이러한 등록 정보 값은 `ldm` 명령을 실행할 권한이 있는 컨트롤 도메인의 사용자와 배치된 도메인에 로그인한 사용자가 볼 수 있습니다.

이 시점에서 도메인은 완전히 구성되고 작동해야 합니다.

Oracle VM Server for SPARC 템플릿 예

이 절에서는 다음 Oracle VM Server for SPARC 템플릿 작업의 예를 보여줍니다.

- 예 1. “Oracle VM Server for SPARC 템플릿 만들기”
- 예 2. “Oracle VM Server for SPARC 템플릿 등록 정보 구성”
- 예 3. “Oracle VM Server for SPARC 템플릿 배치”
- 예 4. “Oracle VM Server for SPARC 템플릿 라이브러리 관리”

예 1 Oracle VM Server for SPARC 템플릿 만들기

다음 `ovmtcreate` 명령은 `ldg1` 도메인을 기반으로 `ovmtcreate_example`이라는 템플릿을 만듭니다. 결과 템플릿 이름에는 `.ova` 접미어가 붙습니다.

```
# ovmtcreate -d ldg1 -o ovmtcreate_example
...
STAGE 1 - EXAMINING SYSTEM AND ENVIRONMENT
-----
Performing platform & prerequisite checks
```

```

Checking user permissions
Checking for required packages
Checking for required services
Checking directory permissions

STAGE 2 - ANALYZING DOMAIN
-----
Retrieving and processing attributes
Checking domain state
Getting domain resource settings
Discovering network topology
Discovering disk topology

STAGE 3 - ARCHIVE CREATION
-----
Checking destination and current directory capacity
Compressing disk image
Creating XML configuration
Calculating manifest checksums
Creating archive file
Checking archive

PROCESS COMPLETED
-----
Started: Tue Aug 18 15:29:14 PDT 2015
Completed: Tue Aug 18 15:41:25 PDT 2015
Elapsed time: 0:12:11

```

예 2 Oracle VM Server for SPARC 템플릿 등록 정보 구성

ovmtconfig 및 ovmtprop 유틸리티를 사용하여 각각 Oracle VM Server for SPARC 템플릿 등록 정보 값과 Oracle Solaris OS 등록 정보 값을 지정할 수 있습니다.

- 다음 ovmtconfig 명령은 ldg1 도메인의 파일 시스템에 직접 구성 작업을 수행합니다.
 - c 옵션은 등록 정보 값을 설정할 /opt/ovmtutils/share/scripts/ovmt_s11_scprofile.sh 스크립트를 지정합니다. -p 옵션은 com.oracle.solaris.network.ipaddr 및 com.oracle.solaris.system.computer-name 등록 정보에 대해 특정 값을 지정합니다.

```

# ovmtconfig -v -d ldg1 -f -s \
-c /opt/ovmtutils/share/scripts/ovmt_s11_scprofile.sh \
-p com.oracle.solaris.network.ipaddr.0=10.153.118.211,\
com.oracle.solaris.system.computer-name=system1
...

```

```

STAGE 1/7 - EXAMINING SYSTEM AND ENVIRONMENT
-----
Checking operating system
Checking platform type
Checking user permissions
Checking packages
Checking host domain name
Checking host domain type
Checking services

```

STAGE 2/7 - PROCESSING COMMAND LINE OPTIONS

```
-----
Parsing individual properties
Creating consolidated properties list
```

STAGE 3/7 - ANALYZING TARGET DOMAIN

```
-----
Stopping domain ldg1
Analyzing domain disk topology for domain ldg1
Discovering 1 volumes for vDisks
Examining 1 backend devices
unbinding domain ldg1
Creating 1 virtual disks for back mount
Created virtual disk 0
```

STAGE 4/7 - PERFORMING BACKMOUNT

```
-----
Finding Solaris device for vdisks
Importing zpools for 1 Solaris devices
Detected conflicting zpool name, attempting rename
Getting boot file system for properties in 1 zpool
Setting properties in 1 zpools
Mounting ZFS file systems
Mounting ZFS found in zpool rpool_1
```

STAGE 5/7 - PERFORMING ACTIONS ON TARGET DOMAIN

STAGE 6/7 - UNMOUNTING AND RESTORING DOMAIN STATE

```
-----
Rolling back commands DEBUG [20150819-07:02:42]: Rolling back 8 /usr/sbin/zfs unmount -f
rpool_1/ROOT/solaris/var
completed
```

STAGE 7/7 - SETTING TARGET DOMAIN ENVIRONMENT

```
-----
Checking 2 properties to set as domain variables
Process completed
```

- 다음 `ovmtprop` 명령은 Oracle Solaris OS 등록 정보 값을 지정합니다.

```
primary# ovmtprop set-prop com.oracle.solaris.system.computer-name=test ldg1
```

`ldm list -l` 명령을 사용하여 `com.oracle.solaris.system.computer-name` 등록 정보 값이 `test`인지 확인합니다.

```

primary# ldm list -l ldg1
NAME          STATE      FLAGS  CONS  VCPU  MEMORY  UTIL  NORM  UPTIME
ldg1          active    -n---- 5000   8     8G     0.0%  0.0%  23h 23m
..
VARIABLES
  auto-boot?=true
  boot-file=-k
  pm_boot_policy=disabled=0;ttfc=2000;ttmr=0;

VMAPI TO GUEST
  com.oracle.solaris.fmri.count=0
  com.oracle.solaris.system.computer-name=test

```

예 3 Oracle VM Server for SPARC 템플리트 배치

다음 `ovmtdploy` 명령은 `ovmtdcreate_example.ova` Oracle VM Server for SPARC 템플리트를 사용하여 `/export/ovmtdploy` 디렉토리에 `ldg1`이라는 도메인을 만듭니다.

```
# ovmtdploy -d ldg1 -o /export/ovmtdploy ovmtdcreate_example.ova
...
```

```

STAGE 1 - EXAMINING SYSTEM AND ENVIRONMENT
-----
Checking user privilege
Performing platform & prerequisite checks
Checking for required services
Named resourced available

STAGE 2 - ANALYZING ARCHIVE & RESOURCE REQUIREMENTS
-----
Checking .ova format and contents
Validating archive configuration
Checking sufficient resources present
WARNING: Virtual switch primary-vsw0 already exists

STAGE 3 - EXTRACTING ARCHIVE
-----
Extracting archive
Validating checksums
Decompressing disk image(s)

STAGE 4 - DEPLOYING DOMAIN
-----
Creating domain and adding resources
Validating deployment
Domain created:

```

`ldm list` 출력은 새로운 `ldg1` 도메인을 만들었음을 보여줍니다.

```

# ldm list
NAME      STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  NORM  UPTIME
primary  active -n-cv-  UART  8     40G    1.4%  1.1%  6d 2h 18m
ldg1     active -n---- 5000   8     8G     41%   38%   28s

```

예 4 Oracle VM Server for SPARC 템플리트 라이브러리 관리

ovmtlibrary 명령은 파일을 구성하고, 데이터베이스에 정보를 저장하고 데이터베이스에서 정보를 검색하고 편집하는 방식으로 Oracle VM Server for SPARC 템플리트에 대한 데이터베이스 및 파일 시스템 기반 저장소를 관리합니다.

- 다음 명령은 export/user1/ovmtlibrary_example에 템플리트 라이브러리를 만듭니다.

```
# ovmtlibrary -c init -l /export/user1/ovmtlibrary_example
...
```

```
Init complete
```

- 다음 명령은 export/user1/ovmtlibrary_example 라이브러리에 sol-11_2-ovm-2-sparc.ova 템플리트를 저장합니다.

```
# ovmtlibrary -c store -d "ovmtlibrary example" -o http://system1.example.com/s11.2/templates/
sol-11_2-ovm-2-sparc.ova -l /export/user1/ovmtlibrary_example
...
```

```
Templates present in path "/export/user1/ovmtlibrary_example"
```

```
event id is 2
```

```
*****
converted 'http://system1.example.com/s11.2/templates/sol-11_2-ovm-2-sparc.ova' (646) ->
'http://system1.example.com/s11.2/templates/sol-11_2-ovm-2-sparc.ova' (UTF-8)
--2015-08-18 16:37:17-- http://system1.example.com/s11.2/templates/sol-11_2-ovm-2-
sparc.ova
Resolving system1.example.com (system1.example.com)... 10.134.127.18
Connecting to system1.example.com (system1.example.com)|10.134.127.18|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1018341888 (971M) [text/plain]
Saving to: '/export/user1/ovmtlibrary_example/repository/templates/1/1/sol-11_2-ovm-2-
sparc.ova'
```

```
/export/user1/ovmtlibrary_example/repo 100%
```

```
[=====>]
```

```
971.17M 6.05MB/s in 5m 37s
```

```
2015-08-18 16:42:55 (2.88 MB/s) - '/export/user1/ovmtlibrary_example/repository/
templates/1/1/sol-11_2-ovm-2-sparc.ova' saved
```

```
[1018341888/1018341888]
```

```
*****
```

```
Download complete
```

```
Extracting the ova file...
```

```
Extract complete
```

```
Decompress file System.img.gz
```

Store complete

- 다음 명령은 export/user1/ovmtlibrary_example 라이브러리의 내용을 나열합니다.

```
# ovmtlibrary -c list -l /export/user1/ovmtlibrary_example
...
```

Templates present in path "/export/user1/ovmtlibrary_example"

ID Name	Version	Description	Date
1	sol-11_2-ovm-2-sparc	1	ovmtlibrary example 2015-08-18

- 다음 명령은 export/user1/ovmtlibrary_example 라이브러리의 상세한 목록을 보여줍니다.

```
# ovmtlibrary -c list -i 1 -o -l /export/user1/ovmtlibrary_example
...
```

Templates present in path "/export/user1/ovmtlibrary_example"

ID Name	Type	Path	Size(bytes)
1	sol-11_2-ovm-2-sparc.ova	ova /export/user1/ovmtlibrary_example/repository/templates/1/1/sol-11_2-ovm-2-sparc.ova	1018341888
2	sol-11_2-ovm-sparc.ovf	ovf /export/user1/ovmtlibrary_example/repository/templates/1/1/sol-11_2-ovm-sparc.ovf	3532
3	sol-11_2-ovm-sparc.mf	mf /export/user1/ovmtlibrary_example/repository/templates/1/1/sol-11_2-ovm-sparc.mf	137
4	System.img	img /export/user1/ovmtlibrary_example/repository/templates/1/1/System.img	21474836480

- 다음 명령은 export/user1/ovmtlibrary_example 라이브러리에서 템플릿 ID 1을 삭제합니다.

```
# ovmtlibrary -c delete -i 1 -l /export/user1/ovmtlibrary_example
```

◆◆◆ 3 장

논리적 도메인 Manager에서 XML 인터페이스 사용

이 장에서는 외부 사용자 프로그램이 Oracle VM Server for SPARC 소프트웨어와 연결하는데 사용할 수 있는 XML(확장성 마크업 언어) 통신 방식에 대해 설명합니다. 다음 기본 항목을 다룹니다.

