

# Oracle® Solaris 11.2의 네트워크 가상화 및 네트워크 리소스 관리

ORACLE®

부품 번호: E53792-02  
2014년 9월

Copyright © 2011, 2014, Oracle and/or its affiliates. All rights reserved.

본 소프트웨어와 관련 문서는 사용 제한 및 기밀 유지 규정을 포함하는 라이선스 계약서에 의거해 제공되며, 지적 재산법에 의해 보호됩니다. 라이선스 계약서 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, 디어셈블리 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 공지 사항이 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 회사들의 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 등록 상표입니다.

본 소프트웨어 혹은 하드웨어와 관련문서(설명서)는 제 3자로부터 제공되는 콘텐츠, 제품 및 서비스에 접속할 수 있거나 정보를 제공합니다. Oracle Corporation과 그 자회사는 제 3자의 콘텐츠, 제품 및 서비스와 관련하여 어떠한 책임도 지지 않으며 명시적으로 모든 보증에 대해서도 책임을 지지 않습니다. Oracle Corporation과 그 자회사는 제 3자의 콘텐츠, 제품 및 서비스에 접속하거나 사용으로 인해 초래되는 어떠한 손실, 비용 또는 손해에 대해 어떠한 책임도 지지 않습니다.

# 목차

---

이 설명서 사용 .....	9
<b>1 네트워크 가상화 및 네트워크 리소스 관리 소개 .....</b>	<b>11</b>
Oracle Solaris 11.2에서 네트워크 가상화 및 네트워크 리소스 관리의 새로운 기능 .....	11
네트워크 가상화 및 네트워크 리소스 관리란? .....	13
가상 네트워크 개요 .....	13
가상 네트워크의 구성 요소 .....	14
가상 네트워크 작동 방법 .....	16
VXLAN(Virtual Extensible Local Area Network) 사용 .....	18
에지 가상 브리징 사용 .....	18
가상 네트워크 구현 대상 .....	18
네트워크 리소스 관리의 개요 .....	19
데이터 링크 등록 정보를 사용하여 네트워크 리소스 관리 .....	20
플로우를 사용한 네트워크 리소스 관리 .....	20
네트워크 리소스 관리의 이점 .....	20
<b>2 가상 네트워크 만들기 및 관리 .....</b>	<b>21</b>
가상 네트워크의 구성 요소 구성 .....	21
가상 네트워크의 구성 요소를 구성하기 위한 명령 .....	21
▼ VNIC 및 Etherstub 구성 방법 .....	22
▼ VLAN ID로 VNIC 구성 방법 .....	24
가상 네트워크 구축 .....	26
▼ 가상 네트워크용으로 영역을 구성하는 방법 .....	26
▼ VNIC를 사용하도록 영역을 재구성하는 방법 .....	28
▼ 영역에서 VNIC를 임시로 만드는 방법 .....	30
▼ 개인 가상 네트워크를 구성하는 방법 .....	32
VNIC 관리 .....	34
VNIC 표시 .....	35
VNIC의 VLAN ID 수정 .....	38
VNIC MAC 주소 수정 .....	40

VNIC 마이그레이션 .....	41
VNIC 삭제 .....	43
VNIC와 함께 단일 루트 I/O 가상화 사용 .....	46
데이터 링크의 SR-IOV 모드 사용 .....	46
VF VNIC 만들기 .....	47
VF VNIC 마이그레이션 .....	48
VF 정보 표시 .....	49
<b>3 VXLAN(Virtual Extensible Local Area Network)을 사용하여 가상 네트워크 구성</b> .....	<b>51</b>
VXLAN 개요 .....	51
VXLAN 사용 시의 이점 .....	52
VXLAN 이름 지정 규칙 .....	52
VXLAN 토폴로지 .....	53
영역과 함께 VXLAN 사용 .....	55
VXLAN 구성 계획 .....	57
VXLAN 요구 사항 .....	57
VXLAN 구성 .....	58
▼ VXLAN을 구성하는 방법 .....	58
VXLAN 정보 표시 .....	62
VXLAN 삭제 .....	62
영역에 VXLAN 지정 .....	62
▼ 영역에 VXLAN을 지정하는 방법 .....	63
사용 사례: 링크 집계를 통해 VXLAN 구성 .....	64
<b>4 에지 가상 브리징을 사용하여 서버-네트워크 에지 가상화 관리</b> .....	<b>69</b>
서버-네트워크 에지 가상화에서 EVB 지원 .....	69
반사 중계 .....	70
네트워크에서 자동화된 VNIC 구성 .....	70
Improving Network and Server Efficiency by Using EVB .....	71
EVB 설치 .....	73
▼ EVB를 설치하는 방법 .....	73
동일한 물리적 포트를 통한 VM 간 전환 제어 .....	74
외부 스위치를 통해 통신하도록 VM을 사용으로 설정 .....	74
LLDP를 사용하여 VM 간 통신 관리 .....	78
VDP를 사용하여 VNIC 정보 교환 .....	79
VDP에서 VNIC 정보를 교환하는 방법 .....	80
VDP와 ECP의 상태 및 통계 표시 .....	81
VDP 상태 및 통계 표시 .....	81

링크 등록 정보 표시 .....	81
ECP 상태 및 통계 표시 .....	82
기본 EVB 구성 변경 .....	82
▼ 기본 EVB 구성을 변경하는 방법 .....	83
<b>5 탄력적 가상 스위치 정보 .....</b>	<b>87</b>
EVS(탄력적 가상 스위치) 기능의 개요 .....	87
Oracle Solaris의 가상 스위치 .....	88
Oracle Solaris 탄력적 가상 스위치 기능이란? .....	89
EVS 사용 이점 .....	91
탄력적 가상 스위치 리소스 .....	92
EVS의 이름 공간 관리 .....	93
EVS 구성 요소 .....	94
EVS 관리자 .....	95
EVS 컨트롤러 .....	95
EVS 클라이언트 .....	97
EVS 노드 .....	97
EVS 관리 명령 .....	97
evsadm 명령 .....	98
evsstat 명령 .....	100
dladm 명령 .....	100
zonecfg 명령 .....	100
탄력적 가상 스위치에 연결된 VNIC 관리에 대한 제한 사항 .....	101
자동으로 생성된 VXLAN 데이터 링크 .....	101
EVS 사용을 위한 필수 패키지 .....	102
EVS의 영역 작업 방법 .....	102
EVS 사용을 위한 보안 요구 사항 .....	103
<b>6 탄력적 가상 스위치 관리 .....</b>	<b>105</b>
EVS 관리 작업 .....	105
탄력적 가상 스위치 구성 계획 .....	106
EVS 컨트롤러 만들기 및 관리 .....	107
EVS 컨트롤러를 위한 필수 패키지 .....	107
EVS 컨트롤러를 구성하기 위한 명령 .....	108
EVS 컨트롤러 구성 .....	110
탄력적 가상 스위치 구성 .....	117
탄력적 가상 스위치를 위한 필수 패키지 .....	117
탄력적 가상 스위치를 구성하기 위한 명령 .....	117
▼ 탄력적 가상 스위치를 구성하는 방법 .....	119

탄력적 가상 스위치에 대한 VNIC 만들기 .....	121
탄력적 가상 스위치, IPnet 및 VPort 관리 .....	123
탄력적 가상 스위치 관리 .....	123
IPnet 구성 관리 .....	127
VPort 구성 관리 .....	129
탄력적 가상 스위치 삭제 .....	133
탄력적 가상 스위치 모니터링 .....	134
탄력적 가상 스위치에 대한 사용 사례 예 .....	136
사용 사례: 탄력적 가상 스위치 구성 .....	137
사용 사례: 테넌트에 대한 탄력적 가상 스위치 구성 .....	142
<b>7 네트워크 리소스 관리 .....</b>	<b>149</b>
데이터 링크 등록 정보를 사용하여 네트워크 리소스 관리 .....	149
데이터 링크에서 리소스를 할당하는 명령 .....	150
NIC 링 관리 .....	150
MAC 클라이언트에서 링 할당 .....	150
VLAN에서 링 할당 .....	151
링을 구성하는 명령 .....	151
데이터 링크에 대한 링 사용 및 링 지정 표시 .....	153
클라이언트 구성 및 링 할당 .....	154
풀 및 CPU 관리 .....	159
풀 및 CPU 작업 .....	159
데이터 링크에 대한 CPU 풀 구성 .....	162
데이터 링크에 CPU 할당 .....	163
플로우를 사용하여 네트워크 리소스 관리 .....	164
플로우에서 리소스를 할당하는 명령 .....	165
플로우 구성 .....	165
사용 사례: 데이터 링크 및 플로우 등록 정보를 설정하여 네트워크 리소스 관리 .....	167
<b>8 네트워크 트래픽 및 리소스 사용 모니터링 .....</b>	<b>173</b>
데이터 링크 및 플로우의 네트워크 트래픽 통계 모니터링 개요 .....	173
네트워크 트래픽 통계를 모니터링하는 명령 .....	175
Displaying Network Traffic Statistics of Links .....	175
네트워크 장치의 네트워크 트래픽 통계 표시 .....	176
데이터 링크의 네트워크 트래픽 통계 표시 .....	179
링크 집계의 네트워크 트래픽 통계 표시 .....	180
브리지의 네트워크 트래픽 통계 표시 .....	180
Displaying Network Traffic Statistics of Flows .....	181
네트워크 트래픽에 대한 네트워크 정산 구성 .....	183

▼ 네트워크 정산 설정 방법 .....	183
네트워크 트래픽에 대한 기록 통계 표시 .....	185
색인 .....	189





## 이 설명서 사용

---

- **개요** - Oracle Solaris 가상 네트워킹 기능을 구성하고 네트워크 트래픽을 모니터링하는 방법에 대해 설명합니다. 또한 네트워크 리소스를 관리하는 데 사용되는 다른 프로세스에 대해 설명합니다.
- **대상** - 시스템 관리자
- **필요한 지식** - 네트워크 관리와 관련한 기본 기술과 일부 고급 기술

## 제품 설명서 라이브러리

이 제품에 대한 최신 정보 및 알려진 문제는 설명서 라이브러리(<http://www.oracle.com/pls/topic/lookup?ctx=E36784>)에서 확인할 수 있습니다.

## Oracle 지원 액세스

Oracle 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.

## 피드백

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



## 네트워크 가상화 및 네트워크 리소스 관리 소개

---

이 장에서는 Oracle Solaris의 네트워크 가상화 및 네트워크 리소스 관리에 대한 개요를 제공합니다.

이 장의 내용:

- “Oracle Solaris 11.2에서 네트워크 가상화 및 네트워크 리소스 관리의 새로운 기능” [11]
- “네트워크 가상화 및 네트워크 리소스 관리란?” [13]
- “가상 네트워크 개요” [13]
- “네트워크 리소스 관리의 개요” [19]

### Oracle Solaris 11.2에서 네트워크 가상화 및 네트워크 리소스 관리의 새로운 기능

기존 고객을 위해 이 절에서는 이 릴리스의 주요 변경 사항을 강조해서 보여줍니다.

- **Oracle Solaris EVS(Elastic Virtual Switch) 기능** - 가상 스위치를 직접 관리할 수 있도록 Oracle Solaris 네트워크 가상화 기능을 확장했습니다. Oracle Solaris EVS 기능은 데이터 센터 또는 멀티 테넌트 클라우드 환경에서 여러 서버에 있는 가상 머신을 상호 연결하는 가상 네트워킹 기반구조를 제공합니다. 동일한 EVS에 연결된 가상 머신은 서로 통신할 수 있습니다. EVS를 사용하면 여러 호스트에 있는 가상 스위치를 중앙에서 관리할 수 있으므로 EVS에 연결된 vNIC를 중앙에서 관리할 수 있습니다. 자세한 내용은 [5장. 탄력적 가상 스위치 정보](#)를 참조하십시오. EVS를 관리하는 방법에 대한 자세한 내용은 [6장. 탄력적 가상 스위치 관리](#)를 참조하십시오.
- **VXLAN(Virtual Extensible Local Area Network) 지원** - Oracle Solaris에서는 대규모 데이터 센터에서 가상화 지원을 위해 격리 방법을 제공하는 VXLAN 기술을 지원합니다. VXLAN 기술을 사용하면 클라우드 환경의 서로 다른 계층 2 네트워크에 속하는 물리적 서버 간에 가상 머신을 마이그레이션할 수 있습니다. 자세한 내용은 [3장. VXLAN\(Virtual Extensible Local Area Network\)을 사용하여 가상 네트워크 구성](#)을 참조하십시오.
- **SR-IOV(Single Root I/O Virtualization) 지원** - 이 기능을 사용하면 SR-IOV를 지원하는 네트워크 장치에서 vNIC를 기반으로 VF(가상 기능)를 만들 수 있습니다. 자세한 내용은 [“vNIC와 함께 단일 루트 I/O 가상화 사용” \[46\]](#)을 참조하십시오.

- **영역에서 임시로 VNIC(Virtual Network Interface Card) 만들기** - 전역 영역에서 비전역 영역에 임시 VNIC를 직접 만들 수 있습니다. 임시 VNIC를 만들려면 `dladm create-vnic` 명령에서 `-t` 옵션을 사용해야 합니다. 임시 VNIC는 영역을 다음에 재부트할 때 까지 지속됩니다. VNIC를 임시로 만드는 이외에 영역에서 VLAN 및 IPoB(IP over InfiniBand) 분할 영역도 임시로 만들 수 있습니다. 자세한 내용은 [영역에서 VNIC를 임시로 만드는 방법 \[30\]](#)을 참조하십시오.
- **외부 스위치를 사용하여 VNIC 간 통신** - Oracle Solaris 11.2 반사 중계 기능을 사용하면 기본 물리적 NIC를 공유하는 로컬 Oracle Solaris 영역 또는 Oracle VM 간 트래픽을 호스트 가상 스위치 대신 물리적 네트워크에 강제로 전송할 수 있습니다. 이러한 엔티티 간 통신에는 반사 중계 기능을 지원하는 외부 스위치에 대해 구성된 정책이 적용됩니다. 자세한 내용은 [“동일한 물리적 포트를 통한 VM 간 전환 제어” \[74\]](#)를 참조하십시오.
- **네트워크 트래픽 통계 모니터를 위한 향상된 기능** - 향상된 `dlstat` 및 `flowstat` 명령을 사용하여 네트워크 트래픽 통계를 효율적으로 모니터할 수 있습니다. 네트워크 트래픽 통계 모니터를 위한 향상된 기능은 다음과 같습니다.
  - 네트워크 트래픽 통계가 현재 시간과 함께 표시됩니다.
  - 지정된 간격과 개수 값을 기반으로 네트워크 트래픽 통계를 표시하고 새로 고칩니다.
  - 지정된 간격 값을 기반으로 초당 속도 단위로 네트워크 트래픽 통계를 표시합니다.

향상된 기능에 대한 자세한 내용은 [“네트워크 장치의 네트워크 트래픽 통계 표시” \[176\]](#) 및 [“Displaying Network Traffic Statistics of Flows” \[181\]](#)를 참조하십시오.

- **플로우 구성 변경** - 향상된 `flowadm add-flow` 명령을 사용하여 추가된 속성 수와 최신 조합을 기반으로 데이터 링크에서 플로우를 구성할 수 있습니다. 그러면 다른 포트, 전송 프로토콜 및 IP 주소에서 수신되는 네트워크 패킷을 선택적으로 구성할 수 있습니다. 자세한 내용은 [“플로우를 사용하여 네트워크 리소스 관리” \[164\]](#)를 참조하십시오.  
 플로우를 관리하는 데 사용되는 대역폭 등록 정보 이외에 `flowadm set-flowprop` 명령을 사용하여 플로우에 대한 `priority` 등록 정보를 설정할 수 있습니다. `priority` 등록 정보를 설정하여 플로우의 우선 순위를 지정할 수 있습니다. 새로운 읽기 전용 등록 정보인 `hwflow`를 사용하여 플로우가 인스턴스화되는 방법을 확인할 수 있습니다. 자세한 내용은 [“플로우 구성” \[165\]](#)을 참조하십시오.
- **VNIC에 연결된 여러 MAC 주소 표시** - 향상된 `dladm show-vnic` 명령을 사용하여 VNIC에 연결된 여러 MAC 주소를 표시할 수 있습니다. 자세한 내용은 [“MAC 주소가 여러 개인 VNIC 표시” \[36\]](#)를 참조하십시오.
- **시스템 생성 VNIC** - `dladm create-vnic` 명령을 사용하여 만들 수 있는 VNIC 이외에 시스템에서도 VNIC를 만듭니다. 이러한 VNIC를 시스템 생성 VNIC라고 합니다. 자세한 내용은 [“가상 네트워크의 구성 요소를 구성하기 위한 명령” \[21\]](#)을 참조하십시오.
- **데이터 링크의 물리적 및 가상 링크 상태 표시** - `dladm show-phys` 및 `dladm show-ether` 명령을 사용하여 데이터 링크의 물리적 링크 상태를 표시할 수 있습니다. 데이터 링크의 가상 링크 상태를 표시하려면 `dladm show-link` 명령을 사용할 수 있습니다. 자세한 내용은 [“데이터 링크의 물리적 링크 상태 및 가상 링크 상태 표시” \[37\]](#)를 참조하십시오.
- **데이터 링크 등록 정보의 유효 값 표시** - `dladm show-linkprop` 명령은 데이터 링크 등록 정보에 대한 `EFFECTIVE` 필드를 표시하도록 향상되었습니다. `EFFECTIVE` 필드의 값은 리소스의 가용성, 기본 장치의 기능 또는 피어와의 협상을 기반으로 시스템에 의해 결정됩니다.

다. 유효 값은 구성된 값과 같지 않아도 됩니다. 등록 정보가 값으로 구성되지 않더라도 데이터 링크 등록 정보는 유효 값을 가질 수 있습니다.

## 네트워크 가상화 및 네트워크 리소스 관리란?

네트워크 가상화는 하드웨어 네트워크 리소스 및 소프트웨어 네트워크 리소스를 단일 관리 단위로 결합하는 프로세스입니다. 이 단일 관리 단위를 가상 네트워크라고 합니다.

네트워크 리소스 관리는 네트워킹 프로세스에 대한 리소스를 관리 및 할당하는 프로세스입니다. 처리 중인 네트워크 트래픽의 양에 따라 리소스를 다르게 지정할 수 있습니다. 실제 요구 사항에 따라 리소스를 관리 및 할당하여 패킷을 처리할 때 시스템의 효율성을 높일 수 있습니다.

네트워크 리소스 관리를 효율적으로 사용할 때 네트워크 가상화가 최적화됩니다. 시스템 및 사용자의 하드웨어 및 소프트웨어 네트워킹 리소스 공유를 제어하여 가상 네트워킹 프로세스의 효율성을 높일 수 있습니다. 네트워크 가상화를 네트워크 리소스 관리와 함께 사용하면 플로우 제어를 관리하고 시스템 성능을 향상시키며 OS(운영 체제) 가상화, 유틸리티 컴퓨팅, 서버 통합 실현 등에 필요한 네트워크 사용률을 구성하는 데 유용합니다.

## 가상 네트워크 개요

가상 네트워크는 물리적 네트워크를 열거하는 네트워크로 하드웨어 및 소프트웨어 네트워크 리소스의 조합입니다. 가상 네트워크는 네트워크 가상화의 최종 결과물입니다.

가상 네트워크는 두 가지 광범위한 유형인 외부와 내부로 분류됩니다.

외부 가상 네트워크는 소프트웨어에서 단일 엔티티로 관리하는 여러 로컬 네트워크로 구성됩니다. 클래식 외부 가상 네트워크의 빌딩 블록은 스위치 하드웨어 및 VLAN(가상 근거리 통신망) 소프트웨어 기술입니다. 외부 가상 네트워크의 예로 대규모 회사 네트워크와 데이터 센터가 있습니다. VLAN에 대한 자세한 내용은 [“Oracle Solaris 11.2의 네트워크 데이터 링크 관리”의 3 장](#), [“가상 LAN을 사용하여 가상 네트워크 구성”](#)을 참조하십시오.

내부 가상 네트워크는 하나 이상의 물리적 NIC에 네트워크 인터페이스가 구성된 가상 시스템 또는 영역을 사용하는 시스템 한 개로 구성됩니다. 이러한 네트워크 인터페이스를 가상 네트워크 인터페이스 카드 또는 vNIC(가상 NIC)라고 부릅니다. 이러한 가상 머신 또는 영역은 동일한 로컬 네트워크에 있는 것처럼 서로 통신하여 단일 호스트에서 효율적으로 가상 네트워크로 작동할 수 있습니다. 이 문서의 장에서는 내부 가상 네트워크를 중심으로 다룹니다.

특별한 유형의 내부 가상 네트워크는 개인 가상 네트워크입니다. 개인 가상 네트워크는 VPN(가상 사설망)과는 다릅니다. VPN은 두 끝점 시스템 간에 보안 P2P 연결을 만듭니다.

개인 가상 네트워크는 외부 네트워크에서 액세스할 수 없는 시스템의 가상 네트워크입니다. 이 내부 네트워크를 다른 외부 네트워크와 격리하려면 *etherstub*이라는 가상 NIC를 통해 VNIC을 구성합니다. 자세한 내용은 “[Etherstub](#)” [15]을 참조하십시오.

## 가상 네트워크의 구성 요소

가상 네트워크의 구성 요소는 다음과 같습니다.

- VNIC(가상 네트워크 인터페이스 카드)
- 가상 스위치
- Etherstub
- 영역

### VNIC(가상 네트워크 인터페이스 카드)

VNIC는 구성될 경우 물리적 NIC처럼 작동하는 L2 엔티티 또는 가상 네트워크 장치입니다. 여러 영역 또는 VM 간에 VNIC를 공유하려면 기본 데이터 링크를 통해 VNIC를 구성합니다. 또한 시스템 리소스가 VNIC를 물리적 NIC처럼 처리합니다. 모든 물리적 이더넷 인터페이스는 VNIC 만들기를 지원합니다. VNIC를 구성하는 방법에 대한 자세한 내용은 [VNIC 및 Etherstub 구성 방법](#) [22]을 참조하십시오.

VNIC에는 자동으로 생성된 MAC 주소가 있습니다. 사용 중인 네트워크 인터페이스에 따라 자동으로 생성된 MAC 주소가 아닌 다른 MAC 주소를 VNIC에 지정할 수 있습니다. 자세한 내용은 “[VNIC MAC 주소 수정](#)” [40]을 참조하십시오.

### 가상 스위치

가상 스위치는 VM(가상 머신) 간에 쉽게 통신할 수 있도록 해주는 엔티티입니다. 가상 스위치는 물리적 시스템 내의 가상 머신 간에 트래픽 루프를 생성하며(VM 간 트래픽) 전송 중에 이 트래픽을 내보내지 않습니다. 가상 스위치는 기본 데이터 링크 위에 VNIC를 만들 때마다 암시적으로 생성됩니다. VM을 통해 구성된 VNIC는 VM 간 통신을 위해 동일한 VLAN 또는 VXLAN에 위치해야 합니다. EVS를 통해 가상 스위치를 관리할 수 있습니다. EVS에 대한 자세한 내용은 [5장. 탄력적 가상 스위치 정보](#)를 참조하십시오.

이더넷 설계에 따라 스위치 포트가 해당 포트에 연결된 호스트로부터 송신 패킷을 받는 경우 해당 패킷이 동일한 포트의 대상으로 이동할 수 없습니다. 가상 네트워크는 동일한 NIC를 공유하므로 이더넷 설계는 가상 네트워크를 통해 구성된 시스템의 제한 사항입니다. 이 이더넷 설계 제한은 VM 간에 서로 통신할 수 있도록 해주는 가상 스위치를 사용하여 해결할 수 있습니다.

시스템에서 VM 간에 통신하기 위해 스위치를 사용해야 하는 경우도 있습니다. 예를 들어, 스위치에 구성된 액세스 제어 목록(ACL)에 VM 간 통신이 종속되어야 할 수 있습니다. 기본적

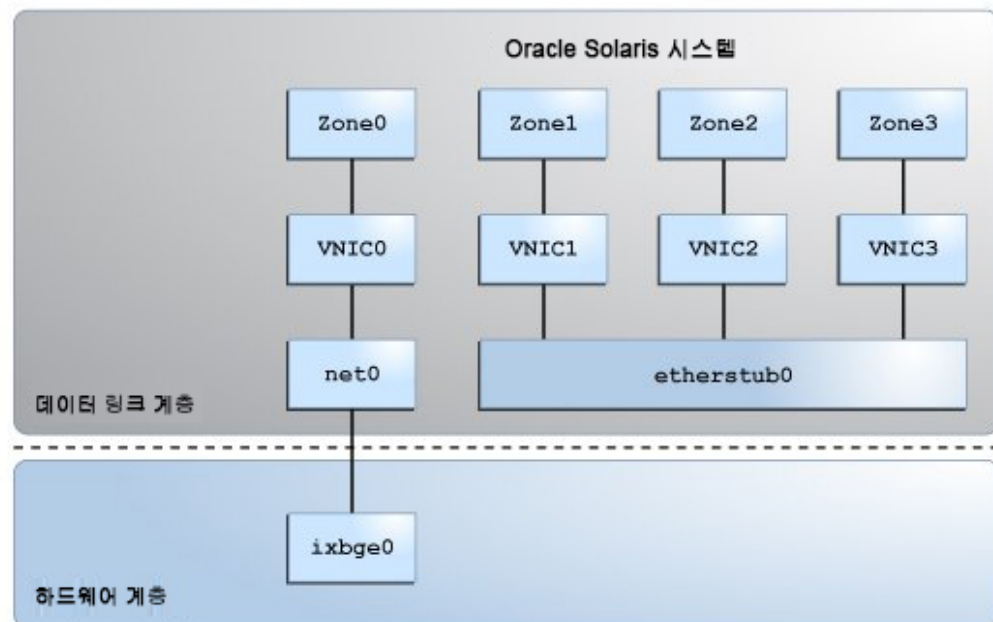
으로 스위치는 패킷을 수신한 해당 포트에서 패킷을 송신할 수 없습니다. 따라서 반사 중계는 스위치를 사용하는 VM 간 통신을 위해 스위치에 대해 사용으로 설정됩니다. 반사 중계를 사용하면 스위치를 통해 패킷을 수신한 해당 포트에서 패킷을 전달할 수 있습니다. 자세한 내용은 [반사 중계](#)를 참조하십시오.

## Etherstub

*etherstub*은 Oracle Solaris 네트워크 스택의 데이터 링크 계층(L2)에서 구성되는 의사 이더넷 NIC입니다. 물리적 NIC 대신 *etherstub*을 통해 vNIC를 만들 수 있습니다. *etherstub*을 사용하면 시스템의 다른 가상 네트워크와 외부 네트워크에서 모두 격리된 개인 가상 네트워크를 생성할 수 있습니다. 예를 들어, *etherstub*을 사용하여 전체 네트워크가 아닌 개발자만으로 액세스를 제한하는 네트워크 환경을 구축할 수 있습니다.

다음 그림은 *etherstub*을 기반으로 개인 가상 네트워크를 보여줍니다.

그림 1-1 개인 가상 네트워크



이 그림은 vNIC1, vNIC2 및 vNIC3을 구성하는 데 사용되는 etherstub0을 보여줍니다. 각 vNIC가 영역에 지정됩니다. etherstub을 기반으로 하는 개인 가상 네트워크는 외부

네트워크에서 액세스할 수 없습니다. 자세한 내용은 [개인 가상 네트워크를 구성하는 방법 \[32\]](#)을 참조하십시오.

## 영역

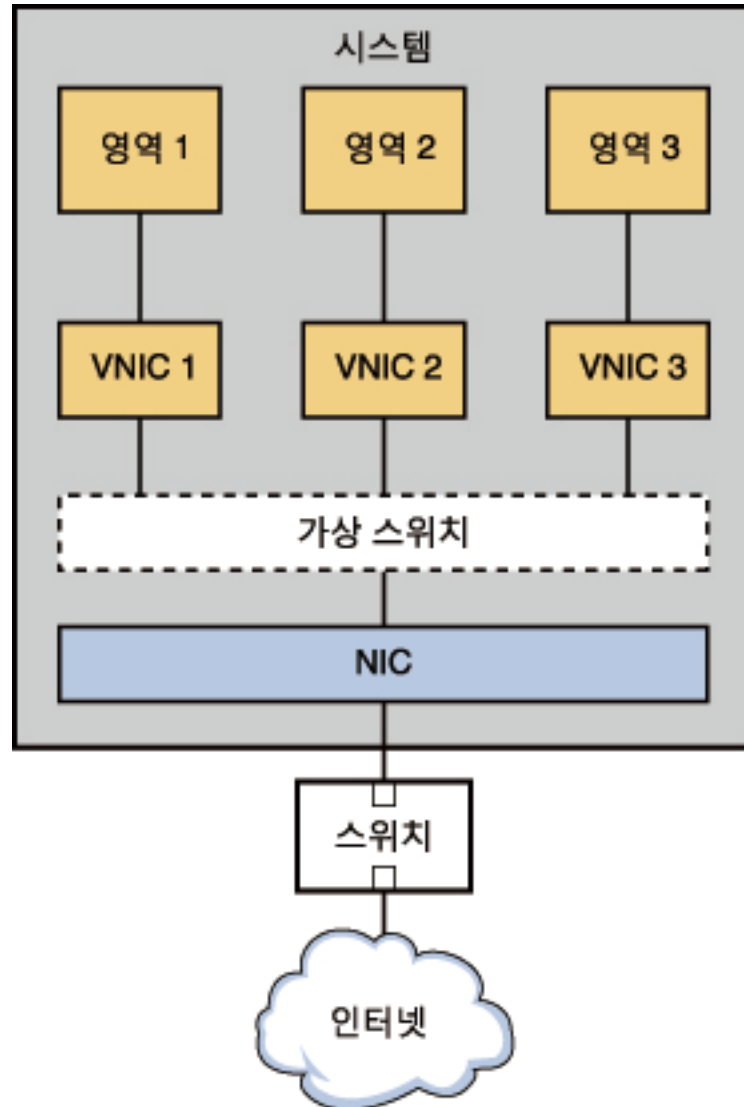
영역은 Oracle Solaris 운영 체제의 단일 인스턴스 내에서 만들어진 가상화된 운영 체제 환경입니다. 영역은 애플리케이션 실행을 위한 격리되고 보호된 환경을 제공합니다. etherstub 및 VNIC는 Oracle Solaris 가상화 기능의 일부일 뿐입니다. Oracle Solaris 영역에서 사용할 VNIC 또는 etherstub을 지정하여 단일 시스템에 네트워크를 만들 수 있습니다. 영역에 대한 자세한 내용은 "[Oracle Solaris 영역 소개](#)"를 참조하십시오.

## 가상 네트워크 작동 방법

다음 그림은 가상 네트워크 작업과 시스템의 해당 구성 요소를 보여줍니다.



그림 1-2 가상 네트워크 작업



이 그림에서는 NIC 한 개가 있는 단일 시스템을 보여줍니다. NIC는 VNIC 세 개로 구성됩니다. 각 VNIC가 영역에 지정됩니다. Zone 1, Zone 2 및 Zone 3은 시스템에서 사용하도록 구성된 세 개의 영역입니다. 영역은 각각의 VNIC를 사용하여 각 영역 간에 그리고 외부 네트워크와 통신합니다. 세 개의 VNIC가 가상 스위치를 통해 기본 물리적 NIC에 연결됩니다. 가상 스위치의 기능은 물리적 스위치의 기능과 동일하며, 두 스위치 모두 시스템에 연결합니다.

가상 네트워크가 구성된 경우 영역에서 가상 네트워크가 없는 시스템과 동일한 방식으로 외부 호스트에 트래픽을 보냅니다. 트래픽은 영역에서 vNIC를 통해 가상 스위치와 물리적 인터페이스 순서로 흐르며, 여기서 데이터를 네트워크로 보냅니다.

영역에 구성된 vNIC가 동일한 VLAN의 일부인 경우 영역은 시스템 내에서 각 영역 간에 트래픽을 교환할 수도 있습니다. 예를 들어, 전용 vNIC 1을 통해 Zone 1에서 패킷을 전달합니다. 그런 다음 트래픽이 가상 스위치를 통해 vNIC 3으로 전달됩니다. 그런 다음 vNIC 3은 트래픽을 Zone 3에 전달합니다. 트래픽이 시스템을 벗어나지 않았으므로 이더넷 제한 사항에 위반되지도 않습니다.

또는 etherstub을 기반으로 가상 네트워크를 만들 수 있습니다. etherstub은 완전히 소프트웨어를 기반으로 하며 가상 네트워크의 기반으로 네트워크 인터페이스가 필요하지 않습니다.

Oracle은 가상 데이터 센터 내에 가상 네트워크를 만드는 기능 등 네트워크 가상화의 일부 측면을 관리하기 위한 Oracle Enterprise Manager Ops Center도 제공합니다. Oracle Enterprise Manager Ops Center에 대한 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=oc122&id=OPCCM>에서 설명서 라이브러리를 참조하십시오.

## VXLAN(Virtual Extensible Local Area Network) 사용

VXLAN(Virtual eXtensible Local Area Network)은 가상 네트워크에 대한 확장성 및 네트워크 격리를 제공하는 네트워크 가상화 기술입니다. VXLAN은 VLAN(Virtual Local Area Network)의 4K 제한을 해결하고 스위치와 같은 물리적 기반구조의 가상화 수요를 줄입니다. 자세한 내용은 [3장. VXLAN\(Virtual Extensible Local Area Network\)을 사용하여 가상 네트워크 구성](#)을 참조하십시오.

## 에지 가상 브리징 사용

호스트는 EVB(에지 가상 브리징)를 사용하여 시스템의 가상 링크와 관련된 정보를 외부 스위치와 교환할 수 있습니다. EVB는 포트 뒤에 있는 모든 가상 링크에 대한 정보를 교환하는 데 사용되고, DCB는 포트에 대한 정보를 교환하는 데 사용됩니다. EVB에 대한 자세한 내용은 [4장. 에지 가상 브리징을 사용하여 서버-네트워크 에지 가상화 관리](#)를 참조하십시오.

## 가상 네트워크 구현 대상

Oracle Sun 서버에서 리소스를 통합해야 하는 경우 가상 네트워크를 구축해 보십시오. 일부 서버에서 다양한 애플리케이션을 통합하여 사용 가능한 리소스를 효율적으로 활용할 수 있습니다. 그런 다음 가상 네트워킹을 사용하여 애플리케이션 간에 연결할 수 있습니다.

ISP, 통신 회사 및 대규모 금융 기관의 통합자는 다음 하드웨어 리소스로 인해 서버를 통합할 수 있습니다.

- 상당한 양의 대역폭과 하드웨어 지원을 제공하는 강력한 NIC(예: NIC 링 및 VF(가상 기능) 지원)
- 대용량 RAM(Random Access Memory) 및 CPU(중앙 처리 장치)를 탑재한 강력한 물리적 시스템

분리, 보안 및 유연성의 큰 손실 없이 여러 영역이나 가상 시스템이 포함된 단일 시스템으로 많은 시스템을 교체할 수 있습니다.

네트워크 가상화의 이점에 대한 데모를 보려면 [Consolidating the Data Center With Network Virtualization \(http://download.oracle.com/otndocs/tech/OTN\\_Demos/data-center-consolidation.html\)](http://download.oracle.com/otndocs/tech/OTN_Demos/data-center-consolidation.html)을 참조하십시오.

## 네트워크 리소스 관리의 개요

Oracle Solaris에서는 네트워크 리소스를 관리하여 QoS(서비스 품질)를 보다 쉽고 동적으로 얻을 수 있습니다. 네트워크 리소스 관리는 트래픽의 전용 레인을 만드는 것에 비유됩니다. 여러 리소스를 결합하여 특정 유형의 네트워크 패킷에 제공하는 경우 해당 리소스가 이러한 패킷의 네트워크 레인을 형성합니다. 각 네트워크 레인에 대해 리소스를 다르게 지정할 수 있습니다. 예를 들어, 네트워크 트래픽이 가장 많은 레인에 리소스를 더 할당할 수 있습니다. 리소스가 실제 요구에 따라 분산되는 네트워크 레인을 구성하면 시스템의 네트워크 패킷 처리 효율성이 증가합니다. 네트워크 레인에 대한 자세한 내용은 “[데이터 링크 및 플로우의 네트워크 트래픽 통계 모니터링 개요](#)” [173]를 참조하십시오.

다음 네트워크 리소스는 시스템의 패킷 처리 효율성을 높이는 데 사용됩니다.

- **대역폭** - 데이터 링크에서 지원되는 네트워킹 프로세스의 실제 요구 사항에 따라 데이터 링크의 대역폭을 제한할 수 있습니다.
- **우선 순위** - 패킷이 처리되는 우선 순위를 지정할 수 있습니다. 우선 순위가 높은 패킷이 다른 패킷보다 빨리 처리되므로 우선 순위가 높을수록 패킷에 대한 대기 시간이 단축됩니다.
- **NIC 링** - NIC가 링 할당을 지원할 경우 전송 및 수신 링을 데이터 링크에서 전용으로 사용할 수 있습니다. 자세한 내용은 “[NIC 링 관리](#)” [150]를 참조하십시오.
- **CPU 풀** - CPU 풀은 생성된 후 특정 영역에 연결됩니다. 이러한 풀을 데이터 링크에 추가 지정하여 연결된 영역의 네트워크 프로세스를 관리할 수 있습니다. 자세한 내용은 “[풀 및 CPU 관리](#)” [159]를 참조하십시오.
- **CPU** - 여러 CPU가 있는 시스템에서는 특정 네트워크 처리를 위해 지정된 개수의 CPU를 전용으로 사용할 수 있습니다. 자세한 내용은 “[풀 및 CPU 관리](#)” [159]를 참조하십시오.

데이터 링크 등록 정보 또는 플로우를 사용하여 시스템의 네트워크 리소스를 관리할 수 있습니다.

## 데이터 링크 등록 정보를 사용하여 네트워크 리소스 관리

데이터 링크 등록 정보를 사용하여 네트워크 리소스를 관리하면 시스템의 패킷 처리 효율성이 향상됩니다. 링크를 만들 때 데이터 링크 등록 정보를 설정할 수 있습니다. 또는 오랫동안 리소스 사용을 조사하고 리소스를 보다 효율적으로 할당하는 방법을 확인한 후와 같이 나중에 데이터 링크 등록 정보를 설정할 수 있습니다. 네트워크 리소스와 관련된 데이터 링크 등록 정보를 설정하여 네트워킹 프로세스에 사용 가능한 지정된 리소스의 양을 결정할 수 있습니다. 리소스 할당 절차는 가상 네트워크와 물리적 네트워크에 모두 적용됩니다. 데이터 링크 등록 정보에 대한 자세한 내용과 데이터 링크 등록 정보를 구성하는 방법은 “[데이터 링크 등록 정보를 사용하여 네트워크 리소스 관리](#)” [149]를 참조하십시오.

## 플로우를 사용한 네트워크 리소스 관리

플로우는 단일 속성 또는 속성 조합을 기반으로 네트워크 패킷을 분류하는 사용자 정의된 방식입니다. 플로우를 만드는 기본 역할을 하는 속성은 네트워크 패킷 헤더의 정보에서 파생됩니다. 네트워크 리소스 관리를 위해 데이터 링크 등록 정보를 설정한 후 플로우를 사용하여 네트워크 패킷 처리를 위해 리소스를 사용하는 방법을 제어할 수 있습니다. 데이터 링크 등록 정보를 설정하지 않고 플로우만 사용하여 네트워크 리소스를 관리할 수도 있습니다.

리소스 관리 플로우 사용 시에는 다음과 같은 단계가 포함됩니다.

1. 단일 속성 또는 속성 조합을 기반으로 플로우 만들기
2. 네트워크 리소스와 관련된 등록 정보를 설정하여 플로우의 리소스 사용 사용자 정의. 현재는 대역폭 및 우선 순위 등록 정보만 플로우로 설정할 수 있습니다.

플로우 구성에 대한 자세한 내용은 “[플로우를 사용하여 네트워크 리소스 관리](#)” [164]를 참조하십시오.

## 네트워크 리소스 관리의 이점

네트워크 리소스 관리를 사용하여 복잡한 QoS 규칙 정의 없이 개별 시스템에서 데이터 트래픽을 격리시키고 우선 순위를 지정하고 추적 및 제어할 수 있습니다.

네트워크 리소스 관리는 다음 작업에 유용합니다.

- 네트워크 프로비저닝
- 서비스 단계 계약 체결
- 클라이언트 청구
- 보안 문제 진단

# ◆◆◆ 2 장

## 가상 네트워크 만들기 및 관리

---

이 장에서는 단일 시스템에서 가상 네트워크의 구성 요소를 구성하고, 가상 네트워크를 구축하고, VNIC을 관리하는 작업에 대해 설명합니다. 가상 네트워크에 대한 소개는 [1장. 네트워크 가상화 및 네트워크 리소스 관리 소개](#)를 참조하십시오.

이 장의 내용:

- “가상 네트워크의 구성 요소 구성” [21]
- “가상 네트워크 구축” [26]
- “VNIC 관리” [34]
- “VNIC와 함께 단일 루트 I/O 가상화 사용” [46]

### 가상 네트워크의 구성 요소 구성

Oracle Solaris에서 VNIC과 etherstub는 가상 네트워크의 기본 구성 요소입니다. 이 단원에서는 가상 네트워크를 구축하기 위한 준비로 이러한 구성 요소를 구성하는 단계를 설명합니다. 이러한 구성 요소에 대한 자세한 내용은 [“가상 네트워크의 구성 요소” \[14\]](#)를 참조하십시오.

### 가상 네트워크의 구성 요소를 구성하기 위한 명령

VNIC를 만들려면 `dladm create-vnic` 명령을 사용합니다.

```
# dladm create-vnic -l link [-v vid] VNIC
```

*link* VNIC를 구성하는 데 사용되는 링크의 이름입니다.

*vid* VLAN ID 또는 VNIC(VNIC를 VLAN으로 만드는 경우)입니다. VLAN ID로 VNIC를 구성하려면 [VLAN ID로 VNIC 구성 방법 \[24\]](#)을 참조하십시오. VLAN에 대한 자세한 내용은 [“Oracle Solaris 11.2의 네트워크 데이터 링크 관리”의 3장, “가상 LAN을 사용하여 가상 네트워크 구성”](#)을 참조하십시오.

VNIC VNIC의 이름입니다. 사용자 정의 이름을 만드는 방법에 대한 자세한 내용은 [“Oracle Solaris 11.2 네트워크 구성 요소의 구성 및 관리”](#)의 [“유효한 링크 이름 규칙”](#)을 참조하십시오.

MAC 주소, VNIC와 연결할 CPU 등과 같은 VNIC에 대한 다른 등록 정보를 구성할 수 있습니다. 다른 등록 정보의 목록은 [dladm\(1M\)](#) 매뉴얼 페이지를 참조하십시오. 특정 등록 정보 수정은 VNIC에서만 작동합니다. 예를 들어, `dladm create-vmnic` 명령의 경우 MAC 주소를 구성할 뿐만 아니라 VLAN ID를 지정하여 VNIC를 VLAN으로 만들 수 있습니다. 하지만 `dladm create-vlan` 명령을 사용하여 VLAN에 대해 MAC 주소를 직접 구성할 수는 없습니다.

VNIC는 데이터 링크를 통해 한 번에 하나씩만 만들 수 있습니다. 데이터 링크와 마찬가지로 VNIC에는 필요에 따라 추가로 구성할 수 있는 링크 등록 정보가 포함됩니다. 다른 유형의 링크 등록 정보에 대한 자세한 내용은 [“데이터 링크 등록 정보를 사용하여 네트워크 리소스 관리”](#) [20]를 참조하십시오.

`dladm create-vmnic` 명령을 사용하여 만들 수 있는 VNIC 이외에 Oracle VM Server for SPARC vnet용 가상 네트워크 I/O에 유용한 VNIC도 생성됩니다. 이를 시스템 생성 VNIC라고 합니다. 시스템 생성 VNIC는 `<entity>-<name>` 이름 지정 규칙을 따릅니다. 여기서 `entity`는 VNIC를 만든 시스템 엔티티를 나타내고, `name`은 시스템 엔티티 내의 VNIC 이름을 나타냅니다. 사용자 생성 VNIC 이름은 하이픈(-)을 포함할 수 없습니다. 시스템 생성 VNIC에만 하이픈(-)을 사용할 수 있으므로, 시스템 생성 VNIC와 사용자 생성 VNIC를 구별하는 데 도움이 됩니다. 시스템 생성 VNIC를 수정, 이름 바꾸기, 연결(plumb) 또는 삭제할 수 없습니다. 자세한 내용은 [Oracle VM Server for SPARC 3.1 Administration Guide](#)를 참조하십시오.

`dlstat` 및 `snoop` 명령을 사용하여 시스템 생성 VNIC에 대한 네트워크 트래픽을 모니터링할 수 있습니다. `flowadm` 명령을 사용하여 시스템 생성 VNIC에 대한 플로우를 만들 수도 있습니다. 플로우를 사용하면 네트워크 리소스를 관리하는 동시에 네트워크 트래픽 통계를 관리할 수 있습니다. `flowstat` 명령을 사용하여 플로우에 대한 네트워크 트래픽 통계를 모니터링할 수 있습니다. 플로우에 대한 자세한 내용은 [“플로우 구성”](#) [165]을 참조하십시오.

etherstub를 만들려면 `dladm create-etherstub` 명령을 사용합니다.

```
# dladm create-etherstub etherstub
```

여기서 `etherstub`은 만들려는 etherstub의 이름입니다.

## ▼ VNIC 및 Etherstub 구성 방법

VNIC는 가상 네트워크를 외부 네트워크에 연결합니다. 또한 VNIC를 사용하면 VNIC로 자동으로 만들어지는 가상 스위치를 통해 영역이 서로 통신할 수 있습니다. 가상 네트워크가 영역 간에 내부적으로, 외부 LAN 및 인터넷에서 트래픽을 호스팅하기 위해서는 각 영역에 고유한 VNIC가 포함되어야 합니다. 따라서 가상 네트워크에 속하는 영역 수만큼 이 절차를 여러 번 반복해야 합니다.

1. 관리자로 로그인합니다.

자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”을 참조하십시오.

2. (옵션) etherstub을 만듭니다.

```
# dladm create-etherstub etherstub
```

이 단계는 개인 가상 네트워크를 만드는 경우에만 수행합니다. 개인 가상 네트워크에 대한 설명은 “가상 네트워크 개요” [13]를 참조하십시오. 개인 가상 네트워크를 구성하는 방법에 대한 자세한 내용은 개인 가상 네트워크를 구성하는 방법 [32]을 참조하십시오.

데이터 링크와 마찬가지로 네트워크 설정에 유용한 방식으로 etherstub 이름을 지정할 수 있습니다. 사용자 정의 이름을 만드는 방법에 대한 자세한 내용은 “Oracle Solaris 11.2 네트워크 구성 요소의 구성 및 관리”의 “유효한 링크 이름 규칙”을 참조하십시오.

3. VNIC를 만듭니다.

```
# dladm create-vnic -l link [-v vid] VNIC
```

개인 가상 네트워크에 대한 VNIC를 만드는 경우 *link*에 대한 *etherstub*을 대체합니다. VNIC를 VLAN으로 만드는 경우에만 명령에 *-v* 옵션을 포함합니다. VNIC를 VLAN으로 만드는 방법에 대한 자세한 내용은 VLAN ID로 VNIC 구성 방법 [24]을 참조하십시오.

4. VNIC를 통해 IP 인터페이스를 만듭니다.

```
# ipadm create-ip interface
```

*interface*            이전 단계에서 만든 VNIC입니다.

5. VNIC 인터페이스에 정적 IP 주소를 지정합니다.

```
# ipadm create-addr -a address interface
```

*-a address*            IP 주소를 CIDR(Classless Inter-Domain Routing) 표기법으로 지정합니다.

정적 IP 주소는 IPv4 또는 IPv6 주소일 수 있습니다. 자세한 내용은 “Oracle Solaris 11.2 네트워크 구성 요소의 구성 및 관리”의 “IPv4 인터페이스를 구성하는 방법”을 참조하십시오.

6. (옵션) 생성된 VNIC를 확인합니다.

```
# dladm show-link
```

예 2-1 VNIC 구성

이 예에서는 *net0* 데이터 링크를 통해 *vnic1*을 구성하는 방법을 보여줍니다.

```
# dladm create-vnic -l net0 vnic1
```

```
# ipadm create-ip vnic1
# ipadm create-addr -a 192.168.0.10/24 vnic1
# dladm show-link
LINK          CLASS      MTU      STATE    OVER
net0          phys       1500     up       --
vnic1         vnic       1500     up       net0
```

## ▼ VLAN ID로 VNIC 구성 방법

VLAN 트래픽을 호스트하도록 VLAN ID를 사용하여 VNIC를 구성할 수 있습니다. VNIC가 VLAN의 일부로 포함되고 해당 VLAN에 대한 트래픽을 수신해야 하는 경우 해당 VLAN의 VLAN ID를 VNIC에 지정해야 합니다. 또한 개별 VNIC의 VLAN 구성을 네트워크에 전파하도록 링크 등록 정보 `vlan-announce`를 설정할 수도 있습니다.

일반 VLAN 링크와 달리 VLAN으로 구성된 VNIC는 고유한 MAC 주소를 갖고 있습니다. 일반 VLAN에 대한 자세한 내용은 [“Oracle Solaris 11.2의 네트워크 데이터 링크 관리”의 3장, “가상 LAN을 사용하여 가상 네트워크 구성”](#)을 참조하십시오.

이 절차에는 VLAN ID로 VNIC를 만들고 VNIC가 VLAN 트래픽을 지원하도록 설정하기 위해 적합한 등록 정보를 설정하는 단계만 포함됩니다. `vlan-announce` 등록 정보를 사용으로 설정하면 중간 포트와 스위치가 자동으로 업데이트되지만 이러한 포인트에서 VLAN을 정의하려면 중간 포트와 스위치를 별도로 구성해야 합니다.

### 1. 관리자로 로그인합니다.

자세한 내용은 [“Oracle Solaris 11.2의 사용자 및 프로세스 보안”](#)의 [“지정된 관리 권한 사용”](#)을 참조하십시오.

### 2. VLAN ID를 사용하여 VNIC를 만듭니다.

```
# dladm create-vnic -l link -v vid VNIC
```

### 3. (옵션) VNIC의 VLAN 구성을 네트워크에 브로드캐스트합니다.

```
# dladm set-linkprop -p vlan-announce=gvrp link
```

이 단계에서는 연결된 스위치에 VLAN ID를 자동으로 등록하는 GVRP(GARP VLAN 등록 프로토콜) 클라이언트 시스템을 사용으로 설정합니다. 기본적으로 `vlan-announce` 등록 정보는 `off`로 설정되며 네트워크에 VLAN 브로드캐스트 메시지가 전송되지 않습니다. 등록 정보를 `gvrp`로 설정한 후에는 네트워크 장치의 자동 VLAN 포트 구성을 사용으로 설정하기 위해 해당 링크에 대한 VLAN 구성이 전달됩니다. 그런 다음 이러한 장치에서 VLAN 트래픽을 수락하고 전달할 수 있습니다. GVRP에 대한 자세한 내용은 [“Configuring GVRP,” in Sun Ethernet Fabric Operating System](#)을 참조하십시오.

### 4. (옵션) VLAN 브로드캐스트 간의 대기 시간을 구성하려면 `gvrp-timeout` 등록 정보를 설정합니다.



```
# dladm set-linkprop -p gvrp-timeout=time link
```

*time* gvrp-timeout 등록 정보의 값을 밀리초 단위로 나타냅니다. 기본값은 250밀리초입니다. 로드가 높은 시스템은 VLAN 정보를 다시 브로드캐스트할 때 간격이 더 짧아야 할 수 있습니다. 이 등록 정보를 사용하면 간격을 조정할 수 있습니다.

##### 5. (옵션) vlan-announce 및 gvrp-timeout 등록 정보의 값을 표시합니다.

```
# dladm show-linkprop -p vlan-announce,gvrp-timeout
```

#### 예 2-2 VNIC를 VLAN으로 구성

이 예에서는 VLAN ID 123을 사용하여 net0 데이터 링크에서 vnic0 VNIC를 만드는 방법과 VLAN 구성이 네트워크에 전달되도록 설정하는 방법을 보여줍니다.

```
# dladm create-vnic -l net0 -v 123 vnic0
# dladm set-linkprop -p vlan-announce=gvrp net0
# dladm set-linkprop -p gvrp-timeout=250 net0
# dladm show-linkprop -p vlan-announce,gvrp-timeout net0
```

LINK	PROPERTY	PERM	VALUE	EFFECTIVE	DEFAULT	POSSIBLE
net0	vlan-announce	rw	gvrp	gvrp	off	off,gvrp
net0	gvrp-timeout	rw	250	250	250	100-100000

출력에는 다음 정보가 표시됩니다.

LINK	이름으로 식별된 물리적 데이터 링크입니다.
PROPERTY	링크의 등록 정보입니다. 하나의 링크가 여러 등록 정보를 가질 수 있습니다.
PERM	등록 정보에 대한 사용 권한이며, 다음 중 하나입니다. <ul style="list-style-type: none"> <li>■ ro는 링크 등록 정보에 대한 읽기 전용 권한을 나타냅니다.</li> <li>■ rw는 링크 등록 정보에 대한 읽기 및 쓰기 권한을 나타냅니다.</li> </ul>
VALUE	현재 또는 지속 링크 등록 정보 값입니다. 값을 설정하지 않은 경우 --로 표시됩니다. 값을 알 수 없는 경우 ?로 표시됩니다.
DEFAULT	링크 등록 정보의 기본값입니다. 링크 등록 정보에 기본값이 없는 경우 --가 표시됩니다.
POSSIBLE	링크 등록 정보에 포함될 수 있는 심표로 구분된 값 목록입니다. 가능한 값을 알 수 없거나 바인드 해제된 경우 --가 표시됩니다.

## 가상 네트워크 구축

가상 네트워크를 구축하려면 영역을 만들어야 합니다. 시스템 지원을 기반으로 영역을 필요한 만큼 만들 수 있습니다. 각 영역은 고유한 가상 인터페이스를 가집니다. 시스템의 영역은 서로 통신할 수 있습니다. 가상 네트워크는 전체적으로 보다 큰 외부 네트워크의 대상에 연결합니다.

가상 네트워크를 구축하려면 etherstub 또는 VNIC를 구성하고 영역을 구성해야 합니다. 이러한 단계는 독립적인 절차 세트이지만 가상 네트워크 구축을 완료하기 위해서는 두 가지를 모두 수행해야 합니다.

이 단원의 절차는 다음과 같은 가정을 기반으로 합니다.

- 시스템의 가상 네트워크는 세 개의 영역으로 구성되며 구성 단계가 서로 다릅니다. 첫 번째 영역은 새 영역으로 만들어졌으며, 두 번째 영역은 이미 시스템에 존재하고 VNIC를 사용하도록 재구성되어야 합니다. 세 번째 영역은 개인 가상 네트워크로 지정되며 시스템 외부로 네트워크 트래픽을 전송하려면 사용으로 설정해야 합니다.
- 시스템의 물리적 인터페이스는 IP 주소 192.168.3.70으로 구성됩니다.
- 라우터의 IP 주소는 192.168.3.25입니다

가상 네트워크를 구축할 때 일부 단계는 전역 영역에서 수행되며, 일부 단계는 비전역 영역에서 수행됩니다. 혼동하지 않도록 각 단계의 예제에서는 특정 명령이 실행되는 영역이 프롬프트로 표시됩니다. 하지만 프롬프트에 표시되는 실제 경로는 시스템에 지정된 프롬프트에 따라 다를 수 있습니다.

가상 네트워크 관리에 대한 데모는 [Configuring a Virtual Network in Oracle Solaris - Part 1](http://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/VirtualDemo_Part1/VirtualDemo_Part1.htm) ([http://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/VirtualDemo\\_Part1/VirtualDemo\\_Part1.htm](http://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/VirtualDemo_Part1/VirtualDemo_Part1.htm)) 및 [Configuring a Virtual Network in Oracle Solaris - Part 2](http://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/VirtualDemo_Part2/VirtualDemo_Part2.htm) ([http://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/VirtualDemo\\_Part2/VirtualDemo\\_Part2.htm](http://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/VirtualDemo_Part2/VirtualDemo_Part2.htm))를 참조하십시오.

### ▼ 가상 네트워크용으로 영역을 구성하는 방법

이 절차에서는 새 VNIC를 사용하여 새 영역을 구성하는 방법을 설명합니다. 이 절차에는 네트워크 가상화와 관련된 단계만 포함되어 있습니다. 영역을 구성하는 방법에 대한 자세한 내용은 “[Oracle Solaris 영역 만들기 및 사용](#)”의 1 장, “[비전역 영역을 계획 및 구성하는 방법](#)”을 참조하십시오.

1. 관리자로 로그인합니다.  
자세한 내용은 “[Oracle Solaris 11.2의 사용자 및 프로세스 보안](#)”의 “[지정된 관리 권한 사용](#)”을 참조하십시오.
2. VNIC를 구성합니다.

자세한 내용은 [VNIC 및 Etherstub 구성 방법 \[22\]](#)을 참조하십시오.

### 3. 영역을 만듭니다.

```
global# zonecfg -z zone
```

`zone`                    영역의 이름을 나타냅니다.

이전에 만든 VNIC를 영역의 물리적 인터페이스로 지정해야 합니다. 기본적으로 영역의 ip-type 매개변수는 `exclusive`로 설정됩니다.

### 4. 구현한 변경 사항을 확인하고 커밋한 다음 영역을 종료합니다.

```
zonecfg:zone> verify
zonecfg:zone> commit
zonecfg:zone> exit
```

### 5. 영역을 설치합니다.

```
global# zoneadm -z zone install
```

### 6. 영역을 시작합니다.

```
global# zoneadm -z zone boot
```

### 7. 영역이 완전히 부트된 후 영역에 로그인합니다.

```
global# zlogin -C zone
```

### 8. 메시지가 표시되면 정보를 지정합니다.

대부분의 정보를 선택 항목 목록에서 선택하여 지정할 수 있습니다. 일반적으로 기본 옵션만으로도 충분합니다. 가상 네트워크를 구성하려면 다음 정보를 지정하거나 확인해야 합니다.

- 영역의 호스트 이름(예: zone1)
- 영역 VNIC의 IP 주소를 기반으로 하는 영역의 IP 주소
- IPv6이 사용으로 설정되었는지 여부
- 가상 네트워크가 있는 시스템이 서브넷의 일부인지 여부
- IP 주소의 넷마스크
- 가상 네트워크가 작성된 물리적 인터페이스의 IP 주소일 수 있는 기본 경로

필요한 정보를 제공한 후 영역이 다시 시작됩니다.

또는 `anet 리소스`라는 자동 VNIC를 사용하여 배타적 IP 영역을 구성할 수 있습니다. 자세한 내용은 [“Oracle Solaris 영역 만들기 및 사용”의 “영역 구성 방법”](#)을 참조하십시오.

#### 예 2-3 가상 네트워크용으로 영역 구성

이 예에서 zone1은 가상 네트워크용으로 생성되고 vnic1은 물리적 인터페이스로 연결됩니다. 가상 네트워크를 만드는 것과 관련된 영역 매개변수만 나열됩니다.

```
global # zonecfg -z zone1
zonecfg:zone1> create
zonecfg:zone1> set zonepath=/export/home/zone1
zonecfg:zone1> set autoboot=true
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=vnic1
zonecfg:zone1:net> end
zonecfg:zone1> verify
zonecfg:zone1> commit
zonecfg:zone1> exit
```

```
global# zoneadm -z zone1 install
.
.
.
global# zoneadm -z zone1 boot
```

```
global# zlogin -C zone1
```

네트워크를 구성하도록 다음과 같은 정보가 제공됩니다.

```
Hostname: zone1
IP address: 192.168.3.80
System part of a subnet: Yes
Netmask: 255.255.255.0
Enable IPv6: No
Default route: 192.168.3.70
Router IP address: 192.168.3.25
```

## ▼ VNIC를 사용하도록 영역을 재구성하는 방법

이 절차에서는 가상 네트워크의 두번째 영역을 참조합니다. 이 영역이 이미 존재하지만 현재 구성으로는 가상 네트워크에 포함될 수 없습니다. 특히 영역의 IP 유형이 공유 유형이고 현재 인터페이스가 net0입니다. 이러한 구성을 모두 변경해야 합니다.

1. 관리자로 로그인합니다.  
자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”을 참조하십시오.

2. VNIC를 만듭니다.

```
global# dladm create-vnic -l link VNIC
```

이 절차의 뒷부분에서 VNIC의 인터페이스를 구성합니다.

3. 영역의 IP 유형을 shared에서 exclusive로 변경합니다.

```
global# zonecfg -z zone
zonecfg:zone> set ip-type=exclusive
```

4. VNIC를 사용하도록 영역의 인터페이스를 변경합니다.

```
zonecfg:ZONE> remove net physical=NIC
zonecfg:ZONE> add net
zonecfg:ZONE:net> set physical=VNIC
zonecfg:ZONE:net> end
```

5. 구현한 변경 사항을 확인하고 커밋한 다음 영역을 종료합니다.

```
zonecfg:ZONE> verify
zonecfg:ZONE> commit
zonecfg:ZONE> exit
```

6. 영역을 재부트합니다.

```
global# zoneadm -z ZONE reboot
```

7. 영역에 로그인합니다.

```
global# zlogin ZONE
```

8. 영역에서 지금 영역에 지정한 VNIC를 통해 IP 인터페이스를 만듭니다.

```
zone# ipadm create-ip interface
```

9. 정적 IP 주소 또는 DHCP(Dynamic Host Configuration Protocol) IP 주소를 사용하여 VNIC를 구성합니다.

■ 정적 IP 주소를 지정합니다.

```
zone# ipadm create-addr -a address interface
```

-a address                    IP 주소를 지정합니다. CIDR 표기법을 사용할 수 있습니다.

■ DHCP IP 주소를 지정합니다.

```
zone# ipadm create-addr -T dhcp interface
```

10. 영역을 종료합니다.

```
zone# exit
```

11. 전역 영역에서 주소 정보를 /etc/hosts 파일에 추가합니다.

예 2-4 VNIC를 사용하도록 영역 재구성

이 예제에서 zone2는 공유 영역으로 존재합니다. 이 영역은 또한 가상 링크가 아닌 시스템의 기본 인터페이스를 사용합니다. vnic2를 사용하도록 zone2를 수정해야 합니다. vnic2를 사용하려면 먼저 zone2의 IP 유형을 exclusive로 변경해야 합니다. 가상 네트워크와 관련된 정보에만 집중할 수 있도록 일부 출력 결과가 잘려 있습니다.

```

global# dladm create-vnic -l net0 vnic2

global# zonecfg -z zone2
zonecfg:zone2> set ip-type=exclusive
zonecfg:zone2> remove net physical=net0
zonecfg:zone2> add net
zonecfg:zone2:net> set physical=vnic2
zonecfg:zone2:net> end
zonecfg:zone2> verify
zonecfg:zone2> commit
zonecfg:zone2> exit
global# zoneadm -z zone2 reboot

global# zlogin zone2
zone2# ipadm create-ip vnic2
zone2# ipadm create-addr -a 192.168.3.85/24 vnic2
ipadm: vnic2/v4

zone2# exit

global# pfedit /etc/hosts
#
::1          localhost
127.0.0.1    localhost
192.168.3.70 loghost    #For net0
192.168.3.80 zone1      #using vnic1
192.168.3.85 zone2      #using vnic2

```

## ▼ 영역에서 VNIC를 임시로 만드는 방법

링크를 *zone/link*로 지정하여 전역 영역에서 비전역 영역에 VNIC를 직접 만들 수 있습니다. 이 방법에서는 비전역 영역의 이름 공간에 VNIC를 직접 만듭니다. `-t` 옵션은 VNIC가 일시적임을 지정하는 데 사용됩니다. 임시 VNIC는 영역을 다음에 재부트할 때까지 지속됩니다. 전역 영역과 다른 비전역 영역이 동일한 이름을 가진 VNIC를 포함할 수도 있습니다. 이 방법을 통해서만 VNIC를 일시적으로 만들 수 있습니다.

VNIC를 일시적으로 만드는 이외에 VLAN 및 IPoIB(IP over InfiniBand) 분할 영역도 일시적으로 만들 수 있습니다. 자세한 내용은 [dladm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

1. 관리자로 로그인합니다.  
자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”을 참조하십시오.
2. 전역 영역에서 비전역 영역을 만들고 부트합니다.  

```
global# zoneadm -z zone boot
```
3. 비전역 영역에 대한 임시 VNIC를 만듭니다.

```
global# dladm create-vnic -t -l link zone/VNIC
```

-t VNIC가 일시적임을 지정합니다. 임시 VNIC는 *zone*을 다음에 재부트할 때까지 지속됩니다. VNIC를 비전역 영역의 이름 공간에 만들 경우 이 옵션을 지정해야 합니다.

-l 링크(물리적 링크 또는 etherstub)를 지정합니다.

전역 영역에서 비전역 영역에 VLAN 또는 IPoIB 분할 영역을 만드는 데 사용하는 명령의 예는 예 2-5. “영역에서 임시로 VNIC, VLAN 및 IPoIB(IP-over-IB) 분할 영역 만들기”를 참조하십시오.

#### 4. VNIC가 영역에 생성되었는지 확인합니다.

```
global# dladm show-link -Z
```

#### 5. 영역에 로그인합니다.

```
global# zlogin zone
```

#### 6. VNIC가 성공적으로 생성되었는지 확인합니다.

```
zone# dladm show-link
```

### 예 2-5 영역에서 임시로 VNIC, VLAN 및 IPoIB(IP-over-IB) 분할 영역 만들기

다음 예에서는 전역 영역에서 비전역 영역에 vnic1이라는 VNIC를 만드는 방법을 보여줍니다.

```
global# zoneadm -z zone1 boot
global# dladm create-vnic -t -l net0 zone1/vnic1
global# dladm show-link -Z
LINK          ZONE    CLASS    MTU    STATE    OVER
net0          global  phys     1500   up       --
zone1/vnic1   zone1   vnic     1500   down    net0
```

다음 예에서는 zone1에서의 dladm show-link 명령 출력을 보여줍니다.

```
zone1# dladm show-link
LINK          CLASS    MTU    STATE    OVER
vnic1         vnic     1500   down    ?
```

다음 예에서는 전역 영역에서 비전역 영역에 vlan3이라는 VLAN을 만드는 방법을 보여줍니다.

```
global# dladm create-vlan -t -l net0 -v 3 zone1/vlan3
```

-v 옵션은 이더넷 링크를 통해 VLAN의 VLAN ID를 지정합니다.

다음 예에서는 전역 영역에서 비전역 영역에 part1이라는 IPoIB 분할 영역을 만드는 방법을 보여줍니다.

```
global# dladm create-part -t -l net1 -P FFFF zone1/part1
```

-P 옵션은 분할 영역 링크를 만드는 데 사용되는 분할 영역 키를 지정합니다.

## ▼ 개인 가상 네트워크를 구성하는 방법

이 절차에서는 개인 가상 네트워크를 만들고 시스템 외부로 네트워크 트래픽을 전송하기 위해 개인 가상 네트워크를 사용으로 설정하는 방법에 대해 설명합니다. 영역이 가상 네트워크에 속하지만 외부 시스템에서 액세스할 수 없습니다. 격리된 영역이 시스템 외부로 네트워크 트래픽을 전송할 수 있도록 하려면 NAT(네트워크 주소 변환)를 사용해야 합니다. NAT는 VNIC의 개인 IP 주소를 물리적 네트워크 인터페이스의 경로 지정 가능한 IP 주소로 변환합니다. 하지만 개인 IP 주소는 외부 네트워크에서 볼 수 없습니다. NAT에 대한 자세한 내용은 [“Oracle Solaris 11.2의 네트워크 보안”의 “IP 필터의 NAT 기능 사용”](#)을 참조하십시오.

etherstub 사용은 가상 네트워크와 개인 가상 네트워크 간에 중요한 차이점이 있습니다. 개인 가상 네트워크에서 영역에 지정되는 VNIC는 etherstub을 통해 구성되며 시스템을 통과하는 네트워크 트래픽과 격리됩니다.

이 절차에서는 영역이 이미 있지만 현재까지는 연결된 인터페이스가 없다고 가정합니다.

### 1. 관리자로 로그인합니다.

자세한 내용은 [“Oracle Solaris 11.2의 사용자 및 프로세스 보안”](#)의 [“지정된 관리 권한 사용”](#)을 참조하십시오.

### 2. etherstub을 만듭니다.

```
global# dladm create-etherstub etherstub
```

### 3. etherstub에 VNIC를 만듭니다.

```
global# dladm create-vnic -l etherstub VNIC
```

이 절차의 뒷부분에서 VNIC의 인터페이스를 구성합니다.

### 4. 영역에 VNIC를 지정합니다.

```
global# zonecfg -z ZONE
zonecfg:ZONE> add net
zonecfg:ZONE:net> set physical=VNIC
zonecfg:ZONE:net> end
```

### 5. 구현한 변경 사항을 확인하고 커밋한 다음 영역을 종료합니다.

```
zonecfg:ZONE> verify
zonecfg:ZONE> commit
zonecfg:ZONE> exit
```



6. 영역을 재부트합니다.

```
global# zoneadm -z ZONE reboot
```

7. 영역에 로그인합니다.

```
global# zlogin zone
```

8. 영역에서 영역에 이제 지정된 VNIC를 통해 IP 인터페이스를 만듭니다.

```
zone# ipadm create-ip interface
```

9. 정적 IP 주소 또는 DHCP IP 주소를 사용하여 VNIC를 구성합니다.

- 정적 IP 주소를 지정합니다.

```
zone# ipadm create-addr -a address interface
```

- DHCP IP 주소를 지정합니다.

```
zone# ipadm create-addr -T dhcp interface
```

10. 영역을 종료합니다.

```
zone# exit
```

11. 전역 영역에서 주소 정보를 `/etc/hosts` 파일에 추가합니다.

12. 전역 영역에서 IP 전달을 수행하도록 기본 인터페이스를 설정합니다.

```
global# ipadm set-ifprop -p forwarding=on -m ipv4 primary-interface
```

---

참고 - Oracle Solaris에서 주 인터페이스는 NIC의 물리적 데이터 링크입니다.

---

13. 전역 영역에서 기본 인터페이스에 대해 `/etc/ipnat.conf` 파일에서 NAT(네트워크 주소 변환)를 구성합니다.

14. NAT를 사용으로 설정하도록 IP 필터 서비스를 시작합니다.

```
global# svcadm enable network/ipfilter
```

15. 영역을 재부트합니다.

```
global# zoneadm -z ZONE reboot
```

#### 예 2-6 개인 가상 네트워크 구성

이 예제에서 zone3은 개인 네트워크로 격리되도록 구성됩니다. NAT 및 IP 전달도 개인 가상 네트워크가 호스트 외부로 패킷을 전송하도록 구성되지만 개인 주소를 외부 네트워크로부터

숨깁니다. 영역은 이미 배타적 IP 유형으로 구성되어 있습니다. 하지만 IP 인터페이스는 영역에 지정되지 않습니다.

```

global# dladm create-etherstub ether0
global# dladm create-vnic -l ether0 vnic3
global# zonecfg -z zone3
zonecfg:zone3> add net
zonecfg:zone3:net> set physical=vnic3
zonecfg:zone3:net> end
zonecfg:zone3> verify
zonecfg:zone3> commit
zonecfg:zone3> exit

global# zoneadm -z zone3 reboot
global# zlogin zone3
zone3# ipadm create-ip vnic3
zone3# ipadm create-addr -a 192.168.0.10/24 vnic3
ipadm: vnic3/v4
zone3# exit

global# pfedit /etc/hosts
::1          localhost
127.0.0.1    localhost
192.168.3.70 loghost   #For net0
192.168.3.80 zone1     #using vnic1
192.168.3.85 zone2     #using vnic2
192.168.0.10 zone3     #using vnic3

global# ipadm set-ifprop -p forwarding=on -m ipv4 vnic3

global# pfedit /etc/ipf/ipnat.conf
map vnic3 192.168.0.0/24 -> 0/32 portmap tcp/udp auto
map vnic3 192.168.0.0/24 -> 0/32

global# svcadm enable network/ipfilter
global# zoneadm -z zone3 reboot

```

## VNIC 관리

이 단원에서는 기본 구성을 수행한 후 VNIC에서 수행할 수 있는 작업을 설명합니다. VNIC의 기본 구성을 수행하는 방법에 대한 자세한 내용은 [VNIC 및 Etherstub 구성 방법 \[22\]](#)을 참조하십시오.

VLAN ID, MAC 주소 및 VNIC의 기본 데이터 링크를 수정할 수 있습니다. 기본 데이터 링크를 수정하면 VNIC가 다른 데이터 링크로 이동됩니다. 데이터 링크의 모든 VNIC 속성을 전역적으로 수정하거나 지정된 VNIC의 속성만 선택적으로 수정할 수 있습니다.

이 단원에서는 다음 항목을 다룹니다.

- “VNIC 표시” [35]
- “VNIC의 VLAN ID 수정” [38]
- “VNIC MAC 주소 수정” [40]
- “VNIC 마이그레이션” [41]
- “VNIC 삭제” [43]

## VNIC 표시

시스템에서 VNIC에 대한 정보를 가져오려면 `dladm show-vnic` 명령을 사용합니다.

예 2-7 시스템의 VNIC 표시

```
# dladm show-vnic
LINK      OVER      SPEED      MACADDRESS      MACADDRTYPE      VIDS
vnic1     net0      1000       2:8:20:c2:39:38  random           123
vnic2     net0      1000       2:8:20:5f:84:ff  random           456
```

출력에는 다음 정보가 표시됩니다.

LINK	이름으로 식별된 가상 데이터 링크입니다.
OVER	VNIC를 구성하는 데 사용되는 물리적 데이터 링크 또는 가상 데이터 링크입니다.
SPEED	VNIC의 초당 메가비트 속도입니다.
MACADDRESS	VNIC의 MAC 주소입니다.
MACADDRTYPE	VNIC의 MAC 주소 유형이며, 다음 중 하나입니다. <ul style="list-style-type: none"> <li>■ random - VNIC에 지정된 임의 주소</li> <li>■ factory - VNIC에 사용되는 NIC의 출하 시 기본 MAC 주소</li> <li>■ fixed - 사용자가 지정한 MAC 주소</li> </ul>
VID	VNIC의 VLAN ID입니다.

VNIC가 시스템에 없는 경우 VNIC에 대한 정보를 포함하려면 데이터 링크에 대한 정보를 표시하는 `dladm` 명령을 사용할 수 있습니다. 예를 들어, `dladm show-link` 명령은 VNIC를 다른 데이터 링크와 함께 표시합니다. `dladm show-linkprop` 명령을 사용하여 VNIC의 등록 정보를 표시할 수 있습니다.

단일 VNIC의 데이터 링크 등록 정보에 대한 정보를 가져오려면 다음 명령 구문으로 VNIC를 지정합니다.

```
# dladm show-linkprop [-p property] vnic
```

예 2-8 영역에 연결된 VNIC 표시

이 예에서는 영역에 연결된 주 데이터 링크 및 VNIC에 대한 정보가 표시됩니다. 주 데이터 링크 net0은 전역 영역에 연결됩니다. vnic1 및 vnic2 VNIC는 각각 zone1 및 zone2 영역에 연결됩니다.

```
# dladm show-link -Z
LINK           ZONE    CLASS    MTU    STATE    OVER
net0           global  phys     1500   up       --
zone1/vnic1    zone1   vnic     1500   up       net0
zone2/vnic2    zone2   vnic     1500   up       net0
```

### MAC 주소가 여러 개인 VNIC 표시

여러 MAC 주소가 Oracle VM Server for SPARC의 시스템 생성 VNIC 및 Oracle Solaris 커널 영역의 anet 리소스에 연결되어 있습니다. Oracle VM Server for SPARC의 게스트 도메인 내에서 VNIC 및 영역을 지원하려면 alt-mac-addr 등록 정보를 사용하여 vnet을 만들어야 합니다. 이 경우 MAC 주소가 여러 개인 VNIC가 자동으로 생성됩니다. 이 여러 개의 MAC 주소는 생성된 vnet에서 가져옵니다. 자세한 내용은 [Oracle VM Server for SPARC 3.1 Administration Guide](#)를 참조하십시오.

커널 영역 내에서 영역 또는 VNIC를 지원하려면 여러 MAC 주소를 사용하여 anet 리소스를 구성합니다. zonecfg 명령을 사용하여 커널 영역에서 네트워크 액세스를 위해 생성된 anet 리소스에 여러 MAC 주소를 지정합니다. 자세한 내용은 [solaris-kz\(5\)](#) 매뉴얼 페이지를 참조하십시오. 커널 영역을 구성하는 방법에 대한 자세한 내용은 “[Oracle Solaris 커널 영역 만들기 및 사용](#)”을 참조하십시오.

여러 MAC 주소가 VNIC에 연결되어 있는 경우 MAC 주소 하나는 가상 네트워크 드라이버에서 사용됩니다. 나머지 MAC 주소를 사용하여 커널 영역 또는 게스트 도메인 내에서 VNIC를 만들 수 있습니다. 예를 들어, VNIC가 세 개의 MAC 주소와 연결되어 있는 경우 MAC 주소 하나는 가상 네트워크 드라이버 용으로 지정됩니다. 따라서 나머지 두 MAC 주소를 사용하여 2개의 VNIC만 만들 수 있습니다.

다음 명령을 사용하여 VNIC에 연결된 여러 MAC 주소를 표시할 수 있습니다.

```
# dladm show-vnic -m
```

예 2-9 커널 영역에서 MAC 주소가 여러 개인 VNIC 표시

```
# dladm show-vnic -m
LINK           OVER     MACADDRESSES    MACADRTYPES    VIDS
gz_vnic0       net0     2:8:20:d7:27:9d  random         0
zone1/net0     net0     2:8:20:70:52:9   random         0
               2:8:20:c9:d:4c   fixed
               2:8:20:70:db:3   random
zone1/net1     net0     0:1:2:3:4:5     fixed         0
               0:1:2:3:4:6     fixed
```

이 예에서 커널 영역 zone1에는 두 개의 anet 리소스 net0 및 net1이 있습니다. 두 리소스는 모두 MAC 주소가 여러 개 구성되어 있습니다. 따라서 커널 영역 zone1 내에서 데이터 링크 net0에 연결된 가상 NIC 드라이버 zvnet 위에 최대 두 개의 VNIC를 만들 수 있습니다. 데이터 링크 net1에 연결된 가상 NIC 드라이버 zvnet 위에 VNIC를 하나만 만들 수 있습니다.

예 2-10 MAC 주소가 여러 개인 시스템 생성 VNIC 표시

```
# dladm show-vnic -m
LINK          OVER      MACADDRESSES  MACADDRTYPES  VIDS
ldoms-vsw0.vport0 net1      0:14:4f:fb:e1:8f  fixed          0,21
              0:14:4f:f8:6b:9   fixed
              0:14:4f:fa:48:7f  fixed
ldoms-vsw0.vport1 net1      0:14:4f:f9:1b:8d  fixed          45,44
              0:14:4f:f9:27:4   fixed
```

이 예에서는 ldoms-vsw0.vport0에 연결된 게스트 도메인의 가상 네트워크 드라이버 vnet 위에 최대 두 개의 VNIC를 만들 수 있습니다. ldoms-vsw0.vport1에 연결된 가상 NIC 드라이버 vnet 위에 최대 한 개의 VNIC를 만들 수 있습니다.

## 데이터 링크의 물리적 링크 상태 및 가상 링크 상태 표시

데이터 링크의 물리적 링크 상태는 물리적 장치가 외부 네트워크에 연결되어 있는지 여부를 식별합니다. 케이블이 연결되어 있고 케이블의 다른 쪽 끝에 있는 포트의 상태가 up인 경우 물리적 장치는 외부 네트워크에 연결되어 있습니다.

다음 명령을 사용하여 데이터 링크의 물리적 링크 상태를 표시할 수 있습니다.

```
# dladm show-phys [link]
# dladm show-ether [link]
```

자세한 내용은 [dladm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

예 2-11 데이터 링크의 물리적 링크 상태 표시

다음 예에서는 dladm show-phys 명령을 사용하여 시스템에 있는 데이터 링크의 물리적 링크 상태를 표시합니다.

```
# dladm show-phys
LINK      MEDIA      STATE    SPEED  DUPLEX  DEVICE
net1      Ethernet  down    0      unknown e1000g1
net2      Ethernet  down    0      unknown e1000g2
net3      Ethernet  down    0      unknown e1000g3
net0      Ethernet  up      1000   full    e1000g0
```

다음 예에서는 dladm show-ether 명령을 사용하여 시스템에 있는 데이터 링크의 물리적 링크 상태를 표시합니다.

```
# dladm show-ether
LINK      PTYPE  STATE  AUTO  SPEED-DUPLEX  PAUSE
net1      current down   yes   0M           bi
net2      current down   yes   0M           bi
net3      current down   yes   0M           bi
net0      current up     yes   1G-f        bi
```

NIC를 통해 여러 VNIC를 만들 경우 VNIC 및 주 데이터 링크가 동일한 VLAN에 있을 때 통신할 수 있도록 가상 스위치가 내부적으로 생성됩니다. 이러한 데이터 링크는 물리적 데이터 링크가 외부 네트워크에 연결되어 있지 않더라도 서로 통신할 수 있습니다. 그러면 데이터 링크의 가상 링크 상태가 결정되고, 이러한 상태는 up, down 또는 unknown이 될 수 있습니다. 데이터 링크의 가상 링크 상태는 물리적 케이블이 분리되어 있는 상태에서도 데이터 링크가 시스템 내에서 내부 네트워크에 연결되는지 여부를 식별합니다.

다음 명령을 사용하여 데이터 링크의 가상 링크 상태를 표시합니다.

```
# dladm show-link [link]
```

#### 예 2-12 데이터 링크의 가상 링크 상태 표시

이 예에서는 시스템에 있는 데이터 링크의 가상 링크 상태를 표시합니다.

```
# dladm show-link
LINK      CLASS  MTU   STATE  OVER
net0      phys   1500  up     --
net2      phys   1500  down   --
net4      phys   1500  down   --
net1      phys   1500  up     --
net5      phys   1500  up     --
vnic0     vnic   1500  up     net5
vnic1     vnic   1500  up     net5
vnic2     vnic   1500  up     net1
```

## VNIC의 VLAN ID 수정

VNIC는 VLAN으로 구성할 수 있습니다. VNIC가 특정 VLAN의 트래픽을 호스트하도록 하려면 데이터 링크에서 VNIC의 VLAN ID를 수정해야 합니다.

사용되는 dladm 하위 명령은 VLAN을 수정하는지 VLAN으로 구성된 VNIC를 수정하는지에 따라 달라집니다.

- dladm create-vlan 명령을 사용하여 만드는 VLAN의 경우 dladm modify-vlan 명령을 사용합니다. 이러한 VLAN을 표시하려면 dladm show-vlan 명령을 사용합니다.
- dladm create-vnic 명령을 사용하여 만드는 VLAN의 경우 dladm modify-vnic 명령을 사용합니다. VLAN ID가 있는 VNIC를 포함하여 이러한 VNIC를 표시하려면 dladm show-vnic 명령을 사용합니다.

데이터 링크에 구성된 단일 VNIC 또는 여러 VNIC의 VLAN ID를 수정할 수 있습니다. VLAN ID를 가진 모든 VNIC를 구성하여 VNIC의 VLAN ID를 그룹으로 수정할 수도 있습니다.

- 데이터 링크에 VNIC가 하나만 구성되어 있는 경우 다음 명령 구문을 사용하여 VNIC의 VLAN ID를 수정합니다.

```
# dladm modify-vnic -v vid -L link
```

여기서 *vid*는 VNIC에 지정된 새로운 VLAN ID입니다.

#### 예 2-13 데이터 링크에서 VNIC의 VLAN ID 수정

이 예에서는 데이터 링크 *net0*을 통해 구성된 *vnic0*의 VLAN ID가 수정됩니다.

```
# dladm modify-vnic -v 123 -L net0
# dladm show-vnic
LINK      OVER      SPEED      MACADDRESS      MACADDRTYPE      VIDS
vnic0     net0      1000       2:8:20:c2:39:38  random           123
```

- 데이터 링크에 여러 VNIC가 구성되어 있는 경우 다음 명령 구문을 사용하여 VNIC의 VLAN ID를 수정합니다.

```
# dladm modify-vnic -v vid VNIC
```

각 VLAN ID는 동일 데이터 링크의 VNIC에 대해 고유하므로 한 번에 하나씩 VLAN ID를 변경해야 합니다.

#### 예 2-14 데이터 링크에서 여러 VNIC의 VLAN ID 수정

이 예에서는 *vnic0*, *vnic1* 및 *vnic2*의 VLAN ID가 수정됩니다.

```
# dladm modify-vnic -v 123 vnic0
# dladm modify-vnic -v 456 vnic1
# dladm modify-vnic -v 789 vnic2
# dladm show-vnic
LINK      OVER      SPEED      MACADDRESS      MACADDRTYPE      VIDS
vnic0     net0      1000       2:8:20:c2:39:38  random           123
vnic1     net0      1000       2:8:20:5f:84:ff  random           456
vnic2     net0      1000       2:8:20:5f:84:ff  random           789
```

- 데이터 링크별로 다른 VNIC가 구성되어 있는 경우 다음 명령 구문을 사용하여 VNIC의 VLAN ID를 그룹으로 수정합니다.

```
# dladm modify-vnic -v vid VNIC,VNIC,[...]
```

#### 예 2-15 VNIC의 VLAN ID를 그룹으로 수정

이 예에서는 *vnic0*, *vnic1* 및 *vnic2*의 VLAN ID를 그룹으로 수정합니다. 이러한 VNIC는 각각 *net0*, *net1* 및 *net2* 데이터 링크를 통해 구성됩니다.

```
# dladm modify-vnic -v 123 vnic0,vnic1,vnic2
# dladm show-vnic
LINK      OVER    SPEED      MACADDRESS      MACADDRTYPE      VIDS
vnic0     net0    1000       2:8:20:c2:39:38  random           123
vnic1     net1    1000       2:8:20:5f:84:ff  random           123
vnic2     net2    1000       2:8:20:5f:84:ff  random           123
```

## VNIC MAC 주소 수정

사용자가 만드는 모든 VNIC는 MAC 주소를 하나만 가질 수 있습니다. `dladm modify-vnic` 명령을 사용하여 MAC 주소를 수정할 수 있습니다. 하나 이상의 MAC 주소를 가진 커널 영역에 대해 생성된 VNIC를 구성할 수 있습니다.

데이터 링크에 구성된 VNIC의 기존 MAC 주소를 수정할 수 있습니다. 모든 VNIC의 MAC 주소를 수정하거나 지정된 VNIC의 MAC 주소를 선택적으로 수정할 수 있습니다. VNIC의 MAC 주소와 VLAN ID를 동시에 수정할 수도 있습니다.

- VNIC의 MAC 주소를 수정하려면 다음 명령 구문을 사용합니다.

```
# dladm modify-vnic -m MAC-address VNIC
```

여기서 *MAC-address*는 VNIC에 지정할 새로운 MAC 주소입니다.

### 예 2-16 VNIC의 MAC 주소 수정

이 예에서는 `vnic0`에 특정 MAC 주소가 지정됩니다.

```
# dladm modify-vnic -m 3:8:20:5f:84:ff vnic0
# dladm show-vnic
LINK      OVER    SPEED      MACADDRESS      MACADDRTYPE      VIDS
vnic0     net0    1000       3:8:20:5f:84:ff  fixed             0
```

- 데이터 링크에 있는 모든 VNIC의 MAC 주소를 수정하려면 다음 명령 구문을 사용합니다.

```
# dladm modify-vnic -m random -L link
```

이 명령 구문에서 `-m random` 옵션은 `-m auto` 옵션과 동일합니다. MAC 주소는 VNIC에 임의로 자동 지정됩니다.

### 예 2-17 데이터 링크에 있는 모든 VNIC의 MAC 주소 수정

이 예에서는 데이터 링크 `net0`을 통해 구성된 모든 VNIC의 MAC 주소가 임의로 자동 수정됩니다.

```
# dladm modify-vnic -m random -L net0
# dladm show-vnic
```



LINK	OVER	SPEED	MACADDRESS	MACADDRTYPE	VIDS
vnic0	net0	1000	2:8:20:22:9d:bb	random	0
vnic1	net0	1000	2:8:20:72:2e:9	random	0
vnic2	net0	1000	2:8:20:2f:e5:83	random	0

- VNIC의 MAC 주소를 선택적으로 수정하려면 다음 명령 구문을 사용합니다.

```
# dladm modify-vnic -m random VNIC,VNIC,[...]
```

전역 수정 및 선택적 수정 모두 -m 옵션에 대해 random을 지정합니다.

예 2-18 데이터 링크에서 VNIC의 MAC 주소를 선택적으로 수정

이 예에서는 데이터 링크 net0을 통해 구성된 vnic0 및 vnic2의 MAC 주소가 선택적으로 수정됩니다.

```
# dladm modify-vnic -m random vnic0,vnic2
# dladm show-vnic
```

LINK	OVER	SPEED	MACADDRESS	MACADDRTYPE	VIDS
vnic0	net0	1000	2:8:20:2f:e5:83	random	0
vnic1	net0	1000	2:8:20:5f:84:ff	fixed	0
vnic2	net0	1000	2:8:20:2f:e5:83	random	0

- VNIC의 MAC 주소와 VLAN ID를 동시에 수정하려면 다음 명령 구문을 사용합니다.

```
# dladm modify-vnic -m random -v vid VNIC
```



주의 - VNIC의 여러 속성을 전역적으로 수정하면 VNIC에서 예기치 않은 동작이 발생할 수 있습니다. 대신 VNIC의 여러 속성을 개별적으로 수정합니다.

예 2-19 VNIC의 MAC 주소 및 VLAN ID 수정

이 예에서는 vnic0의 MAC 주소와 VLAN ID가 동시에 수정됩니다.

```
# dladm modify-vnic -m random -v 123 vnic0
# dladm show-vnic vnic0
```

LINK	OVER	SPEED	MACADDRESS	MACADDRTYPE	VIDS
vnic0	net0	1000	2:8:20:2f:e5:83	random	123

## VNIC 마이그레이션

VNIC를 삭제하고 재구성하지 않고 하나 이상의 VNIC를 한 기본 데이터 링크에서 다른 기본 데이터 링크로 이동할 수 있습니다. 기본 데이터 링크는 물리적 링크, 링크 집계 또는 etherstub일 수 있습니다.

일반적으로 다음과 같은 경우에 VNIC를 마이그레이션합니다.

- 기존 NIC를 새 NIC로 바꾸어야 하는 경우
- 대상 NIC가 기존 NIC보다 대역폭이 많은 경우
- 대상 NIC가 하드웨어에서 특정 기능을 구현하는 경우(예: LRO(Large Receive Offload), LSO(Large Segment Offload), 체크섬)

VNIC를 성공적으로 마이그레이션하기 위해서는 VNIC를 이동할 대상 데이터 링크가 VNIC의 데이터 링크 등록 정보를 수용할 수 있어야 합니다. 해당 등록 정보가 지원되지 않는 경우 마이그레이션이 실패하고 사용자에게 통지됩니다. 마이그레이션을 성공한 후 대상 데이터 링크가 네트워크에 연결되어 있는 경우 VNIC를 사용하는 모든 응용 프로그램이 정상적으로 작동합니다.

데이터 링크 상태, 링크 속도 및 MTU 크기와 같은 특정 하드웨어에 종속된 등록 정보는 VNIC 마이그레이션 이후 변경될 수 있습니다. 이러한 등록 정보의 값은 VNIC가 마이그레이션된 데이터 링크에서 상속됩니다. 데이터 링크를 통해 구성된 모든 VNIC를 마이그레이션하거나 지정된 VNIC를 선택적으로 마이그레이션할 수 있습니다. VNIC를 마이그레이션하고 동시에 VLAN ID를 수정할 수도 있습니다.