- “XML 전송” [21]
- “XML 프로토콜” [22]
- “이벤트 메시지” [27]
- “논리적 도메인 Manager 작업” [32]
- “논리적 도메인 Manager 리소스 및 등록 정보” [34]
- “XML 스키마” [49]

XML 전송

외부 프로그램은 확장성 메시징 및 프레즌스 프로토콜(XMPP – RFC 3920)을 사용하여 논리적 도메인 Manager와 통신할 수 있습니다. XMPP는 로컬 연결과 원격 연결 모두에 대해 지원되며, 기본적으로 설정되어 있습니다. XML 인터페이스는 TLS(전송 계층 보안) 프로토콜 버전 1.2만 지원합니다.

원격 연결을 사용 안함으로 설정하려면 `ldmd/xmpp_enabled` SMF 등록 정보를 `false`로 설정하고 논리적 도메인 Manager를 다시 시작하십시오.

```
# svccfg -s ldom/ldmd setprop ldmd/xmpp_enabled=false
# svcadm refresh ldmd
# svcadm restart ldmd
```

주 - XMPP 서버를 사용 안함으로 설정하면 도메인 마이그레이션 및 메모리 동적 재구성도 수행되지 않습니다.

XMPP 서버

논리적 도메인 Manager는 사용 가능한 여러 XMPP 클라이언트 응용 프로그램과 라이브러리와 통신할 수 있는 XMPP 서버를 구현합니다. 논리적 도메인 Manager에서 사용하는 보안 방식은 다음과 같습니다.

- 클라이언트와 자신과의 통신 채널에 보안을 설정하는 전송 계층 보안
- 인증용 SASL(Simple Authentication and Security Layer). PLAIN은 유일하게 지원되는 SASL 방식입니다. 사용자 이름 및 암호를 서버에 제공해야만 모니터링 또는 관리 작업을 허용하기 전에 권한을 부여 받을 수 있습니다.

로컬 연결

논리적 도메인 Manager는 사용자 클라이언트가 자신과 동일한 도메인에서 실행 중인지 여부를 감지하여, 그럴 경우 해당 클라이언트와 최소한의 XMPP 핸드셰이크를 수행합니다. 특히 TLS를 통해 보안 채널을 설정한 후 SASL 인증 단계를 건너뛸 수 있습니다. 인증 및 권한 부여는 클라이언트 인터페이스를 구현하는 프로세스의 자격 증명을 기준으로 수행됩니다.

클라이언트는 전체 XMPP 클라이언트를 구현하거나 단순히 libxml2 SAX(Simple API for XML) 구문 분석기와 같은 스트리밍 XML 구문 분석기를 실행할 수 있습니다. 어떤 방식을 사용하든지 클라이언트는 TLS 협상 지점에 대해 XMPP 핸드셰이크를 처리해야 합니다. 필요한 순서는 XMPP 사양을 참조하십시오.

XML 프로토콜

통신 초기화가 완료되면 Oracle VM Server for SPARC에서 정의한 XML 메시지가 전송됩니다. 두 가지 일반 유형의 XML 메시지는 다음과 같습니다.

- 요청 및 응답 메시지 - <LDM_interface> 태그를 사용합니다. 이 유형의 XML 메시지는 CLI(명령줄 인터페이스)를 사용하여 명령을 실행하는 것과 비슷하게, 명령을 전달하고 논리적 도메인 Manager에서 다시 결과를 가져오는 데 사용됩니다. 이 태그는 이벤트 등록 및 등록 해제에도 사용됩니다.
- 이벤트 메시지 - <LDM_event> 태그를 사용합니다. 이 유형의 XML 메시지는 논리적 도메인 Manager에서 게시한 이벤트를 비동기적으로 보고하는 데 사용됩니다.

요청 및 응답 메시지

Oracle VM Server for SPARC에 대한 XML 인터페이스는 두 가지 형식입니다.

- 명령을 논리적 도메인 Manager로 보내는 데 사용되는 형식
- 논리적 도메인 Manager가 수신 메시지의 상태 및 해당 메시지 내에서 요청된 작업에 대해 응답하는 데 사용되는 형식

이 두 형식은 여러 가지 공통 XML 구조를 공유하지만, 두 형식의 차이점을 보다 잘 이해할 수 있도록 여기서는 개별적으로 설명합니다.

요청 메시지

가장 기본적인 레벨에서 논리적 도메인 Manager에 대한 수신 XML 요청에는 단일 객체에 대해 작동하는 단일 명령의 설명이 포함됩니다. 보다 복잡한 요청은 여러 개의 명령과 명령당 여러 개의 객체를 처리할 수 있습니다. 다음 예에서는 기본 XML 명령의 구조를 보여줍니다.

예 5 단일 객체에 대해 작동하는 단일 명령의 형식

```
<LDM_interface version="1.3">
  <cmd>
    <action>Place command here</action>
    <options>Place options for certain commands here</options>
    <arguments>Place arguments for certain commands here</arguments>
    <data version="3.0">
      <Envelope>
        <References/>
        <!-- Note a <Section> section can be here instead of <Content>
        <Content xsi:type="ovf:VirtualSystem_Type" id="Domain name">
          <Section xsi:type="ovf:ResourceAllocationSection_type">
            <Item>
              <rasd:OtherResourceType>LDom Resource Type</rasd:OtherResourceType>
              <gprop:GenericProperty
                key="Property name">Property Value</gprop:GenericProperty>
            </Item>
          </Section>
        </Content>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>
```

<LDM_interface> 태그

논리적 도메인 Manager로 전송되는 모든 명령은 <LDM_interface> 태그로 시작해야 합니다. 논리적 도메인 Manager로 전송되는 모든 문서는 <LDM_interface> 태그를 한 개만 포함해야 합니다.

다. <LDM_interface> 태그에는 예 5. “단일 객체에 대해 작동하는 단일 명령의 형식”과 같은 버전 속성이 포함되어야 합니다.

<cmd> 태그

<LDM_interface> 태그 내에서 문서는 <cmd> 태그를 적어도 한 개 포함해야 합니다. 각 <cmd> 섹션에는 <action> 태그가 한 개만 있어야 합니다. <action> 태그는 실행할 명령을 설명하는 데 사용됩니다. 각 <cmd> 태그는 명령이 작동할 객체를 설명하는 <data> 태그를 적어도 한 개 포함해야 합니다.

<cmd> 태그는 일부 명령과 연관된 옵션과 플래그에 사용되는 <options> 태그도 포함할 수 있습니다. 옵션을 사용하는 명령은 다음과 같습니다.

- ldm remove-domain 명령은 -a 옵션을 사용할 수 있습니다.
- ldm bind-domain 명령은 -f 옵션을 사용할 수 있습니다.
- ldm add-vdsdev 명령은 -f 옵션을 사용할 수 있습니다.
- ldm cancel-operation 명령은 migration 또는 reconf 옵션을 사용할 수 있습니다.
- ldm add-spconfig 명령은 -r *autosave-name* 옵션을 사용할 수 있습니다.
- ldm remove-spconfig 명령은 -r 옵션을 사용할 수 있습니다.
- ldm list-spconfig 명령은 -r [*autosave-name*] 옵션을 사용할 수 있습니다.
- ldm stop-domain 명령은 다음 태그를 사용하여 명령 인수를 설정할 수 있습니다.
 - <force>는 -f 옵션을 나타냅니다.
 - <halt>는 -h 옵션을 나타냅니다.
 - <message>는 -m 옵션을 나타냅니다.
 - <quick>은 -q 옵션을 나타냅니다.
 - <reboot>는 -r 옵션을 나타냅니다.
 - <timeout>은 -t 옵션을 나타냅니다.

태그에는 콘텐츠 값이 포함되지 않아야 합니다. 하지만 -t 및 -m 옵션에는 null이 아닌 값이 있어야 합니다(예: <timeout>10</timeout> 또는 <message>Shutting down now</message>).

다음 XML 단편 예에서는 재부트 메시지가 포함된 재부트 요청을 ldm stop-domain 명령으로 전달하는 방법을 보여줍니다.

```
<action>stop-domain</action>
<arguments>
  <reboot/>
  <message>my reboot message</message>
</arguments>
```

<data> 태그

각 <data> 섹션은 지정된 명령과 관련된 객체에 대한 설명을 포함할 수 있습니다. <data> 섹션의 형식은 OVF(Open Virtualization Format) 드래프트 사양의 XML 스키마 부분을 기반으로 합니다. 해당 스키마는 <References> 태그(Oracle VM Server for SPARC에서 사용하지 않음)와 <Content> 및 <Section> 섹션을 포함하는 <Envelope> 섹션을 정의합니다.