- 소스 링크를 통해 구성된 모든 VNIC를 대상 링크에 마이그레이션하려면 다음 명령 구문을 사용합니다.

```
# dladm modify-vnic -l target-link -L source-link
```

-l *target-link*            VNIC를 마이그레이션하는 데 사용되는 링크를 나타냅니다.

-L *source-link*            VNIC를 이전에 구성하는 데 사용된 링크를 나타냅니다.

**예 2-20**            모든 VNIC를 소스 링크에서 대상 링크로 마이그레이션

이 예에서는 소스 링크 ether0의 모든 VNIC가 대상 링크 net1로 이동됩니다.

```
# dladm modify-vnic -l net1 -L ether0
```

```
# dladm show-vnic
```

LINK	OVER	SPEED	MACADDRESS	MACADDRTYPE	VIDS
vnic0	net1	1000	2:8:20:c2:39:38	random	321
vnic1	net1	1000	2:8:20:5f:84:ff	random	656
vnic2	net1	1000	2:8:20:5f:84:ff	random	0

- 소스 링크를 통해 구성된 지정된 VNIC를 대상 링크로 마이그레이션하려면 다음 명령 구문을 사용합니다.

```
# dladm modify-vnic -l target-link VNIC,VNIC,[...]
```

선택적 VNIC 마이그레이션을 수행하려면 대상 링크만 지정해야 합니다.

**예 2-21**            지정된 VNIC를 소스 링크에서 대상 링크로 마이그레이션

이 예에서는 vnic0, vnic1 및 vnic2가 소스 링크 net0에서 대상 링크 net1로 선택적으로 이동됩니다.

```
# dladm modify-vnic -l net1 vnic0,vnic1,vnic2
# dladm show-vnic
LINK      OVER      SPEED      MACADDRESS      MACADDRTYPE      VIDS
vnic0     net1      1000      2:8:20:c2:39:38  random           321
vnic1     net1      1000      2:8:20:5f:84:ff  random           656
vnic2     net1      1000      2:8:20:5f:84:ff  random           0
vnic3     net0      1000      2:8:20:5f:84:ff  random           345
```

- 소스 링크를 통해 구성된 VNIC의 VLAN ID를 수정하는 동시에 대상 링크로 마이그레이션하려면 다음 명령 구문을 사용합니다.

```
# dladm modify-vnic -l target-link -v vid VNIC
```

새 VLAN ID를 지정하려면 VNIC를 한 번에 하나씩 마이그레이션해야 합니다.

#### 예 2-22 VNIC의 VLAN ID 마이그레이션 및 수정

이 예에서는 vnic0, vnic1 및 vnic2가 대상 데이터 링크 net1로 마이그레이션됩니다. 마이그레이션과 동시에 모든 VNIC의 VLAN ID가 수정됩니다.

```
# dladm modify-vnic -l net1 -v 123 vnic0
# dladm modify-vnic -l net1 -v 456 vnic1
# dladm modify-vnic -l net1 -v 789 vnic2
# dladm show-vnic
LINK      OVER      SPEED      MACADDRESS      MACADDRTYPE      VIDS
vnic0     net1      1000      2:8:20:c2:39:38  random           123
vnic1     net1      1000      2:8:20:5f:84:ff  random           456
vnic2     net1      1000      2:8:20:5f:84:ff  random           789
```

VNIC를 소스 링크에서 대상 링크로 마이그레이션할 때 임의로 지정된 MAC 주소는 영향을 받지 않고 마이그레이션 후에도 해당 VNIC에 보존됩니다. [예 2-22. “VNIC의 VLAN ID 마이그레이션 및 수정”](#)을 참조하십시오.

하지만 VNIC가 소스 링크의 출하 시 기본 MAC 주소를 사용 중인 경우에는 MAC 주소가 변경됩니다. 마이그레이션 중에 MAC 주소를 지정하지 않은 경우 VNIC의 출하 시 기본 MAC 주소가 임의로 지정된 MAC 주소로 대체됩니다. 마이그레이션 중에 -m 옵션을 사용하여 MAC 주소를 지정할 경우 VNIC의 출하 시 기본 MAC 주소가 지정된 MAC 주소로 대체됩니다.

커널 영역에서 만든 VNIC에 여러 MAC 주소가 연결되어 있습니다. 커널 영역에서 생성된 VNIC를 마이그레이션할 경우 VNIC에 연결된 여러 MAC 주소가 모두 대상 NIC에 마이그레이션됩니다.

## VNIC 삭제

이 절에서는 VNIC를 삭제하는 방법을 설명합니다.

## ▼ VNIC 삭제 방법

1. 관리자로 로그인합니다.
2. (옵션) VNIC가 사용 중인지 확인합니다.

VNIC는 사용하고 있지 않은 경우에만 삭제할 수 있습니다. VNIC는 여러 가지 이유로 사용 중일 수 있습니다. 다음 단계를 수행하여 VNIC가 사용 중인지 확인해야 합니다.

### ■ VNIC가 IP 주소에 연결되어 있는지 확인합니다.

```
# ipadm show-if
# ipadm show-addr
```

VNIC가 IP 주소에 연결되어 있는 경우 IP 인터페이스를 제거합니다.

```
# ipadm delete-ip interface
```

### ■ VNIC를 통해 구성된 플로우가 있는지 확인합니다.

```
# flowadm
```

플로우가 VNIC를 통해 구성되어 있는 경우 플로우를 제거합니다.

```
# flowadm remove-flow flowname
```

### ■ VNIC가 영역에 지정되어 있는지 확인합니다.

```
# dladm show-link -Z
```

영역에 연결된 VNIC를 삭제하는 방법에 대한 자세한 내용은 [영역에 연결된 VNIC를 삭제하는 방법 \[45\]](#)을 참조하십시오.

### ■ VNIC가 시스템에서 생성되는지 확인합니다.

```
# dladm show-vnic
```

시스템 생성 VNIC에만 하이픈(-)을 사용할 수 있으므로, 시스템 생성 VNIC와 사용자 생성 VNIC를 구별하는 데 도움이 됩니다. 시스템 생성 VNIC를 수정, 이름 바꾸기, 연결(plumb) 또는 삭제할 수 없습니다.

### ■ VNIC가 스누핑에 노출되는지 확인합니다.

```
# snoop
# tshark
```

snoop 명령을 사용하여 VNIC가 스누핑에 노출되는 경우 프로세스를 종료합니다.

```
# pkill snoop
```

tshark 명령을 사용하여 VNIC가 스누핑에 노출되는 경우 프로세스를 종료합니다.

```
# pkill tshark
```

3. VNIC를 삭제합니다.

```
# dladm delete-vnic VNIC
```

## ▼ 영역에 연결된 VNIC를 삭제하는 방법

이 절차에서는 VNIC가 영역에 연결되어 있다고 가정합니다. 이 절차를 수행하려면 사용자가 전역 영역에 있어야 합니다.

1. 영역을 중지합니다.

```
global# zoneadm -z ZONE halt
```

---

참고 - 영역에 사용된 링크를 확인하려면 `dladm show-link` 명령을 사용합니다.

---

2. 영역에서 VNIC를 제거하거나 분리합니다.

```
global# zonecfg -z ZONE remove net physical=VNIC
```

3. 시스템에서 VNIC를 삭제합니다.

```
global# dladm delete-vnic VNIC
```

4. 영역을 재부트합니다.

```
global# zoneadm -z ZONE boot
```

### 예 2-23 영역에 연결된 VNIC 삭제

이 예제에서 `vnic1`은 `zoneB` 및 시스템에서 제거됩니다.

```
global# dladm show-link
LINK          CLASS  MTU   STATE  OVER
net0          phys  1500  up     --
net2          phys  1500  up     --
net1          phys  1500  up     --
net3          phys  1500  up     --
zoneA/net0    vnic   1500  up     net0
zoneB/net0    vnic   1500  up     net0
vnic0         vnic   1500  up     net1
zoneA/vnic0   vnic   1500  up     net1
vnic1         vnic   1500  up     net1
zoneB/vnic1   vnic   1500  up     net1

global# zoneadm -z zoneB halt
```

```
global# zonecfg -z zoneB remove net physical=vnic1
global# dladm delete-vnic vnic1
global# zoneadm -z zoneB boot
```

## VNIC와 함께 단일 루트 I/O 가상화 사용

Oracle Solaris 11.2 릴리스부터 `dladm` 명령을 사용하여 SR-IOV(단일 루트 I/O 가상화)를 지원하는 네트워크 장치를 관리할 수 있습니다. SR-IOV는 가상 머신 간에 PCIe(Peripheral Component Interconnect Express) 장치를 효율적으로 공유할 수 있도록 해주는 표준입니다. 하드웨어에서 구현됩니다. 자세한 내용은 [“Writing Device Drivers for Oracle Solaris 11.2”](#)의 [“Introduction to SR-IOV”](#)를 참조하십시오.

## 데이터 링크의 SR-IOV 모드 사용

Oracle Solaris에서는 SR-IOV를 지원하는 네트워크 장치의 VF(가상 기능)를 VNIC 또는 VLAN과 연결할 수 있습니다. VF VNIC는 전용 VF를 소유한 VNIC입니다. VF VNIC는 리소스 공유에서 일반 VNIC와 다릅니다. 일반 VNIC는 다른 일반 VNIC와 리소스를 공유해야 하지만, VF VNIC는 리소스를 공유할 필요가 없습니다. 각 VF는 VF VNIC에 대한 별도의 하드웨어 리소스입니다.

SR-IOV 모드를 지원하는 데이터 링크에 대해서만 VF VNIC를 만들 수 있습니다. 기본적으로 데이터 링크의 SR-IOV 모드는 사용 안함으로 설정됩니다. `iovs` 등록 정보를 `on`으로 설정하여 데이터 링크의 SR-IOV 모드를 사용으로 설정할 수 있습니다. 데이터 링크의 SR-IOV 모드를 사용으로 설정한 이후에 VF VNIC를 만드는 방법에 대한 자세한 내용은 [“VF VNIC 만들기” \[47\]](#)를 참조하십시오.

`dladm show-linkprop` 명령으로 링크 등록 정보 `iovs`를 지정하여 데이터 링크의 SR-IOV 모드를 확인할 수 있습니다. 출력의 `EFFECTIVE` 열 아래에 있는 값이 `off`인 경우 데이터 링크의 SR-IOV 모드가 사용 안함으로 설정됩니다.

다음 예에서는 데이터 링크 `net0`의 SR-IOV 모드를 확인하는 방법을 보여줍니다.

```
# dladm show-linkprop -p iovs net0
LINK    PROPERTY  PERM    VALUE    EFFECTIVE  DEFAULT  POSSIBLE
net0    iovs      rw      auto     off        auto     auto,on,off
```

이 예에서는 데이터 링크 `net0`의 SR-IOV 모드가 사용 안함으로 설정됩니다. 출력에는 다음 정보가 표시됩니다.

VALUE                    `iovs` 링크 등록 정보에 대해 설정한 값을 지정합니다. `iovs` 링크 등록 정보를 수정하지 않은 경우 `iovs` 링크 등록 정보의 기본값은 `auto`입니다. 값이 `auto`이면 특정 물리적 데이터 링크에서 SR-IOV 모드가 기본적으로 사용으로 설정되는지 여부를 OS에서 결정합니다.

EFFECTIVE 데이터 링크의 실제 SR-IOV 모드입니다. 기본적으로 모든 SRIOV 가능 NIC는 EFFECTIVE 열 아래에 `off` 값이 표시됩니다.

다음과 같이 `iovs` 등록 정보를 `on`으로 설정하여 데이터 링크 `net0`의 SR-IOV 모드를 사용으로 설정할 수 있습니다.

```
# dladm set-linkprop -p iov=on net0
# dladm show-linkprop -p iov net0
LINK      PROPERTY  PERM    VALUE    EFFECTIVE  DEFAULT  POSSIBLE
net0     iov       rw      on       on         auto     auto,on,off
```

마찬가지로 `iovs` 링크 등록 정보를 `off`로 설정하여 데이터 링크의 SR-IOV 모드를 사용 안함으로 설정할 수 있습니다. `dladm` 명령에 대한 자세한 내용은 [dladm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## VF VNIC 만들기

데이터 링크에서 VF VNIC를 만들려면 데이터 링크의 SR-IOV 모드를 사용으로 설정해야 합니다. 자세한 내용은 “[데이터 링크의 SR-IOV 모드 사용](#)” [46]을 참조하십시오. 데이터 링크의 SR-IOV 모드를 사용으로 설정한 후 `dladm create-vnic` 명령을 사용하여 VNIC를 만들면 VF가 VNIC에 자동으로 할당됩니다. 마찬가지로 `dladm create-vlan` 명령을 사용하여 VLAN을 만들면 VF가 VLAN에 자동으로 할당됩니다.

`dladm create-vnic` 또는 `dladm create-vlan` 명령으로 `iovs` VNIC 링크 등록 정보를 지정하여 VF를 VNIC 또는 VLAN에 할당해야 하는지 여부를 명시적으로 지정할 수도 있습니다.

다음 명령 구문을 사용하여 VF VNIC를 명시적으로 만듭니다.

```
# dladm create-vnic [-p iov=value] -l link VNIC
```

VF VNIC를 만들 경우 `iovs` VNIC 링크 등록 정보 지정은 선택 사항입니다. `iovs` VNIC 링크 등록 정보를 지정하지 않은 경우 기본값인 `inherit`이 이 등록 정보에 지정됩니다. `iovs` VNIC 링크 등록 정보에 대해 다음 값을 지정할 수 있습니다.

- `inherit` `iovs` VNIC 링크 등록 정보의 기본값입니다. 기본 데이터 링크의 유효 `iovs` 등록 정보 값을 기반으로 VF를 할당해야 하는지 여부를 결정합니다.
  - `off` - VNIC에 대해 VF를 할당하지 않습니다.
  - `on` - VNIC에 대해 VF를 할당하려고 시도합니다. 가능하지 않은 경우 일반 VNIC가 생성됩니다.
- `on` VF를 할당합니다. VF를 찾을 수 없는 경우 VNIC 만들기가 실패합니다.
- `off` VF를 사용하지 않고 VNIC를 만듭니다.

데이터 링크에 대해 `dladm show-linkprop` 명령을 사용할 경우 데이터 링크 등록 정보의 유효 값은 EFFECTIVE 열 아래에 표시된 값입니다.

iov VNIC 링크 등록 정보와 다른 데이터 링크 등록 정보의 차이점은 VNIC 또는 VLAN을 만들 때에만 iov VNIC 링크 등록 정보를 지정할 수 있다는 것입니다. VNIC 또는 VLAN을 만든 이후에는 iov VNIC 링크 등록 정보를 수정할 수 있습니다.

iov VNIC 링크 등록 정보는 VF가 VNIC 또는 VLAN에 대해 할당되는지 여부를 나타내는 유효 값이 있습니다. EFFECTIVE 열 아래에 on 값이 있으면 VF가 할당되고, EFFECTIVE 열 아래에 off 값이 있으면 VF가 할당되지 않습니다.

#### 예 2-24 VF VNIC 만들기

다음 예에서는 iov VNIC 링크 등록 정보를 명시적으로 지정하여 데이터 링크 net0에서 VF VNIC vfvnic1 및 일반 VNIC vnic1을 만드는 방법을 보여줍니다. 이 예에서는 데이터 링크 net0의 SR-IOV 모드를 사용으로 설정했다고 가정합니다.

```
# dladm show-linkprop -p iov net0
LINK    PROPERTY  PERM   VALUE  EFFECTIVE  DEFAULT  POSSIBLE
net0    iov       rw     on     on         auto     auto,on,off
# dladm create-vnic -l net0 vfvnic1
# dladm show-linkprop -p iov vfvnic1
LINK    PROPERTY  PERM   VALUE  EFFECTIVE  DEFAULT  POSSIBLE
vfvnic1 iov       r-     inherit on         inherit  inherit,on,off
# dladm create-vnic -p iov=off -l net0 vnic1
# dladm show-linkprop -p iov vnic1
LINK    PROPERTY  PERM   VALUE  EFFECTIVE  DEFAULT  POSSIBLE
vnic1   iov       r-     off    off        inherit  inherit,on,off
```

이 예에서는 다음 정보를 제공합니다.

- VF VNIC를 만들기 전에 데이터 링크 net0에 대한 iov 등록 정보를 on으로 설정해야 합니다.
- VNIC를 만들 때 iov 등록 정보의 값을 지정하지 않은 경우 기본값인 inherit이 iov 등록 정보에 지정됩니다. VF VNIC vfvnic1이 VF를 사용하여 생성됩니다.
- VNIC를 만들 때 iov 등록 정보에 대해 off 값을 명시적으로 지정한 경우 기본 데이터 링크 net0의 iov 등록 정보가 on이더라도 일반 VNIC는 VF 없이 생성됩니다. VNIC vnic1이 VF 없이 생성됩니다.

## VF VNIC 마이그레이션

VF VNIC 또는 VF VLAN을 데이터 링크에서 다른 데이터 링크로 이동할 수 있습니다. 다음 요구 사항을 확인합니다.

- 대상 데이터 링크가 SR-IOV를 지원하고 iov 등록 정보가 on으로 설정되어 있어야 합니다. 데이터 링크에 대한 iov 등록 정보의 상태를 확인하는 방법에 대한 자세한 내용은 [“데이터 링크의 SR-IOV 모드 사용” \[46\]](#)을 참조하십시오.
- 대상 데이터 링크에서 VF를 사용할 수 있어야 합니다. 데이터 링크에서 사용 가능한 VF 수를 확인하는 방법에 대한 자세한 내용은 [“VF 정보 표시” \[49\]](#)를 참조하십시오.



이러한 요구 사항이 충족되지 않을 경우 VF VNIC는 대상 데이터 링크에 VF 없는 일반 VNIC로 마이그레이션됩니다.

`iov=inherit`을 지정하여 만든 VF VNIC를 마이그레이션할 경우 대상 데이터 링크에서 `iov` 등록 정보를 지원하지 않거나 `iov` 등록 정보가 사용 안함으로 설정되어 있는 경우에도 마이그레이션은 진행됩니다. `iov=on`을 사용하여 만든 VF VNIC를 마이그레이션할 경우 대상 데이터 링크에서 SR-IOV 모드가 사용으로 설정된 경우에만 마이그레이션이 진행됩니다.

VNIC를 마이그레이션하는 방법에 대한 자세한 내용은 [“VNIC 마이그레이션” \[41\]](#)을 참조하십시오.

## VF 정보 표시

다음 명령을 사용하여 데이터 링크에서 VF의 가용성에 대한 정보를 표시할 수 있습니다.

```
# dladm show-phys -V
```

출력에는 다음 정보가 표시됩니다.

LINK	데이터 링크의 이름입니다.
VFS-AVAIL	VNIC에 지정될 수 있는 데이터 링크에서 사용 가능한 VF 수입니다. 데이터 링크에서 SR-IOV를 지원하지 않는 경우 <code>VFS-AVAIL</code> 이 --로 표시됩니다.
VFS-INUSE	데이터 링크에 사용되는 VF 수입니다. 데이터 링크에서 SR-IOV를 지원하지 않는 경우 <code>VFS-INUSE</code> 가 --로 표시됩니다.
FLAGS	1 플래그는 데이터 링크가 Oracle VM Server for SPARC에서 관리됨을 나타냅니다.

**예 2-25**      데이터 링크에 대한 VF 정보 표시

```
# dladm show-phys -V
LINK      VFS-AVAIL  VFS-INUSE  FLAGS
net0      30         1          -----
net1      0          0          1-----
net2      --         --         -----
```

이 예에서 데이터 링크 `net0`에는 30개의 사용 가능한 VF와 한 개의 사용 중인 VF가 있습니다. 데이터 링크 `net1`은 0개의 사용 가능한 VF가 있고, 현재 Oracle VM Server for SPARC에서 사용되고 있습니다. 데이터 링크 `net2`는 SR-IOV를 지원하지 않습니다.

다음 명령을 사용하여 시스템의 VNIC에 지정된 VF 장치를 표시할 수 있습니다.

```
# dladm show-vnic -V
```

출력에는 다음 정보가 표시됩니다.

LINK	VNIC의 이름입니다.
VF-ASSIGNED	VNIC에 지정된 VF 장치입니다. VNIC에 VF가 없는 경우 VF-ASSIGNED가 --로 표시됩니다.

예 2-26 VNIC에 지정된 VF 장치 표시

```
# dladm show-vnic -V
LINK          VF-ASSIGNED
vnic1        ixgbevf0
vnic2        --
vnic3        ixgbevf1
```

이 예에서는 VF 장치 ixgbevf0이 vnic1에 지정됩니다. VNIC vnic2에는 할당된 VF 장치가 없습니다. VF 장치 ixgbevf1이 vnic3에 지정됩니다.

## VXLAN(Virtual Extensible Local Area Network)을 사용하여 가상 네트워크 구성

---

VLAN(Virtual Local Area Network)과 같은 기존 네트워크 격리 방법은 대규모 데이터 센터에서 가상화를 지원하는 데 적합하지 않습니다. 클라우드 환경은 기본 물리적 네트워크와 긴밀하게 결합되므로 다른 물리적 계층 2 네트워크에 속하는 물리적 서버 간에 가상 머신을 마이그레이션할 수 없습니다. Oracle Solaris는 대규모 가상화된 데이터 센터 또는 클라우드 환경에서 그런 가상화 문제를 해결하는 VXLAN(Virtual Extensible Local Area Network) 기술을 지원합니다.

이 장에서는 VXLAN 배포의 개요를 제공하고 VXLAN을 구성하는 방법을 설명합니다. 또한 영역과 같은 다른 기술과 함께 VXLAN을 사용하는 방법을 설명합니다.

이 장의 내용:

- “VXLAN 개요” [51]
- “VXLAN 사용 시의 이점” [52]
- “VXLAN 이름 지정 규칙” [52]
- “VXLAN 토폴로지” [53]
- “영역과 함께 VXLAN 사용” [55]
- “VXLAN 구성” [58]
- “VXLAN 정보 표시” [62]
- “VXLAN 삭제” [62]
- “영역에 VXLAN 지정” [62]
- “사용 사례: 링크 집계를 통해 VXLAN 구성” [64]

### VXLAN 개요

클라우드 환경에서는 물리적 서버가 서로 다른 계층 2 네트워크에 위치할 수 있습니다. 예를 들어, 클라우드가 지리적 위치가 서로 다른 물리적 서버 간에 걸쳐 있을 수 있습니다. 이 경우 계층 2를 통해 VM(가상 머신) 또는 테넌트를 만들 경우 VM을 프로비저닝하는 데 사용할 수 있는 물리적 서버 수가 제한됩니다. VM 프로비저닝을 위해 서로 다른 계층 2 네트워크에 있는 물리적 서버를 사용할 수 있습니다. 하지만 다른 서버 간의 마이그레이션이 동일한 계층 2 네트워크로 제한되기 때문에 물리적 리소스 사용률이 최적화되지 않습니다.

VXLAN은 계층 3 네트워크 위에 계층 2 네트워크를 만들어서 네트워크 격리를 강화할 수 있는 계층 2 기술입니다. VXLAN은 여러 물리적 계층 2 네트워크를 통해 확장되는 가상 계층 2 네트워크를 제공합니다. 따라서 클라우드 환경의 리소스 프로비저닝은 단일 물리적 계층 2 네트워크로 제한되지 않습니다. 물리적 서버는 IPv4 또는 IPv6 네트워크를 통해 연결될 경우 VXLAN 네트워크의 일부일 수 있습니다.

VXLAN 기술을 Oracle Solaris의 EVS(Elastic Virtual Switch) 기능과 함께 사용하여 많은 가상 네트워크를 만들 수 있습니다. VXLAN을 EVS 기능과 함께 사용하여 가상 네트워크를 만드는 방법에 대한 자세한 내용은 [“사용 사례: 테넌트에 대한 탄력적 가상 스위치 구성” \[142\]](#)을 참조하십시오. 자세한 내용은 [5장. 탄력적 가상 스위치 정보](#) 및 [6장. 탄력적 가상 스위치 관리](#)를 참조하십시오.

VXLAN은 VXLAN 세그먼트 ID 또는 VXLAN VNI(네트워크 식별자)로 식별되는 격리된 계층 2 세그먼트를 제공합니다. 동일한 VXLAN 세그먼트에 있는 모든 VM은 동일한 계층 2 브로드캐스트 도메인에 속합니다.

VXLAN에서의 통신은 격리된 VLAN에서의 통신과 비슷합니다. 따라서 동일한 VXLAN 세그먼트에 있는 VM만 서로 통신할 수 있습니다. 다른 VXLAN 세그먼트에 있는 VM 간에는 서로 통신할 수 없습니다.

## VXLAN 사용 시의 이점

VXLAN은 다음과 같은 이점을 제공합니다.

- VXLAN ID는 24비트이기 때문에 최대 1,600만 개의 격리된 네트워크를 만들 수 있으므로 가상화된 클라우드 환경에서 확장성이 향상됩니다. 따라서 최대 4,096개의 격리된 네트워크를 만들 수 있는 12비트 VLAN ID를 사용하는 VLAN의 제한을 극복합니다.
- 기본 네트워크의 계층 3 기능을 사용할 수 있습니다.
- 가상 계층 2 네트워크가 기본 물리적 네트워크에서 추상화됩니다. 따라서 가상 네트워크는 물리적 네트워크에 표시되지 않고 다음과 같은 이점을 제공합니다.
  - 추가 물리적 기반구조가 필요하지 않습니다. 예를 들어, 서버에서 물리적 포트 뒤에 있는 VM이 증가하더라도 외부 스위치의 전달 테이블은 증가하지 않습니다.
  - MAC 주소 중복 범위가 동일한 VXLAN 세그먼트에 있는 VM으로 축소됩니다. 주소가 동일한 VXLAN 세그먼트에 포함되지 않는 경우 MAC 주소가 겹칠 수 있습니다. VXLAN에서는 동일한 VXLAN 세그먼트 또는 VNI에 속하는 데이터 링크의 MAC 주소만 고유해야 합니다. 이는 VLAN ID 및 MAC 주소 조합이 고유해야 하는 VLAN과 비슷합니다.

## VXLAN 이름 지정 규칙

Oracle Solaris에서 VXLAN 끝점은 VXLAN 데이터 링크로 표시됩니다. 이 VXLAN 데이터 링크는 IP 주소(IPv4 또는 IPv6) 및 VXLAN VNI(네트워크 식별자)와 연결됩니다. 여러

VXLAN 데이터 링크에서 동일한 IP 주소를 사용할 수 있지만 IP 주소와 VNI 조합은 고유해야 합니다. 동일한 VNI에서 피어 VXLAN 끝점을 검색하고 VXLAN 세그먼트 내에서 브로드캐스트를 구현하는 데 사용되는 선택적 멀티캐스트 주소로 VXLAN 데이터 링크를 구성할 수 있습니다. 동일한 VNI의 VXLAN 데이터 링크는 동일한 멀티캐스트 주소로 구성해야 합니다. VXLAN 요구 사항에 대한 자세한 내용은 [“VXLAN 요구 사항” \[57\]](#)을 참조하십시오.

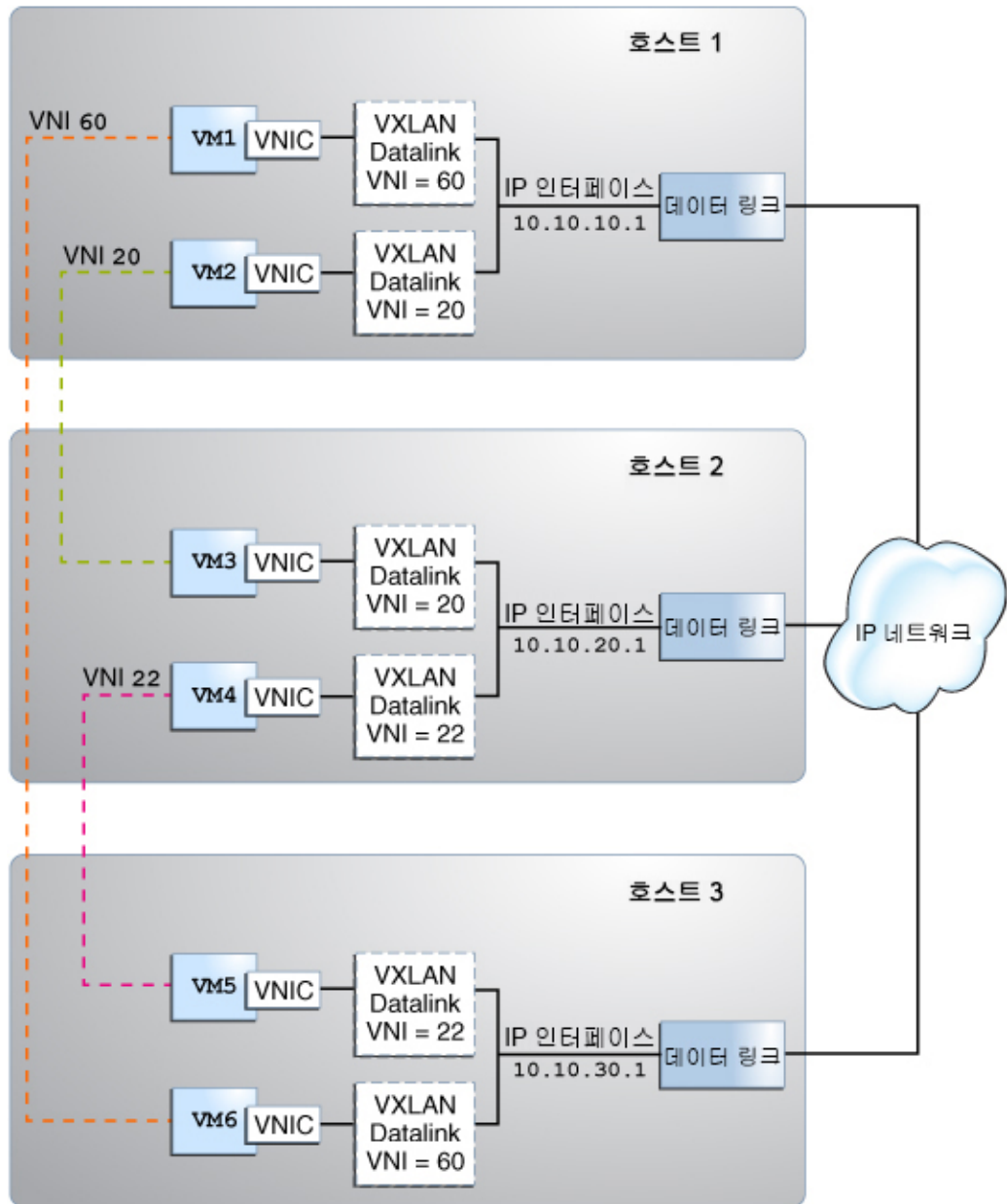
모든 VXLAN 데이터 링크는 VXLAN 세그먼트 ID 또는 VNI에 연결됩니다. VXLAN 데이터 링크 이름 지정 규칙은 링크 또는 VLAN에 사용되는 규칙과 동일합니다. 유효한 데이터 링크 이름을 제공하는 방법에 대한 자세한 내용은 [“Oracle Solaris 11.2 네트워크 구성 요소의 구성 및 관리”](#)의 [“유효한 링크 이름 규칙”](#)을 참조하십시오.

## VXLAN 토폴로지

VXLAN을 사용하면 자체 VXLAN 세그먼트 내의 계층 3 네트워크에서 시스템을 구성할 수 있습니다.

다음 그림은 여러 물리적 서버를 통해 구성되는 VXLAN 네트워크를 보여줍니다.

그림 3-1 VXLAN 토폴로지



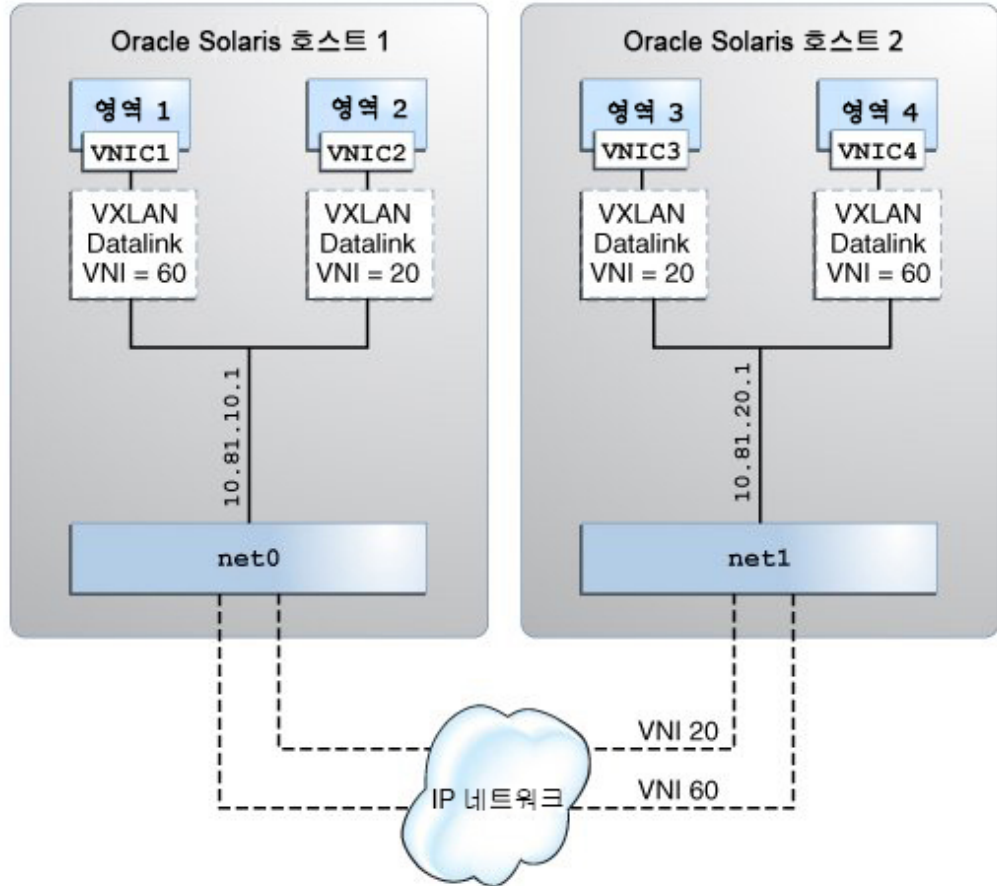
이 그림은 IP 네트워크 기반구조에 연결된 세 개의 가상 호스트를 보여줍니다. VXLAN 세그먼트 ID 또는 VNI(60, 20, 22)로 식별되는 세 개의 VXLAN 오버레이 네트워크가 있습니다. VM1 및 VM6 VM은 VNI 60으로 식별된 오버레이 네트워크에 있고, VM2 및 VM3 VM은 VNI 20으로 식별된 오버레이 네트워크에 있고, VM4 및 VM5 VM은 VNI 22로 식별된 오버레이 네트워크에 있습니다.

## 영역과 함께 VXLAN 사용

VXLAN 데이터 링크를 통해 생성되는 VNIC를 영역에 지정할 수 있습니다. VXLAN 데이터 링크는 VNI를 지정하여 생성되며, 이러한 VXLAN 데이터 링크는 VNI로 식별되는 VXLAN 세그먼트에 속합니다. 예를 들어, VXLAN 데이터 링크를 만들 때 VNI를 20으로 지정한 경우 해당 데이터 링크는 VNI 20으로 식별된 VXLAN 세그먼트에 속합니다. VXLAN 데이터 링크를 통해 생성되는 VNIC는 VXLAN 세그먼트에 포함됩니다.

다음 그림은 VNI 20 및 60으로 식별된 두 개의 VXLAN 오버레이 네트워크를 포함하는 IP 네트워크 기반구조에 연결된 두 개의 가상 Oracle Solaris 호스트를 보여줍니다.

그림 3-2 VXLAN 및 영역



다음과 같은 방법으로 VXLAN 세그먼트에 포함되는 영역을 만들 수 있습니다.

- VXLAN을 통해 VNIC를 만들고 VNIC를 영역에 지정합니다. 자세한 내용은 [“VXLAN 구성” \[58\]](#)을 참조하십시오.
- VXLAN을 영역의 anet(VNIC) 리소스에 대한 기본 링크로 지정합니다. 자세한 내용은 [“영역에 VXLAN 지정” \[62\]](#)을 참조하십시오.

어떤 경우든 영역에서 생성되는 VNIC는 기본 VXLAN 데이터 링크로 식별되는 VXLAN 세그먼트에 포함됩니다. 영역에 대한 자세한 내용은 [“Oracle Solaris 11.2 가상화 환경 소개”](#)를 참조하십시오.



VXLAN 링크에 VNIC를 지정하는 것은 VLAN 링크를 만든 후 영역에 지정하는 것과 비슷합니다. VLAN을 만든 후 VLAN을 영역에 지정하는 방법에 대한 자세한 내용은 [“Oracle Solaris 11.2의 네트워크 데이터 링크 관리”](#)의 [“VLAN을 구성하는 방법”](#)을 참조하십시오.

## VXLAN 구성 계획

VXLAN 구성 계획에는 다음 단계가 포함됩니다.

1. 물리적 네트워크에서 가상 네트워크 토폴로지를 결정합니다. 예를 들어, 서로 다른 서버에 있는 여러 VM으로 구성된 서비스를 호스트 중인 경우 해당 VM에 대해 VXLAN 세그먼트를 지정할 수 있습니다. 이 VXLAN 세그먼트의 VM은 서로 통신할 수 있지만 이 VXLAN 세그먼트에 포함되지 않은 다른 VM과 통신할 수 없습니다.
2. 물리적 서버가 IP 인터페이스를 통해 연결되고 해당 IP 멀티캐스팅이 물리적 네트워크에서 사용으로 설정되어 있는지 확인합니다.
3. VXLAN 세그먼트에 대한 번호 지정 체계를 만듭니다. 예를 들어, VM에 호스트된 응용 프로그램을 기반으로 VXLAN 세그먼트(VNI)를 지정할 수 있습니다.
4. IP 주소 및 VXLAN 세그먼트 ID를 지정하여 VXLAN 데이터 링크를 만듭니다.  
선택적으로 자체 멀티캐스트 주소로 VXLAN 세그먼트를 지정할 수 있습니다.
5. VXLAN 데이터 링크를 통해 VNIC를 만들고 VNIC를 영역에 지정합니다.  
또는 VXLAN 링크를 영역의 anet 링크에 대한 기본 링크로 지정할 수 있습니다.

## VXLAN 요구 사항

VXLAN을 사용하기 전에 다음 요구 사항을 충족해야 합니다.

- 네트워크에서 IP 멀티캐스팅이 지원되는지 확인합니다. IP 멀티캐스팅이 지원되지 않는 경우 VXLAN의 VM 간에 서로 통신할 수 없습니다.
- VXLAN에 다른 IP 서브넷에 있는 서버가 포함되어 있는 경우 전체 서브넷에서 멀티캐스트 라우팅을 지원해야 합니다. 멀티캐스트 경로 지정이 지원되지 않는 경우 동일한 IP 서브넷에 있는 VXLAN 기반 VM만 다른 IP 서브넷에 있는 VXLAN 기반의 다른 VM과 서로 통신할 수 있습니다. 예를 들어, 지리적으로 분산된 데이터 센터 간에는 서로 통신할 수 없습니다.

VXLAN 데이터 링크의 이름 지정 규칙에 대한 자세한 내용은 [“VXLAN 이름 지정 규칙” \[52\]](#)을 참조하십시오.

## VXLAN 구성

다음 절차에서는 영역이 이미 시스템에 생성된 것으로 가정합니다. 영역 구성에 대한 자세한 내용은 “Oracle Solaris 영역 만들기 및 사용”의 1 장, “비전역 영역을 계획 및 구성하는 방법”을 참조하십시오.

### ▼ VXLAN을 구성하는 방법

1. 관리자로 로그인합니다.

자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”을 참조하십시오.

2. 시스템에서 사용할 수 있는 IP 주소를 결정합니다.

```
# ipadm show-addr
```

3. IP 주소 또는 IP 인터페이스를 지정하여 VXLAN 데이터 링크를 만듭니다.

#### ■ IP 주소를 지정하여 VXLAN을 만들려면

```
# dladm create-vxlan -p prop=value VXLAN-LINK
```

-p *prop=value*

생성된 VXLAN 데이터 링크에서 지정된 값으로 설정할 수 있는 쉘표로 구분된 VXLAN 데이터 링크 등록 정보 목록을 지정합니다. 다음 등록 정보를 설정합니다.

- **addr** - VXLAN 네트워크에 대한 IPv4 또는 IPv6 주소를 지정합니다. 이 주소는 특정 주소이거나 주소/접두어 길이 조합일 수 있습니다.
- **vni** - VXLAN 세그먼트의 네트워크 식별자를 지정합니다. 0과 16777215 사이의 숫자를 지정할 수 있습니다.
- **mgroup** - (선택 사항) 멀티캐스트 그룹 이름을 지정합니다. VXLAN 세그먼트에 자체 멀티캐스트 그룹이 있는 경우에만 이 옵션을 지정할 수 있습니다.

*VXLAN-LINK*

VXLAN의 이름입니다.

#### ■ IP 인터페이스를 지정하여 VXLAN을 만들려면 다음을 참조하십시오.

```
# dladm create-vxlan -p prop=value
```

-p *prop=value*

생성된 VXLAN 데이터 링크에서 지정된 값으로 설정할 수 있는 쉘표로 구분된 VXLAN 데이터 링크 등록 정보 목록을 지정합니다. 다음 등록 정보를 설정합니다.

- `interface` - VXLAN 네트워크에 대한 IP 인터페이스를 지정합니다.
- `vni` - VXLAN 세그먼트의 네트워크 식별자를 지정합니다. 0과 16777215 사이의 숫자를 지정할 수 있습니다.

VXLAN                      VXLAN의 이름입니다.

IP 인터페이스와 IP 버전을 지정하면 해당 인터페이스에 지정되는 버전의 사용 가능한 IP 주소를 통해 VXLAN 데이터 링크가 생성됩니다. 예를 들어, `net0`을 통해 IP 주소 `10.10.10.1`을 구성한 경우 VXLAN 데이터 링크는 `10.10.10.1`을 통해 생성됩니다. 기본적으로 IP 버전은 IPv4 주소입니다. 하지만 IPv6 주소가 필요한 경우 `ipvers` 등록 정보를 사용하여 버전을 지정해야 합니다.

---

참고 - 물리적 통합 링크(트렁크 또는 DLMP 집계) 또는 IPoIB 링크에 호스트되는 IP 주소에서 VXLAN 데이터 링크를 만들 수 있습니다. 하지만 IPMP, 가상 네트워크 인터페이스 또는 루프백 인터페이스에 호스트된 IP 주소에는 VXLAN 데이터 링크를 만들 수 없습니다.

---

#### 4. 만든 VXLAN을 확인합니다.

```
# dladm show-vxlan
```

#### 5. VXLAN 데이터 링크를 통해 VNIC를 만듭니다.

```
# dladm create-vnic -l VXLAN-LINK VNIC
```

VXLAN 데이터 링크를 통해 VLAN VNIC를 만들 수 있습니다. VLAN VNIC를 만들려면 `-f`(강제) 옵션을 지정해야 합니다. 자세한 내용은 [VLAN ID로 VNIC 구성 방법 \[24\]](#)을 참조하십시오.

#### 6. VNIC를 영역에 먼저 지정하거나, 직접 VNIC를 통해 IP 인터페이스를 구성합니다.

- VNIC를 통해 IP 인터페이스를 구성합니다.

```
# ipadm create-ip VNIC
```

```
# ipadm create-addr -a address VNIC
```

- 영역에 VNIC를 지정하고 영역 내의 VNIC를 통해 IP 인터페이스를 구성합니다.

##### a. 영역의 인터페이스를 사용하여 VNIC를 지정합니다.

```
zonecfg:ZONE> add net
zonecfg:ZONE:net> set physical=VNIC
zonecfg:ZONE:net> end
```

##### b. 구현한 변경 사항을 확인하고 커밋한 다음 영역을 종료합니다.

```
zonecfg:ZONE> verify
```

```
zonecfg:zone> commit
zonecfg:zone> exit
```

c. 영역을 재부트합니다.

```
global# zoneadm -z zone reboot
```

d. 영역에 로그인합니다.

```
global# zlogin zone
```

e. 영역에서 영역에 이제 지정된 VNIC를 통해 IP 인터페이스를 만듭니다.

```
zone# ipadm create-ip interface
```

f. 유효한 IP 주소로 VNIC를 구성합니다.

VNIC에 정적 주소를 지정하는 경우 다음을 입력합니다.

```
zone# ipadm create-addr -a address interface
```

-a address IP 주소를 지정합니다. CIDR 표기법을 사용할 수 있습니다.

g. 영역을 종료합니다.

dladm 및 ipadm 명령에 대한 자세한 내용은 [dladm\(1M\)](#) 및 [ipadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

예 3-1 VXLAN을 만들고 VXLAN을 통해 생성된 VNIC에 대한 IP 인터페이스 구성

1. 시스템에서 사용 가능한 IP 주소를 확인합니다.

```
# ipadm show-addr net4
ADDROBJ  TYPE  STATE  ADDR
net4/v4  static ok    10.10.11.1/24
```

2. VXLAN 세그먼트 10에서 VXLAN 데이터 링크를 만듭니다.

```
# dladm create-vxlan -p addr=10.10.11.1,vni=10 vxlan1
```

3. 만든 VXLAN 링크를 확인합니다.

```
# dladm show-vxlan
LINK  ADDR      VNI  MGROUP
vxlan1 10.10.11.1 10   224.0.0.1
```

멀티캐스트 주소를 지정하지 않았기 때문에 이 VXLAN 세그먼트에서는 동일한 네트워크 세그먼트에서 모든 호스트의 주소를 지정하는 All Host 멀티캐스트 주소를 사용합니다.

4. VXLAN 링크 정보를 확인합니다.

```
# dladm show-link vxlan1
```

```
LINK CLASS MTU STATE OVER
vxlan1 vxlan 1440 up --
```

vxlan1이 생성되고 링크 상태는 up입니다.