Oracle VM Server for SPARC의 경우, <Content> 섹션은 특정 도메인을 식별하고 설명하는 데 사용됩니다. <Content> 노드의 id= 속성에 있는 도메인 이름으로 도메인을 식별합니다. <Content> 섹션 내에는 특정 명령에 필요한 도메인의 리소스를 설명하는 하나 이상의 <Section> 섹션이 있습니다.

도메인 이름만 식별해야 하는 경우에는 <Section> 태그를 사용할 필요가 없습니다. 반대로, 명령에 도메인 식별자가 필요하지 않은 경우 명령에 필요한 리소스를 설명하는 <Section> 섹션을 <Content> 섹션 외부에 제공하되 <Envelope> 섹션에 포함되도록 해야 합니다.

객체 정보를 추론할 수 있는 경우 <data> 섹션은 <Envelope> 태그를 포함할 필요가 없습니다. 이러한 상황은 작업에 해당하는 모든 객체를 모니터링하는 요청, 이벤트 등록 및 등록 해제 요청에 주로 적용됩니다.

2개의 추가 OVF 유형은 OVF 사양의 스키마를 사용하여 모든 유형의 객체를 제대로 정의할 수 있도록 해줍니다.

- <gprop:GenericProperty> 태그
- <Binding> 태그

<gprop:GenericProperty> 태그는 OVF 사양에 정의가 없는 객체 등록 정보를 처리합니다. 등록 정보 이름은 노드의 key= 속성에 정의되며 등록 정보 값은 노드 콘텐츠입니다. <binding> 태그는 다른 리소스에 바인드된 리소스를 정의하기 위해 `ldm list-bindings` 명령 출력에 사용됩니다.

응답 메시지

발신 XML 응답은 포함된 명령과 객체의 관점에서 수신 요청의 구조와 일치하지만, 요청에 대한 전체 <Response> 섹션과 지정된 각 객체와 명령에 대한 <Response> 섹션이 추가됩니다. <Response> 섹션은 상태 및 메시지 정보를 제공합니다. 다음 예에서는 기본 XML 요청에 대한 응답의 구조를 보여줍니다.

예 6 단일 객체에 대해 작동하는 단일 명령에 대한 응답의 형식

```
<LDM_interface version="1.3">
  <cmd>
    <action>Place command here</action>
    <data version="3.0">
      <Envelope>
```

```

<References/>
<!-- Note a <Section> section can be here instead of <Content>
<Content xsi:type="ovf:VirtualSystem_Type" id="Domain name">
  <Section xsi:type="ovf:ResourceAllocationSection_type">
    <Item>
      <rasd:OtherResourceType>
        LDom Resource Type
      </rasd:OtherResourceType>
      <gprop:GenericProperty
        key="Property name">
        Property Value
      </gprop:GenericProperty>
    </Item>
  </Section>
  <!-- Note: More <Section>
</Content>
</Envelope>
<response>
  <status>success or failure</status>
  <resp_msg>Reason for failure</resp_msg>
</response>
</data>

<response>
  <status>success or failure</status>
  <resp_msg>Reason for failure</resp_msg>
</response>
</cmd>

<response>
  <status>success or failure</status>
  <resp_msg>Reason for failure</resp_msg>
</response>
</LDM_interface>

```

전체 응답

<response> 섹션은 <LDM_interface> 섹션의 직속 하위로, 전체 요청의 전반적인 성공 또는 실패를 나타냅니다. 수신 XML 문서의 형식이 잘못된 경우가 아니라면 <response> 섹션에는 <status> 태그만 포함됩니다. 이 응답 상태가 성공을 나타낸다면 모든 객체에 대한 모든 명령이 성공한 것입니다. 이 응답 상태가 실패이고 <resp_msg> 태그가 없는 경우 원래 요청에 포함된 명령 중 하나가 실패한 것입니다. <resp_msg> 태그는 XML 문서 자체와 관련된 문제를 설명하는 데만 사용됩니다.

명령 응답

<cmd> 섹션 아래에 있는 <response> 섹션은 사용자에게 특정 명령의 성공이나 실패에 대해 알립니다. <status> 태그는 명령 성공 또는 실패를 보여줍니다. 전체 응답의 경우와 마찬가지로, 명령이 실패하면 요청의 <cmd> 섹션 콘텐츠의 형식이 잘못된 경우 <response> 섹션에는 <resp_msg> 태그만 포함됩니다. 그렇지 않은 경우 실패 상태는 명령이 실행된 객체 중 하나가 실패의 원인임을 의미합니다.

객체 응답

끝으로, <cmd> 섹션에 있는 각 <data> 섹션에도 <response> 섹션이 있습니다. 이 섹션은 특정 객체에 대해 실행 중인 명령의 성공 또는 실패를 보여줍니다. 응답 상태가 success일 경우, <response> 섹션에 <resp_msg> 태그가 없습니다. 상태가 FAILURE일 경우 객체에 대해 명령을 실행 중일 때 발생한 오류에 따라 <response> 필드에 <resp_msg> 태그가 하나 이상 표시됩니다. 객체 오류는 명령 실행 시 발견된 문제 또는 형식이 잘못되었거나 알 수 없는 객체로 인해 발생할 수 있습니다.

<data> 섹션은 <response> 섹션 이외에 다른 정보를 포함할 수 있습니다. 이 정보는 오류가 발생한 객체를 설명하는 수신 <data> 필드와 같은 형식으로 되어 있습니다. "[<data> 태그](#)" [25]를 참조하십시오. 이 추가 정보는 특히 다음과 같은 경우에 유용합니다.

- 명령이 특정 <data> 섹션에 대해 실패하지만 추가 <data> 섹션에 대해 성공하는 경우
- 빈 <data> 섹션이 명령에 전달되어 어떤 도메인에서는 실패하고 다른 도메인에서는 성공하는 경우

이벤트 메시지

폴링 대신 특정 상태가 변경될 경우 이벤트 알림을 수신하도록 가입할 수 있습니다. 개별적으로 또는 일괄적으로 가입할 수 있는 이벤트 유형에는 세 가지가 있습니다. 자세한 내용은 "[이벤트 유형](#)" [29]을 참조하십시오.

등록 및 등록 해제

<LDM_interface> 메시지를 사용하여 이벤트에 대해 등록할 수 있습니다. "[<LDM_interface> 태그](#)" [23]를 참조하십시오. <action> 태그는 등록 또는 등록을 해제할 이벤트의 유형을 자세히 설명하며 <data> 섹션은 비어 있습니다.

예 7 이벤트 등록 요청 메시지 예

```
<LDM_interface version="1.3">
  <cmd>
    <action>reg-domain-events</action>
    <data version="3.0"/>
  </cmd>
</LDM_interface>
```

논리적 도메인 Manager는 등록 또는 등록 해제가 성공적인지 여부를 나타내는 <LDM_interface> 응답 메시지로 응답합니다.

예 8 이벤트 등록 응답 메시지 예

```

<LDM_interface version="1.3">
  <cmd>
    <action>reg-domain-events</action>
    <data version="3.0"/>
      <response>
        <status>success</status>
      </response>
    </data>
    <response>
      <status>success</status>
    </response>
  </cmd>
  <response>
    <status>success</status>
  </response>
</LDM_interface>

```

각 이벤트 유형에 대한 작업 문자열은 이벤트 하위 섹션에 나열됩니다.

<LDM_event> 메시지

이벤트 메시지는 수신 <LDM_interface> 메시지와 같은 형식입니다. 단, 메시지의 시작 태그가 <LDM_event>라는 점만 다릅니다. 메시지의 <action> 태그는 이벤트 트리거를 위해 수행된 작업입니다. 메시지의 <data> 섹션은 이벤트와 연관된 객체에 대해 설명하는데, 세부사항은 발생한 이벤트의 유형에 따라 다릅니다.

예 9 <LDM_event> 알림 예

```

<LDM_event version='1.1'>
  <cmd>
    <action>Event command here</action>
    <data version='3.0'>
      <Envelope>
        <References/>
        <Content xsi:type='ovf:VirtualSystem_Type' ovf:id='ldg1' />
          <Section xsi:type="ovf:ResourceAllocationSection_type">
            <Item>
              <rasd:OtherResourceType>LDom Resource Type</rasd:OtherResourceType>
              <gprop:GenericProperty
                key="Property name">Property Value</gprop:GenericProperty>
            </Item>
          </Section>
        </Envelope>
      </data>
    </cmd>
  </LDM_event>

```

이벤트 유형

다음과 같은 이벤트 유형에 가입할 수 있습니다.

- 도메인 이벤트
- 하드웨어 이벤트
- 진행률 이벤트
- 리소스 이벤트

모든 이벤트는 `ldm` 하위 명령에 해당합니다.

도메인 이벤트

도메인 이벤트는 도메인에 대해 직접 수행할 수 있는 작업을 설명합니다. `<LDM_event>` 메시지의 `<action>` 태그에 다음 도메인 이벤트를 지정할 수 있습니다.

- `add-domain`
- `bind-domain`
- `domain-reset`
- `migrate-domain`
- `panic-domain`
- `remove-domain`
- `start-domain`
- `stop-domain`
- `unbind-domain`

해당 이벤트는 항상 OVF `<data>` 섹션에 이벤트가 발생한 도메인을 설명하는 `<Content>` 태그만 포함합니다. 도메인 이벤트에 대해 등록하려면 `<action>` 태그를 `reg-domain-events`로 설정하여 `<LDM_interface>` 메시지를 보내십시오. 해당 이벤트에 대해 등록을 해제하려면 `<action>` 태그를 `unreg-domain-events`로 설정하여 `<LDM_interface>` 메시지를 보내십시오.