5. vxlan1을 통해 VNIC를 만듭니다.

```
# dladm create-vnic -l vxlan1 vnic1
```

6. 만든 VNIC를 확인합니다.

```
# dladm show-vnic
LINK OVER SPEED MACADDRESS MACADRTYPE VIDS
vnic1 vxlan1 10000 2:8:20:fe:58:d4 random 0
```

7. VNIC를 통해 IP 인터페이스를 구성합니다.

```
# ipadm create-ip vnic1
```

```
# ipadm create-addr -T static -a local=10.10.12.1/24 vnic1/v4
```

```
# ipadm show-addr vnic1
```

```
ADDROBJ TYPE STATE ADDR
vnic1/v4 static ok 10.10.12.1/24
```

IP 주소를 지정하여 VXLAN을 성공적으로 만들었습니다. VXLAN을 통해 VNIC를 만들고 IP 인터페이스를 구성했습니다.

### 예 3-2 VXLAN을 통해 만든 VNIC를 영역에 지정하고 IP 인터페이스 구성

이 예에서는 예 3-1. “VXLAN을 만들고 VXLAN을 통해 생성된 VNIC에 대한 IP 인터페이스 구성”에서 1-6단계를 완료했다고 가정합니다.

VNIC를 만든 후 VNIC를 영역에 지정하고 IP 인터페이스를 구성합니다.

```
global# zonecfg -z zone2
zonecfg:zone2> add net
zonecfg:zone2:net> set physical=vnic1
zonecfg:zone2:net> end
zonecfg:zone2> verify
zonecfg:zone2> commit
zonecfg:zone2> exit
global# zoneadm -z zone2 reboot

global# zlogin zone2
zone2# ipadm create-ip vnic1
zone2# ipadm create-addr -a 192.168.3.85/24 vnic1
ipadm: vnic1/v4

zone2# exit
```

VNIC를 영역에 지정한 다음 VNIC를 통해 IP 인터페이스를 구성했습니다.

## VXLAN 정보 표시

dladm show-link 명령을 사용하여 VXLAN 링크에 대한 일반적인 링크 정보를 볼 수 있습니다. VXLAN에 특정한 정보를 보려면 dladm show-vxlan 명령을 사용합니다.

```
# dladm show-link
LINK          CLASS  MTU   STATE  OVER
net6          phys   1500  down   --
net0          phys   1500  up     --
net2          phys   1500  unknown --
net3          phys   1500  unknown --
net1          phys   1500  unknown --
net5          phys   1500  unknown --
net4          phys   1500  up     --
vxlan1       vxlan  1440  up     --
vnic1        vnic   1440  up     vxlan1

# dladm show-vxlan vxlan1
LINK  ADDR      VNI  MGROUP
vxlan1  10.10.11.1  10   224.0.0.1
```

## VXLAN 삭제

VXLAN 링크를 삭제하려면 dladm delete-vxlan 명령을 사용합니다. VXLAN 링크를 삭제하기 전에 dladm show-link 명령을 사용하여 해당 VXLAN 링크에 구성된 VNIC가 없는지 확인해야 합니다.

관리자로 로그인하고 다음 명령을 실행합니다.

```
# dladm delete-vxlan VXLAN
```

예를 들어, vxlan1을 삭제하려면 다음 명령을 입력합니다.

```
# dladm delete-vxlan vxlan1
```

## 영역에 VXLAN 지정

VXLAN을 영역의 anet 리소스에 기본 링크로 지정하여 VXLAN 세그먼트에 포함되는 영역을 만들 수 있습니다. 영역을 구성하는 방법에 대한 자세한 내용은 [“Oracle Solaris 영역 만들기 및 사용”](#)을 참조하십시오.

## ▼ 영역에 VXLAN을 지정하는 방법

1. 관리자로 로그인합니다.  
자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”을 참조하십시오.

2. 시스템에서 사용 가능한 IP 주소를 확인합니다.

```
# ipadm show-addr
```

3. IP 주소를 지정하여 VXLAN을 만듭니다.

```
# dladm create-vxlan -p prop=value VXLAN-LINK
```

4. 만든 VXLAN을 확인합니다.

```
# dladm show-vxlan
```

5. 생성된 VXLAN을 영역의 anet에 대한 기본 링크로 지정하여 영역을 구성합니다.

```
global# zonecfg -z ZONE
zonecfg:zone2> add anet
zonecfg:zone2:net> set linkname=datalink
zonecfg:zone2:net> set lower-link=VXLAN-LINK
zonecfg:zone2:net> end
zonecfg:zone2> verify
zonecfg:zone2> commit
zonecfg:zone2> exit
global# zoneadm -z ZONE reboot
```

VXLAN은 영역의 anet에 대한 기본 링크로 지정됩니다.

### 예 3-3 영역의 anet에 VXLAN 지정

```
# ipadm show-addr net4
ADDROBJ  TYPE  STATE  ADDR
net4/v4  static ok    10.10.11.1/24 2

# dladm create-vxlan -p addr=10.10.11.1,vni=10 vxlan1

# dladm show-vxlan
LINK  ADDR      VNI  MGROUP
vxlan1 10.10.11.1 10   224.0.0.1
```

멀티캐스트 주소를 지정하지 않았기 때문에 이 VXLAN 세그먼트에서는 동일한 네트워크 세그먼트에서 모든 호스트의 주소를 지정하는 All Host 멀티캐스트 주소를 사용합니다.

```
# dladm show-link vxlan1
LINK  CLASS MTU  STATE OVER
```

```
vxlan1 vxlan 1440 up --
```

vxlan1이 생성되고 링크 상태는 up입니다.

```
global# zonecfg -z zone2
zonecfg:zone2> add anet
zonecfg:zone2:net> set linkname=net1
zonecfg:zone2:net> set lower-link=vxlan1
zonecfg:zone2:net> end
zonecfg:zone2> verify
zonecfg:zone2> commit
zonecfg:zone2> exit
global# zoneadm -z zone2 reboot
```

vxlan1은 영역의 anet에 대한 기본 링크로 지정됩니다.

영역이 부트되면 net1이 vxlan1을 통해 zone2에 생성됩니다.

## 사용 사례: 링크 집계를 통해 VXLAN 구성

다음 사용 사례에서는 다음을 수행하는 방법을 보여줍니다.

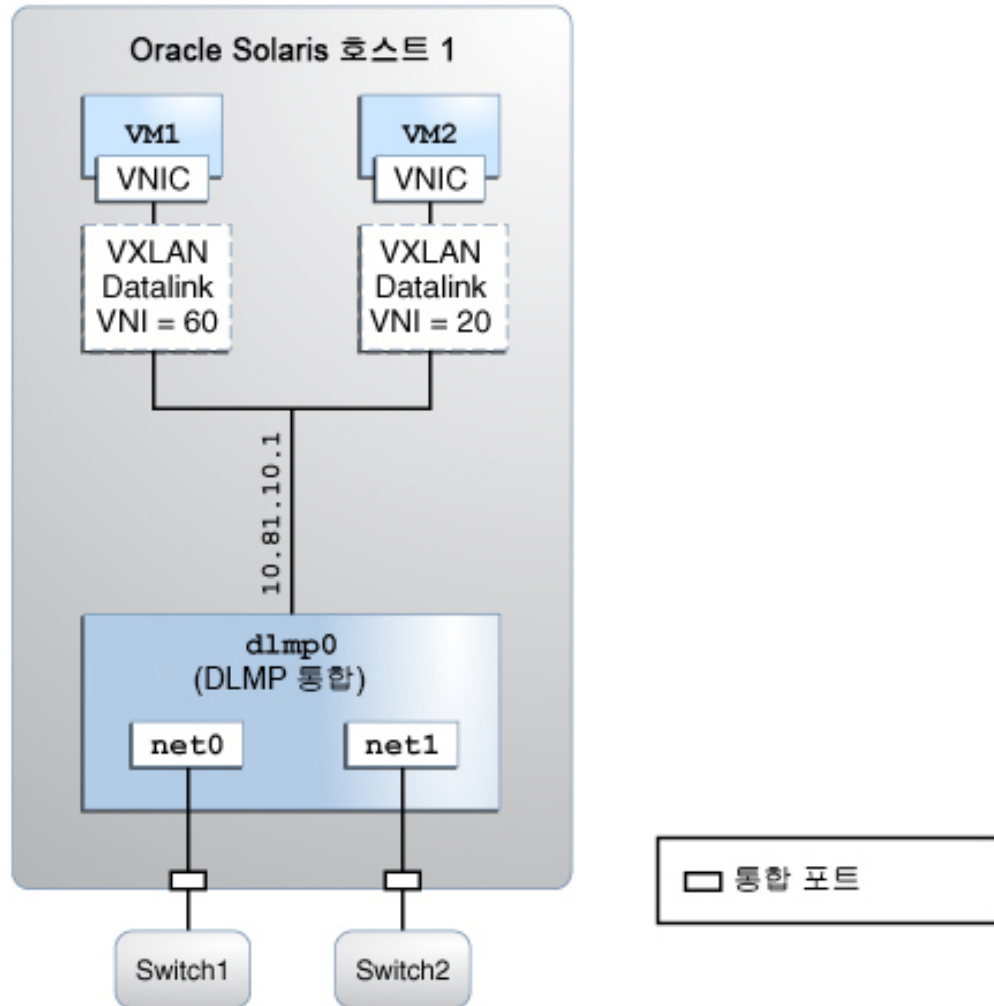
- DLMP 집계 만들기
- 집계를 통해 IP 주소 구성
- 집계를 통해 두 VXLAN 만들기
- VXLAN 데이터 링크를 통해 두 영역을 하위 링크로 구성

링크 집계에 대한 자세한 내용은 “[Oracle Solaris 11.2의 네트워크 데이터 링크 관리](#)”의 2장, “[링크 집계를 사용하여 고가용성 구성](#)”을 참조하십시오.

다음 그림은 DLMP 집계를 통한 VXLAN 구성을 보여줍니다.



그림 3-3 링크 집계를 통한 VXLAN



통합 포트 또는 외부 스위치가 실패하더라도 하나 이상의 포트가 있고 스위치가 작동한다면 집계 기반 VXLAN 데이터 링크가 계속 존재하므로 페일오버 중에도 네트워크에고가용성이 제공됩니다. 예를 들어, net0이 실패할 경우 DLMP 집계에서는 VXLAN 데이터 링크 간에 나머지 포트 net1을 공유합니다. 통합된 포트 간 분산은 사용자에게 투명하고 통합에 연결된 외부 스위치에 독립적으로 발생합니다.

1. 관리자로 로그인합니다.

자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”을 참조하십시오.

2. 통합의 데이터 링크를 식별하기 위해 데이터 링크 정보를 표시합니다.

```
# dladm show-link
LINK      CLASS    MTU     STATE   OVER
net0     phys    1500    up      --
net1     phys    1500    up      --
net2     phys    1500    up      --
```

3. 통합할 데이터 링크에 대해 구성된 IP 인터페이스가 없는지 확인합니다. 링크에 대해 인터페이스가 구성된 경우 해당 인터페이스를 삭제합니다.

```
# ipadm show-if
IFNAME    CLASS      STATE    ACTIVE   OVER
lo0       loopback   ok       yes      --
net0      ip         ok       no       --

# ipadm delete-ip net0
```

4. 링크 net0 및 net1을 사용하여 DLMP 집계를 만듭니다.

```
# dladm create-aggr -m dlmp -l net0 -l net1 dlmp0
```

5. 집계 dlmp0 위에 IP 인터페이스를 구성합니다.

```
# ipadm create-ip dlmp0
# ipadm create-addr -T static -a local=10.10.10.1 dlmp0/v4
```

6. 집계를 통해 구성되는 IP 주소를 지정하여 두 VXLAN을 만들고, VXLAN 세그먼트에 대한 네트워크 식별자인 VNI를 지정합니다.

```
# dladm create-vxlan -p addr=10.10.10.1,vni=20 vxlan20
# dladm create-vxlan -p addr=10.10.10.1,vni=60 vxlan60
```

두 VNI는 모두 기본 멀티캐스트 주소를 사용하여 구성됩니다.

7. VXLAN 데이터 링크 vxlan20을 하위 링크로 사용하여 영역 VM1을 구성합니다.

```
global# zonecfg -z VM1
zonecfg:VM1> add anet
zonecfg:VM1:net> set linkname=net0
zonecfg:VM1:net> set lower-link=vxlan20
zonecfg:VM1:net> end
zonecfg:VM1> verify
zonecfg:VM1> commit
zonecfg:VM1> exit
global# zoneadm -z VM1 reboot
```

8. VXLAN 데이터 링크 vxlan60을 하위 링크로 사용하여 영역 VM2를 구성합니다.

```
global# zonecfg -z VM2
zonecfg:VM2> add anet
```

```

zonecfg:VM2:net> set linkname=net0
zonecfg:VM2:net> set lower-link=vxlan60
zonecfg:VM2:net> end
zonecfg:VM2> verify
zonecfg:VM2> commit
zonecfg:VM2> exit
global# zoneadm -z VM2 reboot
    
```

net0 및 net1 데이터 링크가 DLMP 집계 d1mp0에 통합되고, IP 주소 10.10.10.1이 집계  
에 대해 구성됩니다. vxlan20 및 vxlan60 VXLAN은 집계에 대해 구성되는 지정된 IP 주  
소 10.10.10.1을 통해 생성됩니다. VXLAN vxlan20은 VXLAN 세그먼트 20에 생성되고,  
VXLAN vxlan60은 VXLAN 세그먼트 60에 생성됩니다. 영역 vm1은 VXLAN 데이터 링크  
vxlan20을 하위 링크로 사용하여 구성되고, 영역 vm2는 VXLAN 데이터 링크 vxlan60을 하위  
링크로 사용하여 구성됩니다.



# ◆◆◆ 4 장 4

## 에지 가상 브리징을 사용하여 서버-네트워크 에지 가상화 관리

---

서버-네트워크 에지는 서버 포트와 첫번째 홉 스위치 포트 사이의 연결에 있습니다. VLAN(가상 로컬 영역 네트워크) 및 LACP(Link Aggregation Control Protocol)와 같은 네트워크 구성은 서버 포트와 이 에지의 스위치 포트에서 동일해야 합니다. DCBX(Data Center Bridging Capability Exchange)를 사용하여 서버와 스위치 포트의 구성을 자동화할 수 있습니다. 자세한 내용은 “Oracle Solaris 11.2의 네트워크 데이터 링크 관리”의 6 장, “데이터 센터 브리징을 사용하여 통합 네트워크 관리”를 참조하십시오.

서버 가상화에서는 여러 가상 포트가 스위치 포트에 연결된 하나의 서버 포트만이 아닌 서버 포트 뒤의 가상 머신(VM)과 연관됩니다. 서버 가상화는 서버-네트워크 에지에서 다음과 같은 추가적인 요구 사항을 부여합니다.

- VM 간 트래픽이 스위치에서 구성된 정책을 따르도록 외부 스위치를 통한 가상 머신 간 스위칭
- 가상 포트 등록 정보를 네트워크로 확장

Oracle Solaris는 이러한 요구 사항을 충족하는 최근 IEEE 표준인 에지 가상 브리징(EVB)을 지원합니다.

이 장의 내용:

- “서버-네트워크 에지 가상화에서 EVB 지원” [69]
- “Improving Network and Server Efficiency by Using EVB” [71]
- “EVB 설치” [73]
- “동일한 물리적 포트를 통한 VM 간 전환 제어” [74]
- “VDP를 사용하여 VNIC 정보 교환” [79]
- “VDP와 ECP의 상태 및 통계 표시” [81]
- “기본 EVB 구성 변경” [82]

## 서버-네트워크 에지 가상화에서 EVB 지원

가상화 서버에는 동일한 물리적 링크를 통해 여러 가상 NIC가 포함될 수 있습니다. 이러한 VNIC를 VM에 지정할 수 있습니다. 전통적으로 스위치는 패킷을 수신하는 동일한 링크에서

다시 패킷을 전송하지 않습니다. VM 간 패킷은 호스트 자체 내의 가상 스위치에 의해 다시 루프됩니다. 따라서 외부 스위치에서 구성된 모든 정책은 VM 간 패킷에 적용되지 않습니다. EVB에 대한 지원으로 Oracle Solaris 및 스위치는 VM 간 패킷에 대한 모든 정책을 적용한 후 VM 간 패킷이 외부 스위치에 의해 스위칭되도록 합니다. VNIC에 대한 자세한 내용은 [2 장. 가상 네트워크 만들기 및 관리](#)를 참조하십시오.

또한 EVB를 지원하는 Oracle Solaris는 스위치와 VNIC에 대한 정보를 교환할 수 있습니다. 이 정보 교환을 통해 스위치는 네트워크에서 대역폭 제한, 대역폭 공유 및 MTU와 같은 VNIC 등록 정보를 자동으로 구성할 수 있습니다. 이 기능이 없으면 VNIC가 서버에서 생성, 수정 또는 삭제될 때마다 서버 관리자 및 네트워크 관리자가 서로 조정하여 스위치에서 변경 작업을 수행해야 합니다. VNIC 등록 정보를 네트워크로 확장하면 VNIC 등록 정보를 기준으로 네트워킹 리소스의 효율적인 사용이 가능해집니다. 예를 들어, 호스트에 도달 후 패킷에 대한 대역폭 제한 적용은 패킷이 이미 링크 대역폭을 모두 사용했으므로 별로 도움이 되지 않습니다.

## 반사 중계

반사 중계는 동일한 물리적 NIC를 통해 VNIC를 사용하는 VM이 외부 스위치를 통해 통신할 수 있도록 하는 기능입니다. 스위치가 이 기능을 지원해야 합니다. Oracle Solaris에서 LLDP는 스위치가 반사 중계 기능을 지원하는지 확인하고 스위치에서 반사 중계 기능을 사용 또는 사용 안함으로 설정하는 데 사용되는 EVB TLV(type-length value) 단위를 포함하도록 확장됩니다. 따라서 스위치가 LLDP 및 EVB TLV 단위를 지원하는 경우에만 LLDP를 사용하여 스위치에서 이 기능의 감지 및 구성을 자동화할 수 있습니다. 그렇지 않으면 스위치에서 반사 중계 기능을 수동으로 구성해야 합니다. 반사 중계를 수동으로 구성하는 방법에 대한 자세한 내용은 스위치 제조업체의 설명서를 참조하십시오.

Oracle Solaris의 반사 중계 지원에 대한 자세한 내용은 [“동일한 물리적 포트를 통한 VM 간 전환 제어” \[74\]](#)를 참조하십시오. LLDP TLV 단위에 대한 자세한 내용은 [“Oracle Solaris 11.2의 네트워크 데이터 링크 관리”의 “LLDP 에이전트 공개 정보”](#)를 참조하십시오.

## 네트워크에서 자동화된 VNIC 구성

Oracle Solaris는 IEEE 802.1Qbg에서 정의된 VDP(Virtual Station Interface Discovery and Configuration Protocol)를 사용하여 스위치와 VNIC 정보를 교환합니다. 스위치가 VDP를 지원하는 경우 VNIC 등록 정보가 스위치에서 자동으로 구성됩니다. 이것은 호스트와 스위치가 DCBX를 사용하여 물리적 링크 등록 정보를 교환하는 것과 유사합니다. VNIC가 생성, 수정 또는 삭제되면 호스트와 스위치 사이에 VDP 교환이 시작됩니다. 이 교환을 통해 스위치는 VNIC의 등록 정보를 기준으로 VNIC로 향하는 패킷에 대해 리소스를 할당합니다.

Oracle Solaris에서 시스템과 외부 스위치 간의 VNIC 정보 교환에 대한 자세한 내용은 [“VDP를 사용하여 VNIC 정보 교환” \[79\]](#) 및 [“VDP에서 VNIC 정보를 교환하는 방법” \[80\]](#)을 참조하십시오.

## Improving Network and Server Efficiency by Using EVB

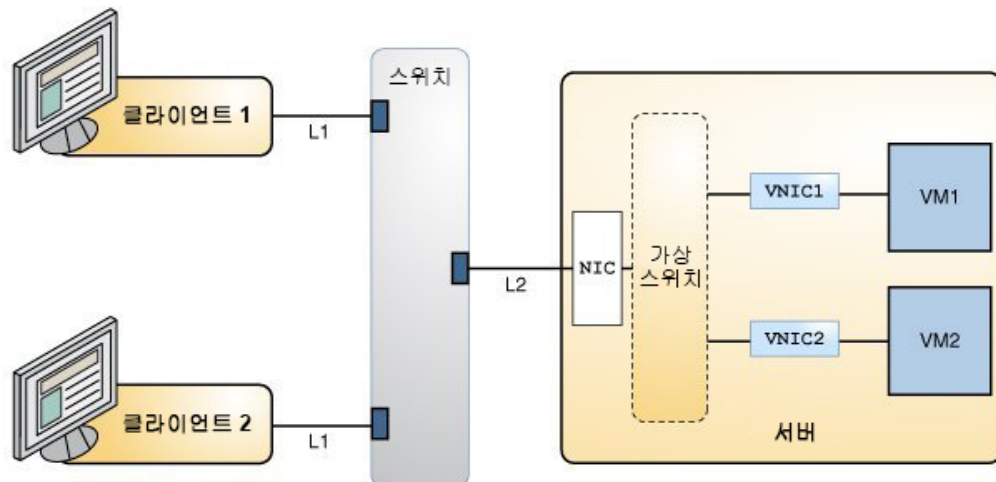
이 절의 예제에서는 서버에서 EVB를 사용으로 설정한 경우에 서버 및 네트워크 효율성을 향상시키는 방법을 보여줍니다.

이 예제에서는 서버에서 두 응용 프로그램을 클라우드 환경의 동일한 물리적 시스템에 호스트한다고 가정합니다.

- 응용 프로그램은 클라우드의 물리적 시스템에 별도의 가상 머신(VM1 및 VM2)으로 호스트됩니다. vNIC vNIC1 및 vNIC2가 VM1 및 VM2에 대해 각각 구성됩니다.
- 계정이 있는 클라이언트(클라이언트 1 및 클라이언트 2)는 응용 프로그램에 액세스할 수 있습니다.
- 가상 머신(VM1 및 VM2)은 물리적 시스템의 리소스와 링크 L2의 대역폭을 공유합니다.
- 클라이언트는 링크 L1을 사용하여 스위치에 연결됩니다. 스위치는 링크 L2를 사용하여 NIC에 연결됩니다.
- 미리 정해진 SLA에 따라 가상 머신에 대한 리소스가 지정됩니다. 다음(L2) 대역폭 사용이 가상 머신의 SLA에 포함됩니다.
  - VM1은 우선 순위가 높은 TCP(Transmission Control Protocol) 서비스를 실행하고 있습니다. 따라서 VM1에 대한 SLA에는 8Gbps의 최대 대역폭 제한이 있습니다.
  - VM2는 우선 순위가 높지 않은 UDP(사용자 데이터그램 프로토콜) 서비스를 실행하고 있습니다. 따라서 VM2에 대한 SLA에는 3Gbps의 최대 대역폭 제한이 있습니다.

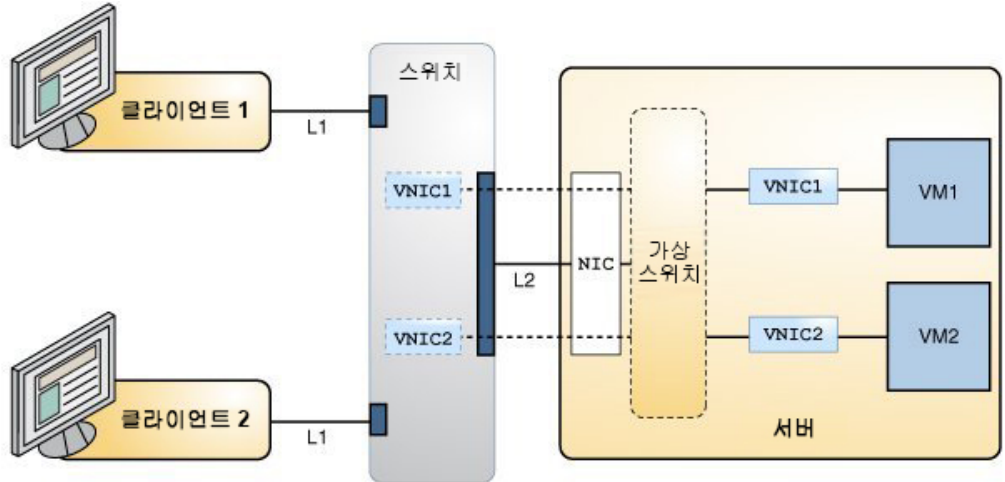
다음 그림은 서버에 호스트된 응용 프로그램을 보여줍니다.

그림 4-1 EVB를 사용하지 않는 응용 프로그램 설정



서버 및 스위치에서 EVB를 사용으로 설정한 경우 서버에서는 다음 그림에 표시된 것과 동일한 물리적 스위치 포트를 통해 스위치와 VNIC 정보를 교환합니다.

그림 4-2 EVB를 사용으로 설정한 응용 프로그램 설정



다음 표는 서버 및 스위치에서 EVB를 사용으로 설정하기 이전과 이후의 서버 효율성을 보여줍니다.

표 4-1 EVB를 사용하지 않는 경우와 사용하는 경우의 서버 효율성

EVB를 사용하는 서버 효율성	EVB를 사용하지 않는 서버 효율성
서버에서는 대역폭 적용을 위해 클라이언트에서 수신되는 트래픽을 조정합니다.	스위치는 서버를 대상으로 하는 트래픽을 조정합니다.
시스템 리소스가 사용되므로 시스템 및 네트워크 성능에 영향을 줍니다.	대역폭을 처리하는 데 시스템 리소스가 사용되지 않으므로, 시스템 효율성이 향상됩니다.
이 예에서 클라이언트(클라이언트 1 및 클라이언트 2)는 서비스를 동시에 활용해야 하는 경우 링크 L2의 대역폭 및 서버 리소스를 사용합니다. 서버에서는 VM1 및 VM2에 VNIC에 대한 SLA를 적용하여 클라이언트의 인바운드 트래픽과 아웃바운드 트래픽을 조정합니다. 하지만 네트워크 성능과 대역폭 사용에 다음과 같이 영향을 줍니다. <ul style="list-style-type: none"> <li>■ 클라이언트(클라이언트 1 및 클라이언트 2)의 트래픽은 링크 L2의 대역폭을 제한 없이 사용합니다. 또한 호스트에 대역폭 제한이 구성되어 있는 경우 L2의 대역폭을 사용하는 패킷이 호스트에서 삭제되어 대역폭을 충분히 사용하지 못하게 될 수 있습니다.</li> <li>■ VM1은 우선 순위가 높은 TCP 서비스를 제공하고 VM2는 우선 순위가 높지 않은 UDP 서비스를 제공합니다. 서버에서 VM1의 대역폭을 조정하면 TCP가 응답하므로 링크 L2에 대한 VM1의 대역폭</li> </ul>	서버 및 스위치에서 EVB를 사용으로 설정하면 시스템 효율성이 다음과 같이 향상됩니다. <ul style="list-style-type: none"> <li>■ 서버의 VNIC에 대해 구성된 SLA가 스위치에 적용됩니다.</li> <li>■ 스위치는 구성된 대역폭을 기반으로 VM1 및 VM2에 대한 트래픽을 조정하므로 링크 L2의 대역폭을 적절히 활용할 수 있으며, 결과적으로 네트워크 효율성이 높아집니다.</li> </ul> 스위치가 대역폭을 조정하기 때문에 서버는 수신측에서 대역폭을 처리할 필요가 없으며, 결과적으로 서버 효율성이 높아집니다.



EVB를 사용하는 서버 효율성	EVB를 사용하지 않는 서버 효율성
사용에 영향을 줍니다. 하지만 서버에서 VM2 서비스를 조정하면 링크 L2의 대역폭 사용에 영향을 주지 않습니다. 따라서 링크 L2를 사용하는 다른 서비스에 영향을 줍니다.	
이 예에서 서버에 인바운드되는 UDP 및 TCP 서비스에 대한 네트워크 트래픽은 링크 L2에서 사용 가능한 대역폭을 제한 없이 사용합니다. 서버는 네트워크 트래픽을 수신한 후 구성된 대역폭 제한에 따라 네트워크 트래픽을 조정합니다.	구성된 대역폭 제한(3Gbps 및 8Gbps)은 서버 및 스위치에 의해 조정됩니다. 따라서 공유 링크 L2는 구성된 대역폭 제한을 기반으로 사용됩니다.

## EVB 설치

시스템에서 EVB를 사용하려면 EVB 패키지를 설치해야 합니다.

### ▼ EVB를 설치하는 방법

1. 관리자로 로그인합니다.  
자세한 내용은 [“Oracle Solaris 11.2의 사용자 및 프로세스 보안”](#)의 [“지정된 관리 권한 사용”](#)을 참조하십시오.
2. EVB 패키지가 설치되었는지 확인합니다.  

```
# pkg info evb
```
3. EVB 패키지가 설치되지 않은 경우 패키지를 설치합니다.  

```
# pkg install evb
```
4. 서비스가 사용으로 설정되었는지 확인합니다.  

```
# svcs vdp
```
5. 서비스가 사용으로 설정되지 않은 경우 서비스를 사용으로 설정합니다.  

```
# svcadm enable vdp
```

EVB 패키지 설치 후 기본 EVB 구성이 자동으로 사용으로 설정됩니다. 기본 EVB 구성을 적용하면 시스템에 구성된 VNIC에 대한 정보를 외부 스위치와 즉시 교환할 수 있습니다.

- 참조
- VNIC 정보 교환, VNIC 정보를 교환하는 데 사용되는 프로토콜 및 EVB 구성 요소에 대한 자세한 내용은 [“VDP를 사용하여 VNIC 정보 교환” \[79\]](#)을 참조하십시오.
  - 시스템에서 EVB를 사용으로 설정한 경우의 물리적 이더넷 링크의 VSI 검색 및 구성 프로토콜(VDP) 상태에 대한 정보를 표시하고 VNIC에 대해 VDP 패킷이 교환되는지를 확인하려면 [“VDP와 ECP의 상태 및 통계 표시” \[81\]](#)를 참조하십시오.

- 기본 EVB 구성을 변경하려면 “기본 EVB 구성 변경” [82]을 참조하십시오.
- 기본 EVB 구성 정보를 표시하려면 예 4-2. “물리적 링크의 EVB 관련 데이터 링크 등록 정보 표시”를 참조하십시오.

## 동일한 물리적 포트를 통한 VM 간 전환 제어

vswitchmode 데이터 링크 등록 정보를 사용하여 동일한 물리적 포트를 통한 VM 간 전환을 제어할 수 있습니다. 가능한 값은 다음 세 가지입니다.

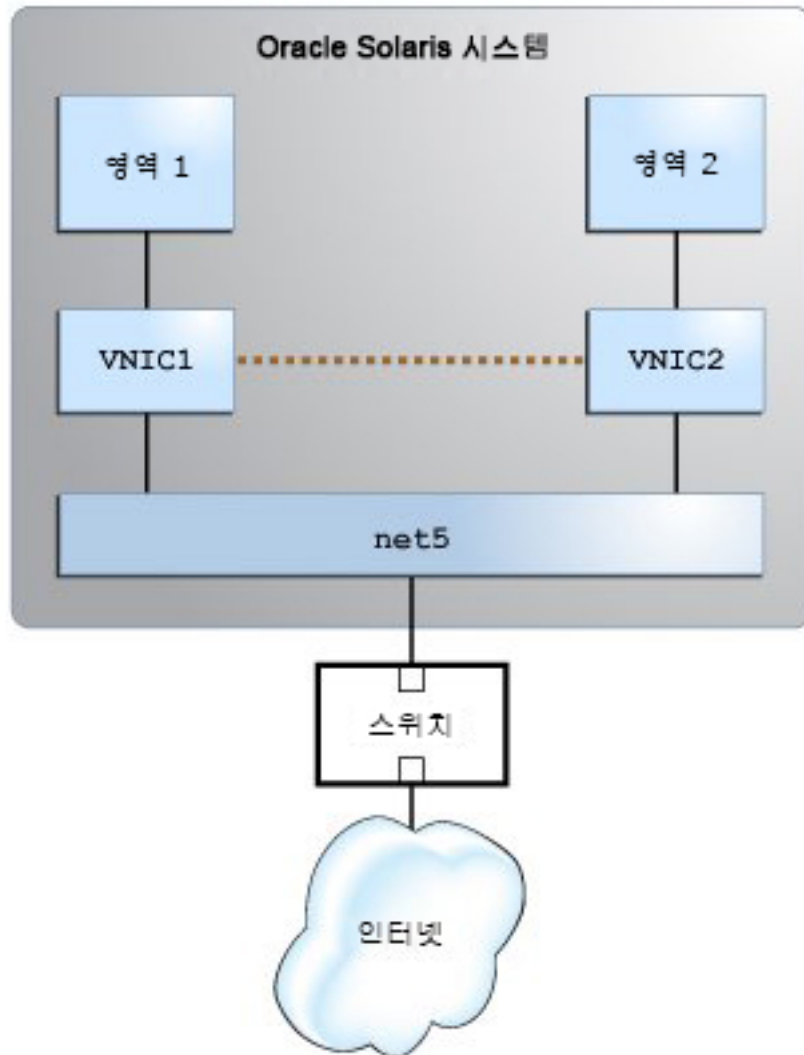
- local - 동일한 물리적 NIC를 통한 VM 간 네트워크 트래픽을 내부적으로 교환할 수 있습니다. 기본 모드입니다.
- remote - 동일한 물리적 NIC를 통한 VM 간 네트워크 트래픽을 외부 스위치를 통해 교환할 수 있습니다.
- auto - LLDP를 사용하여 외부 스위치에서 반사 중계가 지원되는지를 확인합니다. 반사 중계가 외부 스위치에서 지원되는 경우 외부 스위치를 통해 VM 간 네트워크 트래픽을 교환할 수 있습니다. 그렇지 않으면 VM 간 네트워크 트래픽이 내부적으로 교환됩니다.

## 외부 스위치를 통해 통신하도록 VM을 사용으로 설정

동일한 물리적 NIC를 통해 여러 vNIC를 구성한 경우 vswitchmode 데이터 링크 등록 정보를 remote로 설정하여 스위치를 통해 네트워크 트래픽을 외부적으로 전송할 수 있습니다. 하지만 반사 중계 모드에서 외부 스위치를 구성해야 합니다. 반사 중계를 사용으로 설정하는 스위치 구성은 스위치 유형에만 한정됩니다. 자세한 내용은 스위치 제조업체의 설명서를 참조하십시오.

다음 그림은 10G 이더넷 링크가 외부 스위치에 연결되어 있고 동일한 고객에 대한 서비스를 실행 중인 두 영역(VM)을 호스트하는 동일한 시스템을 보여줍니다.

그림 4-3 영역 간 내부 통신



두 영역 Zone1 및 Zone2에서 동일한 고객에 대한 서비스를 실행하고 있으므로, 두 영역 간의 통신은 내부적으로 제한 없이 수행할 수 있습니다. 따라서 VNIC1 및 VNIC2 사이의 트래픽을 내부적으로 교환할 수 있습니다.

물리적 NIC net5에 대한 vswitchmode 등록 정보의 기존 값을 다음과 같이 확인합니다.

```
# dladm show-linkprop -p vswitchmode net5
LINK PROPERTY PERM VALUE EFFECTIVE DEFAULT POSSIBLE
net4 vswitchmode rw local local local local,remote,auto
```

출력에는 VALUE 및 EFFECTIVE 필드에 대한 local 값이 표시됩니다. 이 값은 영역 간 통신이 내부적으로 수행됨을 나타냅니다.

이 예에서는 두 영역 Zone1 및 Zone2에서 서로 다른 고객에 대한 서비스를 실행해야 하고 외부 스위치에는 이러한 서비스에 대한 네트워크 트래픽을 제어하는 액세스 제어 목록(ACL)이 구성되어 있습니다. 따라서 이들이 내부적으로 통신하면 안되고 vNIC1 및 vNIC2 간의 네트워크 트래픽은 스위치를 통해 외부적으로 교환해야 합니다.

따라서 다음과 같이 vswitchmode 등록 정보를 remote로 설정하여 영역 간의 내부 통신을 사용 안함으로 설정해야 합니다.

```
# dladm set-linkprop -p vswitchmode=remote net5

# dladm show-linkprop -p vswitchmode net5
LINK PROPERTY PERM VALUE EFFECTIVE DEFAULT POSSIBLE
net5 vswitchmode rw remote remote local local,remote,auto
```

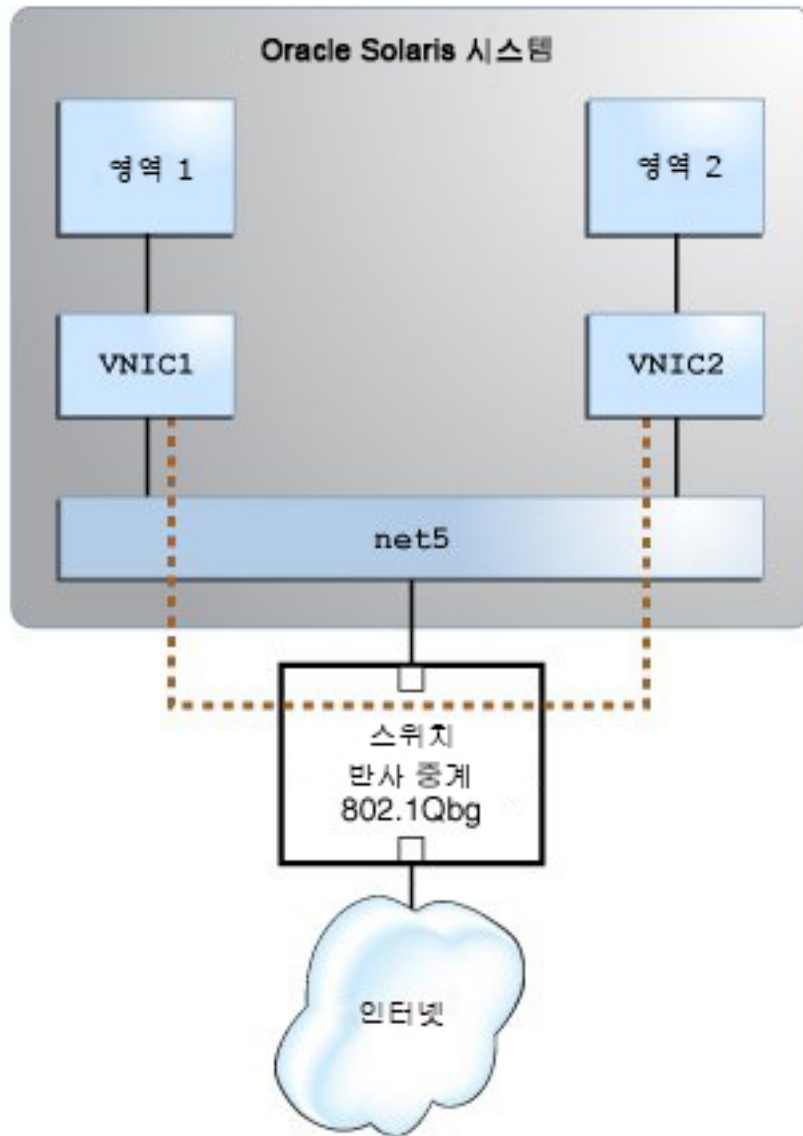
---

참고 - vswitchmode를 remote로 설정하기 전에 반사 중계에 대해 외부 스위치를 구성해야 합니다.

---

vswitchmode 등록 정보를 remote로 설정하여 vNIC의 내부 통신을 사용 안함으로 설정하므로, 다음 그림에 표시된 대로 vNIC 간의 네트워크 트래픽은 외부 스위치를 통해 송신됩니다.

그림 4-4 외부 스위치를 사용하여 영역 간 통신



## LLDP를 사용하여 VM 간 통신 관리

LLDP를 사용하여 VM 간 통신을 자동으로 구성할 수 있습니다. LLDP는 외부 스위치에서 반사 중계를 지원하는지에 따라 네트워크 트래픽 교환을 내부 또는 외부로 구성합니다. LLDP를 사용하려면 vswitchmode 데이터 링크 등록 정보를 auto로 설정합니다. 먼저 다음을 확인해야 합니다.

- LLDP 패키지가 설치되었는지 여부

LLDP 패키지가 설치되었는지 확인하려면 다음 명령을 사용합니다.

```
# pkg info lldp
```

- LLDP 서비스가 온라인 상태인지 여부

LLDP 서비스가 온라인 상태인지 확인하려면 다음 명령을 사용합니다.

```
# svcs lldp
```

```
STATE          STIME      FMRI
online         Jul_13    svc:/network/lldp:default
```

- dot1-tlv TLV 단위에서 EVB가 사용으로 설정되었는지 여부

- NIC에 대한 LLDP 모드가 both인지 여부

이 예에서 dot1-tlv TLV 단위에서 EVB가 사용으로 설정되어 있고 LLDP 모드가 both인지 확인하려면 다음 명령을 사용합니다.

```
# lladm show-agentprop -p mode,dot1-tlv net5
```

```
AGENT  PROPERTY  PERM  VALUE  DEFAULT  POSSIBLE
net5   mode      rw    both   disable  txonly,rxonly,both,disable
net5   dot1-tlv  rw    evb    none     none,vlanname,pvid,linkaggr,pfc,
                                           appln,evb,etscfg,etsreco,all
```

vswitchmode 데이터 링크 등록 정보를 auto로 설정하려는 경우:

```
# dladm set-linkprop -p vswitchmode=auto net5
```

vswitchmode 데이터 링크 등록 정보를 auto로 설정한 경우 dladm show-linkprop 명령의 출력을 사용하여 VM 간 통신이 내부적인지 외부 스위치를 통해 수행되는지 확인할 수 있습니다.

```
# dladm show-linkprop -p vswitchmode net5
```

```
LINK  PROPERTY  PERM  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
net5  vswitchmode  rw    auto   remote     local    local,remote,auto
```

출력의 EFFECTIVE 필드 값이 remote이면 LLDP가 외부 스위치에 대한 반사 중계를 사용으로 설정한 것이므로 VM 간 통신이 외부 스위치를 통해 수행됩니다.

LLDP에 대한 자세한 내용은 “Oracle Solaris 11.2의 네트워크 데이터 링크 관리”의 5 장, “Link Layer Discovery Protocol을 사용하여 네트워크 연결 정보 교환”을 참조하십시오.

## VDP를 사용하여 VNIC 정보 교환

VNIC(VSI) 정보는 VSI VDP(검색 및 구성 프로토콜)를 사용하여 시스템(스테이션)과 외부 스위치(브리지) 간에 교환됩니다. VDP TLV(Type-Length Value) 단위는 ECP(Edge Control Protocol)를 사용하여 교환됩니다. ECP는 피어 간의 VDP 패킷을 안정적으로 전송합니다. VNIC를 만들거나 삭제하면 VDP TLV 단위가 교환됩니다.

다음 EVB 구성 요소를 사용하여 VNIC(VSI) 정보를 외부 스위치에 알릴 수 있습니다.

- VSI 프로파일은 특정 VNIC에 대해 구성된 링크 등록 정보로 구성됩니다. 따라서 시스템은 구성된 VNIC 수만큼 VSI 프로파일을 보유할 수 있습니다.
- VSI 식별자는 VSI 인스턴스를 고유하게 식별합니다. Oracle Solaris에서 이 VSI 인스턴스는 VNIC(VSI)의 MAC 주소입니다. VSI 유형 ID 및 VSI 버전은 주어진 VSI 관리자 ID에서 프로파일을 식별합니다.
- VSI 관리자는 특정 VNIC 등록 정보 세트를 사용하여 VSI 유형 ID - VSI 버전을 매핑하여 시스템에서 여러 VSI 프로파일을 관리합니다. Oracle Solaris에서는 기본 VSI 관리자인 `oracle_v1`이 3바이트 인코딩으로 정의되어 있습니다. 이 3바이트 인코딩은 Oracle Solaris 호스트가 VDP 패킷에서 VSI 유형 ID로 사용합니다.
- VSI 관리자 ID는 특정 VSI 유형 ID - VSI 버전 쌍과 관련된 VSI 관리자를 식별합니다. VSI 관리자 ID는 IPv6 주소로 표시됩니다. Oracle Solaris에서는 기본 VSI 관리자 ID가 `ORACLE_VSIMGR_V1`로 정의되어 있습니다.

---

참고 - 현재는 VSI 프로파일 및 특정 등록 정보를 정의하는 표준이 정의되어 있지 않습니다. VSI 유형 정의는 판매자별로 다르며 VSI 관리자 ID와 밀접하게 연결되어 있습니다.

---

이 `oracle_v1` 인코딩은 다음 등록 정보를 지원합니다.

- 대역폭 제한
- 대역폭 공유
- 기본 링크의 링크 속도
- VNIC의 MTU(최대 전송 단위)

Oracle Solaris에서는 `oracle_v1` 인코딩을 사용하여 링크 정보를 인코딩한 다음 해당 정보를 외부 스위치로 전송합니다. 스위치는 정보를 수신한 다음 동일한 `oracle_v1` 인코딩을 사용하여 인코딩된 정보를 디코딩합니다.

기본적으로 Oracle Solaris 호스트는 다음 요소를 외부 스위치에 전송합니다.

- Oracle VSI 관리자 - `oracle_v1`
- VSI 유형 ID - `oracle_v1` 인코딩을 사용하여 인코딩된 VNIC 등록 정보
- VSI 버전 - 항상 0

Oracle Solaris에서 VNIC 정보 교환 방식은 다음과 같습니다.

1. 외부 스위치는 Oracle VSI 관리자 `oracle_v1`을 지원하도록 구성됩니다.

2. 외부 스위치는 `oracle_v1`을 사용하여 VSI 유형 ID로 인코딩되는 등록 정보를 결정합니다.
3. 외부 스위치는 해당 VNIC에 대한 패킷에 등록 정보 구성을 적용합니다.

Oracle 구성별 OUI TLV 단위는 VSI 관리자 ID TLV에 따라 Oracle 관련 VSI 관리자 ID임을 나타냅니다. 스위치의 응답에 Oracle 관련 TLV 단위가 없으면 스위치가 Oracle VSI 관리자 (인코딩)를 지원하지 않는 Oracle Solaris 호스트임을 나타냅니다. Oracle Switch ES1-24는 Oracle VSI 관리자 `oracle_v1`을 지원합니다. Oracle Switch ES1-24의 EVB 구성에 대한 자세한 내용은 [Sun Ethernet Fabric Operating System, EVB Administration Guide](#)를 참조하십시오.

---

참고 - VDP 및 ECP 프로토콜을 지원하는 이외에 Oracle Solaris 시스템과 통합하려면 외부 스위치는 `ORACLE_VSIMGR_V1`(기본 Oracle VSI 관리자 ID)과 Oracle OUI(Organizationally Unique Identifier) TLV(하위 유형 `VDP_ORACLEOUI_VSIMGR_SUBTYPE`, 인코딩 정보를 전달하는 데 사용됨)도 지원해야 합니다.

---

## VDP에서 VNIC 정보를 교환하는 방법

VNIC 정보를 교환하는 방법은 다음과 같습니다.

VNIC 및 연관된 프로파일을 지정하여 연관 요청(ASSOC)을 외부 스위치에 전송합니다. 외부 스위치는 성공 또는 실패 응답으로 연관 요청에 응답합니다. 그런 다음 시스템에서 VNIC에 대한 연관을 제거하는 연관 해제 요청(DEASSOC) 외부 스위치로 전송할 수 있습니다. VNIC에 대한 요청 상태를 표시 및 가져오는 방법에 대한 자세한 내용은 [“VDP와 ECP의 상태 및 통계 표시” \[81\]](#)를 참조하십시오.

VNIC를 만들 때 VDP 교환이 다음과 같이 수행됩니다.

1. VNIC에 대한 정보를 포함하는 VDP 연관(ASSOC) 요청 TLV 단위가 시스템에 의해 외부 스위치로 전송됩니다.
2. 외부 스위치는 VDP(ASSOC) TLV 단위를 수신하고 VSI 유형 ID, VSI 버전 및 VSI 관리자 ID를 사용하여 VNIC 정보를 가져옵니다.
3. 외부 스위치는 VNIC에 대한 등록 정보 구성을 적용합니다.
4. 외부 스위치는 VDP 연관(ASSOC) 응답 TLV 단위를 시스템에 전송하여 외부 스위치가 해당 VNIC에 대한 등록 정보 구성을 완료했음을 알립니다.

VNIC를 삭제할 때 VDP 교환이 다음과 같이 수행됩니다.

1. VSI ID를 포함하는 VDP 연관 해제(DEASSOC) 요청 TLV 단위가 시스템에 의해 외부 스위치에 전송됩니다.
2. 외부 스위치는 VDP(DEASSOC) TLV 단위를 수신하고 삭제되는 VSI의 VSI ID를 가져옵니다.
3. 외부 스위치는 삭제된 VNIC에 대한 구성을 제거합니다.
4. 외부 스위치는 VDP 연관 해제(DEASSOC) 응답 TLV 단위를 시스템에 전송합니다.



참고 - Oracle Solaris에서 VDP는 ASSOC 및 DEASSOC VDP 요청만 지원합니다.

## VDP와 ECP의 상태 및 통계 표시

시스템에서 EVB를 사용으로 설정한 경우와 VNIC에 대한 VDP 패킷을 교환 중인 경우에 물리적 이더넷 링크의 VDP 상태에 대한 정보를 표시할 수 있습니다. 단일 링크에 대한 정보만 표시하려면 명령에 해당 링크를 지정하십시오. 그렇지 않은 경우 모든 이더넷 링크에 대한 VDP 정보가 표시됩니다.

### VDP 상태 및 통계 표시

VDP 상태를 표시하려면 다음 명령을 입력합니다.

```
# dladm show-ether -P vdp
VSI      LINK      VSIID          VSI-TYPEID  VSI-STATE  CMD-PENDING
vnic1    net0      2:8:20:22:3c:6b  98/0        ASSOC      NONE
vnic2    net0      2:8:20:90:7f:ef  96/0        ASSOC      NONE
```

VSI-STATE는 피어와의 VDP 교환 상태를 표시합니다. 가능한 값은 다음과 같습니다.

- TIMEDOUT - 피어가 VDP 요청에 응답하지 않았습니다.
- ASSOC - 피어가 요청을 성공적으로 처리했습니다.
- DEASSOC - 호스트 또는 피어가 요청을 거부했습니다. 피어는 지정된 등록 정보 또는 프로 파일을 확인할 수 없는 경우 요청을 거부할 수 있습니다. 호스트는 oracle\_v1 인코딩을 사용 중이고 피어가 응답에 Oracle OUI를 포함하지 않은 경우 VDP 패킷 교환을 거부할 수 있습니다.

net0 링크에 대해 VSI(VNIC) 2개가 구성되어 있다고 샘플 출력에 표시됩니다. 해당 특정 VSI ID는 해당 MAC 주소를 참조합니다. VNIC에 대한 VSI-TYPE ID인 vnic1 및 vnic2가 해당 등록 정보(대역폭 제한 및 MTU)에서 생성되고 oracle\_v1에 의해 인코딩이 정의됩니다.

발신 또는 수신 VDP 패킷에 대한 통계를 가져오려면 다음 명령을 입력합니다.

```
# dlstat show-ether -P vdp net1
LINK  IPKTS  OPKTS  KeepAlives
net1  3      2      1
```

### 링크 등록 정보 표시

링크 등록 정보를 표시하려면 dladm show-linkprop 명령의 -p 옵션을 사용합니다.

다음 예에서는 vnic1 및 vnic2에 대한 링크 등록 정보를 표시하는 방법을 보여줍니다.

```
# dladm show-linkprop -p maxbw,mtu vnic1
LINK      PROPERTY  PERM  VALUE    EFFECTIVE  DEFAULT  POSSIBLE
vnic1     maxbw     rw    100     100       --       --
vnic1     mtu       rw    1500    1500     1500    1500
```

```
# dladm show-linkprop -p maxbw,mtu vnic2
LINK      PROPERTY  PERM  VALUE    EFFECTIVE  DEFAULT  POSSIBLE
vnic2     maxbw     rw    20      20        --       --
vnic2     mtu       rw    1500    1500     1500    1500
```

## ECP 상태 및 통계 표시

VDP는 ECP를 사용하여 메시지를 교환합니다. 다음 예에서는 물리적 링크 net0에 특정한 ECP 상태를 표시합니다.

```
# dladm show-ether -P ecp net0
LINK      MAX-RETRIES  TIMEOUT
net0      3            164
```

MAX-RETRIES            피어로부터 응답을 받지 못한 경우에 ECP가 패킷을 전송하는 횟수를 지정합니다.

TIMEOUT                패킷을 재전송하는 간격(밀리초)을 지정합니다. ECP가 패킷을 재전송하기 전에 응답을 대기하는 시간 간격입니다.

물리적 링크에 대한 통계를 가져오려면 다음 명령을 입력합니다.

```
# dlstat show-ether -P ecp
LINK      IPKTS  OPKTS  IERRORS  OERRORS  RETRANSMITS  TIMEOUTS
net0      3      2      0        0        1            0
```

## 기본 EVB 구성 변경

기본적으로 기본 EVB 구성을 변경할 필요가 없습니다. 대부분의 경우 EVB를 설치하고 기본 EVB 구성을 사용하여 시스템에 구성된 모든 VNIC에 대한 정보를 외부 스위치와 교환할 수 있습니다. 하지만 호스트와 네트워크에서 EVB 구성을 완전히 제어하고 관리하려면 기본 구성을 변경할 수 있습니다.

기본 Oracle Solaris VSI 관리자 ID인 ORACLE\_VSIMGR\_V1을 사용하는 경우 만드는 VNIC에 대한 VSI 유형 ID가 자동으로 생성됩니다. 따라서 데이터 링크 등록 정보(예: vsi-typeid 및 vsi-vers)를 설정할 필요가 없습니다. 하지만 기본 VSI 관리자 ID를 사용하지 않을 경우 dladm set-linkprop 명령을 사용하여 EVB와 관련된 데이터 링크 등록 정보를 설정해야 합

니다. EVB와 관련된 데이터 링크 등록 정보를 설정하려면 외부 스위치가 시스템과 통신하고 주어진 VSI 유형 ID 및 VSI 버전 세트에 대한 등록 정보를 검색할 수 있어야 합니다.

Oracle VSI 관리자가 시스템의 VSI 프로파일에 대한 VSI 유형 ID와 VSI 버전을 자동으로 생성할 수 있도록 EVB를 사용할 경우 기본 Oracle VSI 관리자 ID를 사용합니다.

EVB와 관련된 다음 데이터 링크 등록 정보를 구성할 수 있습니다.

- `vsi-mgrid` - 물리적 링크 또는 VNIC에 대해 설정되는 VSI 관리자 ID를 지정합니다. VNIC에 대해 이 등록 정보를 설정하지 않은 경우 기본 물리적 링크의 기본값인 `ORACLE_VSIMGR_V1`이 사용됩니다.  
`vsi-mgrid` 등록 정보를 명시적으로 설정한 경우 VSI 유형 ID와 VSI 버전도 명시적으로 설정해야 합니다. 또한 데이터 링크에 대해 이러한 등록 정보를 명시적으로 구성해야 합니다.

참고 - Oracle Solaris에서 VSI 관리자 ID, VSI 유형 ID 및 VSI 버전을 수동으로 구성한 경우에는 해당 VNIC 등록 정보가 자동으로 구성되지 않습니다.

- `vsi-mgrid-enc` - VSI 관리자 ID와 연결된 인코딩을 나타냅니다. 기본적으로 이 등록 정보는 `oracle_v1`로 설정됩니다. `oracle_v1`을 VSI 관리자 ID와 연결하지 않으려면 이 등록 정보 값을 `none`으로 설정하십시오. 값을 `none`으로 설정한 경우 VSI 관리자 ID, VSI 유형 ID 및 VSI 버전이 자동으로 생성되지 않으므로 이러한 등록 정보를 수동으로 구성해야 합니다.
- `vsi-typeid` - VSI 유형 ID를 지정합니다. VSI 유형 ID는 VSI 프로파일과 연결되는 VSI 버전과 쌍을 이룹니다. `vsi-mgrid` 및 `vsi-mgrid-enc`의 기본값을 사용하는 경우 3바이트 값이 자동으로 생성됩니다. 그렇지 않은 경우 이 등록 정보의 값을 명시적으로 지정해야 합니다.
- `vsi-vers` - VSI 버전을 지정합니다. VSI 버전은 VSI 프로파일과 연결되는 VSI 유형 ID와 쌍을 이룹니다. `vsi-mgrid` 및 `vsi-mgrid-enc`의 기본값을 사용하는 경우 1바이트 값이 자동으로 생성됩니다. 그렇지 않은 경우 이 등록 정보의 값을 명시적으로 지정해야 합니다.

`dladm show-linkprop` 명령을 사용하여 EVB 관련 등록 정보를 표시할 수 있습니다. 등록 정보의 해당 EFFECTIVE 필드 값에서 VNIC 관련 링크 등록 정보의 유효 값을 가져올 수 있습니다. 자세한 내용은 예 4-2. “물리적 링크의 EVB 관련 데이터 링크 등록 정보 표시”를 참조하십시오.

EVB 구성 요소에 대한 자세한 내용은 “VDP를 사용하여 VNIC 정보 교환” [79]을 참조하십시오. EVB에 대한 자세한 내용은 [evb\(7P\)](#) 매뉴얼 페이지를 참조하십시오.

## ▼ 기본 EVB 구성을 변경하는 방법

물리적 링크에 대해 `vsi-mgrid` 및 `vsi-mgrid-enc` 등록 정보만 구성해야 합니다. 다른 EVB 관련 등록 정보(예: `vsi-typeid` 및 `vsi-vers`)는 VNIC에 대해 구성해야 합니다.

1. 관리자로 로그인합니다.  
자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”을 참조하십시오.
2. 프로파일 데이터베이스에 언급된 데이터 링크 등록 정보를 사용하여 VNIC를 만듭니다.  
`# dladm create-vnic -l datalink -p maxbw=maxbw-value,priority=priority-value VNIC`
3. 기본 Oracle VSI 관리자 ID를 사용하지 않을 것이므로 물리적 링크에서 VSI 관리자 ID와 연관된 인코딩을 none으로 설정합니다.  
`# dladm set-linkprop -p vsi-mgrid-enc=none datalink`
4. IPv6 주소를 사용하여 물리적 링크에 대한 VSI 관리자 ID를 설정합니다.  
`# dladm set-linkprop -p vsi-mgrid=IPv6-address datalink`
5. 생성된 VNIC에 대한 VSI 유형 ID 및 VSI 버전을 설정합니다.  
`# dladm set-linkprop -p vsi-typeid=VSI-Type-ID,vsi-vers=VSI-Version VNIC`
6. VNIC에 대해 설정된 등록 정보를 확인합니다.  
`# dladm show-linkprop VNIC`

예 4-1 EVB 관련 데이터 링크 등록 정보 설정

다음 예는 EVB와 관련된 데이터 링크 등록 정보를 설정하는 방법을 보여줍니다. 이 예에서는 IPv6 주소 IP1을 사용하여 액세스할 수 있는 프로파일이 있는 시스템을 사용합니다.

VSI 관리자 ID인 IP1에 다음과 같은 프로파일이 정의되어 있다고 가정합니다.

- VSI 유형 ID: 2
- VSI 버전: 1
- 데이터 링크 등록 정보: maxbw=20, priority=5

1. 프로파일에 언급된 데이터 링크 등록 정보를 사용하여 VNIC를 만듭니다.  
`# dladm create-vnic -l net0 -p maxbw=20,priority=5 vnic1`
2. 기본 Oracle VSI 관리자 ID를 사용하지 않을 것이므로 물리적 링크 net0에서 VSI 관리자 ID와 연관된 인코딩을 none으로 설정합니다.  
`# dladm set-linkprop -p vsi-mgrid-enc=none net0`
3. IPv6 주소가 IP1인 물리적 링크 net0에서 VSI 관리자 ID를 설정합니다.  
`# dladm set-linkprop -p vsi-mgrid=IP1 net0`
4. vnic1에 대한 VSI 유형 ID 및 VSI 버전을 설정합니다.  
`# dladm set-linkprop -p vsi-typeid=2,vsi-vers=1 vnic1`

5. vnic1에 대해 설정된 등록 정보를 확인합니다.

```
# dladm show-linkprop vnic1
LINK      PROPERTY          PERM VALUE    EFFECTIVE  DEFAULT    POSSIBLE
...
vnic1     vsi-typeid        rw  2           2          --         --
vnic1     vsi-vers          rw  1           1          --         --
vnic1     vsi-mgrid         rw  IP1          IP1        --         --
vnic1     vsi-mgrid-enc     rw  --          none       oracle_v1  none,oracle_v1
...
```

vnic1에 대한 VDP ASSOC TLV 단위에는 다음 정보가 포함되어 있습니다.

- VSI 관리자 ID = IP1
- VSI 유형 ID = 2
- VSI 버전 = 1

예 4-2 물리적 링크의 EVB 관련 데이터 링크 등록 정보 표시

다음 예는 물리적 링크의 EVB 관련 등록 정보를 보여줍니다.

```
# dladm show-linkprop -p vsi-mgrid,vsi-mgrid-enc net4
LINK      PROPERTY          PERM VALUE    EFFECTIVE  DEFAULT    POSSIBLE
net4      vsi-mgrid         rw  --          --         ::         --
net4      vsi-mgrid-enc     rw  --          --         oracle_v1  none,oracle_v1
```

출력에는 Oracle Solaris의 EVB 기본 구성이 표시됩니다. oracle\_v1 인코딩을 사용하여 VNIC에 대해 구성된 등록 정보에서 VSI 유형 ID와 VSI 버전을 자동으로 생성합니다.

예 4-3 VNIC의 EVB 관련 등록 정보 표시

다음 예는 VNIC의 EVB 관련 등록 정보를 보여줍니다.

```
# dladm show-linkprop vnic0
LINK      PROPERTY          PERM VALUE    EFFECTIVE  DEFAULT    POSSIBLE
...
vnic0     vsi-typeid        rw  --          94         --         --
vnic0     vsi-vers          rw  --          0          --         --
vnic0     vsi-mgrid         rw  --          ::         --         --
vnic0     vsi-mgrid-enc     rw  --          oracle_v1  oracle_v1  none,oracle_v1
...
```

출력에는 vnic0에 대한 유효 인코딩이 oracle\_v1로 표시됩니다. vsi-typeid 94에 대한 EFFECTIVE 값은 자동으로 생성되고 vnic0에 대해 유효합니다.



# ◆◆◆ 5 장

## 탄력적 가상 스위치 정보

---

Oracle Solaris 11.2 릴리스부터는 Oracle Solaris EVS(탄력적 가상 스위치) 기능을 사용하여 여러 물리적 시스템에 걸쳐 있는 여러 가상 스위치를 관리할 수 있습니다. 이 장에서는 Oracle Solaris의 탄력적 가상 스위치 기능에 대한 개요를 제공하고 다음 내용에 대해 다룹니다.

- “EVS(탄력적 가상 스위치) 기능의 개요” [87]
- “EVS 구성 요소” [94]
- “EVS 관리 명령” [97]
- “EVS 사용을 위한 필수 패키지” [102]
- “EVS의 영역 작업 방법” [102]
- “EVS 사용을 위한 보안 요구 사항” [103]

### EVS(탄력적 가상 스위치) 기능의 개요

현대의 데이터 센터에는 네트워크 패브릭을 통해 연결되는 여러 VM(가상 머신)을 호스트하는 여러 대의 물리적 서버가 포함되어 있습니다. 데이터 센터에서 VM에 대해 네트워킹을 프로비전하는 것이 관리자의 주요 과제이며, VM 간 가상 네트워킹, MAC 주소 및 IP 주소 관리, VLAN 및 VXLAN 관리도 포함됩니다. 또 다른 과제로는 VM에 대한 내부 및 외부 네트워크 연결을 제외하고 VM 및 VM 내의 응용 프로그램에 대한 SLA(서비스 단계 계약)를 프로비전하고 적용하는 것입니다. 이러한 SLA에는 대역폭 제한 및 우선 순위가 포함됩니다. 또한 데이터 센터 관리자는 공통 네트워크 기반구조를 공유하는 여러 테넌트를 각각 격리해야 합니다.

이러한 요구 사항을 충족하기 위해 관리자는 Oracle Solaris 네트워크 가상화 기능을 사용하여 데이터 센터에서 가상 스위치를 관리할 수 있습니다. 가상 스위치는 첫번째 클래스 운영체제 추상화로 노출됩니다. 이러한 가상 스위치(탄력적 가상 스위치라고도 함)는 여러 물리적 서버에 걸쳐 있으며, 시스템 관리자가 해당 물리적 서버를 단일 가상 스위치로 관리할 수 있도록 해줍니다.

## Oracle Solaris의 가상 스위치

가상 스위치는 가상 머신 간의 원활한 통신을 가능하게 하는 엔티티입니다. Oracle Solaris에서 가상 스위치는 링크 집계, 물리적 NIC, Etherstub와 같은 데이터 링크를 통해 vNIC를 만들 때 자동으로 또는 암시적으로 생성됩니다. 가상 스위치는 물리적 시스템 내의 VM 간에 트래픽 루프를 생성하며(VM 간 트래픽) 전송 중에 이 트래픽을 내보내지 않습니다. 모든 VM은 동일한 계층 2 세그먼트에 있어야 서로 통신할 수 있습니다. 자세한 내용은 “[가상 스위치](#)” [14]를 참조하십시오.

Oracle Solaris 11.2 이전 릴리스에서는 vNIC를 생성하는 데 사용된 데이터 링크를 통해 가상 스위치를 간접적으로 관리했습니다. Oracle Solaris 11.2 릴리스부터는 EVS를 통해 가상 스위치를 관리할 수 있습니다. 가상 스위치를 명시적으로 만든 후 이름을 지정하고, 가상 스위치에 가상 포트(vPort)를 지정한 다음 가상 스위치를 IP 주소 블록과 연결할 수 있습니다. 가상 포트에 대한 우선 순위, 최대 대역폭, 서비스 클래스(CoS), MAC 주소 및 IP 주소와 같은 등록 정보를 설정할 수 있습니다. 가상 스위치 단위로 기본 SLA를 구성할 수도 있습니다.

---

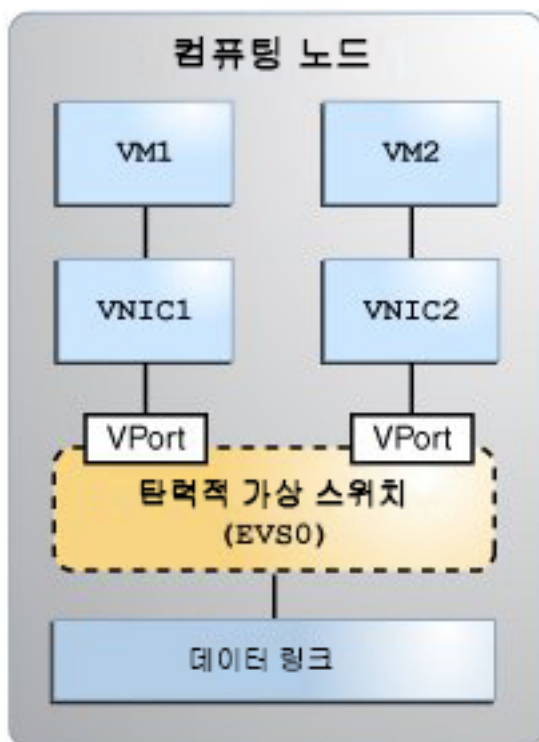
**참고** - vNIC를 만드는 과정에서 암시적으로 생성되는 가상 스위치는 이 릴리스에서도 그대로 유지되며 이전 릴리스에서와 동일한 기능을 수행합니다. EVS는 기존 암시적 가상 스위치를 대체하지 않습니다.

---

다음 그림은 단일 컴퓨팅 노드의 탄력적 가상 스위치 `EVS0`을 보여줍니다.



그림 5-1 컴퓨팅 노드의 탄력적 가상 스위치



## Oracle Solaris 탄력적 가상 스위치 기능이란?

Oracle Solaris EVS(탄력적 가상 스위치) 기능을 사용하면 하나 이상의 계산 노드에 걸쳐 있는 가상 스위치를 만들고 관리할 수 있습니다. 이러한 계산 노드는 VM을 호스트하는 물리적 시스템입니다. 탄력적 가상 스위치는 동일한 L2(계층 2) 세그먼트에 속하는 명시적으로 생성된 가상 스위치를 나타내는 엔티티입니다. 탄력적 가상 스위치는 네트워크 내의 어디서나 가상 스위치에 연결된 VM 간의 네트워크 연결을 제공합니다.

참고 - EVS에서 가상 머신(VM) 용어에 대한 모든 참조는 구체적으로 Oracle Solaris 영역 및 Oracle Solaris 커널 영역을 가리킵니다.

탄력적 가상 스위치는 여러 호스트에 걸쳐 있을 수 있습니다. 이러한 가상 스위치는 호스트로 들어가고 호스트에서 나올 수 있는 기능을 가지므로 "탄력적"이라고 말할 수 있습니다. 탄력적 가상 스위치는 호스트의 VNIC를 탄력적 가상 스위치에 연결할 때 호스트로 들어갑니다. 이러한 VNIC를 삭제하면 탄력적 가상 스위치가 호스트에서 나옵니다.

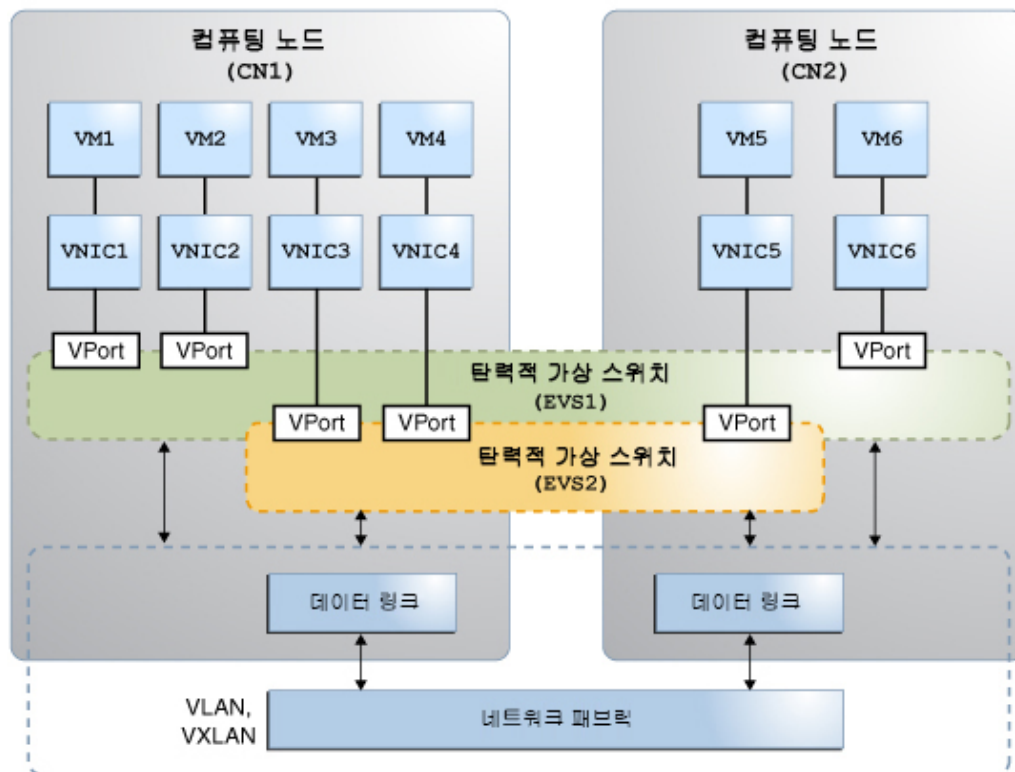
탄력적 가상 스위치는 격리된 L2 세그먼트를 나타내고, 격리는 VLAN 또는 VXLAN을 통해 구현됩니다. VLAN을 통해 탄력적 가상 스위치를 구현하는 방법에 대한 자세한 내용은 [“사용 사례: 탄력적 가상 스위치 구성” \[137\]](#)을 참조하십시오. VXLAN을 통해 탄력적 가상 스위치를 구현하는 방법에 대한 자세한 내용은 [“사용 사례: 테넌트에 대한 탄력적 가상 스위치 구성” \[142\]](#)을 참조하십시오.

VLAN 관리에 대한 자세한 내용은 [“Oracle Solaris 11.2의 네트워크 데이터 링크 관리”의 3장, “가상 LAN을 사용하여 가상 네트워크 구성”](#)을 참조하십시오. VXLAN 관리에 대한 자세한 내용은 [3장. VXLAN\(Virtual Extensible Local Area Network\)을 사용하여 가상 네트워크 구성](#)을 참조하십시오.

모든 탄력적 가상 스위치는 이름, 가상 포트 및 IP 주소 블록에 연결됩니다. 가상 스위치 리소스를 생성, 모니터링 및 제어할 수 있습니다. 자세한 내용은 [6장. 탄력적 가상 스위치 관리](#)를 참조하십시오.

다음 그림은 두 계산 노드 사이의 두 탄력적 가상 스위치(EVS1 및 EVS2)를 보여줍니다. 이러한 계산 노드에서 프로비전되는 VM은 두 계산 노드에 걸쳐 있는 탄력적 가상 스위치를 통해 연결됩니다. 각 컴퓨팅 노드는 데이터 링크를 통해 동일한 네트워크 패브릭에 연결됩니다. 데이터 링크는 업링크 포트라고도 합니다. 이러한 컴퓨팅 노드의 데이터 링크에서는 가상 스위치를 외부 네트워크에 연결합니다. VNIC는 가상 포트 (VPort)를 통해 탄력적 가상 스위치에 연결됩니다. VNIC는 가상 포트와 연결되는 등록 정보(예: MAC 주소, IP 주소, SLA)를 상속합니다.

그림 5-2 컴퓨팅 노드 간의 탄력적 가상 스위치



이 그림에서 VM VM1, VM2 및 VM6는 탄력적 가상 스위치 EVS1을 통해 서로 통신할 수 있습니다. VM VM3, VM4 및 VM5는 탄력적 가상 스위치 EVS2를 통해 서로 통신할 수 있습니다. 자세한 내용은 [탄력적 가상 스위치를 구성하는 방법 \[119\]](#)을 참조하십시오.

## EVS 사용 이점

여러 가상 머신을 호스트하는 데이터 센터 환경에서 EVS는 다음과 같은 이점을 제공하여 일부 네트워크 관리 작업을 간소화해 줍니다.

- 네트워크 연결을 제공하여 여러 서버에 있는 VM 간의 가상 네트워크 생성
- 사용자 정의 SLA를 통해 가상 포트 추가 지원
- VLAN 또는 VXLAN을 사용하여 네트워크 격리 제공

- 동일한 기본 기반구조를 공유하는 다중 테넌트 가상 네트워크 지원
- Oracle Solaris 영역 및 Oracle Solaris 커널 영역과 통합
- 중앙화된 관리 제공:
  - 가상 포트에 대한 MAC 주소 및 IP 주소
  - 가상 스위치 또는 가상 포트 단위 SLA
  - 가상 포트의 런타임 네트워크 트래픽 통계 모니터링

## 탄력적 가상 스위치 리소스

탄력적 가상 스위치는 IP 네트워크 및 가상 포트와 같은 주요 리소스와 연결됩니다.

### IP 네트워크

IP 네트워크(IPnet이라고도 함)는 IPv4 또는 IPv6 주소 블록을 블록에 대한 기본 라우터와 함께 나타냅니다. 이 IPv4 또는 IPv6 주소 블록은 서브넷이라고도 합니다. 탄력적 가상 스위치에 IPnet을 하나만 연결할 수 있습니다. 가상 포트를 통해 탄력적 가상 스위치에 연결되는 모든 VM에는 탄력적 가상 스위치와 연결된 IPnet에서 IP 주소가 지정됩니다.

VPort에 대한 IP 주소 등록 정보 `ipaddr`를 설정하여 VM에 IP 주소를 수동으로 지정할 수도 있습니다. IP 주소는 IPnet의 서브넷 범위 내에 있어야 합니다. IPnet을 탄력적 가상 스위치에 추가하는 방법에 대한 자세한 내용은 [탄력적 가상 스위치를 구성하는 방법 \[119\]](#)을 참조하십시오.

### 가상 포트

가상 포트(VPort라고도 함)는 VNIC과 탄력적 가상 스위치 간의 연결 지점을 나타냅니다. VNIC가 VPort에 연결되는 경우 VNIC는 VPort에서 캡슐화하는 다음과 같은 네트워크 구성 매개변수를 상속합니다.

- SLA 매개변수(예: 최대 대역폭, 서비스 클래스, 우선 순위)
- MAC 주소
- IP 주소

VPort를 만들 때 임의로 생성된 MAC 주소와 연결된 IPnet에서 다음으로 사용 가능한 IP 주소가 VPort에 지정됩니다. 임의로 생성된 MAC 주소는 유효한 IEEE OUI 및 로컬 비트 세트로 구성된 기본 접두어를 가집니다. `evsadm add-vport` 명령을 사용하여 VPort를 추가할 때 IP 주소와 MAC 주소를 지정할 수도 있습니다. VPort를 추가하는 방법에 대한 자세한 내용은 [탄력적 가상 스위치를 구성하는 방법 \[119\]](#)을 참조하십시오.

**참고** - 탄력적 가상 스위치에 가상 포트를 항상 추가할 필요는 없습니다. VNIC가 생성되는 경우 VNIC를 연결할 탄력적 가상 스위치의 이름만 지정할 수 있습니다. 이 경우 EVS 컨트롤러에서 시스템 가상 포트를 생성합니다. 이러한 가상 포트는 이름 지정 규칙 `sys-vportname`을 따릅니다(예: `sys-vport0`). 시스템 가상 포트는 탄력적 가상 스위치 등록 정보를 상속합니다.

다음 표에서는 VPort 등록 정보를 보여줍니다.

표 5-1 VPort 등록 정보

VPort 등록 정보	설명	가능한 값	기본값
cos	VPort의 아웃바운드 패킷에 대한 802.1p 우선 순위를 지정합니다.	0 - 7	--
maxbw	VPort에 대한 전이중 대역폭을 지정합니다.	--	--
priority	VPort에 대한 상대 우선 순위를 지정합니다.	high, medium 또는 low	medium
ipaddr	가상 포트와 연결되는 IP 주소를 지정합니다. VPort를 만드는 경우에만 IP 주소를 지정할 수 있습니다.	--	VPort에 대한 IP 주소를 지정하지 않은 경우 EVS 컨트롤러가 탄력적 가상 스위치에 연결된 IPnet에서 IP 주소를 자동으로 선택합니다.
macaddr	VPort에 연결된 MAC 주소를 지정합니다. VPort를 만드는 경우에만 MAC 주소를 지정할 수 있습니다.	--	VPort에 대한 MAC 주소를 지정하지 않은 경우 EVS 컨트롤러에서 VPort에 대한 MAC 주소를 임의로 생성합니다.
evs	VPort가 연결되는 탄력적 가상 스위치를 나타내는 읽기 전용 등록 정보입니다.	--	--
tenant	VPort가 연결되는 테넌트를 나타내는 읽기 전용 등록 정보입니다.	--	--

등록 정보 `evs` 및 `tenant`는 읽기 전용 등록 정보이므로 수정할 수 없습니다. VPort 등록 정보에 대한 자세한 내용은 [evsadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## EVS의 이름 공간 관리

탄력적 가상 스위치와 해당 리소스는 논리적으로 함께 그룹화됩니다. 각 논리 그룹을 테넌트라고 합니다. 테넌트 내의 탄력적 가상 스위치에 대해 정의된 리소스는 해당 테넌트의 이름 공간 외부에서 보이지 않습니다. 테넌트는 모든 테넌트 리소스를 함께 보관하는 컨테이너 역할을 합니다. 테넌트를 사용하여 탄력적 가상 스위치를 만드는 방법에 대한 자세한 내용은 [탄력적 가상 스위치를 구성하는 방법 \[119\]](#)을 참조하십시오.

EVS 작업에 대한 테넌트 이름을 지정할 필요가 없습니다. 기본 테넌트 이름은 `sys-global`이고 모든 EVS 작업은 이 이름 공간에서 발생합니다.

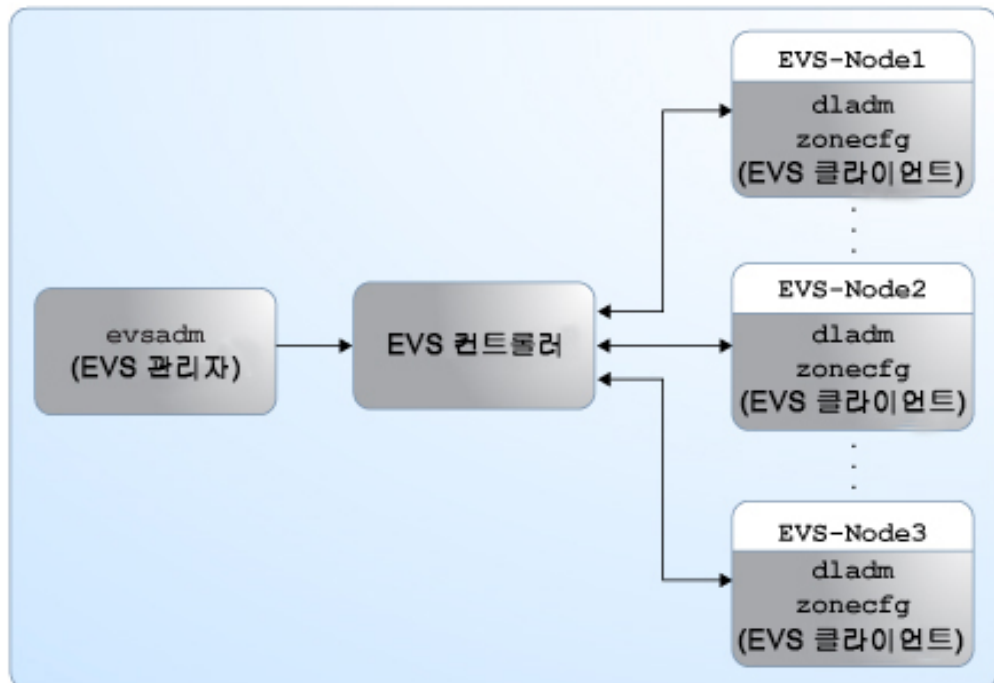
## EVS 구성 요소

EVS의 구성 요소는 다음과 같습니다.

- EVS 관리자
- EVS 컨트롤러
- EVS 클라이언트
- EVS 노드

다음 그림에서는 EVS의 구성 요소를 보여줍니다.

그림 5-3 EVS 구성 요소



이 그림에서 EVS 관리자와 EVS 컨트롤러는 두 개의 별도 호스트입니다. EVS 노드 `evs-Node1`, `evs-Node2` 및 `evs-Node3`은 VNIC 또는 영역의 VNIC `anet` 리소스가 탄력적 가상 스위치에 연결되는 세 개의 호스트입니다.

## EVS 관리자

EVS 관리자는 EVS 컨트롤러와 통신하여 L2 네트워크 토폴로지와 이 L2 네트워크에 사용되는 IP 주소를 정의하는 엔티티입니다. EVS 관리자는 `evsadm` 명령을 사용하여 EVS 컨트롤러와 통신합니다. EVS 관리자와 EVS 컨트롤러가 동일한 컴퓨팅 노드에 있을 수도 있습니다.

---

**참고** - L2 네트워크 토폴로지는 네트워크 세그먼트이고, 각 세그먼트는 VLAN 또는 VXLAN을 사용하여 구현되는 단일 브로드캐스트 도메인을 형성합니다.

---

`service/network/evs` 패키지를 설치하고 `evsadm set-prop` 명령에서 `controller` 등록 정보를 사용하여 EVS 컨트롤러를 지정한 다음 EVS 관리자에서 EVS 작업을 수행할 수 있습니다. `controller` 등록 정보는 `ssh://[user@]example-controller.com` 형식으로 지정됩니다. 자세한 내용은 [6장. 탄력적 가상 스위치 관리](#)를 참조하십시오.

## EVS 컨트롤러

EVS 컨트롤러는 탄력적 가상 스위치 및 모든 연결된 리소스를 구성하고 관리하기 위한 기능을 제공합니다. 데이터 센터에서 하나의 물리적 시스템만 EVS 컨트롤러로 설정해야 합니다.

`evsadm set-prop` 명령에서 `controller` 등록 정보를 사용하여 EVS 컨트롤러를 지정합니다. `controller` 등록 정보는 `svc:/network/evs:default` SMF 서비스에 저장되므로 시스템을 부트해도 영구적으로 유지됩니다.

EVS 컨트롤러는 `evsadm set-controlprop` 명령을 사용하여 구성할 수 있는 등록 정보와 연결됩니다. 물리적 시스템을 통해 L2 세그먼트를 구현하려면 사용 가능한 VLAN ID, 사용 가능한 VXLAN 세그먼트 ID, 각 EVS 노드에 대한 업링크 포트와 같은 정보를 사용하여 EVS 컨트롤러의 등록 정보를 구성해야 합니다. EVS 컨트롤러를 구성하고 등록 정보를 설정하는 방법에 대한 자세한 내용은 [“EVS 컨트롤러 만들기 및 관리” \[107\]](#)를 참조하십시오.

---

**참고** - SMF 사이트 프로파일과 AI(자동 설치) 서비스를 사용하여 EVS 컨트롤러 정보를 데이터 센터의 각 EVS 노드에 푸시할 수도 있습니다. SMF에 대한 자세한 내용은 [“Oracle Solaris 11.2의 시스템 서비스 관리”](#)를 참조하십시오. AI 서비스에 대한 자세한 내용은 [“Oracle Solaris 11.2 시스템 설치”](#)의 [“설치 서비스 작업”](#)을 참조하십시오.

---

다음 표는 EVS 컨트롤러에 대해 구성할 수 있는 등록 정보를 보여줍니다.

표 5-2 EVS 컨트롤러 등록 정보

EVS 컨트롤러 등록 정보	설명	가능한 값	기본값
l2-type	물리적 시스템을 통해 탄력적 가상 스위치를 구현하는 방법을 정의합니다. 참고 - l2-type 등록 정보를 변경 하더라도 변경 이전에 만든 탄력적 가상 스위치에는 영향을 주지 않습니다. 변경 이후에 만든 탄력적 가상 스위치만 업데이트된 l2-type 등록 정보를 가집니다. 이 동작은 VLAN과 VXLAN을 기반으로 하는 L2 세그먼트가 EVS 컨트롤러에서 함께 사용될 수 있다는 것을 의미합니다.	vlan 또는 vxlan	vlan
vlan-range	탄력적 가상 스위치를 만드는 데 사용되는 심프로 구분된 VLAN ID 범위 목록입니다. 각 탄력적 가상 스위치에 VLAN ID가 하나씩 연결됩니다.	1 - 4094	--
vxlan-range	탄력적 가상 스위치를 만드는 데 사용되는 심프로 구분된 VXLAN 세그먼트 번호 범위 목록입니다. 각 탄력적 가상 스위치에 VXLAN 세그먼트 번호가 하나씩 연결됩니다.	0 - 16777215	--
vxlan-addr	VXLAN 데이터 링크를 만드는 데 사용되는 IP 주소를 지정합니다. vxlan-addr 등록 정보를 서브넷으로 설정할 수도 있습니다.	--	--
vxlan-mgroup	VXLAN 데이터 링크를 만드는 동안 사용해야 하는 멀티캐스트 주소를 지정합니다.	--	멀티캐스트 주소를 지정하지 않은 경우 VXLAN 데이터 링크에서 All Host 주소를 사용합니다.
vxlan-ipvers	VXLAN 데이터 링크를 호스트하는 IP 인터페이스에 대해 사용해야 하는 주소의 IP 버전을 지정합니다.	v4 또는 v6	v4
uplink-port	VLAN 또는 VXLAN에 대해 사용해야 하는 데이터 링크를 지정합니다.	--	--

EVS 컨트롤러에 대해 설정하는 컨트롤러 등록 정보는 전체 데이터 센터에 적용될 수 있습니다. 하지만 호스트 단위로 컨트롤러 등록 정보 uplink-port 및 vxlan-addr의 값을 대체할 수 있습니다.

예를 들어, 컨트롤러 등록 정보를 설정할 때 uplink-port 등록 정보를 데이터 링크 net2로 설정하면 이 데이터 링크가 데이터 센터의 모든 EVS 노드에서 VNIC 또는 VXLAN을 만드는 데



사용됩니다. 하지만, 데이터 센터의 EVS 노드에 유일한 인터페이스로 데이터 링크 net1만 있는 경우 다음과 같이 전역 값 net2를 호스트 단위 값으로 대체해야 합니다.

```
# evsadm set-controlprop -h host1 -p uplink-port=net1
```

자세한 내용은 [EVS 컨트롤러를 구성하는 방법 \[115\]](#)을 참조하십시오.

컨트롤러 등록 정보에 대한 값을 지정하지 않을 경우 [예 6-2. “EVS 컨트롤러에 대한 등록 정보 재설정”](#)에 표시된 대로 해당 등록 정보는 기본값으로 재설정됩니다. EVS 컨트롤러 등록 정보에 대한 자세한 내용은 [evsadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## EVS 클라이언트

dladm 및 zonecfg 명령은 EVS 클라이언트입니다. 탄력적 가상 스위치, IPnet 및 VPort를 사용하여 evsadm 명령을 통해 L2 네트워크 토폴로지를 정의할 수 있습니다. dladm 명령을 사용하여 VNIC를 L2 네트워크 토폴로지에 연결하거나, zonecfg 명령을 사용하여 VNIC anet 리소스를 연결하여 영역을 L2 네트워크 토폴로지에 연결할 수 있습니다.

---

**참고** - evsadm 명령은 L2 네트워크 토폴로지를 정의하는 EVS 관리자입니다.

---

dladm 또는 zonecfg 명령을 사용하여 탄력적 가상 스위치에 대한 VNIC를 생성하는 경우 EVS 컨트롤러에서 VNIC에 대한 구성 정보를 검색합니다.

service/network/evs 패키지를 설치하고 evsadm set-prop 명령에서 controller 등록 정보를 사용하여 EVS 컨트롤러를 지정한 다음 EVS 클라이언트에서 EVS 작업을 수행할 수 있습니다. controller 등록 정보는 ssh://[user@]example-controller.com 형식으로 지정됩니다. 자세한 내용은 [6장. 탄력적 가상 스위치 관리](#)를 참조하십시오.

## EVS 노드

EVS 노드는 VNIC 또는 영역의 VNIC anet 리소스가 탄력적 가상 스위치에 연결되는 호스트입니다. dladm 및 zonecfg와 같은 명령을 사용하여 탄력적 가상 스위치에 연결해야 하는 VNIC를 지정할 수 있습니다. 자세한 내용은 [“탄력적 가상 스위치에 대한 VNIC 만들기” \[121\]](#)를 참조하십시오.

## EVS 관리 명령

다음 관리 명령을 사용하여 탄력적 가상 스위치를 관리합니다.

- evsadm

- `evsstat`
- `dladm`
- `zonecfg`

탄력적 가상 스위치를 구성하는 방법에 대한 자세한 내용은 [탄력적 가상 스위치를 구성하는 방법 \[119\]](#)을 참조하십시오.

## evsadm 명령

`evsadm` 명령을 사용하여 EVS 컨트롤러와 통신하고 탄력적 가상 스위치, IPnet 및 VPort를 관리합니다. 이 절에서는 이 명령과 함께 작업을 수행하는 데 사용할 하위 명령을 설명합니다. 자세한 내용은 [evsadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

### 탄력적 가상 스위치를 관리하기 위한 evsadm 하위 명령

가상 스위치를 관리하기 위한 `evsadm` 하위 명령은 다음과 같습니다.

<code>create-evs</code>	탄력적 가상 스위치를 만듭니다.
<code>delete-evs</code>	탄력적 가상 스위치를 삭제합니다.
<code>show-evs</code>	탄력적 가상 스위치에 대한 정보를 표시합니다.
<code>set-evsprop</code>	탄력적 가상 스위치에 대해 <code>maxbw</code> 및 <code>priority</code> 등록 정보를 설정하도록 합니다. 이러한 등록 정보에 대한 자세한 내용은 <a href="#">“탄력적 가상 스위치에 대한 등록 정보 설정” [125]</a> 을 참조하십시오.
<code>show-evsprop</code>	탄력적 가상 스위치의 등록 정보를 표시합니다.