하드웨어 이벤트

하드웨어 이벤트는 물리적 시스템 하드웨어 변경과 관련됩니다. Oracle VM Server for SPARC 소프트웨어의 경우 유일한 하드웨어 변경은 SP(서비스 프로세서) 구성을 추가, 제거 또는 설정할 때 SP에 대해 수행되는 변경입니다. 현재 이 유형의 이벤트는 세 가지입니다.

- `add-spconfig`
- `set-spconfig`

- remove-spconfig

하드웨어 이벤트는 항상 OVF <data> 섹션에 이벤트가 발생 중인 SP 구성을 설명하는 <Section> 태그만 포함합니다. 해당 이벤트에 대해 등록하려면 <action> 태그를 reg-hardware-events로 설정하여 <LDM_interface> 메시지를 보내십시오. 해당 이벤트에 대해 등록을 해제하려면 <action> 태그를 unreg-hardware-events로 설정하여 <LDM_interface> 메시지를 보내십시오.

진행률 이벤트

진행률 이벤트는 도메인 마이그레이션과 같이 장기 실행 중인 명령에 대해 발행됩니다. 이 이벤트는 명령 수명 기간 동안 진행된 정도를 보고합니다. 이때 migration-process 이벤트만 보고됩니다.

진행률 이벤트는 항상 OVF <data> 섹션에 이벤트의 영향을 받는 SP 구성을 설명하는 <Section> 태그만 포함합니다. 이 이벤트에 대해 등록하려면 <action> 태그를 reg-hardware-events로 설정하여 <LDM_interface> 메시지를 전송하십시오. 해당 이벤트에 대해 등록을 해제하려면 <action> 태그를 unreg-hardware-events로 설정하여 <LDM_interface> 메시지를 보내십시오.

진행률 이벤트의 <data> 섹션은 영향을 받는 도메인을 설명하는 <content> 섹션으로 구성됩니다. 이 <content> 섹션은 ldom_info <Section> 태그를 사용하여 진행률을 업데이트합니다. ldom_info 섹션에 표시되는 일반 등록 정보는 다음과 같습니다.

- --progress – 명령에 의해 수행된 진행률입니다.
- --status – 명령 상태를 나타내며, ongoing, failed, done 중 하나일 수 있습니다.
- --source – 진행률을 보고하는 시스템입니다.

리소스 이벤트

리소스 이벤트는 도메인에서 리소스가 추가, 제거 또는 변경될 경우에 발생합니다. 해당 이벤트 중 일부에 대한 <data> 섹션에는 OVF <data> 섹션에 서비스 이름을 제공하는 <Section> 태그와 함께 <Content> 태그가 포함됩니다.

<LDM_event> 메시지의 <action> 태그에 다음 이벤트를 지정할 수 있습니다.

- add-vdiskserverdevice
- remove-vdiskserverdevice
- set-vdiskserverdevice
- remove-vdiskserver
- set-vconscon
- remove-vconscon
- set-vswitch

- remove-vswitch

다음 리소스 이벤트는 항상 OVF <data> 섹션에 이벤트가 발생한 도메인을 설명하는 <Content> 태그만 포함합니다.

- add-vcpu
- add-crypto
- add-memory
- add-io
- add-variable
- add-vconscn
- add-vdisk
- add-vdiskserver
- add-vnet
- add-vsan
- add-vswitch
- remove-crypto
- remove-io
- remove-memory
- remove-variable
- remove-vcpu
- remove-vdisk
- remove-vnet
- set-crypto
- set-memory
- set-variable
- set-vconsole
- set-vcpu
- set-vdisk
- set-vnet

리소스 이벤트에 대해 등록하려면 <action> 태그를 reg-resource-events로 설정하여 <LDM_interface> 메시지를 보내십시오. 해당 이벤트에 대해 등록을 해제하려면 <action> 태그가 unreg-resource-events로 설정된 <LDM_interface> 메시지가 필요합니다.

모든 이벤트

각 이벤트에 대해 개별적으로 등록할 필요 없이 세 가지 유형의 이벤트 전체를 수신하도록 등록할 수도 있습니다. 세 가지 모든 유형의 이벤트에 대해 동시에 등록하려면 <action> 태그를

reg-all-events로 설정하여 <LDM_interface> 메시지를 보내십시오. 해당 이벤트에 대해 등록을 해제하려면 <action> 태그가 unreg-all-events로 설정된 <LDM_interface> 메시지가 필요합니다.

논리적 도메인 Manager 작업

<action> 태그에 지정된 명령(--events 명령 제외)은 ldm 명령줄 인터페이스에 지정된 명령에 해당합니다. ldm 하위 명령에 대한 자세한 내용은 [ldm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

주 - XML 인터페이스는 논리적 도메인 Manager CLI에서 지원하는 동사 또는 명령 별명을 지원하지 않습니다.

<action> 태그에 지원되는 문자열은 다음과 같습니다.

- add-domain
- add-io
- add-mau
- add-memory
- add-sponfig
- add-variable
- add-vconscon
- add-vcpu
- add-vdisk
- add-vdiskserver
- add-vdiskserverdevice
- add-vhba
- add-vnet
- add-vsant
- add-vswitch
- bind-domain
- cancel-operation
- list-bindings
- list-constraints
- list-dependencies
- list-devices
- list-domain
- list-hba
- list-rsrc-group
- list-services

- list-spconfig
- list-variable
- list-vmapi
- migrate-domain
- reg-all-events
- reg-domain-events
- reg-hardware-events
- reg-resource-events
- remove-domain
- remove-io
- remove-mau
- remove-memory
- remove-reconf
- remove-spconfig
- remove-variable
- remove-vconscon
- remove-vcpu
- remove-vdisk
- remove-vdiskserver
- remove-vdiskserverdevice
- remove-vhba
- remove-vmapi
- remove-vnet
- remove-vsan
- remove-vswitch
- rescan-vhba
- set-domain
- set-mau
- set-memory
- set-spconfig
- set-variable
- set-vconscon
- set-vconsole
- set-vcpu
- set-vhba
- set-vmapi
- set-vnet
- set-vswitch

- start-domain
- stop-domain
- unbind-domain
- unreg-all-events
- unreg-domain-events
- unreg-hardware-events
- unreg-resource-events

논리적 도메인 Manager 리소스 및 등록 정보

이 절에서는 논리적 도메인 Manager 리소스 및 해당 리소스 각각에 대해 정의할 수 있는 등록 정보의 예를 제공합니다. 리소스와 등록 정보는 XML 예제에서 굵게 표시됩니다. 이 예는 바인드 출력이 아닌 리소스를 보여줍니다. 도메인 마이그레이션 출력을 제외하고 제약 조건 출력은 논리적 도메인 Manager 작업에 대한 입력을 만드는 데 사용할 수 있습니다. [“도메인 마이그레이션” \[48\]](#)을 참조하십시오. 각 리소스는 <Section> OVF 섹션에 정의되며 <rasd:OtherResourceType> 태그로 지정됩니다.

도메인 정보(ldom_info) 리소스

다음 예에서는 ldom_info 리소스의 선택적 등록 정보를 보여줍니다.

예 10 ldom_info XML 출력 예

다음 예에서는 여러 ldom_info 등록 정보(예: uuid, hostid 및 Address)에 대해 지정된 값을 보여줍니다.

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="primary">
    <Section xsi:type="ovf:ResourceAllocationSection_type">
      <Item>
        <rasd:OtherResourceType>ldom_info</rasd:OtherResourceType>
        <uuid>c2c3d93b-a3f9-60f6-a45e-f35d55c05fb6</uuid>
        <rasd:Address>00:03:ba:d8:ba:f6</rasd:Address>
        <gprop:GenericProperty key="hostid">83d8baf6</gprop:GenericProperty>
        <gprop:GenericProperty key="master">plum</gprop:GenericProperty>
        <gprop:GenericProperty key="failure-policy">reset</gprop:GenericProperty>
        <gprop:GenericProperty key="extended-mapin-space">on</gprop:GenericProperty>
        <gprop:GenericProperty key="progress">45%</gprop:GenericProperty>
        <gprop:GenericProperty key="status">ongoing</gprop:GenericProperty>
        <gprop:GenericProperty key="source">system1</gprop:GenericProperty>
        <gprop:GenericProperty key="rc-add-policy"></gprop:GenericProperty>
        <gprop:GenericProperty key="perf-counters">global</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

```

    </Section>
  </Content>
</Envelope>

```

ldom_info 리소스는 항상 <Content> 섹션 내에 있습니다. ldom_info 리소스 내에 있는 다음 등록 정보는 선택적 등록 정보입니다.

- <uuid> 태그 - 도메인의 UUID를 지정합니다.
 - <rasd:Address> 태그 - 도메인에 지정할 MAC 주소를 지정합니다.
 - <gprop:GenericProperty key="extended-mapin-space"> 태그 - 도메인에 대한 확장 Mapin 공간을 사용으로 설정할지(on) 아니면 사용 안함으로 설정할지(off) 지정합니다. 기본값은 off입니다.
 - <gprop:GenericProperty key="failure-policy"> 태그 - 마스터 도메인에서 실패가 발생할 경우 슬레이브 도메인의 작동 방식을 지정합니다. 기본값은 ignore입니다. 유효한 등록 정보 값은 다음과 같습니다.
 - ignore - 마스터 도메인의 오류를 무시합니다(슬레이브 도메인이 영향을 받지 않음).
 - panic으로 설정하면 마스터 도메인이 실패할 경우 슬레이브 도메인에 패닉이 발생합니다.
 - reset으로 설정하면 마스터 도메인이 실패할 경우 슬레이브 도메인이 재설정됩니다.
 - stop으로 설정하면 마스터 도메인이 실패할 경우 슬레이브 도메인이 중지됩니다.
 - <gprop:GenericProperty key="hostid"> 태그 - 도메인에 지정할 호스트 ID를 지정합니다.
 - <gprop:GenericProperty key="master"> 태그 - 마스터 도메인 이름을 심프로 구분하여 4개까지 지정합니다.
 - <gprop:GenericProperty key="progress"> 태그 - 명령에 의해 수행된 진행률을 지정합니다.
 - <gprop:GenericProperty key="source"> 태그 - 명령의 진행률에 대해 보고하는 시스템을 지정합니다.
 - <gprop:GenericProperty key="status"> 태그 - 명령의 상태(done, failed 또는 ongoing)를 지정합니다.
 - <gprop:GenericProperty key="rc-add-policy"> 태그 - 지정된 도메인에 추가될 수 있는 루트 컴플렉스에서 직접 I/O 및 SR-IOV I/O 가상화 작업을 사용으로 설정할지 아니면 사용 안함으로 설정할지 지정합니다. 유효한 값은 iov 및 값 없음(rc-add-policy=)입니다.
 - <gprop:GenericProperty key="perf-counters"> 태그 - 액세스할 성능 레지스터 세트(global, htstrand, strand)를 지정합니다.
- 플랫폼에 성능 액세스 기능이 없으면 perf-counters 등록 정보 값이 무시됩니다.

CPU(cpu) 리소스

cpu 리소스에 대한 할당 단위 등록 정보 <rasd:AllocationUnits>는 항상 코어 수가 아닌 가상 CPU 수를 지정합니다.

cpu 리소스는 항상 <Content> 섹션 내에 있습니다.

예 11 `ldm list-bindings` 명령의 `cpu` XML 섹션 출력

다음 예에서는 `ldm list-bindings` 명령을 사용하여 `<cpu>` 섹션의 XML 출력을 보여줍니다.

```
<?xml version="1.0"?>
<LDM_interface
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ovf="./schemas/envelope"
xmlns:rasd="./schemas/CIM_ResourceAllocationSettingData"
xmlns:vssd="./schemas/CIM_VirtualSystemSettingData"
xmlns:gprop="./schemas/GenericProperty"
xmlns:bind="./schemas/Binding"
version="1.3"
xsi:noNamespaceSchemaLocation="./schemas/combined-v3.xsd">
  <cmd>
    <action>list-bindings</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="primary">
          <Section xsi:type="ovf:ResourceAllocationSection_Type">
            <Item>
              <rasd:OtherResourceType>ldom_info</rasd:OtherResourceType>
              <uuid>1e04cbbd-472a-e8b9-ba4c-d3eee86e7725</uuid>
              <rasd:Address>00:21:28:f5:11:6a</rasd:Address>
              <gprop:GenericProperty key="hostid">0x8486632a</gprop:GenericProperty>
              <failure-policy>fff</failure-policy>
              <wcore>0</wcore>
              <extended-mapin-space>0</extended-mapin-space>
              <cpu-arch>native</cpu-arch>
              <rc-add-policy/>
              <gprop:GenericProperty key="state">active</gprop:GenericProperty>
            </Item>
          </Section>
          <Section xsi:type="ovf:VirtualHardwareSection_Type">
            <Item>
              <rasd:OtherResourceType>cpu</rasd:OtherResourceType>
              <rasd:AllocationUnits>8</rasd:AllocationUnits>
              <bind:Binding>
                <Item>
                  <rasd:OtherResourceType>cpu</rasd:OtherResourceType>
                  <gprop:GenericProperty key="vid">0</gprop:GenericProperty>
                  <gprop:GenericProperty key="pid">0</gprop:GenericProperty>
                  <gprop:GenericProperty key="cid">0</gprop:GenericProperty>
                  <gprop:GenericProperty key="strand_percent">100</gprop:GenericProperty>
                  <gprop:GenericProperty key="util_percent">1.1%</gprop:GenericProperty>
                  <gprop:GenericProperty key="normalized_utilization">0.1%</gprop:
GenericProperty>
                </Item>
              </Section>
            </Content>
          </Envelope>
        </data>
      </cmd>
    </LDM_interface>
```

예 12 `ldm list-domain` 명령의 `cpu` XML 섹션 출력

다음 예에서는 `ldm list-domain` 명령을 사용하여 `<cpu>` 섹션의 XML 출력을 보여줍니다.

```
<?xml version="1.0"?>
```

```

<LDM_interface
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ovf="./schemas/envelope"
xmlns:rasd="./schemas/CIM_ResourceAllocationSettingData"
xmlns:vssd="./schemas/CIM_VirtualSystemSettingData"
xmlns:gprop="./schemas/GenericProperty"
xmlns:bind="./schemas/Binding"
version="1.3"
xsi:noNamespaceSchemaLocation="./schemas/combined-v3.xsd">
  <cmd>
    <action>list-domain</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="primary">
          <Section xsi:type="ovf:ResourceAllocationSection_Type">
            <Item>
              <rasd:OtherResourceType>ldom_info</rasd:OtherResourceType>
              <gprop:GenericProperty key="state">active</gprop:GenericProperty>
              <gprop:GenericProperty key="flags">-n-cv-</gprop:GenericProperty>
              <gprop:GenericProperty key="utilization">0.7%</gprop:GenericProperty>
              <gprop:GenericProperty key="uptime">3h</gprop:GenericProperty>
              <gprop:GenericProperty key="normalized_utilization">0.1%</gprop:
GenericProperty>
            </Item>
          </Section>
        </Content>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>

```

MAU(mau) 리소스

mau 리소스는 항상 <Content> 섹션 내에 있습니다. 유일한 등록 정보는 MAU 또는 다른 암호화 단위의 개수를 나타내는 <rasd:AllocationUnits> 태그입니다.

주 - mau 리소스는 지원되는 서버에서 지원되는 암호화 단위입니다. 현재 지원되는 두 개의 암호화 단위는 MAU(Modular Arithmetic Unit) 및 CWQ(Control Word Queue)입니다.

예 13 mau XML 예

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>mau</rasd:OtherResourceType>
        <rasd:AllocationUnits>1</rasd:AllocationUnits>
      </Item>
    </Section>
  </Content>
</Envelope>

```

메모리(memory) 리소스

memory 리소스는 항상 <Content> 섹션 내에 있습니다. 유일한 등록 정보는 메모리의 양을 나타내는 <rasd:AllocationUnits> 태그입니다.

예 14 memory XML 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>memory</rasd:OtherResourceType>
        <rasd:AllocationUnits>4G</rasd:AllocationUnits>
      </Item>
    </Section>
  </Content>
</Envelope>
```

가상 SAN(vsan) 리소스

가상 SAN(vsan) 리소스는 <Content> 섹션에 포함될 수 있습니다. 다음 키를 포함하는 <gprop:GenericProperty> 태그를 사용해야 합니다.

- service_name – 가상 SAN의 이름
- vsan_iport – 가상 SAN의 개시자 포트

vsan_volume 섹션에 단일 vol_name 등록 정보를 포함하는 항목이 있어야 합니다. 이 vol_name 등록 정보는 *로 설정해야 합니다.

예 15 vsan XML 예

```
<Envelope>
  <References/>
  <Section xsi:type="ovf:VirtualHardwareSection_Type">
    <Item>
      <rasd:OtherResourceType>vsan</rasd:OtherResourceType>
      <gprop:GenericProperty key="service_name">vs1</gprop:GenericProperty>
      <gprop:GenericProperty key="vsan_iport">
        /pci@400/pci@1/pci@0/pci@8/SUNW,emlxs@0/fp@0,0</gprop:GenericProperty>
      <bind:Binding>
        <Item>
          <rasd:OtherResourceType>vsan_volume</rasd:OtherResourceType>
          <gprop:GenericProperty key="vol_name">*</gprop:GenericProperty>
        </Item>
      </bind:Binding>
    </Item>
  </Section>
</Envelope>
```

가상 디스크 서버(vds) 리소스

가상 디스크 서버(vds) 리소스는 도메인 설명의 일부로 <Content> 섹션에 표시되거나, <Envelope> 섹션에 자체적으로 표시될 수 있습니다. 유일한 등록 정보는 설명할 vds 리소스의 이름을 포함하는 service_name 키가 있는 <gprop:GenericProperty> 태그입니다.