### IPnet을 관리하기 위한 evsadm 하위 명령

IPnet을 관리하기 위한 `evsadm` 하위 명령은 다음과 같습니다.

<code>add-ipnet</code>	IPnet을 탄력적 가상 스위치에 추가하고 <code>subnet</code> 및 <code>defrouter</code> 등록 정보를 설정하도록 합니다. 이러한 등록 정보에 대한 자세한 내용은 <a href="#">“탄력적 가상 스위치에 IPnet 추가” [118]</a> 를 참조하십시오.
<code>remove-ipnet</code>	IPnet을 제거합니다.

`show-ipnet` IPnet에 대한 정보를 표시합니다.

## VPort를 관리하기 위한 `evsadm` 하위 명령

가상 포트를 관리하기 위한 `evsadm` 하위 명령은 다음과 같습니다.

`add-vport` VPort를 추가합니다.

`remove-vport` VPort를 제거합니다.

`show-vport` VPort에 대한 정보를 표시합니다.

`set-evsprop` VPort에 대해 다음 등록 정보를 설정하도록 합니다.

- `cos`
- `maxbw`
- `priority`

이러한 등록 정보에 대한 자세한 내용은 표 5-1. “VPort 등록 정보”를 참조하십시오.

`show-vportprop` VPort의 등록 정보를 표시합니다.

`reset-vport` VPort를 재설정합니다.

## EVS 클라이언트 등록 정보를 관리하기 위한 `evsadm` 하위 명령

EVS 클라이언트 등록 정보를 관리하기 위한 `evsadm` 하위 명령은 다음과 같습니다.

`set-prop` controller 등록 정보를 설정하도록 합니다.

`show-prop` EVS 클라이언트 등록 정보를 표시합니다.

## EVS 컨트롤러 등록 정보를 관리하기 위한 `evsadm` 하위 명령

EVS 컨트롤러 등록 정보를 관리하기 위한 `evsadm` 하위 명령은 다음과 같습니다.

`set-controlprop` 컨트롤러에 대해 다음 등록 정보를 설정하도록 합니다.

- `l2-type`
- `vlan-range`
- `vxlان-range`

- vxlan-mgroup
- vxlan-addr
- vxlan-ipvers
- uplink-port

이러한 등록 정보에 대한 자세한 내용은 표 5-2. “EVS 컨트롤러 등록 정보”를 참조하십시오.

`show-controlprop` EVS 컨트롤러의 등록 정보를 표시합니다.

## evsstat 명령

`evsstat` 명령은 데이터 센터의 모든 VPort와 지정된 탄력적 가상 스위치의 모든 VPort에 대한 네트워크 트래픽 통계를 표시합니다. 또한 VPort와 연결된 VNIC에 대한 통계를 보고합니다. 자세한 내용은 “탄력적 가상 스위치 모니터링” [134]을 참조하십시오. `evsstat` 명령에 대한 자세한 내용은 `evsstat(1M)` 매뉴얼 페이지를 참조하십시오.

## dladm 명령

다음 `dladm` 명령을 사용하여 탄력적 가상 스위치에 연결된 VNIC를 관리할 수 있습니다.

- `dladm create-vnic` 명령 - VNIC를 만들고 VNIC에 연결해야 하는 탄력적 가상 스위치 이름을 지정할 수 있습니다. 선택적으로 탄력적 가상 스위치의 VPort를 지정할 수 있습니다.
- `dladm show-vnic` 명령 - 특정 VNIC에 대한 탄력적 가상 스위치 정보를 표시할 수 있습니다. 또한 `dladm show-vnic` 명령의 출력에는 TENANT, EVS 및 VPORT 필드가 표시됩니다. 하지만 이러한 필드는 영역 내에 표시되지 않습니다.

자세한 내용은 `dladm(1M)` 매뉴얼 페이지를 참조하십시오.

탄력적 가상 스위치에 대한 VNIC를 구성하는 방법에 대한 자세한 내용은 [탄력적 가상 스위치에 대한 VNIC를 만드는 방법 \[121\]](#)을 참조하십시오.

## zonecfg 명령

항상된 `zonecfg` 명령을 사용하여 탄력적 가상 스위치에 대한 영역의 VNIC `anet` 리소스를 구성합니다. VNIC `anet` 리소스에 대한 다음 등록 정보를 설정할 수 있습니다.

- `tenant` - 테넌트의 이름을 지정합니다. 영역을 구성할 때 값을 지정하지 않을 경우 기본 값 `sys-global`이 지정됩니다.

- vport - VPort의 이름을 지정합니다. 영역을 구성할 때 값을 지정하지 않은 경우 탄력적 가상 스위치에 대한 VPort가 생성되고 VPort는 탄력적 가상 스위치 등록 정보를 상속합니다.
- evs - VNIC anet 리소스를 연결해야 하는 탄력적 가상 스위치의 이름을 지정합니다.

anet 리소스에 대한 자세한 내용은 “[Oracle Solaris 영역 소개](#)”의 “[리소스 유형 등록 정보](#)”의 anet 설명을 참조하십시오.

---

참고 - 영역 구성에는 테넌트 이름, 탄력적 가상 스위치 이름 및 데이터 센터에서 VPort를 고유하게 식별하는 VPort 이름을 포함해야 합니다. 영역 구성에 대한 자세한 내용은 “[Oracle Solaris 영역 만들기 및 사용](#)”을 참조하십시오.

---

탄력적 가상 스위치의 VNIC anet 리소스를 구성하는 방법에 대한 자세한 내용은 “[탄력적 가상 스위치에 대한 VNIC anet 리소스 만들기](#)” [122]를 참조하십시오. zonecfg 명령에 대한 자세한 내용은 [zonecfg\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## 탄력적 가상 스위치에 연결된 VNIC 관리에 대한 제한 사항

다음은 dladm create-vnic 명령 또는 zonecfg 명령을 사용하여 만들고 탄력적 가상 스위치에 연결하는 VNIC에 적용되는 제한 사항입니다.

- dladm rename-link 명령을 사용하여 VNIC의 이름을 바꿀 수 없습니다.
- dladm set-linkprop 또는 dladm reset-linkprop 명령을 사용하여 해당 VNIC의 등록 정보를 변경할 수 없습니다.
- dladm modify-vnic 명령을 사용하여 해당 VNIC를 수정할 수 없습니다.

## 자동으로 생성된 VXLAN 데이터 링크

VXLAN을 사용하여 탄력적 가상 스위치에 대한 계층 2 세그먼트를 구현한 경우 EVS는 탄력적 가상 스위치에 대한 VNIC를 호스트하는 EVS 노드에 VXLAN 데이터 링크를 자동으로 만듭니다. 이러한 데이터 링크는 자동으로 생성된 VXLAN 데이터 링크라고 하며 이름 지정 규칙 `evs-vxlansegment-ID`를 따릅니다. 여기서 evs는 데이터 링크를 만든 엔티티입니다. 예를 들어, `evs-vxlan200` 이름은 200이 VXLAN ID이고 evs가 이 데이터 링크를 만든 엔티티임을 나타냅니다. `dladm show-vxlan` 명령을 사용하여 자동으로 생성된 VXLAN 데이터 링크를 표시할 수 있습니다. 자세한 내용은 “[VXLAN 정보 표시](#)” [62]를 참조하십시오.

자동으로 생성된 VXLAN 데이터 링크에 대해 `dladm` 하위 명령을 사용하여 데이터 링크를 삭제하거나 이름을 바꿀 수 없습니다. 하지만 `dladm set-linkprop` 명령 및 `dladm reset-linkprop` 명령을 사용하여 데이터 링크 등록 정보를 일시적으로 설정할 수 있습니다.

## EVS 사용을 위한 필수 패키지

EVS를 사용하기 전에 다음 패키지를 설치해야 합니다.

- pkg:/service/network/evs

EVS 관리자, EVS 컨트롤러 및 EVS 노드에 pkg:/service/network/evs 핵심 패키지를 설치해야 합니다. 이 패키지는 다음 구성 요소를 포함합니다.

- evsadm

- evsstat

- SMF 서비스(svc:/network/evs:default) - 이 SMF 서비스는 호스트 이름 또는 EVS 컨트롤러의 IP 주소를 보관하는 controller 등록 정보가 있습니다. EVS 클라이언트는 호스트 이름 또는 IP 주소를 사용하여 EVS 컨트롤러와 통신합니다. evsadm set-prop 명령을 사용하여 controller 등록 정보를 관리합니다.

pkg:/service/network/evs 패키지를 설치하면 새 사용자인 evsuser가 생성됩니다. evsuser는 탄력적 가상 스위치 관리 권한 프로파일을 가지는 특정 사용자입니다. 이 프로파일은 EVS 작업을 수행하는 데 필요한 모든 권한 부여 및 권한을 제공합니다.

- pkg:/system/management/rad/module/rad-*evs-controller*

EVS 컨트롤러 역할을 수행하는 시스템에만 이 패키지를 설치해야 합니다. 컨트롤러 하나만 사용하여 데이터 센터 내의 모든 탄력적 가상 스위치를 관리해야 합니다. 이 패키지는 SMF 서비스 svc:/network/evs-controller:default를 포함합니다. 이 SMF 서비스에는 물리적 시스템에서 L2 세그먼트를 구현하는 데 필요한 정보를 캡처하는 등록 정보가 있습니다. evsadm set-controlprop 명령을 사용하여 컨트롤러 등록 정보를 관리합니다.

자세한 내용은 “EVS 컨트롤러를 위한 필수 패키지” [107]를 참조하십시오.

## EVS의 영역 작업 방법

zonecfg 명령에 연결된 등록 정보를 사용하여 VNIC anet 리소스를 탄력적 가상 스위치에 연결할 수 있습니다. Oracle Solaris 영역 및 Oracle Solaris 커널 영역에서는 EVS 기능을 지원합니다.

커널 영역에서는 탄력적 가상 스위치에 대해 생성되는 VNIC를 지원합니다. 커널 영역 내부에서 만드는 VNIC는 zvnet 드라이버에 연결된 출하 시 기본 MAC 주소를 사용하는 경우에만 작동합니다. 탄력적 가상 스위치에 대해 생성되는 VNIC는 탄력적 가상 스위치의 VPort에 연결된 MAC 주소를 상속하므로, macaddr 등록 정보를 zvnet 드라이버의 출하 시 기본 MAC 주소로 설정하여 탄력적 가상 스위치에 대한 VPort를 만들어야 합니다.

다음 명령 구문을 사용하여 출하 시 기본 MAC 주소를 명시적으로 지정합니다.

```
# evsadm add-vport -p macaddr=factory-MAC-addr-zvnet EVS-name/VPort-name
```

커널 영역에서는 이 명령을 사용하여 만드는 VPort에 VNIC를 연결할 수 있습니다. 커널 영역에 대한 자세한 내용은 [“Oracle Solaris 커널 영역 만들기 및 사용”](#)을 참조하십시오.

## EVS 사용을 위한 보안 요구 사항

EVS 작업을 수행하려면 슈퍼 유저 또는 탄력적 가상 스위치 관리 권한 프로파일을 가진 사용자여야 합니다. 사용자를 만들고 해당 사용자에게 탄력적 가상 스위치 관리 권한 프로파일을 지정할 수도 있습니다. 자세한 내용은 [“Oracle Solaris 11.2의 사용자 및 프로세스 보안”](#)을 참조하십시오.

---

**참고** - 다중 테넌트 EVS 설정에서는 각 사용자에 대한 테넌트 단위 사용자 권한 부여가 지원되지 않으므로 개별 테넌트는 자신의 탄력적 가상 스위치 및 해당 리소스를 관리할 수 없습니다. 전체 EVS 도메인에는 모든 테넌트의 리소스를 관리하는 단일 관리자가 있어야 합니다.

---

다음 예에서는 탄력적 가상 스위치 권한 프로파일을 가진 user1을 만드는 방법을 보여줍니다.

```
# useradd -P "Elastic Virtual Switch Administration" user1
```

다음 예에서는 탄력적 가상 스위치 권한 프로파일을 기존 사용자 user1에 추가하는 방법을 보여줍니다.

```
# usermod -P +"Elastic Virtual Switch Administration" user1
```

EVS 컨트롤러를 설정할 때 탄력적 가상 스위치 관리 권한 프로파일을 가진 사용자를 지정해야 합니다. 예를 들어, EVS 컨트롤러를 설정할 때 다음과 같이 user1을 지정해야 합니다.

```
# evsadm set-prop -p controller=ssh://user1@example-controller.com
```

자세한 내용은 [“EVS 컨트롤러 구성” \[110\]](#)을 참조하십시오.

---

**참고** - pkg:/service/network/evs 패키지를 설치하면 생성되는 evsuser를 사용할 수도 있습니다. evsuser 사용자에게는 탄력적 가상 스위치 관리 권한 프로파일이 지정됩니다. 이 프로파일은 EVS 작업을 수행하는 데 필요한 모든 권한 부여 및 권한을 제공합니다.

---





# ◆◆◆ 6 장 6

## 탄력적 가상 스위치 관리

---

이 장에서는 탄력적 가상 스위치와 리소스를 관리하는 작업에 대해 설명합니다. 일반적인 정보는 5장. 탄력적 가상 스위치 정보를 참조하십시오.

이 장의 내용:

- “EVS 관리 작업” [105]
- “탄력적 가상 스위치 구성 계획” [106]
- “EVS 컨트롤러 만들기 및 관리” [107]
- “탄력적 가상 스위치 구성” [117]
- “탄력적 가상 스위치, IPnet 및 VPort 관리” [123]
- “탄력적 가상 스위치 모니터링” [134]
- “탄력적 가상 스위치에 대한 사용 사례 예” [136]

### EVS 관리 작업

이 절에서는 관련 EVS 관리 작업에 대한 다음 정보를 제공합니다.

- EVS 컨트롤러를 구성하는 방법 [115]
- 탄력적 가상 스위치를 구성하는 방법 [119]
- “탄력적 가상 스위치에 대한 VNIC 만들기” [121]
- “탄력적 가상 스위치 정보 표시” [123]
- “탄력적 가상 스위치에 대한 등록 정보 설정” [125]
- “탄력적 가상 스위치에 대한 등록 정보 표시” [125]
- “IPnet 제거” [127]
- “IPnet 표시” [127]
- “VPort에 대한 등록 정보 설정” [129]
- “VPort의 등록 정보 표시” [130]
- “VPort 표시” [131]
- “VPort 제거” [132]

- 탄력적 가상 스위치를 삭제하는 방법 [133]
- “탄력적 가상 스위치 모니터링” [134]

## 탄력적 가상 스위치 구성 계획

탄력적 가상 스위치 구성 계획에는 다음 작업이 포함됩니다.

1. EVS 컨트롤러, EVS 관리자 및 EVS 노드에 필수 패키지 설치. 각 구성 요소에 대해 필수 패키지를 별도로 설치해야 합니다. 자세한 내용은 “EVS 사용을 위한 필수 패키지” [102]를 참조하십시오.
2. EVS 설정에서 다음 구성 요소 사이에 `evsuser`에 대해 미리 공유한 공개 키를 사용하여 SSH 인증을 설정합니다.
  - EVS 관리자 및 EVS 컨트롤러
  - 각 EVS 노드 및 EVS 컨트롤러
  - EVS 컨트롤러 및 각 EVS 노드자세한 내용은 “SSH 인증 설정” [110]을 참조하십시오.
3. `controller` 등록 정보를 설정하여 EVS 컨트롤러를 지정합니다. EVS 노드, EVS 관리자 및 EVS 컨트롤러에서 EVS 컨트롤러의 호스트 이름 또는 IP 주소를 지정해야 합니다. 자세한 내용은 “EVS 컨트롤러 구성” [110]을 참조하십시오.
4. EVS 컨트롤러 구성으로 포함되는 작업은 다음과 같습니다.
  - a. EVS 컨트롤러에 대한 등록 정보 설정
  - b. EVS 컨트롤러에 대해 설정된 등록 정보 확인자세한 내용은 [EVS 컨트롤러를 구성하는 방법 \[115\]](#)을 참조하십시오.
5. EVS 관리자를 사용하여 탄력적 가상 스위치를 구성하며, 여기에 포함되는 작업은 다음과 같습니다.
  - a. 탄력적 가상 스위치 만들기
  - b. 탄력적 가상 스위치에 IPnet 추가
  - c. 탄력적 가상 스위치에 VPort 추가
  - d. 구성된 탄력적 가상 스위치 확인자세한 내용은 [탄력적 가상 스위치를 구성하는 방법 \[119\]](#)을 참조하십시오.
6. EVS 노드에서 VNIC를 만들고 탄력적 가상 스위치에 VNIC 연결. 포함되는 작업은 다음과 같습니다.
  - a. `dladm` 명령을 사용하여 VNIC 만들기 또는 `zonecfg` 명령을 사용하여 VNIC `anet` 리소스를 만든 후 탄력적 가상 스위치에 연결
  - b. 탄력적 가상 스위치에 연결된 VNIC 확인자세한 내용은 “탄력적 가상 스위치에 대한 VNIC 만들기” [121]를 참조하십시오.

## EVS 컨트롤러 만들기 및 관리

EVS 컨트롤러는 탄력적 가상 스위치 및 모든 연결된 리소스를 구성하고 관리하기 위한 기능을 제공합니다. EVS 컨트롤러에 대해 물리적 시스템에서 계층 2 세그먼트를 구현하는 데 필요한 정보를 캡처한 등록 정보를 설정해야 합니다. 자세한 내용은 “EVS 컨트롤러” [95]를 참조하십시오.

EVS 컨트롤러 계획에 대한 고려 사항은 다음과 같습니다.

- VLAN, VXLAN 또는 둘 다 사용하여 탄력적 가상 스위치를 구현할지 결정합니다.
  - VLAN을 사용하여 탄력적 가상 스위치를 구현할 경우 uplink-port 및 vlan-range 등록 정보를 설정해야 합니다.
  - VXLAN을 사용하여 탄력적 가상 스위치를 구현할 경우 vxlan-range 등록 정보와 uplink-port 또는 vxlan-addr 등록 정보를 설정해야 합니다. 선택적으로 vxlan-mgroup 및 vxlan-ipvers 등록 정보를 설정할 수도 있습니다.

---

참고 - 탄력적 가상 스위치를 만든 후에는 해당 탄력적 가상 스위치에 대한 EVS 컨트롤러 등록 정보를 수정할 수 없습니다. EVS 컨트롤러 등록 정보에 대한 모든 수정 사항은 새로 만드는 탄력적 가상 스위치에 반영됩니다.

---

- 컴퓨팅 노드에 동일한 데이터 링크가 없는 경우 모든 컴퓨팅 노드에 대해 uplink-port 등록 정보에 대한 데이터 링크를 지정해야 합니다.  
 예를 들어, 데이터 링크 net2를 가진 host1과 데이터 링크 net3을 가진 host2의 두 컴퓨팅 노드를 생각해 보겠습니다. uplink-port 등록 정보를 다음과 같이 설정할 경우 두 호스트 모두의 데이터 링크를 지정해야 합니다.

```
# evsadm set-controlprop -h host1 -p uplink-port=net2
# evsadm set-controlprop -h host2 -p uplink-port=net3
```

## EVS 컨트롤러를 위한 필수 패키지

컨트롤러 하나만 사용하여 데이터 센터 내의 모든 탄력적 가상 스위치를 관리해야 합니다. EVS 컨트롤러 역할을 하는 시스템에 pkg:/service/network/evs 및 pkg:/system/management/rad/module/rad-evs-controller 패키지를 설치해야 합니다.

다음 명령을 사용하여 패키지를 설치합니다.

```
# pkg install evs
# pkg install rad-evs-controller
```

rad-evs-controller 패키지를 설치한 후 다음 명령으로 rad:local 서비스를 다시 시작하여 EVS 컨트롤러를 로드해야 합니다.

```
# svcadm restart rad:local
```

## EVS 컨트롤러를 구성하기 위한 명령

이 절에서는 EVS 컨트롤러에 대해 다음 작업을 수행하는 방법을 설명합니다.

- EVS 컨트롤러 설정
- EVS 컨트롤러 표시
- EVS 컨트롤러에 대한 등록 정보 설정
- EVS 컨트롤러의 등록 정보 표시

### EVS 컨트롤러 설정

`evsadm set-prop` 명령을 사용하여 호스트에서 EVS 컨트롤러를 설정합니다. 명령 구문은 다음과 같습니다.

```
# evsadm set-prop -p controller=[value[...]]
```

이 명령은 명령이 실행되는 호스트에 대한 등록 정보 값을 설정합니다. `controller` 등록 정보만 지원되며, `ssh://[user@]evs-controller-host-name` 또는 `ssh://[user@]evs-controller-IP-address` 형식을 따릅니다.

### EVS 컨트롤러 표시

`evsadm show-prop` 명령을 사용하여 EVS 컨트롤러를 표시합니다. 명령 구문은 다음과 같습니다.

```
# evsadm show-prop [[-c] -o field[,...]] [-p controller[,...]]
```

<code>-p controller</code>	RAD 클라이언트가 연결해야 하는 EVS 컨트롤러를 지정합니다.
<code>-o field[,...]</code>	표시할 출력 필드의 대소문자를 무시한 심표로 구분된 목록을 지정합니다. 출력에 열로 표시되는 다음 필드를 지정할 수 있습니다.
<code>all</code>	모든 출력 필드 표시
<code>PROPERTY</code>	등록 정보 이름
<code>PERM</code>	등록 정보에 대한 권한( <code>rw</code> 또는 <code>r-</code> )
<code>VALUE</code>	등록 정보의 값
<code>DEFAULT</code>	등록 정보의 기본값

`-c` 안정적인 컴퓨터에서 구문 분석 가능한 형식을 사용하여 표시합니다.  
`-o` 옵션을 `-c` 옵션과 함께 지정해야 합니다.

EVS 컨트롤러를 표시하는 방법을 보여주는 예는 [예 6-1. “EVS 컨트롤러 구성”](#)을 참조하십시오.

## EVS 컨트롤러에 대한 등록 정보 설정

`evsadm set-controlprop` 명령을 사용하여 EVS 컨트롤러에 대한 등록 정보를 설정합니다. 명령 구문은 다음과 같습니다.

```
# evsadm set-controlprop [-h host] -p prop=[value[...]]
```

`-h host` 등록 정보를 설정할 호스트를 지정합니다.

`-p prop` EVS 컨트롤러에 대해 설정되는 컨트롤러 등록 정보의 이름을 지정합니다. 등록 정보에 값이 여러 개 사용되는 경우 쉼표를 분리자로 사용하여 값을 지정해야 합니다. 등록 정보를 한 번에 하나씩 지정해야 합니다. 값을 지정하지 않을 경우 등록 정보는 기본값으로 재설정됩니다. EVS 컨트롤러에 대해 설정할 수 있는 등록 정보에 대한 자세한 내용은 [표 5-2. “EVS 컨트롤러 등록 정보”](#)를 참조하십시오.

## EVS 컨트롤러의 등록 정보 표시

`evsadm show-controlprop` 명령을 사용하여 EVS 컨트롤러의 등록 정보를 표시합니다. 명령 구문은 다음과 같습니다.

```
# evsadm show-controlprop [[-c] -o field[,...]] [-p prop[,...]]
```

이 명령은 EVS 컨트롤러에 대한 하나 이상의 등록 정보의 현재 값을 표시합니다. EVS 컨트롤러에 대한 등록 정보를 지정하지 않은 경우 컨트롤러에 대한 기존 등록 정보가 모두 표시됩니다. 컨트롤러 등록 정보에 대한 자세한 내용은 [표 5-2. “EVS 컨트롤러 등록 정보”](#)를 참조하십시오.

`-o field[,...]` 표시할 출력 필드의 대소문자를 무시한 쉼표로 구분된 목록을 지정합니다. 출력에 열로 표시되는 다음 필드를 지정할 수 있습니다.

<code>all</code>	모든 출력 필드를 표시합니다.
<code>PROPERTY</code>	등록 정보의 이름입니다.
<code>PERM</code>	등록 정보에 대한 권한( <code>rw</code> 또는 <code>r-</code> )입니다.
<code>VALUE</code>	등록 정보의 값입니다.

DEFAULT	등록 정보의 기본값입니다.
HOST	값이 --이면 전역 등록 정보이고 모든 호스트에 적용될 수 있습니다. 그렇지 않으면 등록 정보를 특정 호스트에 적용할 수 있습니다.

EVS 컨트롤러에 대한 등록 정보를 표시하는 방법을 보여주는 예는 [예 6-1. “EVS 컨트롤러 구성”](#)을 참조하십시오.

## EVS 컨트롤러 구성

EVS 노드가 EVS 컨트롤러와 통신할 수 있도록 네트워크에서 컴퓨팅 노드를 하나만 EVS 컨트롤러로 구성한 다음 각 EVS 노드에서 EVS 컨트롤러를 설정해야 합니다. 하지만 EVS 컨트롤러와 통신할 수 있는 노드에서 한 번만 EVS 컨트롤러에 대한 등록 정보를 설정해야 합니다. `evsadm set-controlprop` 명령을 사용하여 EVS 컨트롤러에 대한 등록 정보를 설정합니다. 자세한 내용은 [EVS 컨트롤러를 구성하는 방법 \[115\]](#)을 참조하십시오.

EVS 컨트롤러에 대한 등록 정보를 재설정할 수도 있습니다. [예 6-2. “EVS 컨트롤러에 대한 등록 정보 재설정”](#)은 EVS 컨트롤러에 대한 등록 정보를 재설정하는 방법을 보여줍니다. EVS 컨트롤러 및 등록 정보에 대한 자세한 내용은 [“EVS 컨트롤러” \[95\]](#)를 참조하십시오.

탄력적 가상 스위치의 구성을 간소화하려면 `evsuser`로 연결해야 합니다. 필수 EVS 패키지 (`service/network/evs`)를 설치하면 특별 사용자인 `evsuser`가 만들어지고 탄력적 가상 스위치 관리 권한 프로파일이 지정됩니다. 이 프로파일에는 EVS 작업을 수행할 수 있는 모든 권한 부여 및 권한이 포함됩니다. `evsuser`를 사용하려면 `controller` 등록 정보를 다음과 같이 설정해야 합니다.

```
# evsadm set-prop -p controller=ssh://evsuser@evs-controller-hostname-or-IP-address
```

또한 `evsadm` 명령을 실행하는 호스트와 EVS 컨트롤러 간에 미리 공유한 공개 키를 사용하여 SSH 인증을 설정해야 합니다.

---

**참고** - EVS 작업을 수행하려면 슈퍼 유저이거나 탄력적 가상 스위치 관리 권한 프로파일을 가진 사용자여야 합니다. 자세한 내용은 [“EVS 사용을 위한 보안 요구 사항” \[103\]](#)을 참조하십시오.

---

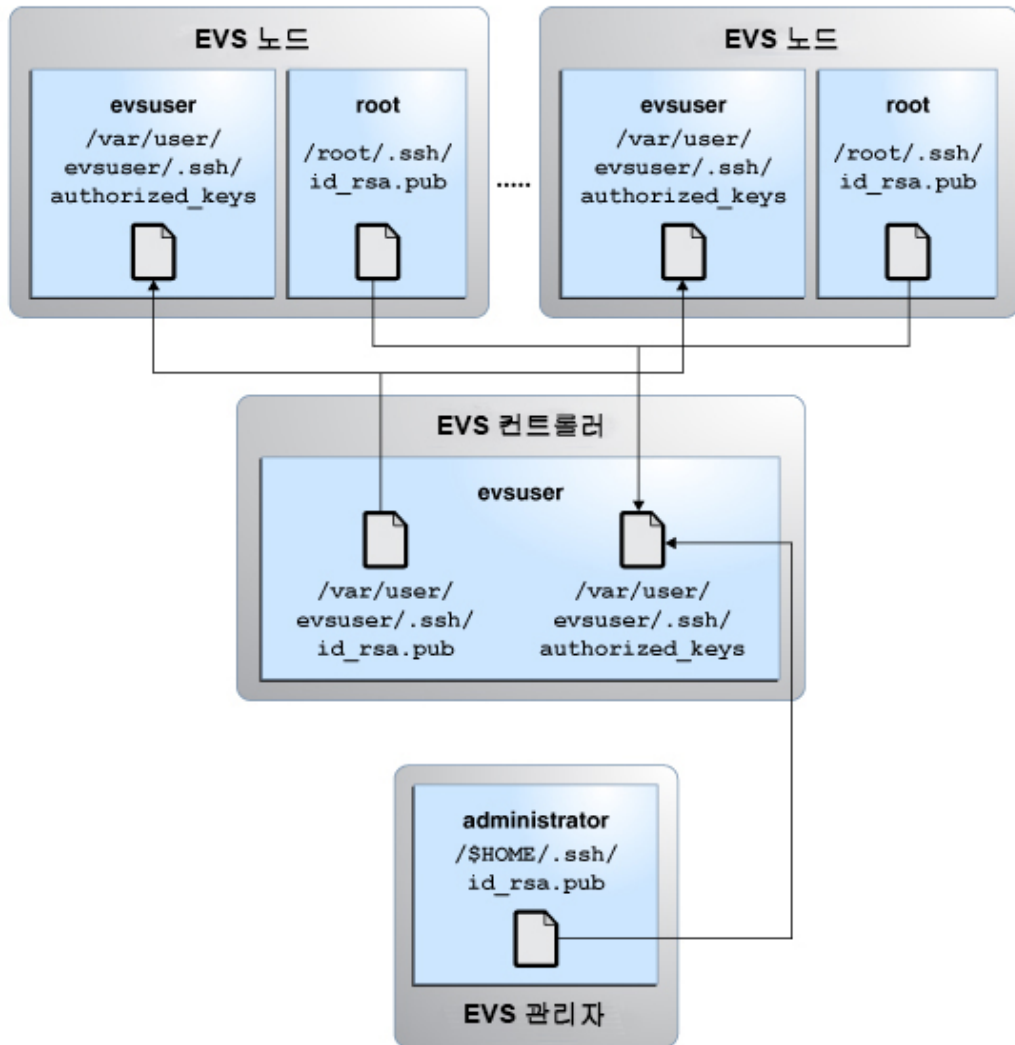
## SSH 인증 설정

EVS 컨트롤러와 비대화식으로 안전하게 통신하려면 `evsadm` 명령에 대해 미리 공유한 공개 키를 사용하는 SSH 인증이 필요합니다. EVS 설정에서 다음 구성 요소 사이에 `evsuser`에 대해 미리 공유한 공개 키를 사용하여 SSH 인증을 설정해야 합니다.

- **EVS 관리자 및 EVS 컨트롤러** - EVS 관리자에서 관리자 또는 `evsadm` 명령을 실행하는 사용자의 공개 키를 EVS 컨트롤러의 `/var/user/evsuser/.ssh/authorized_keys` 파일에 추가합니다.
- **EVS 노드 및 EVS 컨트롤러** - 각 EVS 노드에서 `root` 사용자의 공개 키를 EVS 컨트롤러의 `/var/user/evsuser/.ssh/authorized_keys` 파일에 추가합니다. `zoneadmd` 데몬이 루트로 실행되므로 이러한 공개 키를 추가해야 합니다. 이 데몬은 EVS 컨트롤러에 연결하고 VNIC `anet` 리소스에 대한 구성 정보를 검색합니다. 자세한 내용은 [zoneadmd\(1M\)](#) 매뉴얼 페이지를 참조하십시오.
- **EVS 컨트롤러 및 EVS 노드** - VPort 등록 정보를 설정하기 위해 EVS 컨트롤러가 각 EVS 노드와 통신하므로, EVS 컨트롤러에 대한 `evsuser`의 공개 키를 각 EVS 노드의 `/var/user/evsuser/.ssh/authorized_keys` 파일에 추가합니다.

다음 그림은 EVS 구성 요소 간의 SSH 인증 설정을 보여줍니다.

그림 6-1 EVS 설정에서 SSH 인증



SSH 인증을 설정한 후 EVS 컨트롤러를 지정해야 합니다. EVS 노드, EVS 관리자 및 EVS 컨트롤러에서 controller 등록 정보가 `ssh://evsuser@evs-controller.example.com`으로 설정되었다고 가정합니다.

다음 절차에서는 SSH 인증을 설정하는 방법을 보여줍니다.



## ▼ EVS 노드와 EVS 컨트롤러 간의 SSH 인증을 설정하는 방법

1. 관리자로 로그인합니다.

자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”을 참조하십시오.

2. EVS 노드에서 RSA 키 쌍을 생성합니다.

```
evs-node# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
a0:64:de:3d:c8:26:59:cb:4a:46:b9:1d:17:04:7d:bf root@evs-node
```

3. EVS 노드의 `/root/.ssh/id_rsa.pub` 파일에서 공개 키를 EVS 컨트롤러의 `/var/user/evsuser/.ssh/authorized_keys` 파일에 복사합니다.

4. EVS 노드에서 `evsuser`로 EVS 컨트롤러에 로그인하여 SSH 인증이 설정되었는지 여부를 확인합니다.

```
evs-node# ssh evsuser@evs-controller
The authenticity of host 'evs-controller (192.168.100.10)' can't be established.
RSA key fingerprint is 73:66:81:15:0d:49:46:e0:1d:73:32:77:4f:7c:24:a5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'evs-controller' (RSA) to the list of known hosts.
Last login: Wed Jun 11 14:36:28 2014 from evs-controller
Oracle Corporation      SunOS 5.11      11.2      April 2014
evsuser@evs-controller$
```

EVS 노드에서 암호 없이 `evsuser`로 EVS 컨트롤러에 로그인할 수 있다는 메시지가 출력됩니다.

## ▼ EVS 관리자와 EVS 컨트롤러 간의 SSH 인증을 설정하는 방법

1. 관리자로 로그인합니다.

자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”을 참조하십시오.

2. EVS 관리자에서 RSA 키 쌍을 생성합니다.

```
evs-manager# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

```
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
a0:64:de:3d:c8:26:59:cb:4a:46:b9:1d:17:04:7e:bf root@evs-manager
```

3. EVS 관리자의 `/root/.ssh/id_rsa.pub` 파일에서 공개 키를 EVS 컨트롤러의 `/var/user/evsuser/.ssh/authorized_keys` 파일에 복사합니다.
4. EVS 관리자에서 `evsuser`로 EVS 컨트롤러에 로그인하여 SSH 인증이 설정되었는지 여부를 확인합니다.

```
evs-manager# ssh evsuser@evs-controller
The authenticity of host 'evs-controller (192.168.100.10)' can't be established.
RSA key fingerprint is 73:66:81:15:0d:49:46:e0:1d:73:32:77:4f:7c:24:a5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'evs-controller' (RSA) to the list of known hosts.
Last login: Wed Jun 11 14:38:28 2014 from evs-controller
Oracle Corporation SunOS 5.11 11.2 April 2014
evsuser@evs-controller$
```

EVS 관리자에서 암호 없이 `evsuser`로 EVS 컨트롤러에 로그인할 수 있다는 메시지가 출력됩니다.

### ▼ EVS 컨트롤러와 EVS 노드 간의 SSH 인증을 설정하는 방법

1. 관리자로 로그인합니다.  
자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”을 참조하십시오.

2. EVS 컨트롤러에서 사용자 `evsuser`로 전환합니다.

```
evs-controller# su - evsuser
```

자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”을 참조하십시오.

3. EVS 컨트롤러에서 `evsuser`에 대한 RSA 키 쌍을 생성합니다.

```
evsuser@evs-controller$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/var/user/evsuser/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/user/evsuser/.ssh/id_rsa.
Your public key has been saved in /var/user/evsuser/.ssh/id_rsa.pub.
The key fingerprint is:
a0:64:de:3d:c8:26:59:cb:4a:46:b9:1e:17:04:7d:bf evsuser@evs-controller
```

4. EVS 컨트롤러의 `/var/user/evsuser/.ssh/id_rsa.pub` 파일에서 공개 키를 EVS 노드의 `/var/user/evsuser/.ssh/authorized_keys` 파일에 복사합니다.

5. EVS 컨트롤러에서 `evsuser`로 EVS 노드에 로그인하여 SSH 인증이 설정되었는지 여부를 확인합니다.

```

evsuser@evs-controller$ ssh evsuser@evs-node
The authenticity of host 'evs-node (192.168.100.20)' can't be established.
RSA key fingerprint is 73:66:89:15:0d:49:46:e0:1d:73:32:77:4f:7c:24:a5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'evs-node' (RSA) to the list of known hosts.
Last login: Wed Jun 11 14:40:28 2014 from evs-node
Oracle Corporation      SunOS 5.11      11.2      April 2014
evsuser@evs-node$

```

EVS 컨트롤러에서 암호 없이 `evsuser`로 EVS 노드에 로그인할 수 있다는 메시지가 출력됩니다.



주의 - EVS 설정 중 SSH 인증을 설정하지 않을 경우 `evsadm` 명령이 EVS 컨트롤러와 비대화 식으로 안전하게 통신할 수 없습니다.

## ▼ EVS 컨트롤러를 구성하는 방법

시작하기 전에 `evsadm` 명령을 실행하는 호스트와 EVS 컨트롤러 간에 미리 공유한 키를 사용하여 SSH 인증을 설정합니다.

1. 관리자 또는 탄력적 가상 스위치 관리 프로파일을 가진 사용자로 전환합니다.  
자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”을 참조하십시오.

2. EVS 컨트롤러를 설정합니다.

```
# evsadm set-prop -p controller=[value[...]]
```

이 명령은 명령이 실행되는 호스트에 대한 등록 정보 값을 설정합니다. `controller` 등록 정보만 지원되며, `ssh://[user@]evs-controller-host-name` 또는 `ssh://[user@]evs-controller-IP-address` 형식을 따릅니다.

3. (옵션) 구성된 EVS 컨트롤러를 표시합니다.

```
# evsadm show-prop [[-c] -o field[...]] [-p controller[...]]
```

자세한 내용은 “EVS 컨트롤러 표시” [108]를 참조하십시오.

4. EVS 컨트롤러에 대한 등록 정보를 설정합니다.

```
# evsadm set-controlprop [-h host] -p prop=[value[...]]
```

자세한 내용은 “EVS 컨트롤러에 대한 등록 정보 설정” [109]을 참조하십시오.

5. (옵션) EVS 컨트롤러의 등록 정보를 표시합니다.

```
# evsadm show-controlprop [[-c] -o field[,...]] [-p prop[,...]]
```

자세한 내용은 “EVS 컨트롤러의 등록 정보 표시” [109]를 참조하십시오.

#### 예 6-1 EVS 컨트롤러 구성

다음 예에서는 s11-server 호스트를 EVS 컨트롤러로 구성하는 방법을 보여줍니다. VXLAN을 사용하여 이 컨트롤러의 L2 세그먼트가 생성됩니다.

```
# evsadm set-prop -p controller=ssh://evsuser@s11-server
# evsadm show-prop
PROPERTY      PERM  VALUE                                DEFAULT
controller    rw    ssh://evsuser@s11-server            --
# evsadm set-controlprop -p l2-type=vxlan
# evsadm set-controlprop -p vxlan-range=10000-20000
# evsadm set-controlprop -p vxlan-addr=192.168.10.0/24
# evsadm set-controlprop -h s11-server -p uplink-port=net3
# evsadm set-controlprop -h s11-client -p uplink-port=net4
# evsadm show-controlprop
PROPERTY      PERM  VALUE                                DEFAULT      HOST
l2-type       rw    vxlan                                vxlan        --
uplink-port   rw    net3                                --           s11-server
uplink-port   rw    net4                                --           s11-client
vlan-range    rw    --                                  --           --
vlan-range-avail r-    --                                  --           --
vxlan-addr    rw    192.168.10.0/24                    0.0.0.0     --
vxlan-ipvers  rw    v4                                  v4           --
vxlan-mgroup  rw    0.0.0.0                            0.0.0.0     --
vxlan-range   rw    10000-20000                        --           --
vxlan-range-avail r-    10000-20000                        --           --
```

이 예에서 vxlan-range-avail 등록 정보는 탄력적 가상 스위치를 구현하는 데 사용할 수 있는 VXLAN ID(10000-20000)를 표시합니다. 192.168.10.0/24 서브넷의 일부인 IP 인터페이스는 EVS 노드에서 VXLAN 링크를 만드는 데 사용됩니다.

다음 예에서는 IP 주소가 192.168.100.1인 호스트를 EVS 컨트롤러로 구성하는 방법을 보여줍니다. VLAN을 사용하여 이 컨트롤러의 L2 세그먼트가 생성됩니다.

```
# evsadm set-prop -p controller=ssh://evsuser@192.168.100.1
# evsadm set-controlprop -p l2-type=vlan
# evsadm set-controlprop -p vlan-range=200-300,400-500
# evsadm set-controlprop -p uplink-port=net2
# evsadm set-controlprop -h host2.example.com -p uplink-port=net3
# evsadm set-controlprop -h host3.example.com -p uplink-port=net4
```

탄력적 가상 스위치에 대해 200-300 및 400-500 VLAN ID를 사용할 수 있다는 메시지가 출력됩니다. net2 데이터 링크는 host2.example.com 및 host3.example.com을 제외한 모든 호스트에서 uplink-port입니다. host2에서는 net3 데이터 링크가 uplink-port로 사용되고, host3에서는 net4 데이터 링크가 uplink-port로 사용됩니다.

## 예 6-2 EVS 컨트롤러에 대한 등록 정보 재설정

다음 예에서는 uplink-port 컨트롤러 등록 정보를 재설정하는 방법을 보여줍니다.

```
# evsadm show-controlprop -p uplink-port
PROPERTY          PERM    VALUE    DEFAULT    HOST
uplink-port       rw      net2     --         --
# evsadm set-controlprop -p uplink-port=
# evsadm show-controlprop -p uplink-port
PROPERTY          PERM    VALUE    DEFAULT    HOST
uplink-port       rw      --       --         --
```

## 탄력적 가상 스위치 구성

탄력적 가상 스위치는 하나 이상의 물리적 시스템에 걸쳐 있는 가상 스위치로, 격리된 L2 세그먼트를 나타냅니다. 격리는 VLAN 또는 VXLAN을 통해 구현됩니다. EVS 노드의 VNIC 또는 anet 리소스를 탄력적 가상 스위치에 연결하여 EVS 노드 간에 네트워크 연결을 제공할 수 있습니다. 자세한 내용은 [“Oracle Solaris 탄력적 가상 스위치 기능이란?” \[89\]](#)을 참조하십시오.

탄력적 가상 스위치를 구성하려면 가상 토폴로지를 이해해야 합니다. 필요한 L2 세그먼트 수, 각 네트워크에 대한 IPnet 정보(예: 서브넷, 기본 라우터) 등을 결정합니다. 또한 탄력적 가상 스위치에 대해 구성해야 하는 가상 포트의 수와 가상 포트에 대해 지정해야 하는 등록 정보를 결정할 수 있습니다.

## 탄력적 가상 스위치를 위한 필수 패키지

EVS 클라이언트 및 EVS 노드 역할을 하는 시스템에 pkg:/service/network/evs 패키지를 설치해야 합니다.

다음 명령을 사용하여 패키지를 설치합니다.

```
# pkg install evs
```

## 탄력적 가상 스위치를 구성하기 위한 명령

이 절에서는 탄력적 가상 스위치를 구성하기 위해 다음 작업을 수행하는 방법을 설명합니다.

- 탄력적 가상 스위치 만들기
- 탄력적 가상 스위치에 IPnet 추가

■ 탄력적 가상 스위치에 VPort 추가

## 탄력적 가상 스위치 만들기

evsadm create-evs 명령을 사용하여 탄력적 가상 스위치를 만듭니다. 명령 구문은 다음과 같습니다.

```
# evsadm create-evs [-T tenant-name] [-p {prop=value[,...]}[,...]] EVS-switch-name
```

**-T tenant-name**      테넌트를 지정합니다. 테넌트를 지정하면 탄력적 가상 스위치가 테넌트의 이름 공간 내에 생성됩니다. 그렇지 않으면 탄력적 가상 스위치가 기본 테넌트 `sys-global`에 생성됩니다. 테넌트는 탄력적 가상 스위치가 연결되는 테넌트를 나타내는 읽기 전용 등록 정보입니다.

**-p prop**            탄력적 가상 스위치에서 지정된 값으로 설정할 수 있는 심표로 구분된 등록 정보 목록을 지정합니다. 다음 등록 정보를 설정할 수 있습니다.

- **maxbw** - 탄력적 가상 스위치의 포트에 대한 전이중 대역폭을 설정합니다. 대역폭은 k(Kbps), m(Mbps), g(Gbps) 스케일 접미어를 가진 정수로 지정됩니다. 단위를 지정하지 않은 경우 입력 값은 Mbps로 표시됩니다. 기본 대역폭 제한은 없습니다.
- **priority** - 탄력적 가상 스위치의 포트에 대한 상대 우선 순위를 설정합니다. 가능한 값은 `high`, `medium` 또는 `low`입니다. 기본값은 `medium`입니다. 우선 순위는 전송 중인 프로토콜 우선 순위 필드에는 반영되지 않지만, 시스템 내의 패킷 처리 예약에는 사용됩니다. 시스템 리소스의 가용성에 따라 VPort의 우선 순위가 높을수록 대기 시간이 더 길어집니다.

**EVS-switch-name**    탄력적 가상 스위치의 이름을 지정합니다.

탄력적 가상 스위치를 만드는 방법을 보여주는 예는 [예 6-3. “탄력적 가상 스위치 구성”](#)을 참조하십시오.

## 탄력적 가상 스위치에 IPnet 추가

evsadm add-ipnet 명령을 사용하여 탄력적 가상 스위치에 IPnet을 추가합니다. 명령 구문은 다음과 같습니다.