예 16 vds XML 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>vds</rasd:OtherResourceType>
        <gprop:GenericProperty
          key="service_name">vdstmp</gprop:GenericProperty>
        </Item>
      </Section>
    </Content>
  </Envelope>
```

가상 디스크 서버 볼륨(vds_volume) 리소스

vds_volume 리소스는 도메인 설명의 일부로 <Content> 섹션에 표시되거나, <Envelope> 섹션에 자체적으로 표시될 수 있습니다. 다음 키를 포함하는 <gprop:GenericProperty> 태그가 있어야 합니다.

- vol_name – 볼륨 이름입니다.
- service_name – 이 볼륨이 바인드될 가상 디스크 서버의 이름입니다.
- block_dev – 이 볼륨과 연관될 파일 또는 장치 이름입니다.

선택적으로 vds_volume 리소스는 다음과 같은 등록 정보를 포함할 수도 있습니다.

- vol_opts – {ro, slice, excl}과 같이 한 문자열 내에 쉼표로 구분된 하나 이상의 옵션입니다.
- mpgroup – 다중 경로(페일오버) 그룹의 이름입니다.

예 17 vds_volume XML 예

```
<Envelope>
  <References/>
  <Section xsi:type="ovf:VirtualHardwareSection_Type">
    <Item>
      <rasd:OtherResourceType>vds_volume</rasd:OtherResourceType>
      <gprop:GenericProperty key="vol_name">vdsdev0</gprop:GenericProperty>
      <gprop:GenericProperty key="service_name">primary-vds0</gprop:GenericProperty>
      <gprop:GenericProperty key="block_dev">
        opt/SUNWldm/domain_disks/testdisk1</gprop:GenericProperty>
    </Item>
  </Section>
</Envelope>
```

```

        <gprop:GenericProperty key="vol_opts">ro</gprop:GenericProperty>
        <gprop:GenericProperty key="mpgroup">mpgroup-name</gprop:GenericProperty>
    </Item>
</Section>
</Envelope>

```

디스크(disk) 리소스

disk 리소스는 항상 <Content> 섹션 내에 있습니다. 다음 키를 포함하는 <gprop:GenericProperty> 태그가 있어야 합니다.

- vdisk_name – 가상 디스크의 이름입니다.
- service_name – 이 가상 디스크가 바인드될 가상 디스크 서버의 이름입니다.
- vol_name – 이 가상 디스크와 연관될 가상 디스크 서비스 장치입니다.

선택적으로 disk 리소스는 timeout 등록 정보를 포함할 수도 있습니다. 이 등록 정보는 가상 디스크 클라이언트(vdc)와 가상 디스크 서버(vds) 간에 연결을 설정하기 위한 시간 초과 값(초)입니다. 가상 디스크(vdisk) 경로가 여러 개 있을 경우 vdc가 다른 vds에 대한 연결을 시도할 수 있습니다. 시간 초과는 지정된 시간 이내에 vds에 대한 연결이 설정되도록 합니다.

예 18 disk XML 예

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>disk</rasd:OtherResourceType>
        <gprop:GenericProperty key="vdisk_name">vdisk0</gprop:GenericProperty>
        <gprop:GenericProperty key="service_name">primary-vds0</gprop:GenericProperty>
        <gprop:GenericProperty key="vol_name">vdsdev0</gprop:GenericProperty>
        <gprop:GenericProperty key="timeout">60</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>

```

가상 스위치(vsw) 리소스

vsw 리소스는 도메인 설명의 일부로 <Content> 섹션에 표시되거나, <Envelope> 섹션에 자체적으로 표시될 수 있습니다. 가상 스위치에 지정될 이름인 service_name 키를 포함하는 <gprop:GenericProperty> 태그가 있어야 합니다.

선택적으로 vsw 리소스는 다음과 같은 등록 정보를 포함할 수도 있습니다.

- <rasd:Address> – 가상 스위치에 MAC 주소를 지정합니다.

- `default-vlan-id` – 태그 지정 모드에서 가상 네트워크 장치 또는 가상 스위치가 멤버로 속해 있어야 하는 기본 VLAN(Virtual Local Area Network) 을 지정합니다. 첫번째 VLAN ID(`vid1`)는 `default-vlan-id`용으로 예약되었습니다.
- `dev_path` – 이 가상 스위치와 연관될 네트워크 장치의 경로입니다.
- `id` – 새 가상 스위치 장치의 ID를 지정합니다. 기본적으로 ID 값은 자동으로 생성되므로, OS의 기존 장치 이름과 일치해야 하는 경우 이 등록 정보를 설정하십시오.
- `inter_vnet_link` – Vnet 간 통신을 위해 LDC 채널을 지정할지 여부를 지정합니다. 유효한 값은 `on`, `off` 및 `auto`입니다. 기본값은 `auto`입니다.
- `linkprop` – 가상 장치가 물리적 링크 상태 업데이트를 가져오도록 지정합니다(기본값: `phys-state`). 값이 비어 있을 경우, 가상 장치가 물리적 링크 상태 업데이트를 가져오지 않습니다.
- `mode` – Oracle Solaris Cluster 하트비트 지원의 경우 `sc`입니다.
- `pvid` – 포트 VLAN(Virtual Local Area Network) ID(식별자)는 태그 미지정 모드에서 가상 네트워크가 멤버로 속해 있어야 하는 VLAN을 나타냅니다.
- `mtu` – 가상 스위치나 가상 스위치에 바인드된 가상 네트워크 장치 또는 둘 다의 MTU(최대 전송 단위)를 지정합니다. 유효한 값의 범위는 1500-16000입니다. 잘못된 값이 지정된 경우 `ldm` 명령이 오류를 발행합니다.
- `vid` – VLAN(Virtual Local Area Network) ID(식별자)는 태그 지정 모드에서 가상 네트워크와 가상 스위치가 멤버로 속해 있어야 하는 VLAN을 나타냅니다.

예 19 vsw XML 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg2">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>vsw</rasd:OtherResourceType>
        <rasd:Address>00:14:4f:fb:ec:00</rasd:Address>
        <gprop:GenericProperty key="service_name">test-vsw1</gprop:GenericProperty>
        <gprop:GenericProperty key="inter_vnet_link">auto</gprop:GenericProperty>
        <gprop:GenericProperty key="default-vlan-id">1</gprop:GenericProperty>
        <gprop:GenericProperty key="pvid">1</gprop:GenericProperty>
        <gprop:GenericProperty key="mtu">1500</gprop:GenericProperty>
        <gprop:GenericProperty key="dev_path">switch00</gprop:GenericProperty>
        <gprop:GenericProperty key="id">0</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

네트워크(network) 리소스

network 리소스는 항상 <Content> 섹션 내에 있습니다. 다음 키를 포함하는 <gprop:GenericProperty> 태그가 있어야 합니다.

- `linkprop` – 가상 장치가 물리적 링크 상태 업데이트를 가져오도록 지정합니다(기본값: `phys-state`). 값이 비어 있을 경우, 가상 장치가 물리적 링크 상태 업데이트를 가져오지 않습니다.

- `vnet_name` – 가상 네트워크(`vnet`)의 이름입니다.
- `service_name` – 이 가상 네트워크가 바인드될 가상 스위치(`vswitch`)의 이름입니다.
- `custom` – 신뢰할 수 있는 호스트에서 가상 네트워크 장치에 지정될 수 있는 최대 VLAN 및 MAC 주소 수에 대한 사용자정의 설정을 사용으로 설정할지 아니면 사용 안함으로 설정할지 지정합니다. 기본값은 `disable`입니다.
- `custom/max-mac-addr`s – 신뢰할 수 있는 호스트에서 가상 네트워크 장치에 지정될 수 있는 최대 MAC 주소 수를 지정합니다. 기본값은 4096입니다.
- `custom/max-vlans` – 신뢰할 수 있는 호스트에서 가상 네트워크 장치에 지정될 수 있는 최대 VLAN 수를 지정합니다. 기본값은 4096입니다.

선택적으로 `network` 리소스는 다음과 같은 등록 정보를 포함할 수도 있습니다.

- `<rasd:Address>` – 가상 스위치에 MAC 주소를 지정합니다.
- `pvid` – 포트 VLAN(Virtual Local Area Network) ID(식별자)는 태그 미지정 모드에서 가상 네트워크가 멤버로 속해 있어야 하는 VLAN을 나타냅니다.
- `vid` – VLAN(Virtual Local Area Network) ID(식별자)는 태그 지정 모드에서 가상 네트워크와 가상 스위치가 멤버로 속해 있어야 하는 VLAN을 나타냅니다.

예 20 `network` XML 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>network</rasd:OtherResourceType>
        <gprop:GenericProperty key="linkprop">phys-state</gprop:GenericProperty>
        <gprop:GenericProperty key="custom">enable</gprop:GenericProperty>
        <gprop:GenericProperty key="custom/max-mac-addr">4096</gprop:GenericProperty>
        <gprop:GenericProperty key="custom/max-vlans">12</gprop:GenericProperty>
        <gprop:GenericProperty key="vnet_name">ldg1-vnet0</gprop:GenericProperty>
        <gprop:GenericProperty key="service_name">primary-vsw0</gprop:GenericProperty>
        <rasd:Address>00:14:4f:fc:00:01</rasd:Address>
      </Item>
    </Section>
  </Content>
</Envelope>
```

가상 콘솔 집중기(`vcc`) 리소스

`vcc` 리소스는 도메인 설명의 일부로 `<Content>` 섹션에 표시되거나, `<Envelope>` 섹션에 자체적으로 표시될 수 있습니다. 다음 키를 포함하는 `<gprop:GenericProperty>` 태그가 있을 수 있습니다.

- `service_name` – 가상 콘솔 집중기 서비스에 지정할 이름입니다.
- `min_port` – 이 `vcc`와 연관될 최소 포트 번호입니다.
- `max_port` – 이 `vcc`와 연관될 최대 포트 번호입니다.

예 21 vcc XML 예

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>vcc</rasd:OtherResourceType>
        <gprop:GenericProperty key="service_name">vcc1</gprop:GenericProperty>
        <gprop:GenericProperty key="min_port">6000</gprop:GenericProperty>
        <gprop:GenericProperty key="max_port">6100</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>

```

변수(var) 리소스

var 리소스는 항상 <Content> 섹션 내에 있습니다. 다음 키를 포함하는 <gprop:GenericProperty> 태그가 있을 수 있습니다.

- name – 변수의 이름입니다.
- value – 변수 값입니다.

예 22 var XML 예

```

<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>var</rasd:OtherResourceType>
        <gprop:GenericProperty key="name">test_var</gprop:GenericProperty>
        <gprop:GenericProperty key="value">test1</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>

```

물리적 I/O 장치(physio_device) 리소스

physio_device 리소스는 항상 <Content> 섹션 내에 있습니다. add-io, set-io, remove-io, create-vf, destroy-vf 및 set-domain 하위 명령을 사용하여 이 리소스를 수정할 수 있습니다.

예 23 physio_device XML 예

다음 예에서는 가상 기능, 물리적 기능 및 루트 컴플렉스에 대해 작업을 수행하는 방법을 보여줍니다.

- 다음 XML 단편 예에서는 `ldm add-io` 명령을 사용하여 `/SYS/MB/NET0/IOVNET.PF0.VF0` 가상 기능을 `ldg1` 도메인에 추가하는 방법을 보여줍니다.

```
<LDM_interface version="1.3">
  <cmd>
    <action>add-io</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="ldg1">
          <Section xsi:type="ovf:VirtualHardwareSection_Type">
            <Item>
              <rasd:OtherResourceType>physio_device</rasd:OtherResourceType>
              <gprop:GenericProperty key="name">
                /SYS/MB/NET0/IOVNET.PF0.VF0</gprop:GenericProperty>
            </Item>
          </Section>
        </Content>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>
```

- 다음 XML 단편 예에서는 `ldm set-io` 명령을 사용하여 `pci_1` 루트 컴플렉스에 대한 `iov_bus_enable_iov` 등록 정보 값을 `on`으로 설정하는 방법을 보여줍니다.

```
<LDM_interface version="1.3">
  <cmd>
    <action>set-io</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Section xsi:type="ovf:VirtualHardwareSection_Type">
          <Item>
            <rasd:OtherResourceType>physio_device</rasd:OtherResourceType>
            <gprop:GenericProperty key="name">pci_1</gprop:GenericProperty>
            <gprop:GenericProperty key="iov_bus_enable_iov">
              on</gprop:GenericProperty>
          </Item>
        </Section>
      </Envelope>
    </data>
```

```
</cmd>
</LDM_interface>
```

- 다음 XML 단편 예에서는 `ldm set-io` 명령을 사용하여 `/SYS/MB/NET0/IOVNET.PF1` 물리적 기능에 대한 `unicast-slots` 등록 정보 값을 6으로 설정하는 방법을 보여줍니다.

```
<LDM_interface version="1.3">
  <cmd>
    <action>set-io</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Section xsi:type="ovf:VirtualHardwareSection_Type">
          <Item>
            <rasd:OtherResourceType>physio_device</rasd:OtherResourceType>
            <gprop:GenericProperty key="name">
              /SYS/MB/NET0/IOVNET.PF1</gprop:GenericProperty>
            <gprop:GenericProperty key="unicast-slots">6</gprop:GenericProperty>
          </Item>
        </Section>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>
```

- 다음 XML 단편 예에서는 `ldm create-vf` 명령을 사용하여 다음 등록 정보 값으로 `/SYS/MB/NET0/IOVNET.PF1.VF0` 가상 기능을 만드는 방법을 보여줍니다.

- `unicast-slots=6`
- `pvid=3`
- `mtu=1600`

```
<LDM_interface version="1.3">
  <cmd>
    <action>create-vf</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Section xsi:type="ovf:VirtualHardwareSection_Type">
          <Item>
            <rasd:OtherResourceType>vf_device</rasd:OtherResourceType>
            <gprop:GenericProperty key="iov_pf_name">
              /SYS/MB/NET0/IOVNET.PF1</gprop:GenericProperty>
            <gprop:GenericProperty key="unicast-slots">6</gprop:GenericProperty>
            <gprop:GenericProperty key="pvid">3</gprop:GenericProperty>
            <gprop:GenericProperty key="mtu">1600</gprop:GenericProperty>
          </Item>
        </Section>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>
```

```

        </Section>
    </Envelope>
</data>
</cmd>
</LDM_interface>

```

- 다음 XML 단편 예에서는 `ldm create-vf` 명령을 사용하여 `/SYS/MB/NET0/IOVNET.PF1` 물리적 기능과 함께 `iov_pf_repeat_count_str` 값(3)으로 지정된 개수의 가상 기능을 만드는 방법을 보여줍니다. `iov_pf_repeat_count_str` 등록 정보로 여러 가상 기능을 만들 때 다른 등록 정보 값은 지정할 수 없습니다.

```

<LDM_interface version="1.3">
  <cmd>
    <action>create-vf</action>
    <data version="3.0">
      <Envelope>
        <References/>
        <Section xsi:type="ovf:VirtualHardwareSection_Type">
          <Item>
            <rasd:OtherResourceType>vf_device</rasd:OtherResourceType>
            <gprop:GenericProperty key="iov_pf_name">
              /SYS/MB/NET0/IOVNET.PF1</gprop:GenericProperty>
            <gprop:GenericProperty key="iov_pf_repeat_count_str">
              3</gprop:GenericProperty>
          </Item>
        </Section>
      </Envelope>
    </data>
  </cmd>
</LDM_interface>

```

SP 구성(spconfig) 리소스

SP(서비스 프로세서) 구성(spconfig) 리소스는 항상 `<Envelope>` 섹션에 자체적으로 표시됩니다. 다음 키를 포함하는 `<gprop:GenericProperty>` 태그가 있을 수 있습니다.

- `spconfig_name` – SP에 저장될 구성의 이름입니다.
- `spconfig_status` – 특정 SP 구성의 현재 상태입니다. 이 등록 정보는 `ldm list-spconfig` 명령의 출력에 사용됩니다.

예 24 spconfig XML 예

```

<Envelope>
  <Section xsi:type="ovf:ResourceAllocationSection_type">
    <Item>

```

```

    <rasd:OtherResourceType>spconfig</rasd:OtherResourceType>
    <gprop:GenericProperty
      key="spconfig_name">primary</gprop:GenericProperty>
    <gprop:GenericProperty
      key="spconfig_status">current</gprop:GenericProperty>
    <gprop:GenericProperty
      key="spconfig_aux_status">degraded</gprop:GenericProperty>
  </Item>
</Section>
</Envelope>

```

DRM 정책 구성(policy) 리소스

DRM 정책(policy) 리소스는 <Envelope> 섹션에 표시되며 다음 키를 포함하는 <gprop:GenericProperty> 태그가 있을 수 있습니다.

- policy_name – DRM 정책의 이름입니다.
- policy_enable – DRM 정책을 사용 또는 사용 안함으로 설정할지 여부를 지정합니다.
- policy_priority – DRM 정책의 우선 순위입니다.
- policy_vcpu_min – 도메인의 최소 가상 CPU 리소스 수입니다.
- policy_vcpu_max – 도메인의 최대 가상 CPU 리소스 수입니다.
- policy_util_lower – 정책 분석이 트리거되는 하한 사용량 레벨입니다.
- policy_util_upper – 정책 분석이 트리거되는 상한 사용량 레벨입니다.
- policy_tod_begin – DRM 정책의 유효 시작 시간입니다.
- policy_tod_end – DRM 정책의 유효 중지 시간입니다.
- policy_sample_rate – 초 단위의 주기 시간인 샘플 비율입니다.
- policy_elastic_margin – CPU 사용량의 상한 및 하한 간 버퍼 양입니다.
- policy_attack – 한 리소스 제어 주기 중에 추가될 최대 리소스 양입니다.
- policy_decay – 한 리소스 제어 주기 중에 제거될 최대 리소스 양입니다.

예 25 policy XML 예

```

<Envelope>
  <Section xsi:type="ovf:VirtualHardwareSection_Type">
    <Item>
      <rasd:OtherResourceType>policy</rasd:OtherResourceType>
      <gprop:GenericProperty key="policy_name">test-policy</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_enable">on</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_priority">1</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_vcpu_min">12</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_vcpu_max">13</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_util_lower">8</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_util_upper">9</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_tod_begin">07:08:09</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_tod_end">09:08:07</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_sample_rate">1</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_elastic_margin">8</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_attack">8</gprop:GenericProperty>
      <gprop:GenericProperty key="policy_decay">9</gprop:GenericProperty>
    </Item>
  </Section>
</Envelope>

```

```
</Section>
</Envelope>
```

콘솔(console) 리소스

console 리소스는 항상 <Content> 섹션에 포함됩니다. 다음 키를 포함하는 <gprop:GenericProperty> 태그가 있을 수 있습니다.

- port – 이 가상 콘솔(console)을 변경할 포트입니다.
- service_name – 이 console을 바인드할 가상 콘솔 집중기(vcc) 서비스입니다.
- group – 이 console을 바인드할 그룹의 이름입니다.
- enable-log – 이 콘솔에 대한 가상 콘솔 로깅을 사용 또는 사용 안함으로 설정합니다.