```
# evsadm add-ipnet [-T tenant-name] -p subnet=value[{,prop=value[,...]}[,...]]\
EVS-switch-name/IPnet-name
```

**-T tenant-name**      테넌트의 이름을 지정합니다. 테넌트 이름을 지정한 경우 IPnet은 테넌트 이름 공간에 있는 EVS와 연결됩니다.

<code>-p prop</code>	<p>특정 탄력적 가상 스위치에 대해 설정해야 하는 IPnet 등록 정보의 심표로 구분된 목록입니다.</p> <p>IPnet에 대해 지원되는 등록 정보는 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>■ <code>subnet</code> - 필수 항목입니다. IPv4 또는 IPv6 주소의 블록을 나타냅니다. IPnet을 추가할 때 <code>subnet</code> 등록 정보를 지정해야 합니다. 그렇지 않으면 IPnet 추가가 실패합니다.</li> <li>■ <code>defrouter</code> - 선택 사항입니다. 지정된 서브넷에 대한 게이트웨이의 IP 주소를 지정합니다. <code>defrouter</code>를 지정하지 않은 경우 범위 내의 첫번째 주소가 기본 라우터 IP 주소로 선택됩니다.</li> </ul>
<code>EVS-switch-name/IPnet-name</code>	연결된 IPnet을 사용하여 탄력적 가상 스위치의 이름을 지정합니다.

IPnet 등록 정보에 대한 자세한 내용은 [evsadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오. IPnet을 탄력적 가상 스위치에 추가하는 방법을 보여주는 예는 [예 6-3. “탄력적 가상 스위치 구성”](#)을 참조하십시오.

## 탄력적 가상 스위치에 VPort 추가

`evsadm add-vport` 명령을 사용하여 탄력적 가상 스위치에 VPort를 추가합니다. 명령 구문은 다음과 같습니다.

```
# evsadm add-vport [-T tenant-name] [-p {prop=value[,...]}[,...]] EVS-switch-name/VPort-name
```

<code>-p prop</code>	<p>VPort에 대해 설정할 수 있는 심표로 구분된 VPort 등록 정보 목록을 지정합니다. 지원되는 VPort 등록 정보에 대한 자세한 내용은 <a href="#">표 5-1. “VPort 등록 정보”</a>을 참조하십시오.</p>
<code>EVS-switch-name/VPort-name</code>	연결된 VPort를 사용하여 탄력적 가상 스위치의 이름을 지정합니다.

VPort를 탄력적 가상 스위치에 추가하는 방법을 보여주는 예는 [예 6-3. “탄력적 가상 스위치 구성”](#)을 참조하십시오.

## ▼ 탄력적 가상 스위치를 구성하는 방법

시작하기 전에 탄력적 가상 스위치를 구성할 컴퓨팅 노드에서 EVS 컨트롤러를 설정해야 합니다. 자세한 내용은 [EVS 컨트롤러를 구성하는 방법 \[115\]](#)의 2단계를 참조하십시오.

1. 관리자 또는 탄력적 가상 스위치 관리 프로파일을 가진 사용자로 전환합니다.

자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”을 참조하십시오.

2. 탄력적 가상 스위치를 만듭니다.

```
# evsadm create-evs [-T tenant-name] [-p {prop=value[,...]}[,...]] EVS-switch-name
```

자세한 내용은 “탄력적 가상 스위치 만들기” [118]를 참조하십시오.

---

참고 - 가상 포트에 대한 등록 정보를 명시적으로 설정한 경우 이 등록 정보 값이 해당 탄력적 가상 스위치 등록 정보 값을 대체합니다.

---

3. 탄력적 가상 스위치에 IPnet을 추가합니다.

```
# evsadm add-ipnet [-T tenant-name] -p subnet=value[,{,prop=value[,...]}[,...]]
\
EVS-switch-name/IPnet-name
```

자세한 내용은 “탄력적 가상 스위치에 IPnet 추가” [118]를 참조하십시오.

4. (옵션) 탄력적 가상 스위치에 VPort를 추가합니다.

```
# evsadm add-vport [-T tenant-name] [-p {prop=value[,...]}[,...]] EVS-switch-
name/VPort-name
```

VPort는 탄력적 가상 스위치에 추가되면 IPnet 주소 범위 내에서 임의의 MAC 주소와 IP 주소가 지정됩니다. 따라서 먼저 IPnet을 탄력적 가상 스위치에 추가한 다음 VPort를 추가해야 합니다. evsadm add-vport 명령에 대한 자세한 내용은 “탄력적 가상 스위치에 VPort 추가” [119]를 참조하십시오.

---

참고 - 탄력적 가상 스위치에 가상 포트를 항상 추가할 필요는 없습니다. VNIC가 생성되는 경우 VNIC를 연결할 탄력적 가상 스위치의 이름만 지정할 수 있습니다. 이 경우 EVS 컨트롤러에서 시스템 가상 포트를 생성합니다. 이러한 가상 포트는 이름 지정 규칙 sys-vportname을 따릅니다(예: sys-vport0). 시스템 가상 포트는 탄력적 가상 스위치 등록 정보를 상속합니다.

---

5. (옵션) 구성된 탄력적 가상 스위치를 표시합니다.

```
# evsadm
```

예 6-3 탄력적 가상 스위치 구성

다음 예에서는 탄력적 가상 스위치 ORA를 만들고, IPnet ora\_ipnet을 추가하고, VPort vport0을 탄력적 가상 스위치에 추가하는 방법을 보여줍니다.

```
# evsadm create-evs ORA
# evsadm add-ipnet -p subnet=192.168.10.0/24 ORA/ora_ipnet
# evsadm add-vport ORA/vport0
# evsadm
```



NAME	TENANT	STATUS	VNIC	IP	HOST
ORA	sys-global	idle	--	ora_ipnet	--
vport0	--	free	--	192.168.10.2/24	--

다음 예에서는 테넌트 tenantA를 사용하여 탄력적 가상 스위치 ORA를 만들고, IPnet ora\_ipnet을 추가하고, VPort vport0을 탄력적 가상 스위치에 추가하는 방법을 보여줍니다.

```
# evsadm create-evs -T tenantA ORA
# evsadm add-ipnet -T tenantA -p subnet=192.168.10.0/24 ORA/ora_ipnet
# evsadm add-vport -T tenantA ORA/vport0
# evsadm
```

NAME	TENANT	STATUS	VNIC	IP	HOST
ORA	tenantA	idle	--	ora_ipnet	--
vport0	--	free	--	192.168.10.2/24	--

## 탄력적 가상 스위치에 대한 VNIC 만들기

이제 dladm 및 zonecfg 명령을 통해 탄력적 가상 스위치에 대한 VNIC를 만들 수 있습니다.

### ▼ 탄력적 가상 스위치에 대한 VNIC를 만드는 방법

시작하기 전에 evsadm set-prop 명령을 사용하여 EVS 노드에서 controller 등록 정보를 설정해야 합니다. 자세한 내용은 [EVS 컨트롤러를 구성하는 방법 \[115\]](#)을 참조하십시오.

1. 관리자 또는 탄력적 가상 스위치 관리 프로파일을 가진 사용자로 전환합니다.  
자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”을 참조하십시오.

2. 탄력적 가상 스위치에 대한 VNIC를 구성합니다.

```
# dladm create-vnic -t -c EVS-switch-name[/VPort-name] [-T tenant-name] VNIC-name
```

-t	VNIC가 일시적임을 지정합니다.
-c EVS-switch-name[/VPort-name]	VNIC를 연결해야 하는 탄력적 가상 스위치의 이름을 지정합니다. VPort의 이름을 지정하면 VNIC가 지정된 VPort에 연결됩니다. VPort 이름을 지정하지 않은 경우 VPort가 자동으로 생성되어 VNIC에 지정됩니다. VNIC가 탄력적 가상 스위치에 연결된 이후에 VNIC는 지정된 탄력적 가상 스위치 또는 VPort에서 등록 정보를 상속합니다.
-T tenant-name	탄력적 가상 스위치를 소유한 테넌트의 이름을 지정합니다. 테넌트를 지정하지 않은 경우 기본 sys-global 테넌트라고 간주합니다.
VNIC-name	VNIC의 이름입니다.

### 3. (옵션) 탄력적 가상 스위치에 연결된 VNIC에 대한 정보를 표시합니다.

```
# dladm show-vnic -c
```

-c 옵션은 탄력적 가상 스위치에 연결된 VNIC에 대한 정보를 표시합니다.

#### 예 6-4 탄력적 가상 스위치에 대한 VNIC 만들기

이 예에서는 임시 VNIC vnic1을 만들고 이 VNIC를 탄력적 가상 스위치 ORA 및 VPort vport0에 연결하는 방법을 보여줍니다.

```
# dladm create-vnic -t -c ORA/vport0 vnic1
```

```
# dladm show-vnic -c
```

LINK	TENANT	EVS	VPORT	OVER	MACADDRESS	VIDS
vnic1	sys-global	ORA	vport0	evs-vxlan10000	2:8:20:b0:6e:63	0

## 탄력적 가상 스위치에 대한 VNIC anet 리소스 만들기

향상된 zonecfg 명령을 사용하여 탄력적 가상 스위치에 대한 영역의 VNIC anet 리소스를 구성할 수 있습니다.

영역을 구성할 때 anet 리소스에 대해 다음 등록 정보를 설정할 수 있습니다.

- tenant - 테넌트의 이름을 지정합니다. 영역을 구성할 때 값을 지정하지 않을 경우 기본 값인 sys-global 테넌트가 지정됩니다.
- vport - VPort의 이름을 지정합니다. 영역을 구성할 때 값을 지정하지 않을 경우 시스템 VPort가 탄력적 가상 스위치에 대해 자동으로 생성되고 VPort는 탄력적 가상 스위치 등록 정보를 상속합니다.
- evs - anet 리소스를 연결해야 하는 탄력적 가상 스위치의 이름을 지정합니다.

데이터 센터의 VPort는 테넌트 이름, 탄력적 가상 스위치 이름 및 VPort 이름으로 고유하게 식별됩니다. 자세한 내용은 [“Oracle Solaris 영역 만들기 및 사용”](#)을 참조하십시오.

#### 예 6-5 탄력적 가상 스위치에 대한 VNIC anet 리소스 만들기

이 예에서는 테넌트 tenantA의 ORA 및 vport0에 연결되는 VNIC anet 리소스 evszone/net1을 포함하는 영역을 만드는 방법을 보여줍니다.

```
# zonecfg -z evszone
```

```
Use 'create' to begin configuring a new zone
```

```
zonecfg:evszone> create
```

```
create: Using system default template 'SYSdefault'
```

```
zonecfg:evszone> set zonpath=/export/zones/evszone
```

```
zonecfg:evszone> set tenant=tenantA
```

```
zonecfg:evszone> add anet
```

```
zonecfg:evszone:net> set evs=ORA
```

```
zonecfg:evszone:net> set vport=vport0
```

```

zonecfg:evszone:net> end
zonecfg:evszone> exit
# zoneadm -z evszone install
# zoneadm -z evszone boot
# zlogin -C evszone
# dladm show-vnic -c
LINK          TENANT  EVS  VPORT  OVER  MACADDRESS      VIDS
evszone/net1  tenantA  ORA  vport0 net2  2:8:20:89:a1:97  200

```

evszone이 부트되면 VNIC anet evszone/net1은 VPort ORA/vport0의 MAC 주소, IP 주소 및 SLA 등록 정보에 연결됩니다. 탄력적 가상 스위치에 대한 영역의 VNIC anet 리소스를 구성하는 방법에 대한 자세한 내용은 “[사용 사례: 탄력적 가상 스위치 구성](#) [137]을 참조하십시오.

## 탄력적 가상 스위치, IPnet 및 VPort 관리

이 절에서는 탄력적 가상 스위치, IPnet 및 VPort를 관리하는 방법을 설명합니다. 탄력적 가상 스위치, IPnet 및 VPort를 구성하는 방법에 대한 자세한 내용은 “[탄력적 가상 스위치 구성](#) [117]을 참조하십시오.

### 탄력적 가상 스위치 관리

이 절에서는 탄력적 가상 스위치에 대해 다음 작업을 수행하는 방법을 설명합니다.

- 탄력적 가상 스위치에 대한 정보 표시
- 탄력적 가상 스위치에 대한 등록 정보 설정
- 탄력적 가상 스위치 등록 정보 표시

### 탄력적 가상 스위치 정보 표시

evsadm show-evs 명령을 사용하여 탄력적 가상 스위치 정보를 표시합니다. 명령 구문은 다음과 같습니다.

```
# evsadm show-evs [-f {fname=value[,...]}[,...]] [-L] [[-c] -o field[,...]] [EVS-switch-name]
```

-f {fname=value[,...]}[,...]  
출력(행 선택)을 필터링하는 데 사용되는 심표로 구분된 이름-값 쌍입니다. 여러 필터를 지정한 경우 출력에는 필터 중 AND 연산의 결과가 표시됩니다. 필터가 다중 값을 갖는 경우 출력에는 필터 값 중 OR 연산의 결과가 표시됩니다. 지원되는 필터는 다음과 같습니다.

- tenant
- evs

	<ul style="list-style-type: none"> <li>■ host</li> <li>■ ipnet</li> <li>■ vport</li> </ul>
-L	탄력적 가상 스위치에 연결된 VLAN ID 또는 VXLAN 세그먼트 ID를 표시합니다.
-o <i>field</i> [,...]	표시할 출력 필드의 대소문자를 무시한 심표로 구분된 목록을 지정합니다. 출력에 열로 표시되는 다음 필드를 지정할 수 있습니다.
all	모든 출력 필드를 표시합니다.
EVS	탄력적 가상 스위치의 이름입니다.
TENANT	탄력적 가상 스위치를 소유한 테넌트의 이름입니다.
STATUS	탄력적 가상 스위치의 상태이며, 유휴 상태인지 사용 중인지를 나타냅니다. VNIC가 연결된 VPort가 하나 이상 있으면 탄력적 가상 스위치가 사용 중인 것입니다.
NVPORTS	탄력적 가상 스위치에 연결된 가상 포트 수입니다.
IPNETS	EVS에 연결된 IP 네트워크 목록입니다. 현재는 탄력적 가상 스위치에 IP 네트워크를 하나만 연결할 수 있습니다.
HOST	탄력적 가상 스위치가 여러 서버에 걸쳐 있는 호스트 목록입니다.

**예 6-6** 탄력적 가상 스위치 정보 표시

다음 예에서는 탄력적 가상 스위치 ORA에 대한 정보를 표시합니다.

```
# evsadm show-evs ORA
EVS      TENANT      STATUS NVPORTS IPNETS      HOST
ORA      sys-global  busy   1      ora_ipnet   s11-client
```

다음 예에서는 탄력적 가상 스위치 ORA에 연결된 VLAN ID를 표시합니다.

```
# evsadm show-evs -L
EVS      TENANT      VID  VNI
ORA      tenantA     200  --
```

출력에는 다음 정보가 표시됩니다.

EVS	탄력적 가상 스위치의 이름
TENANT	탄력적 가상 스위치를 소유한 테넌트의 이름
VID	탄력적 가상 스위치를 구현하는 데 사용되는 VLAN ID
VNI	탄력적 가상 스위치를 구현하는 데 사용되는 VXLAN 세그먼트 ID

## 탄력적 가상 스위치에 대한 등록 정보 설정

evsadm set-evsprop 명령을 사용하여 탄력적 가상 스위치에 대한 등록 정보를 설정합니다. 명령 구문은 다음과 같습니다.

```
# evsadm set-evsprop [-T tenant-name] -p prop=value[,...] EVS-switch-name
```

-p prop	지정된 탄력적 가상 스위치에 대한 등록 정보 값을 설정합니다. EVS는 다음 등록 정보를 지원합니다. <ul style="list-style-type: none"> <li>■ maxbw - 지정된 탄력적 가상 스위치에 연결되는 모든 가상 포트에 대한 전이중 대역폭을 설정합니다. 대역폭은 k(Kbps), M(Mbps), G(Gbps) 스케일 접미어를 가진 정수로 지정됩니다. 단위를 지정하지 않은 경우 입력 값은 Mbps로 표시됩니다. 기본적으로 대역폭 제한이 없습니다.</li> <li>■ priority - 지정된 탄력적 가상 스위치에 연결되는 모든 가상 포트에 대한 기본 우선 순위를 설정합니다. 가능한 값은 high, medium 또는 low입니다. 기본값은 medium입니다. 우선 순위는 전송 중인 프로토콜 우선 순위 필드에는 반영되지 않지만, 시스템 내의 패킷 처리 예약에는 사용됩니다. 시스템 리소스의 가용성에 따라 VPort의 우선 순위가 높을수록 향상된 대기 시간을 제공합니다.</li> </ul>
---------	---

**예 6-7** 탄력적 가상 스위치에 대한 등록 정보 설정

이 예에서는 탄력적 가상 스위치 ORA에 대한 등록 정보를 설정하는 방법을 보여줍니다.

```
# evsadm set-evsprop -p maxbw=200 ORA
# evsadm set-evsprop -p priority=high ORA
```

## 탄력적 가상 스위치에 대한 등록 정보 표시

evsadm show-evsprop 명령을 사용하여 탄력적 가상 스위치의 등록 정보를 표시합니다. 명령 구문은 다음과 같습니다.

```
# evsadm show-evsprop [-f {fname=value[,...]}[,...]] [[-c] -o field[,...]] \
```

`[-p prop[,...]] [EVS-switch-name]`

`-f {fname=value[,...]} [,...]` 출력(행 선택)을 필터링하는 데 사용되는 심표로 구분된 이름-값 쌍입니다. 여러 필터를 지정한 경우 출력에는 필터 중 AND 연산의 결과가 표시됩니다. 필터가 다중 값을 갖는 경우 출력에는 필터 값 중 OR 연산의 결과가 표시됩니다. 지원되는 필터는 다음과 같습니다.

- tenant - 탄력적 가상 스위치 등록 정보를 테넌트 이름별로 필터링
- evs - 탄력적 가상 스위치 등록 정보를 탄력적 가상 스위치 이름별로 필터링
- host - 탄력적 가상 스위치 등록 정보를 호스트 이름별로 필터링

예 6-8. “탄력적 가상 스위치 등록 정보 표시”은 필터 값을 기준으로 출력을 보여줍니다.

`-o field[,...]` 표시할 출력 필드의 대소문자를 무시한 심표로 구분된 목록을 지정합니다. 출력에 열로 표시되는 다음 필드를 지정할 수 있습니다.

<code>all</code>	모든 출력 필드를 표시합니다.
<code>EVS</code>	탄력적 가상 스위치의 이름입니다.
<code>TENANT</code>	탄력적 가상 스위치를 소유한 테넌트의 이름입니다.
<code>PROPERTY</code>	탄력적 가상 스위치 등록 정보의 이름입니다.
<code>PERM</code>	등록 정보에 대한 읽기 또는 쓰기 권한입니다. 표시되는 값은 <code>r-</code> 또는 <code>rw</code> 입니다.
<code>VALUE</code>	현재 등록 정보 값입니다. 값을 설정하지 않은 경우 <code>--</code> 로 표시됩니다. 값을 알 수 없는 경우 <code>?</code> 로 표시됩니다.
<code>DEFAULT</code>	등록 정보의 기본값입니다. 등록 정보에 기본값이 없는 경우 <code>--</code> 가 표시됩니다.
<code>POSSIBLE</code>	등록 정보에 대해 심표로 구분된 가능한 값 목록입니다. 가능한 값을 알 수 없거나 바인드 해제된 경우 <code>--</code> 가 표시됩니다.

**예 6-8** 탄력적 가상 스위치 등록 정보 표시

다음 예에서는 탄력적 가상 스위치 ORA에 대해 구성된 등록 정보를 표시합니다.

```
# evsadm show-evsprop ORA
EVS    TENANT    PROPERTY  PERM VALUE    DEFAULT  POSSIBLE
ORA    sys-global maxbw    rw    200     --      --
```

```
ORA sys-global priority rw high medium low,medium,high
ORA sys-global tenant r- -- -- --
```

다음 예에서는 탄력적 가상 스위치 HR 및 ORA에 대한 출력을 표시합니다. 이 예에서 `evs` 필터는 탄력적 가상 스위치 HR 및 ORA에 대한 출력을 가져오도록 지정되었습니다.

```
# evsadm show-evsprop -f evs=HR,ORA
EVS TENANT PROPERTY PERM VALUE DEFAULT POSSIBLE
HR tenantA maxbw rw 300 -- --
HR tenantA priority rw -- medium low,medium,high
HR tenantA tenant r- -- -- --
ORA sys-global maxbw rw -- -- --
ORA sys-global priority rw -- medium low,medium,high
ORA sys-global tenant r- -- -- --
```

## IPnet 구성 관리

이 절에서는 탄력적 가상 스위치에 대해 IPnet을 추가한 후 IPnet에 대해 다음 작업을 수행하는 방법을 설명합니다.

- 탄력적 가상 스위치에 대해 구성된 IPnet 제거
- IPnet에 대한 정보 표시

### IPnet 제거

`evsadm remove-ipnet` 명령을 사용하여 탄력적 가상 스위치에 대해 구성된 IPnet을 제거합니다. 명령 구문은 다음과 같습니다.

```
# evsadm remove-ipnet [-T tenant-name] EVS-switch-name/IPnet-name
```

이 명령은 지정된 탄력적 가상 스위치에서 지정된 IPnet을 제거합니다. VPort 중 하나라도 사용 중이면 IPnet을 제거할 수 없습니다. 연결된 VNIC가 있으면 VPort가 사용 중인 것입니다.

**예 6-9** 탄력적 가상 스위치에 대해 구성된 IPnet 제거

이 예에서는 탄력적 가상 스위치 ORA에서 IPnet `ora_ipnet`을 제거하는 방법을 보여줍니다.

```
# evsadm remove-ipnet ORA/ora_ipnet
```

### IPnet 표시

`evsadm show-ipnet` 명령을 사용하여 EVS 컨트롤러에서 관리되는 IPnet 또는 지정된 IPnet에 대한 정보를 표시합니다. 명령 구문은 다음과 같습니다.

```
# evsadm show-ipnet [-f {fname=value[,...]}[,...]] [[-c] -o field[,...]] [IPnet-name]
```

**-f {fname=value[,...]}[,...]** 출력(행 선택)을 필터링하는 데 사용되는 심표로 구분된 이름-값 쌍입니다. 여러 필터를 지정한 경우 출력에는 필터 중 AND 연산의 결과가 표시됩니다. 필터가 다중 값을 갖는 경우 출력에는 필터 값 중 OR 연산의 결과가 표시됩니다. 지원되는 필터는 tenant, evs, ipnet 및 host입니다.

**-o field[,...]** 표시할 출력 필드의 대소문자를 무시한 심표로 구분된 목록을 지정합니다. 출력에 열로 표시되는 다음 필드를 지정할 수 있습니다.

all	모든 출력 필드를 표시합니다.
NAME	IPnet의 이름 및 연결된 탄력적 가상 스위치의 이름입니다.
IPNET	IPnet의 이름입니다.
EVS	탄력적 가상 스위치의 이름입니다.
TENANT	탄력적 가상 스위치를 소유한 테넌트의 이름입니다.
SUBNET	이 IPnet에 대한 서브넷(IPv4 또는 IPv6)을 나타냅니다.
START	IP 주소 범위의 시작 주소입니다.
END	IP 주소 범위의 끝 주소입니다.
DEFROUTER	제공된 IPnet에 대한 기본 라우터의 IP 주소입니다.
AVAILRANGE	VPort에 지정될 수 있는 심표로 구분된 사용 가능한 IP 주소 목록입니다.

**예 6-10** 탄력적 가상 스위치에 대한 IPnet 표시

이 예에서는 탄력적 가상 스위치 ora에 대해 구성된 IPnet을 표시합니다.

```
# evsadm show-ipnet
NAME          TENANT      SUBNET      DEFROUTER  AVAILRANGE
ORA/ora_ipnet sys-global 192.168.10.0/24 192.168.10.1 192.168.10.3-192.168.10.254
```



## VPort 구성 관리

이 절에서는 VPort에 대해 다음 작업을 수행하는 방법을 설명합니다.

- VPort에 대한 등록 정보 설정
- VPort에 연결된 등록 정보 표시
- VPort에 대한 정보 표시
- VPort 재설정
- VPort 제거

### VPort에 대한 등록 정보 설정

`evsadm set-vportprop` 명령을 사용하여 VPort에 대한 등록 정보를 설정합니다. 명령 구문은 다음과 같습니다.

```
# evsadm set-vportprop [-T tenant-name] -p prop=value[,...] EVS-switch-name/VPort-name
```

`-T tenant-name`      테넌트의 이름을 지정합니다.

`-p prop=value[,...]`      지정된 VPort에 대한 등록 정보 값을 지정합니다. VPort에 연결된 VNIC가 있는 경우 해당 VPort에 대해 등록 정보를 설정하면 VNIC의 등록 정보가 변경됩니다. VPort 등록 정보에 대한 자세한 내용은 [표 5-1. “VPort 등록 정보”](#)을 참조하십시오.

---

**참고** - 시스템 VPort의 등록 정보를 변경할 수 없습니다. 시스템 VPort에 대한 자세한 내용은 [탄력적 가상 스위치를 구성하는 방법 \[119\]](#)을 참조하십시오.

---

`EVS-switch-name/VPort-name`      등록 정보를 설정할 탄력적 가상 스위치 또는 VPort의 이름을 지정합니다.

---

**참고** - VPort를 만든 이후에는 `ipaddr`, `macaddr`, `evs` 및 `tenant` 등록 정보를 수정할 수 없습니다.

---

**예 6-11**      VPort에 대한 등록 정보 설정

이 예에서는 `HR/vport0`에 대해 최대 대역폭 등록 정보를 1G로 설정하는 방법을 보여줍니다.

```
# evsadm set-vportprop -p maxbw=1G HR/vport0
```

## VPort의 등록 정보 표시

evsadm show-vportprop 명령을 사용하여 VPort의 등록 정보를 표시합니다. 명령 구문은 다음과 같습니다.

```
# evsadm show-vportprop [-f {fname=value[,...]}[,...]] [[-c] -o field[,...]] \
[-p prop[,...]] [[EVS-switch-name]/[VPort-name]]
```

이 명령은 모든 VPort 또는 지정된 VPort에 대한 하나 이상의 등록 정보의 현재 값을 표시합니다. VPort 등록 정보를 지정하지 않은 경우 모든 사용 가능한 VPort 등록 정보가 표시됩니다. VPort 등록 정보에 대한 자세한 내용은 표 5-1. “VPort 등록 정보”을 참조하십시오.

[-f {fname=value[,...]}[,...]]	출력(행 선택)을 필터링하는 데 사용되는 심표로 구분된 이름-값 쌍입니다. 여러 필터를 지정한 경우 출력에는 필터 중 AND 연산의 결과가 표시됩니다. 필터가 다중 값을 갖는 경우 출력에는 필터 값 중 OR 연산의 결과가 표시됩니다. 지원되는 필터는 다음과 같습니다. <ul style="list-style-type: none"> <li>■ tenant - VPort 등록 정보를 테넌트 이름별로 필터링</li> <li>■ EVS - VPort 등록 정보를 탄력적 가상 스위치 이름별로 필터링</li> <li>■ vport - VPort 등록 정보를 VPort 이름별로 필터링</li> <li>■ host - VPort 등록 정보를 호스트 이름별로 필터링</li> </ul>														
-o field[,...]	표시할 출력 필드의 대소문자를 무시한 심표로 구분된 목록을 지정합니다. 출력에 열로 표시되는 다음 필드를 지정할 수 있습니다. <table border="0" style="margin-top: 10px;"> <tr> <td style="padding-right: 20px;">all</td> <td>모든 출력 필드를 표시합니다.</td> </tr> <tr> <td>NAME</td> <td>VPort가 연결된 탄력적 가상 스위치의 이름과 VPort의 이름을 함께 나타내며 <i>EVS-switch-name/VPort-name</i> 형식을 따릅니다.</td> </tr> <tr> <td>TENANT</td> <td>탄력적 가상 스위치를 소유한 테넌트의 이름입니다.</td> </tr> <tr> <td>PROPERTY</td> <td>VPort 등록 정보의 이름입니다.</td> </tr> <tr> <td>PERM</td> <td>등록 정보에 대한 읽기 또는 쓰기 권한입니다. 표시되는 값은 r- 또는 rw입니다.</td> </tr> <tr> <td>VALUE</td> <td>현재 등록 정보 값입니다. 값을 설정하지 않은 경우 --로 표시됩니다. 값을 알 수 없는 경우 ?로 표시됩니다.</td> </tr> <tr> <td>DEFAULT</td> <td>등록 정보의 기본값입니다. 등록 정보에 기본값이 없는 경우 --가 표시됩니다.</td> </tr> </table>	all	모든 출력 필드를 표시합니다.	NAME	VPort가 연결된 탄력적 가상 스위치의 이름과 VPort의 이름을 함께 나타내며 <i>EVS-switch-name/VPort-name</i> 형식을 따릅니다.	TENANT	탄력적 가상 스위치를 소유한 테넌트의 이름입니다.	PROPERTY	VPort 등록 정보의 이름입니다.	PERM	등록 정보에 대한 읽기 또는 쓰기 권한입니다. 표시되는 값은 r- 또는 rw입니다.	VALUE	현재 등록 정보 값입니다. 값을 설정하지 않은 경우 --로 표시됩니다. 값을 알 수 없는 경우 ?로 표시됩니다.	DEFAULT	등록 정보의 기본값입니다. 등록 정보에 기본값이 없는 경우 --가 표시됩니다.
all	모든 출력 필드를 표시합니다.														
NAME	VPort가 연결된 탄력적 가상 스위치의 이름과 VPort의 이름을 함께 나타내며 <i>EVS-switch-name/VPort-name</i> 형식을 따릅니다.														
TENANT	탄력적 가상 스위치를 소유한 테넌트의 이름입니다.														
PROPERTY	VPort 등록 정보의 이름입니다.														
PERM	등록 정보에 대한 읽기 또는 쓰기 권한입니다. 표시되는 값은 r- 또는 rw입니다.														
VALUE	현재 등록 정보 값입니다. 값을 설정하지 않은 경우 --로 표시됩니다. 값을 알 수 없는 경우 ?로 표시됩니다.														
DEFAULT	등록 정보의 기본값입니다. 등록 정보에 기본값이 없는 경우 --가 표시됩니다.														

POSSIBLE

등록 정보에 대해 심표로 구분된 가능한 값 목록입니다. 값이 숫자 범위에 걸쳐 있는 경우 최소값 - 최대값으로 간단히 표시될 수 있습니다. 가능한 값을 알 수 없거나 바인드 해제된 경우 --가 표시됩니다.

#### 예 6-12 VPort 등록 정보 표시

이 예에서는 VPort vport0에 대한 VPort 등록 정보를 표시합니다.

```
# evsadm show-vportprop ORA/vport0
NAME          TENANT      PROPERTY  PERM VALUE          DEFAULT  POSSIBLE
ORA/vport0    sys-global cos        rw  --                0        0-7
ORA/vport0    sys-global maxbw     rw  --                --        --
ORA/vport0    sys-global priority  rw  --                medium   low,medium,high
ORA/vport0    sys-global ipaddr   r-  192.168.10.2/24  --        --
ORA/vport0    sys-global macaddr  r-  2:8:20:b0:6e:63 --        --
ORA/vport0    sys-global evs       r-  ORA              --        --
ORA/vport0    sys-global tenant   r-  sys-global      --        --
```

## VPort 표시

evsadm show-vport 명령을 사용하여 VPort를 표시합니다. 명령 구문은 다음과 같습니다.

```
# evsadm show-vport [-f {fname=value[,...]}[,...]] [[-c] -o field[,...]] \
[[EVS-switch-name/][VPort-name]]
```

**-f {fname=value[,...]}[,...]** 출력(행 선택)을 필터링하는 데 사용되는 심표로 구분된 이름-값 쌍입니다. 여러 필터를 지정한 경우 출력에는 필터 중 AND 연산의 결과가 표시됩니다. 필터가 다중 값을 갖는 경우 출력에는 필터 값 중 OR 연산의 결과가 표시됩니다. 지원되는 필터는 다음과 같습니다.

- tenant - VPort 목록을 테넌트 이름별로 필터링
- evs - VPort 목록을 탄력적 가상 스위치 이름별로 필터링
- vport - VPort 목록을 VPort 이름별로 필터링
- host - VPort 목록을 호스트 이름별로 필터링

**-o field[,...]** 표시할 출력 필드의 대소문자를 무시한 심표로 구분된 목록을 지정합니다. 출력에 열로 표시되는 다음 필드를 지정할 수 있습니다.

**all** 모든 출력 필드를 표시합니다.

**NAME** VPort가 연결된 탄력적 가상 스위치의 이름과 VPort의 이름을 함께 나타내며 *EVS-switch-name/VPort-name* 형식을 따릅니다.

TENANT	탄력적 가상 스위치를 소유한 테넌트의 이름입니다.
STATUS	VPort가 사용 중인지 사용 가능한지를 표시합니다. VPort가 VNIC에 연결되어 있으면 VPort가 사용 중입니다. 그렇지 않으면 VPort가 사용 가능합니다.
VNIC	VPort에 연결된 VNIC의 이름입니다.
HOST	VPort에 연결된 VNIC가 있는 호스트의 이름입니다.

**예 6-13** VPort 정보 표시

이 예에서는 VPort vport0에 대한 정보를 표시합니다.

```
# evsadm show-vport
NAME          TENANT      STATUS VNIC      HOST
ORA/vport0    sys-global  used   vnic1     s11-client
```

## VPort 재설정

VPort와 연결된 VNIC를 삭제할 경우 VPort의 상태는 free가 됩니다. 다음 상황에서는 VPort와 연결된 VNIC를 삭제하더라도 VPort가 used 상태가 될 수 있습니다.

- EVS 노드의 VNIC를 삭제할 경우 EVS 노드가 EVS 컨트롤러에 도달할 수 없습니다.
- EVS 노드를 재부트하기 전에 VPort와 연결된 VNIC가 삭제되지 않았습니다.

VPort의 상태를 free로 재설정하려면 `evsadm reset-vport` 명령을 사용합니다. 명령 구문은 다음과 같습니다.

```
# evsadm reset-vport [-T tenant-name] EVS-switch-name/VPort-name
```

## VPort 제거

VNIC가 VPort에 연결되어 있는 경우 VPort 제거가 실패합니다. 따라서 먼저 `evsadm show-vport` 명령을 사용하여 제거하려는 VPort에 VNIC가 연결되어 있는지 확인해야 합니다. `evsadm remove-vport` 명령을 사용하여 탄력적 가상 스위치에서 VPort를 제거합니다. 명령 구문은 다음과 같습니다.

```
# evsadm remove-vport [-T tenant-name] EVS-switch-name/VPort-name
```

이 명령은 지정된 VPort를 제거합니다. VPort가 제거되면 VPort에 연결된 MAC 주소와 IP 주소가 해제됩니다.

예 6-14 VPort 제거

이 예에서는 탄력적 가상 스위치 ORA에 대해 구성된 VPort vport0을 제거하는 방법을 보여줍니다.

```
# evsadm remove-vport -T tenantA ORA/vport0
```

## 탄력적 가상 스위치 삭제

이 절에서는 탄력적 가상 스위치를 삭제하는 방법을 설명합니다. 탄력적 가상 스위치의 모든 VPort가 사용 가능한 경우에만 탄력적 가상 스위치를 삭제할 수 있습니다. 따라서 VPort가 VNIC와 연결되지 않아야 합니다.

### ▼ 탄력적 가상 스위치를 삭제하는 방법

1. 관리자 또는 탄력적 가상 스위치 관리 프로파일을 가진 사용자로 전환합니다.  
자세한 내용은 [“Oracle Solaris 11.2의 사용자 및 프로세스 보안”](#)의 [“지정된 관리 권한 사용”](#)을 참조하십시오.

2. VPort가 Elastic 가상 스위치에서 사용되는지 여부를 확인합니다.

```
# evsadm show-evs
```

You cannot delete an elastic virtual switch if a VPort is in use. VNIC가 VPort에 연결되어 있으면 VPort가 사용 중입니다. `evsadm show-evs` 명령 출력의 STATUS 필드에는 탄력적 가상 스위치가 사용 중인지 유휴 상태인지 여부가 표시됩니다.

VPort가 사용 중인 경우 다음과 같이 VPort와 연결된 VNIC를 삭제해야 합니다.

```
# dladm delete-vnic VNIC
```

3. 탄력적 가상 스위치를 삭제합니다.

```
# evsadm delete-evs [-T tenant-name] EVS-switch-name
```

이 명령은 지정된 탄력적 가상 스위치와 탄력적 가상 스위치에 연결된 모든 VPort 및 IPnet을 삭제합니다.

예 6-15 탄력적 가상 스위치 삭제

다음 예에서는 탄력적 가상 스위치 ORA를 삭제하는 방법을 보여줍니다.

```
# evsadm show-evs
```

```

EVS          TENANT          STATUS NVPORTS IPNETS      HOST
ORA          sys-global    idle  0      ora_ipnet  --
# evsadm delete-evs ORA
# evsadm show-evs ORA
evsadm: failed to show EVS(s): evs not found

```

다음 예에서는 사용 중인 탄력적 가상 스위치 EVS1을 삭제하는 방법을 보여줍니다.

```

# evsadm show-evs EVS1
EVS          TENANT          STATUS NVPORTS IPNETS      HOST
EVS1        sys-global    busy  1      evs1_ipnet s11-server
# evsadm show-vport EVS1/vport1
NAME          TENANT          STATUS VNIC          HOST
EVS1/vport1  sys-global    used  vnic1         s11-server
# dladm delete-vnic vnic1
# evsadm show-evs EVS1
EVS          TENANT          STATUS NVPORTS IPNETS      HOST
EVS1        sys-global    idle  1      evs1_ipnet  --
# evsadm delete-evs EVS1
# evsadm show-evs EVS1
evsadm: failed to show EVS(s): evs not found

```

## 탄력적 가상 스위치 모니터링

탄력적 가상 스위치의 가상 포트에 대한 네트워크 트래픽 통계를 모니터링하여 다음 정보를 가져올 수 있습니다.

- VM에서 송수신되는 네트워크 트래픽의 양이며, VM의 작업 로드에 대한 정보를 제공합니다.
- 인바운드(idrops) 및 아웃바운드(odrops)로 삭제되는 패킷의 수입니다. 이 값은 잘못된 네트워크에 대한 정보를 제공합니다.
- 모든 VM에서 계산 모드로 전송 및 수신한 네트워크 트래픽의 양이며, 용량 계획을 수행하는 데 도움이 됩니다.

evsstat 명령을 사용하여 탄력적 가상 스위치를 모니터링합니다. evsstat 명령은 탄력적 가상 스위치의 각 VPort에 대한 실행 시간 통계를 보고합니다. 또한 VPort와 연결된 VNIC에 대한 통계를 보고합니다. EVS 및 가상 포트에 대한 자세한 내용은 [evsadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

evsstat 명령은 RAD(원격 관리 데몬) 클라이언트이며, 원격 EVS 컨트롤러와 통신하여 모든 evsstat 하위 명령을 실행합니다. evsstat 명령을 사용하기 전에 evsadm set-prop 명령을 사용하여 EVS 컨트롤러의 분석 가능한 호스트 이름 또는 IP 주소를 지정해야 합니다. 명령 구문은 다음과 같습니다.

```
# evsadm set-prop -p controller=ssh://[username@]hostname-or-IP-address
```

또한 evsstat 명령을 실행하는 호스트와 EVS 컨트롤러 간에 미리 공유한 공개 키를 사용하여 SSH 인증을 설정해야 합니다. EVS 컨트롤러와 비대화식으로 안전하게 통신하려면

evsstat 명령에 대해 미리 공유한 공개 키를 사용하는 SSH 인증이 필요합니다. 자세한 내용은 “SSH 인증 설정” [110]을 참조하십시오.

evsstat의 명령 구문은 다음과 같습니다.

```
# evsstat [-f {fname=value[,...]}[,...]] [[-c] -o field[,...]] [-u R|K|M|G|T|P] \
[EVS-switch-name[/VPort-name]] [interval] [count]
```

<b>EVS-switch-name</b>	통계를 모니터링할 탄력적 가상 스위치의 이름을 지정합니다. 탄력적 가상 스위치의 이름을 지정하지 않은 경우 모든 탄력적 가상 스위치에 대한 통계가 표시됩니다.
<b>VPort-name</b>	통계를 모니터링할 VPort의 이름을 지정합니다. 지정된 VPort에 연결된 VNIC에 대해서만 통계가 표시됩니다. 탄력적 가상 스위치의 이름을 지정한 다음 VPort의 이름을 지정해야 합니다.
<b>-f {fname=val[,...]}[,...]</b>	출력(행 선택)을 필터링하는 데 사용되는 쉼표로 구분된 이름-값 쌍입니다. 여러 필터를 지정한 경우 출력에는 필터 중 AND 연산의 결과가 표시됩니다. 필터가 다중 값을 갖는 경우 출력에는 필터 값 중 OR 연산의 결과가 표시됩니다. 지원되는 필터는 tenant, evs 및 host입니다.
<b>-o field[,...]</b>	표시할 출력 필드의 대소문자를 무시한 쉼표로 구분된 목록을 지정합니다. 출력에 열로 표시되는 다음 필드를 지정할 수 있습니다. <ul style="list-style-type: none"> <li>■ vport</li> <li>■ evs</li> <li>■ tenant</li> <li>■ vnic</li> <li>■ host</li> <li>■ ipkts</li> <li>■ rbytes</li> <li>■ opkts</li> <li>■ idrops</li> <li>■ odrops</li> </ul>
<b>-u R K M G T P</b>	표시되는 통계의 단위를 지정합니다. 지정하지 않은 경우 다양한 단위를 적절하게 사용하여 xy.zu 형식으로 통계를 표시합니다. 여기서 x, y 및 z는 숫자이고 u는 적절한 단위입니다. 지원되는 단위는 다음과 같습니다. <ul style="list-style-type: none"> <li>■ R - 행 수</li> <li>■ K - 킬로비트</li> <li>■ M - 메가비트</li> <li>■ G - 기가비트</li> <li>■ T - 테라비트</li> <li>■ P - 페타비트</li> </ul>

*interval*                    네트워크 통계를 새로 고칠 간격(초)을 지정합니다.

*count*                      통계를 새로 고치는 횟수를 지정합니다. 간격을 지정한 다음 개수를 지정합니다.

**예 6-16**                    탄력적 가상 스위치 모니터링

다음 예에서는 모든 탄력적 가상 스위치에 대한 통계를 표시합니다.

```
# evsstat
VPORT      EVS      TENANT      IPKTS      RBYTES      OPKTS      OBYTES
sys-vport0  ORA      sys-global  101.88K    32.86M      40.16K    4.37M
sys-vport2  ORA      sys-global  4.50M     6.78G       1.38M     90.90M
sys-vport0  HR       sys-global  132.89K    12.25M      236       15.82K
sys-vport1  HR       sys-global  144.47K    13.32M      247       16.29K
```

다음 예에서는 지정된 탄력적 가상 스위치 *evs0*에 대한 통계를 표시합니다.

```
# evsstat ORA
VPORT      EVS      TENANT      IPKTS      RBYTES      OPKTS      OBYTES
sys-vport0  ORA      sys-global  101.88K    32.86M      40.16K    4.37M
sys-vport2  ORA      sys-global  4.50M     6.78G       1.38M     90.90M
```

다음 예에서는 지정된 VPort *evs0/sys-vport2*에 대한 통계를 표시합니다.

```
# evsstat ORA/sys-vport2
VPORT      EVS      TENANT      IPKTS      RBYTES      OPKTS      OBYTES
sys-vport2  ORA      sys-global  4.50M     6.78G       1.38M     90.90M
```

다음 예에서는 간격 값 1초 및 개수 값 3을 사용하여 VPort에 대한 통계를 표시합니다. 통계를 1초 간격으로 세 번 새로 고칩니다.

```
# evsstat ORA/sys-vport2 1 3
VPORT      EVS      TENANT      IPKTS      RBYTES      OPKTS      OBYTES
sys-vport2  ORA      sys-global  4.50M     6.78G       1.38M     90.90M
sys-vport2  ORA      sys-global  4.50M     6.78G       1.38M     90.90M
sys-vport2  ORA      sys-global  4.50M     6.78G       1.38M     90.90M
```

다음 예에서는 지정된 출력 필드에 대한 통계를 표시합니다.