예 26 console XML 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" id="ldg1">
    <Section xsi:type="ovf:VirtualHardwareSection_Type">
      <Item>
        <rasd:OtherResourceType>console</rasd:OtherResourceType>
        <gprop:GenericProperty key="port">6000</gprop:GenericProperty>
        <gprop:GenericProperty key="service_name">vcc2</gprop:GenericProperty>
        <gprop:GenericProperty key="group">group-name</gprop:GenericProperty>
        <gprop:GenericProperty key="enable-log">on</gprop:GenericProperty>
      </Item>
    </Section>
  </Content>
</Envelope>
```

도메인 마이그레이션

이 예에서는 ldm migrate-domain 명령의 <data> 섹션에 포함된 내용을 보여줍니다.

- 첫번째 <Content> 노드(<ldom_info> 섹션 없음)는 마이그레이션할 소스 도메인입니다.
- 두번째 <Content> 노드(<ldom_info> 섹션 있음)는 마이그레이션할 대상 도메인입니다. 소스 및 대상 도메인 이름은 같아도 됩니다.
- 대상 도메인의 <ldom_info> 섹션은 마이그레이션할 시스템 및 해당 시스템으로 마이그레이션하는 데 필요한 세부정보를 설명합니다.
 - target-host - 마이그레이션할 대상 시스템입니다.
 - user-name - 대상 시스템에 대한 로그인 사용자 이름으로, SASL 64비트로 인코딩되어야 합니다.
 - password - 대상 시스템에 로그인하는 데 사용할 암호로, SASL 64비트로 인코딩되어야 합니다.

주 - 논리적 도메인 Manager는 대상 사용자 이름과 암호를 디코딩하는 데 `sas1_decode64()`를 사용하고, 이러한 값을 인코딩하는 데 `sas1_encode64()`를 사용합니다. SASL 64 인코딩은 base64 인코딩에 해당합니다.

예 27 migrate-domain <data> 섹션의 예

```
<Envelope>
  <References/>
  <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="ldg1"/>
  <Content xsi:type="ovf:VirtualSystem_Type" ovf:id="ldg1"/>
  <Section xsi:type="ovf:ResourceAllocationSection_Type">
    <Item>
      <rasd:OtherResourceType>ldom_info</rasd:OtherResourceType>
      <gprop:GenericProperty key="target">target-host</gprop:GenericProperty>
      <gprop:GenericProperty key="username">user-name</gprop:GenericProperty>
      <gprop:GenericProperty key="password">password</gprop:GenericProperty>
    </Item>
  </Section>
</Content>
</Envelope>
```

XML 스키마

논리적 도메인 Manager에 사용되는 XML 스키마는 `/opt/SUNWldm/bin/schemas` 디렉토리에 있습니다. 파일 이름은 다음과 같습니다.

- `cim-common.xsd` - `cim-common.xsd` 스키마
- `cim-rasd.xsd` - `cim-rasd.xsd` 스키마
- `cim-vssd.xsd` - `cim-vssd.xsd` 스키마
- `cli-list-constraint-v3.xsd` - `cli-list-constraint-v3.xsd` 스키마
- `combined-v3.xsd` - `LDM_interface` XML 스키마
- `event-v3.xsd` - `LDM_Event` XML 스키마
- `ldmd-binding.xsd` - `Binding_Type` XML 스키마
- `ldmd-property.xsd` - `GenericProperty` XML 스키마
- `ovf-core.xsd` - `ovf-core.xsd` 스키마
- `ovf-envelope.xsd` - `ovf-envelope.xsd` 스키마
- `ovf-section.xsd` - `ovf-section.xsd` 스키마
- `ovf-strings.xsd` - `ovf-strings.xsd` 스키마
- `ovfenv-core.xsd` - `ovfenv-core.xsd` 스키마
- `ovfenv-section.xsd` - `ovfenv-section.xsd` 스키마

◆◆◆ 4 장

논리적 도메인 Manager 검색

이 장에서는 서버넷의 시스템에 실행 중인 논리적 도메인 Manager 검색에 대한 정보를 제공합니다.

논리적 도메인 Manager를 실행 중인 시스템 검색

멀티캐스트 메시지를 사용하여 서버넷에서 논리적 도메인 Manager를 검색할 수 있습니다. `ldmd` 데몬은 특정 멀티캐스트 패킷을 네트워크에서 수신할 수 있습니다. 해당 멀티캐스트 메시지가 특정 유형인 경우 `ldmd`이 호출자에게 회신합니다. 이렇게 하면 Oracle VM Server for SPARC를 실행 중인 시스템에서 `ldmd`를 검색할 수 있습니다.

멀티캐스트 통신

이 검색 방식은 `ldmd` 데몬에서 사용된 것과 동일한 멀티캐스트 네트워크를 사용하여 MAC 주소를 자동으로 지정할 때 충돌을 감지합니다. 멀티캐스트 소켓을 구성하려면 다음 정보를 제공해야 합니다.

```
#define MAC_MULTI_PORT 64535
#define MAC_MULTI_GROUP "239.129.9.27"
```

기본적으로, 멀티캐스트 패킷만 컴퓨터가 연결된 서버넷에 전송할 수 있습니다. `ldmd` 데몬에 `ldmd/hops` SMF 등록 정보를 설정하여 동작을 변경할 수 있습니다.

메시지 형식

검색 메시지는 다른 메시지와 혼동되지 않도록 명확히 표시해야 합니다. 다음 멀티캐스트 메시지 형식은 검색 수신 프로세스에서 검색 메시지를 구별할 수 있습니다.

```
#include <netdb.h> /* Used for MAXHOSTNAMELEN definition */
#define MAC_MULTI_MAGIC_NO 92792004
#define MAC_MULTI_VERSION 1

enum {
```

```
SEND_MSG = 0,
RESPONSE_MSG,
LDMD_DISC_SEND,
LDMD_DISC_RESP,
};

typedef struct {
uint32_t version_no;
uint32_t magic_no;
uint32_t msg_type;
uint32_t resv;
union {
mac_lookup_t Mac_lookup;
ldmd_discovery_t Ldmd_discovery;
} payload;
#define lookup payload.Mac_lookup
#define discovery payload.Ldmd_discovery
} multicast_msg_t;

#define LDMD_VERSION_LEN 32

typedef struct {
uint64_t mac_addr;
char source_ip[INET_ADDRSTRLEN];
} mac_lookup_t;

typedef struct {
char ldmd_version[LDMD_VERSION_LEN];
char hostname[MAXHOSTNAMELEN];
struct in_addr ip_address;
int port_no;
} ldmd_discovery_t;
```

▼ 서브넷에서 실행 중인 논리적 도메인 Manager를 검색하는 방법

1. 멀티캐스트 소켓을 엽니다.

“멀티캐스트 통신” [51]에 지정된 포트 및 그룹 정보를 사용해야 합니다.

2. 소켓을 통해 multicast_msg_t 메시지를 전송합니다.

다음은 메시지에 포함해야 합니다.

- version_no의 유효한 값 - MAC_MULTI_VERSION에 정의된 대로 1입니다.
- magic_no의 유효한 값 - MAC_MULTI_MAGIC_NO에 정의된 대로 92792004입니다.
- msg_type을 LDMD_DISC_SEND로 설정

3. 논리적 도메인 Manager의 응답을 멀티캐스트 소켓에서 수신합니다.

응답은 다음 정보가 포함된 multicast_msg_t 메시지여야 합니다.

- version_no의 유효한 값
- magic_no의 유효한 값

- msg_type을 LDMD_DISC_RESP로 설정
- ldmd_discovery_t 구조로 구성된 페이로드. 다음 정보를 포함합니다.
 - ldmd_version – 시스템에 실행 중인 논리적 도메인 Manager의 버전
 - hostname – 시스템의 호스트 이름
 - ip_address – 시스템의 IP 주소
 - port_no – 통신을 위해 논리적 도메인 Manager에서 사용 중인 포트 번호. XMPP 포트 6482여야 합니다.

논리적 도메인 Manager에서 응답을 수신할 때 auto-allocation MAC collision-detection 패킷이 폐기되었는지 확인합니다.

가상 도메인 정보 명령 및 API 사용

이 장에서는 가상 도메인 정보 명령 및 API에 대해 설명합니다.

가상 도메인 정보 명령 사용

`virtinfo` 명령을 사용하여 실행 중인 가상 도메인에 대한 정보를 수집할 수 있습니다. 다음 목록에서는 명령 또는 API를 사용하여 가상 도메인에 대해 수집할 수 있는 몇 가지 정보를 보여줍니다.

- 도메인 유형(구현, 컨트롤, 게스트, I/O, 서비스, 루트)
- Virtual Domain Manager로 확인된 도메인 이름
- 도메인의 UUID(Universally Unique Identifier)
- 도메인의 컨트롤 도메인에 대한 네트워크 노드 이름
- 도메인이 실행 중인 새시 일련 번호

`virtinfo` 명령에 대한 자세한 내용은 `virtinfo(1M)` 매뉴얼 페이지를 참조하십시오.

가상 도메인 정보 API 사용

가상 도메인 정보 API를 사용하여 가상 도메인 관련 정보를 수집할 프로그램을 만들 수도 있습니다. `libv12n(3LIB)` 및 `v12n(3EXT)` 매뉴얼 페이지를 참조하십시오.

색인

번호와 기호

Oracle VM Server for SPARC 템플릿 유틸리티
설치, 12

virtinfo

가상 도메인 정보, 55

XML

<LDM_event> 메시지, 28

객체 응답, 27

논리적 도메인 Manager 리소스 및 등록 정보, 34

도메인 마이그레이션, 48

명령 응답, 26

스키마, 49

요청 메시지, 23

요청 및 응답 메시지, 23

응답 메시지, 25

작업, 논리적 도메인 Manager, 32

전체 응답, 26

XML 리소스

console, 48

cpu, 35

disk, 40

ldom_info, 34

mau, 37

memory, 38

network, 41

physio_device, 43

policy, 47

spconfig, 46

var, 43

vcc, 42

vds, 39

vds_volume, 39

vsan, 38

vsw, 40

XML 스키마, 49

논리적 도메인 Manager에서 사용, 21

XML 이벤트

도메인, 29

등록 및 등록 해제, 27

리소스, 30

메시지, 27

모두, 31

유형, 29

진행률, 30

하드웨어, 29

XML 전송 프레임, 21

XML 태그

<cmd>, 24

<data>, 25

<LDM_interface>, 23

XML 프로토콜, 22

XMPP

로컬 연결, 22

서버, 22

ㄱ

가상 도메인 정보

API, 55

virtinfo, 55

ㄴ

논리적 도메인 Manager

XML 스키마 사용, 21

검색 방식, 51

ㄷ

설치

Oracle VM Server for SPARC 템플릿 유틸리
티, 12