```
# evsstat -o vport,evs,vnic,host,ipkts,opkts
VPORT      EVS      VNIC      HOST      IPKTS      OPKTS
sys-vport0  ORA      vnic0     host1     101.88K    40.16K
sys-vport2  ORA      vnic0     host2     4.50M     1.38M
sys-vport0  HR       vnic1     host1     132.89K    236
sys-vport1  HR       vnic1     host2     144.47K    247
```

## 탄력적 가상 스위치에 대한 사용 사례 예

이 절에서는 탄력적 가상 스위치를 구성하는 방법을 설명하는 사용 사례의 예를 제공합니다.

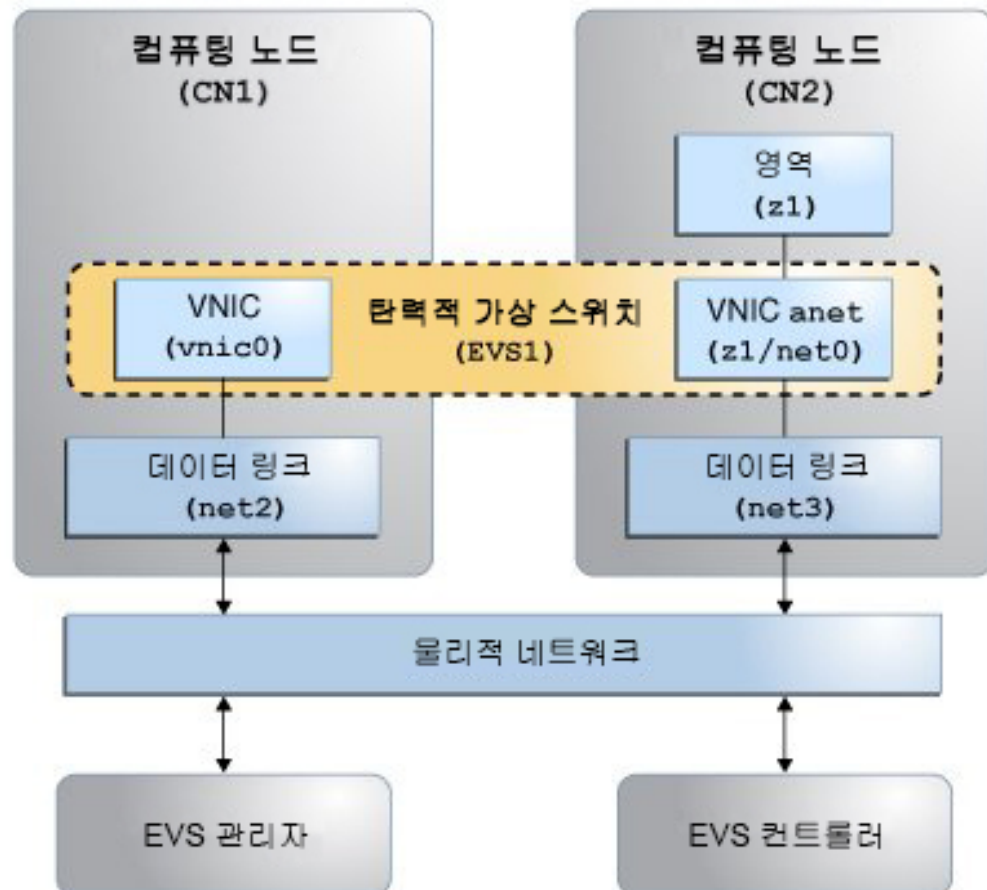


## 사용 사례: 탄력적 가상 스위치 구성

**목표** - 이 사용 사례에서는 두 컴퓨팅 노드를 교차하는 탄력적 가상 스위치(EVS1)를 설정하는 방법을 보여줍니다.

이 사용 사례에서는 CN1의 VNIC vnic0과 영역 z1의 VNIC anet이 동일한 L2 세그먼트에 포함되고 VLAN을 통해 서로 통신할 수 있도록 탄력적 가상 스위치 EVS1에 연결합니다. 다음 그림은 두 컴퓨팅 노드를 교차하는 탄력적 가상 스위치(EVS1)를 보여줍니다.

그림 6-2 탄력적 가상 스위치 구성



이 그림은 다음 구성 요소를 포함하는 네 개의 노드를 가진 네트워크를 보여줍니다.

- 두 컴퓨팅 노드(CN1 및 CN2)
- VNIC anet 리소스(z1/net0)를 포함하는 CN2의 영역 z1
- CN1의 VNIC vnic0
- EVS 컨트롤러(avs-controller.example.com) 역할을 하는 노드
- evsadm 명령을 실행해야 하는 EVS 관리자(MANAGER) 역할을 하는 노드
- 탄력적 가상 스위치 EVS1을 구현하는 VLAN
- VLAN에 사용되는 데이터 링크를 지정하는 uplink-port

---

참고 - 네 개 노드가 모두 동일한 시스템에 있을 수 있습니다. EVS 컨트롤러와 EVS 관리자가 동일한 시스템에 있을 수 있습니다.

---

## 탄력적 가상 스위치 설정 계획

1. 필수 EVS 패키지를 설치합니다.  
필수 패키지에 대한 자세한 내용은 [“EVS 사용을 위한 필수 패키지” \[102\]](#)를 참조하십시오.

---

참고 - evsuser는 pkg:/service/network/avs 패키지를 설치하면 생성되는 특정 사용자입니다. evsuser 사용자에게는 탄력적 가상 스위치 관리 권한 프로파일이 지정됩니다. 이 프로파일은 EVS 작업을 수행하는 데 필요한 모든 권한 부여 및 권한을 제공합니다.

---

2. EVS 설정에서 다음 구성 요소 사이에 evsuser에 대해 미리 공유한 공개 키를 사용하여 SSH 인증을 설정합니다.
  - EVS 관리자 및 EVS 컨트롤러
  - 각 EVS 노드 및 EVS 컨트롤러
  - EVS 컨트롤러 및 각 EVS 노드

자세한 내용은 [“SSH 인증 설정” \[110\]](#)을 참조하십시오.

---

참고 - 이 사용 사례에서는 EVS 노드, EVS 관리자 및 EVS 컨트롤러에서 controller 등록 정보가 ssh://avsuser@avs-controller.example.com으로 설정되었다고 가정합니다.

---

3. EVS 컨트롤러를 구성합니다.
  - a. 계산 노드가 EVS 컨트롤러와 통신할 수 있도록 네트워크에서 계산 노드를 EVS 컨트롤러로 지정한 다음 각 계산 노드에서 EVS 컨트롤러를 설정합니다. EVS 컨트롤러와 통신할 수 있는 계산 노드에서 컨트롤러 등록 정보를 설정할 수 있습니다. 자세한 내용은 [“탄력적 가상 스위치 구성” \[117\]](#)을 참조하십시오.

- b. l2-type, vlan-range, uplink-port 등록 정보를 지정합니다. 그렇지 않으면 탄력적 가상 스위치를 만들 수 없습니다.
4. 탄력적 가상 스위치를 만듭니다. IPnet을 연결하고 VPort를 탄력적 가상 스위치에 추가해야 합니다.
5. CN1에서 임시 VNIC를 만들고 VNIC를 탄력적 가상 스위치의 VPort에 연결합니다.
6. z1 영역에서 VNIC anet 리소스를 만든 후 이 리소스를 탄력적 가상 스위치에 연결합니다.

## EVS 관리자 작업

1. EVS 컨트롤러를 설정합니다.

```
MANAGER# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. EVS 컨트롤러 등록 정보를 설정합니다.

- a. 탄력적 가상 스위치에 사용해야 하는 L2 토폴로지 유형을 설정합니다.

```
MANAGER# evsadm set-controlprop -p l2-type=vlan
```

- b. VLAN 범위를 설정합니다.

```
MANAGER# evsadm set-controlprop -p vlan-range=200-300
```

- c. VLAN에 사용되는 데이터 링크(uplink-port)를 지정합니다.

```
MANAGER# evsadm set-controlprop -p uplink-port=net2
```

```
MANAGER# evsadm set-controlprop -h CN2 -p uplink-port=net3
```

---

참고 - EVS 컨트롤러에 연결할 수 있고 필요한 권한이 있는 경우 데이터 센터의 노드에서 EVS 컨트롤러를 구성할 수 있습니다. 자세한 내용은 [“EVS 사용을 위한 보안 요구 사항” \[103\]](#)을 참조하십시오.

---

3. 컨트롤러 등록 정보를 확인합니다.

```
MANAGER# evsadm show-controlprop -p l2-type,vlan-range,uplink-port
```

NAME	VALUE	DEFAULT	HOST
l2-type	vlan	vlan	--
vlan-range	200-300	--	--
uplink-port	net2	--	--
uplink-port	net3	--	CN2

4. EVS1이라는 탄력적 가상 스위치를 만듭니다.

```
MANAGER# evsadm create-evs EVS1
```

5. IPnet EVS1\_ipnet을 EVS1에 추가합니다.

```
MANAGER# evsadm add-ipnet -p subnet=192.168.100.0/24 EVS1/EVS1_ipnet
```

6. VPort vport0을 EVS1에 추가합니다.

```
MANAGER# evsadm add-vport EVS1/vport0
```

탄력적 가상 스위치에 가상 포트를 항상 추가할 필요는 없습니다. VNIC가 생성되는 경우 VNIC를 연결할 탄력적 가상 스위치의 이름만 지정할 수 있습니다. 이 경우 EVS 컨트롤러에서 시스템 가상 포트를 생성합니다. 이러한 가상 포트는 이름 지정 규칙 `sys-vportname`을 따릅니다(예: `sys-vport0`). 시스템 가상 포트는 탄력적 가상 스위치 등록 정보를 상속합니다.

7. 생성되는 탄력적 가상 스위치를 확인합니다.

```
MANAGER# evsadm
```

NAME	TENANT	STATUS	VNIC	IP	HOST
EVS1	sys-global	--	--	EVS1_ipnet	--
vport0	--	free	--	192.168.100.2/24	--

**참고** - 테넌트 이름을 지정하지 않았기 때문에 기본 테넌트 이름인 `sys-global`이 탄력적 가상 스위치 EVS1에서 사용됩니다. 탄력적 가상 스위치를 만들 때 `-t` 옵션을 사용하여 테넌트 이름을 지정할 수 있습니다. 자세한 내용은 [탄력적 가상 스위치를 구성하는 방법 \[119\]](#)을 참조하십시오.

8. MAC 주소와 EVS1/vport0에 연결된 IP 주소를 확인합니다.

```
MANAGER# evsadm show-vportprop -p macaddr,ipaddr EVS1/vport0
```

NAME	TENANT	PROPERTY	PERM	VALUE	DEFAULT	POSSIBLE
EVS1/vport0	sys-global	ipaddr	r-	192.168.100.2/24	--	--
EVS1/vport0	sys-global	macaddr	r-	2:8:20:3c:78:bd	--	--

vport0에 연결된 VNIC는 IP 주소와 MAC 주소를 상속합니다. vport0에 지정되는 IP 주소는 IPnet EVS1\_ipnet에서 다음으로 사용 가능한 IP 주소이고, MAC 주소는 vport0에 대해 임의로 생성됩니다.

9. 탄력적 가상 스위치 EVS1에 연결된 VLAN ID를 확인합니다.

```
MANAGER# evsadm show-evs -L
```

EVS	TENANT	VID	VNI
EVS1	sys-global	200	--

## 컴퓨팅 노드 CN1 작업

1. EVS 컨트롤러를 지정합니다.

```
CN1# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. 임시 VNIC vnic0을 만들어 EVS1/vport0에 연결합니다.

```
CN1# dladm create-vnic -t -c EVS1/vport0 vnic0
```

3. 생성된 VNIC를 확인합니다.

```
CN1# dladm show-vnic -c
LINK      TENANT      EVS      VPORT      OVER      MACADDRESS      VIDS
vnic0     sys-global  EVS1     vport0     net2      2:8:20:3c:78:bd  200
```

vnic0의 MAC 주소는 VPort의 MAC 주소에 매핑됩니다.

4. vnic0에 대해 허용된 IP 주소를 확인합니다.

```
CN1# dladm show-linkprop -p allowed-ips vnic0
LINK      PROPERTY    VALUE          EFFECTIVE      DEFAULT    POSSIBLE
vnic0     allowed-ips 192.168.100.2 192.168.100.2 --           --
```

allowed-ips 등록 정보는 VPort에 연결된 IP 주소로 설정됩니다. 이 설정을 사용하면 vnic0에서 192.168.100.2 이외의 IP 주소를 만들 수 없습니다.

5. vnic0에 대한 IP 인터페이스를 만들고 192.168.100.2를 IP 주소로 지정합니다.

```
# ipadm create-ip -t vnic0
# ipadm create-addr -t -a 192.168.100.2 vnic0
```

## 컴퓨팅 노드 CN2 작업

1. EVS 컨트롤러를 지정합니다.

```
CN2# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. z1 영역에 대해 VNIC anet 리소스를 구성하고 탄력적 가상 스위치에 연결합니다.

```
CN2# zonecfg -z z1
Use 'create' to begin configuring a new zone
zonecfg:z1> create
create: Using system default template 'SYSdefault'
zonecfg:z1> set zonepath=/export/zones/z1
zonecfg:z1> select anet linkname=net0
zonecfg:z1:anet> set evs=EVS1
zonecfg:z1:anet> end
zonecfg:z1> commit
zonecfg:z1> exit
```

3. z1 영역을 설치하고 부트합니다.

```
CN2# zoneadm -z z1 install
CN2# zoneadm -z z1 boot
```

4. z1 영역에 로그인하고 영역 구성을 완료합니다.

```
CN2# zlogin -C z1
```

영역 구성에 대한 자세한 내용은 [“Oracle Solaris 영역 만들기 및 사용”](#)을 참조하십시오.

5. 생성되는 VNIC anet를 확인합니다.

```
CN2# dladm show-vnic -c
LINK      TENANT    EVS    VPORT      OVER  MACADDRESS    VIDS
z1/net0   sys-global EVS1   sys-vport0 net2   2:8:20:1a:c1:e4  200
```

VNIC anet 리소스를 만들 때 VPort를 지정하지 않았기 때문에 EVS 컨트롤러는 VNIC anet 리소스에 대해 시스템 VPort sys-vport0을 만듭니다.

6. VPort와 관련된 정보를 표시합니다.

```
CN2# evsadm show-vport -o all
NAME          TENANT    STATUS VNIC    HOST MACADDR      IPADDR
EVS1/sys-vport0 sys-global used   z1/net0 CN2  2:8:20:1a:c1:e4  192.168.100.3/24
```

VNIC anet 리소스가 연결되고 VPort의 IP 주소가 지정됩니다.

7. VNIC anet 리소스 z1/net0의 IP 주소를 확인합니다.

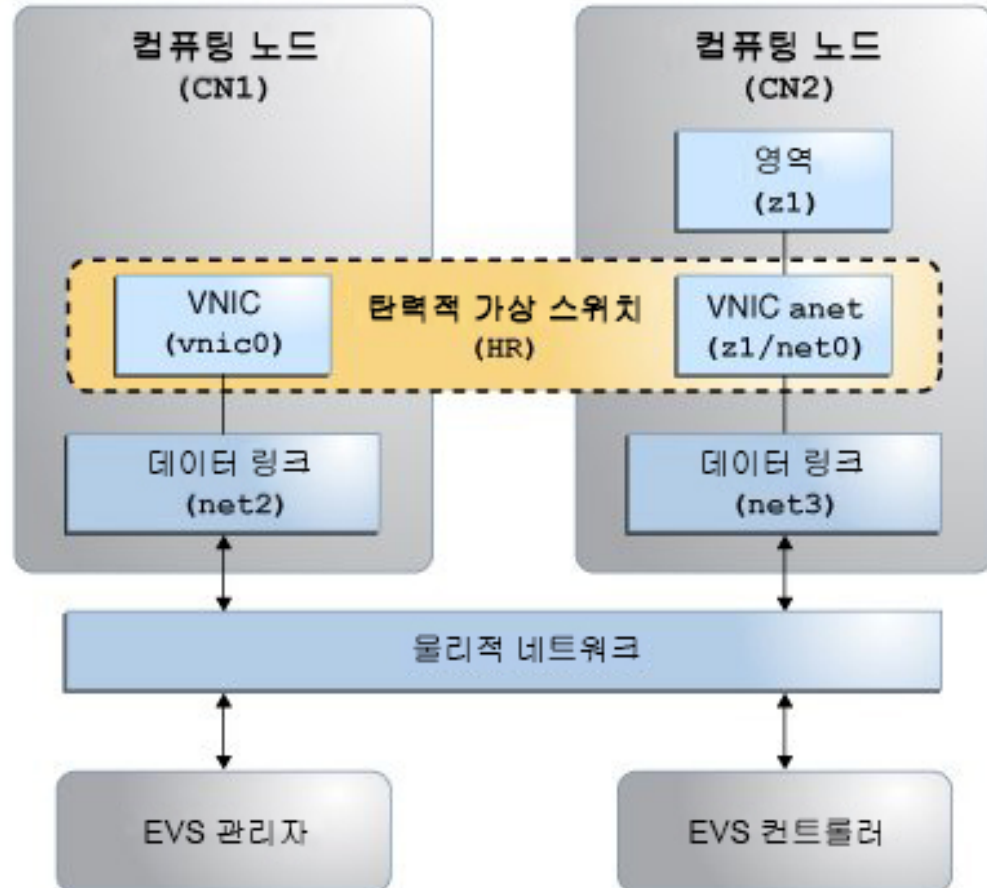
```
CN2# zlogin z1 ipadm
NAME          CLASS/TYPE  STATE    UNDER    ADDR
lo0           loopback   ok       --        --
  lo0/v4      static     ok       --        127.0.0.1/8
  lo0/v6      static     ok       --        ::1/128
net0          ip         ok       --        --
  net0/v4    inherited  ok       --        192.168.100.3/24
```

## 사용 사례: 테넌트에 대한 탄력적 가상 스위치 구성

**목표** - 이 사용 사례에서는 테넌트에 대한 두 컴퓨팅 노드를 교차하는 탄력적 가상 스위치 (HR)를 설정하는 방법을 보여줍니다.

이 사용 사례에서는 CN1의 VNIC vnic0과 영역 z1의 VNIC anet이 동일한 L2 세그먼트에 포함되고 VXLAN을 통해 서로 통신할 수 있도록 탄력적 가상 스위치 HR에 연결합니다. VNIC는 tenantA 테넌트의 일부입니다. 다음 그림은 EVS 설정을 보여줍니다.

그림 6-3 테넌트에 대한 탄력적 가상 스위치 구성



이 그림은 다음 구성 요소를 포함하는 네 개의 노드를 가진 네트워크를 보여줍니다.

- 두 컴퓨팅 노드(CN1 및 CN2)
- VNIC anet 리소스를 포함하는 CN2의 영역 z1
- CN1의 VNIC vnic0
- EVS 컨트롤러 CONTROLLER 역할을 하는 노드
- evsadm 명령을 실행해야 하는 EVS 관리자(MANAGER) 역할을 하는 노드
- 탄력적 가상 스위치 HR을 구현하는 VXLAN
- VXLAN에 사용되는 데이터 링크를 지정하는 uplink-port

## 탄력적 가상 스위치 설정 계획

1. 필수 EVS 패키지를 설치합니다. 필수 패키지에 대한 자세한 내용은 “[EVS 사용을 위한 필수 패키지](#)” [102]를 참조하십시오.

---

**참고** - `evsuser`는 `pkg:/service/network/evs` 패키지를 설치하면 생성되는 특정 사용자입니다. `evsuser` 사용자에게는 탄력적 가상 스위치 관리 권한 프로파일이 지정됩니다. 이 프로파일은 EVS 작업을 수행하는 데 필요한 모든 권한 부여 및 권한을 제공합니다.

---

2. EVS 설정에서 다음 구성 요소 사이에 `evsuser`에 대해 미리 공유한 공개 키를 사용하여 SSH 인증을 설정합니다.

- EVS 관리자 및 EVS 컨트롤러
- 각 EVS 노드 및 EVS 컨트롤러
- EVS 컨트롤러 및 각 EVS 노드

자세한 내용은 “[SSH 인증 설정](#)” [110]을 참조하십시오.

---

**참고** - 이 사용 사례에서는 각 EVS 노드, EVS 관리자 및 EVS 컨트롤러에서 `controller` 등록 정보가 `ssh://evsuser@evs-controller.example.com`으로 설정되었다고 가정합니다.

---

3. EVS 컨트롤러를 구성하고 컨트롤러 등록 정보를 설정합니다.
  - a. 모든 컴퓨팅 노드에서 EVS 컨트롤러를 설정한 다음 컴퓨팅 노드를 교차하여 탄력적 가상 스위치를 구현하는 방법을 지정하는 컨트롤러 등록 정보를 설정합니다.
  - b. `l2-type`, `vxlان-range`, `uplink-port` 등록 정보를 지정합니다. 그렇지 않으면 탄력적 가상 스위치를 만들 수 없습니다.
4. 탄력적 가상 스위치를 만듭니다. IPnet을 연결하고 VPort를 탄력적 가상 스위치에 추가해야 합니다.
5. CN1에서 임시 VNIC를 만들고 VNIC를 탄력적 가상 스위치의 VPort에 연결합니다.
6. z1 영역에서 VNIC anet을 만들고 VNIC anet 리소스를 탄력적 가상 스위치에 연결합니다.

## EVS 관리자 작업

1. EVS 컨트롤러를 설정합니다.

```
MANAGER# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. EVS 컨트롤러 등록 정보를 설정합니다.

- a. 탄력적 가상 스위치에 사용해야 하는 L2 토폴로지 유형을 설정합니다. 이 예에서는 VXLAN을 사용합니다.

```
MANAGER# evsadm set-controlprop -p l2-type=vxlan
```



- b. VXLAN 범위를 설정합니다.

```
MANAGER# evsadm set-controlprop -p vxlan-range=200-300
```

- c. VXLAN에 사용되는 데이터 링크(uplink-port)를 지정합니다.

```
MANAGER# evsadm set-controlprop -p uplink-port=net2
```

```
MANAGER# evsadm set-controlprop -h CN2 -p uplink-port=net3
```

---

참고 - EVS 컨트롤러에 연결할 수 있고 필요한 권한이 있는 경우 데이터 센터의 노드에서 컨트롤러를 구성할 수 있습니다. 자세한 내용은 “EVS 사용을 위한 보안 요구 사항” [103]을 참조하십시오.

---

3. EVS 컨트롤러 등록 정보를 확인합니다.

```
MANAGER# evsadm show-controlprop -p l2-type,vxlan-range,uplink-port
```

NAME	VALUE	DEFAULT	HOST
l2-type	vxlan	vlan	--
vxlan-range	200-300	--	--
uplink-port	net2	--	--
uplink-port	net3	--	CN2

4. 테넌트 tenantA에 대한 탄력적 가상 스위치 HR을 만듭니다.

```
MANAGER# evsadm create-eva -T tenantA HR
```

5. IPnet hr\_ipnet을 탄력적 가상 스위치 HR에 추가합니다.

```
MANAGER# evsadm add-ipnet -T tenantA -p subnet=192.168.100.0/24 HR/hr_ipnet
```

6. VPort vport0을 탄력적 가상 스위치 HR에 추가합니다.

```
MANAGER# evsadm add-vport -T tenantA HR/vport0
```

7. tenantA 테넌트에 대해 생성된 탄력적 가상 스위치를 확인합니다.

```
MANAGER# evsadm
```

NAME	TENANT	STATUS	VNIC	IP	HOST
HR	tenantA	--	--	hr_ipnet	--
vport0	--	free	--	192.168.100.2/24	--

8. MAC 주소와 HR/vport0에 연결된 IP 주소를 확인합니다.

```
MANAGER# evsadm show-vportprop -p macaddr,ipaddr HR/vport0
```

NAME	TENANT	PROPERTY	PERM	VALUE	DEFAULT	POSSIBLE
HR/vport0	tenantA	ipaddr	r-	192.168.100.2/24	--	--
HR/vport0	tenantA	macaddr	r-	2:8:20:d8:da:10	--	--

9. 탄력적 가상 스위치 HR에 연결된 VXLAN 세그먼트 ID를 확인합니다.

```
MANAGER# evsadm show-eva -L
```

EVS	TENANT	VID	VNI
-----	--------	-----	-----

```
HR          tenantA      -- 200
```

## 컴퓨팅 노드 CN1 작업

1. EVS 컨트롤러를 지정합니다.

```
CN1# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. 임시 VNIC vnic0을 만들어 탄력적 가상 스위치 HR/vport0에 연결합니다.

```
CN1# dladm create-vnic -t -T tenantA -c HR/vport0 vnic0
```

3. 생성된 VNIC를 확인합니다.

```
CN1# dladm show-vnic -c
LINK  TENANT  EVS  VPORT  OVER  MACADDRESS  VIDS
vnic0 tenantA HR    vport0 evs-vxlan200 2:8:20:d8:da:10 0
```

vnic0의 MAC 주소는 VPort의 MAC 주소에 매핑됩니다.

4. vnic0에 대해 허용된 IP 주소를 확인합니다.

```
CN1# dladm show-linkprop -p allowed-ips vnic0
LINK  PROPERTY  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
vnic0  allowed-ips 192.168.100.2 192.168.100.2 -- --
```

allowed-ips 등록 정보는 VPort에 연결된 IP 주소로 설정됩니다. 이 출력은 vnic0에 대해 192.168.100.2 이외의 IP 주소를 만들 수 없음을 의미합니다.

5. vnic0에 대한 IP 인터페이스를 만들고 192.168.100.2를 IP 주소로 지정합니다.

```
# ipadm create-ip -t vnic0
# ipadm create-addr -t -a 192.168.100.2 vnic0
```

6. 자동으로 생성된 VXLAN 데이터 링크를 확인합니다.

```
CN1# dladm show-vxlan
LINK  ADDR  VNI  MGROUP
evs-vxlan200 0.0.0.0 200 224.0.0.1
```

## 컴퓨팅 노드 CN2 작업

1. EVS 컨트롤러를 지정합니다.

```
CN2# evsadm set-prop -p controller=ssh://evsuser@evs-controller.example.com
```

2. z1 영역에 대해 VNIC anet을 구성하여 탄력적 가상 스위치에 연결합니다.

```
CN2# zonecfg -z z1
Use 'create' to begin configuring a new zone
zonecfg:z1> create
```

```

create: Using system default template 'SYSdefault'
zonecfg:z1> set zonepath=/export/zones/z1
zonecfg:z1> set tenant=tenantA
zonecfg:z1> select anet linkname=net0
zonecfg:z1:anet> set evs=HR
zonecfg:z1:anet> end
zonecfg:z1> commit
zonecfg:z1> exit

```

3. z1 영역을 설치하고 부트합니다.

```

CN2# zoneadm -z z1 install
CN2# zoneadm -z z1 boot

```

4. z1 영역에 로그인하고 영역 구성을 완료합니다.

```

CN2# zlogin -C z1

```

영역 구성에 대한 자세한 내용은 [“Oracle Solaris 영역 만들기 및 사용”](#)을 참조하십시오.

5. 생성된 VNIC anet 리소스를 확인합니다.

```

CN2# dladm show-vnic -c
LINK      TENANT  EVS  VPORT      OVER      MACADDRESS      VIDS
z1/net0   tenantA HR    sys-vport0 evs-vxlan200 2:8:20:1a:c1:e4  0

```

VPort를 지정하지 않기 때문에 EVS 컨트롤러는 VNIC anet 리소스에 대해 시스템 VPort sys-vport0을 만듭니다.

6. VPort와 관련된 정보를 표시합니다.

```

CN2# evsadm show-vport -o all
NAME          TENANT  STATUS VNIC      HOST MACADDR      IPADDR
HR/sys-vport0 tenantA used   z1/net0  CN2  2:8:20:1a:c1:e4  192.168.100.3/24

```

VNIC anet 리소스가 연결되고 VPort의 IP 주소가 지정됩니다.

7. VNIC anet z1/net0의 IP 주소를 확인합니다.

```

CN2# zlogin z1 ipadm
NAME          CLASS/TYPE STATE      UNDER  ADDR
lo0           loopback  ok         --      --
  lo0/v4      static    ok         --      127.0.0.1/8
  lo0/v6      static    ok         --      ::1/128
net0          ip        ok         --      --
  net0/v4    inherited ok         --      192.168.100.3/24

```



## 네트워크 리소스 관리

---

이 장에서는 데이터 링크 등록 정보와 플로우를 사용하여 네트워크 리소스를 관리 및 할당하는 방법을 설명합니다. 네트워크 리소스를 관리하여 가상 네트워크와 물리적 네트워크의 성능을 향상시키는 IP QoS(서비스 품질)를 구현할 수 있습니다. 네트워크 리소스 관리에 대한 소개는 “네트워크 리소스 관리의 개요” [19]를 참조하십시오.

이 장의 내용:

- “데이터 링크 등록 정보를 사용하여 네트워크 리소스 관리” [149]
- “NIC 링 관리” [150]
- “풀 및 CPU 관리” [159]
- “플로우를 사용하여 네트워크 리소스 관리” [164]
- “사용 사례: 데이터 링크 및 플로우 등록 정보를 설정하여 네트워크 리소스 관리” [167]

### 데이터 링크 등록 정보를 사용하여 네트워크 리소스 관리

데이터 링크에 네트워크 리소스를 할당하여 시스템의 패킷 처리 효율성을 높일 수 있습니다. 데이터 링크를 만들 때 데이터 링크 등록 정보를 설정하여 네트워크 리소스를 할당할 수 있습니다. 또는 데이터 링크 등록 정보를 기존 데이터 링크로 설정할 수 있습니다. `dladm` 명령을 사용하여 데이터 링크에 네트워크 리소스를 할당하려면 다음 데이터 링크 등록 정보를 설정할 수 있습니다.

- `maxbw` - 데이터 링크에 할당할 수 있는 최대 대역폭 양을 지정합니다. 자세한 내용은 “사용 사례: 데이터 링크 및 플로우 등록 정보를 설정하여 네트워크 리소스 관리” [167]를 참조하십시오.
- `rxrings` 및 `txrings` - 특정 데이터 링크에 지정할 수 있는 NIC의 수신 링 및 전송 링의 수를 지정합니다. 자세한 내용은 “NIC 링 관리” [150]를 참조하십시오.
- `pool` - 네트워크 프로세스를 효율적으로 관리하기 위해 데이터 링크에 지정할 수 있는 CPU 세트를 포함하는 CPU 풀의 이름을 지정합니다. 자세한 내용은 “풀 및 CPU 관리” [159]를 참조하십시오.
- `cpus` - 데이터 링크에 지정할 수 있는 CPU의 이름을 지정합니다. 자세한 내용은 “풀 및 CPU 관리” [159]를 참조하십시오.

Oracle Solaris에서 네트워크 리소스 관리에 대한 데모는 [Managing Network Resources Using Oracle Solaris \(http://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/ManagingNetworkResources/ManagingNetworkResources.htm\)](http://www.oracle.com/webfolder/technetwork/tutorials/tutorial/solaris/11/ManagingNetworkResources/ManagingNetworkResources.htm)를 참조하십시오.

## 데이터 링크에서 리소스를 할당하는 명령

다음은 데이터 링크에서 네트워크 리소스를 할당하는 데 사용되는 명령입니다.

- 가상 링크를 만들고 동시에 리소스를 할당하려면 다음 명령 구문을 사용합니다.

```
# dladm create-vnic -l link -p prop=value[,...] VNIC
```

*link* 링크(물리적 링크 또는 가상 링크)의 이름을 나타냅니다.

*prop* 데이터 링크 등록 정보를 나타냅니다. 리소스 할당을 위해 설정할 수 있는 다양한 데이터 링크 등록 정보에 대한 자세한 내용은 “[데이터 링크 등록 정보를 사용하여 네트워크 리소스 관리](#)” [149]를 참조하십시오.

- 기존 링크에 대한 등록 정보를 설정하려면 다음 명령 구문을 사용합니다.

```
# dladm set-linkprop -p prop=value[,...] link
```

자세한 내용은 [dladm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## NIC 링 관리

NIC에서 수신(Rx) 링과 전송(Tx) 링은 각각 시스템이 네트워크 패킷을 받고 보내는 하드웨어 리소스입니다. 네트워크 트래픽에 따라 링을 관리 및 할당하여 시스템의 패킷 처리 효율성을 향상시킵니다. 예를 들어 더 많은 패킷을 수신하는 링에 더 많은 수신(Rx) 링을 할당할 수 있습니다.

## MAC 클라이언트에서 링 할당

NIC를 통해 MAC 클라이언트(예: 물리적 데이터 링크 및 VNIC)를 구성하여 시스템과 다른 네트워크 노드 간에 통신할 수 있습니다. MAC 클라이언트는 하드웨어 기반 클라이언트이거나 소프트웨어 기반 클라이언트일 수 있습니다.

## 하드웨어 기반 클라이언트

하나 이상의 NIC 링을 배타적으로 사용하도록 하는 클라이언트를 하드웨어 기반 클라이언트라고 합니다. NIC에서 지원되는 링 할당에 따라 하드웨어 기반 클라이언트에서 배타적으로 사용하도록 링을 할당할 수 있습니다.

## 소프트웨어 기반 클라이언트

NIC 링을 배타적으로 사용하지 않는 클라이언트를 소프트웨어 기반 클라이언트라고 합니다. 소프트웨어 기반 클라이언트는 링을 다른 기존의 소프트웨어 기반 클라이언트 또는 기본 클라이언트와 공유합니다. 소프트웨어 기반 클라이언트가 사용하는 링은 링 할당에서 우선순위를 갖는 하드웨어 기반 클라이언트 수에 따라 달라집니다.

## VLAN에서 링 할당

VLAN에서의 링 할당은 VLAN을 만드는 방법에 따라 다릅니다.

다음과 같은 방법으로 VLAN을 만들 수 있습니다.

### ■ dladm create-vlan 명령 사용:

```
# dladm create-vlan -l link -v vid VLAN
```

dladm create-vlan 명령을 사용하여 VLAN을 만들 경우 이 VLAN은 기본 데이터 링크와 동일한 MAC 주소를 공유합니다. 따라서 이 VLAN은 기본 데이터 링크의 Rx 및 Tx 링도 공유합니다. VLAN 구성에 대한 자세한 내용은 [“Oracle Solaris 11.2의 네트워크 데이터 링크 관리”](#)의 [“VLAN 구성”](#)을 참조하십시오.

### ■ dladm create-vnic 명령 사용:

```
# dladm create-vnic -l link -v vid VNIC
```

dladm create-vnic 명령을 사용하여 VLAN을 VNIC로 만들 경우 이 VLAN에는 기본 인터페이스와 다른 MAC 주소가 있습니다. 이 유형의 VLAN에 대한 링 할당은 기본 데이터 링크에 대한 할당과 독립적입니다. 따라서 NIC가 하드웨어 기반 클라이언트를 지원하는 경우 이 VLAN에 고유한 전용 링을 지정할 수 있습니다. 클라이언트에 링을 지정하는 방법에 대한 자세한 내용은 [“클라이언트 구성 및 링 할당” \[154\]](#)을 참조하십시오.

## 링을 구성하는 명령

데이터 링크의 링을 구성하려면 다음 dladm 하위 명령을 사용합니다.

### ■ # dladm show-linkprop link

Rx 및 Tx 링을 포함하여 데이터 링크 등록 정보의 현재 값을 표시합니다. 예는 예 7-1. “데이터 링크에 대한 링 사용 및 링 지정”을 참조하십시오.

다음 표에서는 `dladm show-linkprop` 명령을 사용하면 표시되는 링 등록 정보에 대해 설명합니다.

링 등록 정보	사용 권한	설명
<code>rxringsavail</code>	읽기 전용	물리적 데이터 링크에서 하드웨어 기반 클라이언트에 할당할 수 있는 Rx 링 수를 나타냅니다.
<code>rxhwcntavail</code>	읽기 전용	물리적 데이터 링크에서 만들 수 있는 하드웨어 기반 Rx 클라이언트 수를 나타냅니다.
<code>rxrings</code>	읽기 및 쓰기	<p>데이터 링크에서 배타적으로 사용되는 Rx 링 수를 나타냅니다. 이 등록 정보를 세 가지 가능한 값 중 하나로 설정할 수 있습니다.</p> <ul style="list-style-type: none"> <li>■ <code>hw</code>는 하드웨어 기반 클라이언트를 구성하고 있음을 나타냅니다. 기본 물리적 링크에서 하드웨어 기반 Rx 클라이언트 수(<code>rxhwcntavail</code>)가 0보다 큰 경우에 이 값을 설정할 수 있습니다.</li> <li>■ <code>number</code>는 데이터 링크에 지정할 수 있는 링 수를 나타냅니다. 기본 물리적 링크에서 Rx 링 수(<code>rxringsavail</code>)가 0보다 큰 경우에 이 값을 설정할 수 있습니다.</li> <li>■ <code>sw</code>는 데이터 링크가 소프트웨어 기반 클라이언트임을 나타냅니다.</li> </ul>
<code>txringsavail</code>	읽기 전용	물리적 데이터 링크에서 하드웨어 기반 클라이언트에 할당할 수 있는 Tx 링 수를 나타냅니다.
<code>txhwcntavail</code>	읽기 전용	물리적 데이터 링크에서 만들 수 있는 하드웨어 기반 Tx 클라이언트 수를 나타냅니다.
<code>txrings</code>	읽기 및 쓰기	<p>데이터 링크에서 배타적으로 사용되는 Tx 링 수를 나타냅니다. 이 등록 정보를 세 가지 가능한 값 중 하나로 설정할 수 있습니다.</p> <ul style="list-style-type: none"> <li>■ <code>hw</code>는 하드웨어 기반 클라이언트를 구성하고 있음을 나타냅니다. 기본 물리적 링크에서 하드웨어 기반 Tx 클라이언트 수(<code>txhwcntavail</code>)가 0보다 큰 경우에 이 값을 설정할 수 있습니다.</li> <li>■ <code>number</code>는 데이터 링크에 지정할 수 있는 링 수를 나타냅니다. 기본 물리적 링크에서 Tx 링 수(<code>txringsavail</code>)가 0보다 큰 경우에 이 값을 설정할 수 있습니다.</li> </ul>



링 등록 정보	사용 권한	설명
		■ sw는 데이터 링크가 소프트웨어 기반 클라이언트임을 나타냅니다.

■ # dladm show-phys -H *link*

기존 클라이언트에서 물리적 데이터 링크의 링이 현재 어떻게 사용되고 있는지를 표시합니다.

■ # dladm create-vnic -p *ring-properties* -l *link* *VNIC*

-p *ring-properties* 값을 설정할 수 있는 링 등록 정보를 나타냅니다.  
특정 수의 Rx 또는 Tx 링을 사용하여 클라이언트를 만듭니다.

■ # dladm set-linkprop -p *ring-properties* *VNIC*

링을 사용할 수 있고 링 할당이 지원되는 경우 특정 클라이언트에 링을 할당합니다.

## 데이터 링크에 대한 링 사용 및 링 지정 표시

데이터 링크 Rx 링 및 Tx 링의 가능한 값, 구성된 값 및 유효 값을 표시하려면 다음 명령 구문을 사용합니다.

```
# dladm show-linkprop -p rxrings,txrings link
```

클라이언트에서 물리적 데이터 링크의 링이 현재 어떻게 사용되고 있는지를 표시하려면 다음 명령 구문을 사용합니다.

```
# dladm show-phys -H link
```

예 7-1 데이터 링크에 대한 링 사용 및 링 지정

다음 예에서는 데이터 링크 net4에 대한 링 지정을 보여줍니다.

```
# dladm show-linkprop net4
LINK      PROPERTY      PERM VALUE      EFFECTIVE      DEFAULT      POSSIBLE
...
net4      rxrings       rw  1             --            --           sw,hw,<1-7>
net4      txrings       rw  1             --            --           sw,hw,<1-11>
net4      txringsavail  r- 10            10            --           --
net4      rxringsavail  r-  7             7             --           --
net4      rxhwc1ntavail r-  3             3             --           --
net4      txhwc1ntavail r-  3             3             --           --
...
```

데이터 링크 net4에서 Rx 링 하나와 Tx 링 하나를 배타적으로 사용한다는 출력이 표시됩니다. 데이터 링크 net4에는 클라이언트에 할당할 수 있는 7개의 Rx 링과 10개의 Tx 링이 있습니다.

니다. 데이터 링크 net4를 통해 세 개의 하드웨어 기반 Rx 클라이언트와 세 개의 하드웨어 기반 Tx 클라이언트를 만들 수 있습니다.

다음 예에서는 데이터 링크 net0에 대한 링 사용을 보여줍니다.

```
# dladm show-phys -H net0
LINK  RINGTYPE  RINGS  CLIENTS
net0  RX        0-1    <default,mcast>
net0  TX        0-7    <default>net0
net0  RX        2-3    net0
net0  RX        4-5    --
net0  RX        6-7    --
```

출력을 기준으로 net0에 할당되는 두 개의 Rx 링은 링 2와 3입니다. Tx 링의 경우 net0은 링 0~7을 사용합니다.

## 클라이언트 구성 및 링 할당

이 절에서는 링 할당에 대한 지원 유형을 기반으로 데이터 링크에 클라이언트를 구성하는 방법을 설명합니다.

### ▼ 클라이언트 구성 및 링 할당 방법

“링을 구성하는 명령” [151]에 설명한 대로 데이터 링크 링 등록 정보를 표시하는 dladm 명령의 출력을 해석할 수 있는지 확인합니다. 이 정보는 클라이언트를 구성하고 링을 할당하는 데 도움이 됩니다.

1. 관리자로 로그인합니다.  
자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”을 참조하십시오.

2. 기본 물리적 데이터 링크의 등록 정보를 표시합니다.

```
# dladm show-linkprop -p rxringsavail,txringsavail,rxhwcntavail,txhwcntavail link
```

명령의 출력에서 다음 정보를 확인합니다.

- NIC가 하드웨어 기반 클라이언트를 지원는지 여부
- 하드웨어 기반 클라이언트에 할당할 링의 가용성
- 링크에 구성할 수 있는 하드웨어 기반 클라이언트의 가용성

3. 이전 단계의 정보에 따라 다음 중 하나를 수행합니다.

- 다음 구문을 사용하여 하드웨어 기반 클라이언트를 만듭니다.

```
# dladm create-vmnic -p rxrings=value[,txrings=value] -l link VMNIC
```

여기서 *value*는 다음 중 하나일 수 있습니다.

- *hw* - 하드웨어 기반 클라이언트를 구성하고 있음을 나타냅니다.
- *number* - 하드웨어 기반 클라이언트만 구성하고 있음을 나타냅니다. 이 숫자는 배타적 사용을 위해 클라이언트에 할당할 수 있는 링의 수량을 나타냅니다.

- 다음 구문을 사용하여 소프트웨어 기반 클라이언트를 만듭니다.

```
# dladm create-vnic -p rxrings=sw[,txrings=sw] -l link VNIC
```

클라이언트를 이미 만든 경우 `dladm set-linkprop` 명령을 사용하여 링 등록 정보를 설정할 수 있습니다.

4. (옵션) 생성된 클라이언트의 링 정보를 확인합니다.

```
# dladm show-linkprop -p rxrings,txrings VNIC
```

5. (옵션) 여러 클라이언트 간에 배포되는 링크의 링을 확인합니다.

```
# dladm show-phys -H link
```

#### 예 7-2 nxge 장치에서 클라이언트 구성 및 링 할당

이 예는 `nxge` 장치를 기반으로 하며 데이터 링크 `net5`에 대해 클라이언트를 구성하고 링을 할당하는 방법을 보여줍니다. 이 예에서는 다음 클라이언트를 만드는 방법을 보여줍니다.

- VNIC `vnic2` - Rx 및 Tx 링을 배타적으로 사용하는 하드웨어 기반 클라이언트
- VNIC `vnic3` - NIC 드라이버의 초기 구성에 따라 설정되는 고정된 수의 링을 포함하는 하드웨어 기반 클라이언트
- VNIC `vnic4` - 소프트웨어 기반 클라이언트

1. 물리적 데이터 링크 `net5`가 클라이언트에 대한 링 할당을 지원하는지 여부를 확인합니다.

```
# dladm show-linkprop -p rxringsavail,txringsavail net5
```

LINK	PROPERTY	PERM	VALUE	EFFECTIVE	DEFAULT	POSSIBLE
net5	rxringsavail	r-	7	7	--	--
net5	txringsavail	r-	11	11	--	--

물리적 데이터 링크 `net5`를 통해 클라이언트에 지정할 수 있는 7개의 Rx 링과 11개의 Tx 링이 물리적 데이터 링크 `net5`에 있다는 메시지가 출력됩니다.

2. 물리적 데이터 링크 `net5`를 통해 만들 수 있는 하드웨어 기반 클라이언트의 가용성을 확인합니다.

```
# dladm show-linkprop -p rxhwcIntavail,txhwcIntavail net5
```

LINK	PROPERTY	PERM	VALUE	EFFECTIVE	DEFAULT	POSSIBLE
net5	rxhwcIntavail	r-	3	3	--	--
net5	txhwcIntavail	r-	4	4	--	--

데이터 링크 net5를 통해 3개의 하드웨어 기반 Rx 클라이언트와 4개의 하드웨어 기반 Tx 클라이언트를 만들 수 있다는 메시지가 표시됩니다.

- 물리적 데이터 링크 net5를 통해 기존 링 사용을 확인합니다.

```
# dladm show-phys -H net5
LINK      RINGTYPE  RINGS    CLIENTS
nxge1    RX         0-7      <default,mcast>
nxge1    TX         0-11     <default>
```

nxge1 장치에 8개의 Rx 링(0-7)과 12개의 Tx 링(0-11)이 있다는 메시지가 출력됩니다. nxge1 장치에 데이터 링크가 없으므로 Rx 링과 Tx 링은 데이터 링크에 지정되지 않습니다. CLIENTS 열의 <default> 값은 소프트웨어 기반 클라이언트에서 Tx 링을 사용한다는 것을 의미합니다. CLIENTS 열 아래의 <default,mcast> 값은 소프트웨어 기반 클라이언트와 비유니캐스트 패킷에서 Rx 링을 사용한다는 것을 의미합니다.

- Rx 링과 Tx 링이 각각 두 개인 데이터 링크 net5를 통해 VNIC vnic2를 만듭니다.

```
# dladm create-vnic -l net5 -p rxrings=2,txrings=2 vnic2
```

- VNIC vnic2에 지정되는 링을 확인합니다.

```
# dladm show-linkprop -p rxrings,txrings vnic2
LINK      PROPERTY  PERM  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
vnic2     rxrings   rw    2      2          --       sw,hw,<1-7>
vnic2     txrings   rw    2      2          --       sw,hw,<1-11>
```

- 물리적 데이터 링크 net5에서 링 사용을 확인합니다.

```
# dladm show-phys -H net5
LINK      RINGTYPE  RINGS    CLIENTS
nxge1    RX         0,3-7    <default,mcast>
nxge1    TX         0,3-11   <default>
nxge1    RX         1-2      vnic2
nxge1    TX         1-2      vnic2
```

vnic2에 할당된 Rx 링은 1 및 2라는 메시지가 출력됩니다. Tx 링의 경우 vnic2는 1 및 2 링을 사용합니다.

- 물리적 데이터 링크 net5를 통해 추가 하드웨어 기반 클라이언트를 만들 수 있는지 여부를 확인합니다.

```
# dladm show-linkprop -p rxhwclntavail,txhwclntavail net5
LINK      PROPERTY  PERM  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
net5      rxhwclntavail  r-    2      2          --       --
net5      txhwclntavail  r-    3      3          --       --
```

물리적 데이터 링크 net5를 통해 두 개의 하드웨어 기반 Rx 클라이언트와 세 개의 하드웨어 기반 Tx 클라이언트를 만들 수 있다는 메시지가 출력됩니다.

- 하드웨어 기반 클라이언트인 VNIC vnic3을 만듭니다.

```
# dladm create-vnic -l net5 -p rxrings=hw,txrings=hw vnic3
```

9. VNIC vnic3에 지정되는 링을 확인합니다.

```
# dladm show-linkprop -p rxrings,txrings vnic3
```

LINK	PROPERTY	PERM	VALUE	EFFECTIVE	DEFAULT	POSSIBLE
vnic3	rxrings	rw	--	1	--	sw, hw, <1-7>
vnic3	txrings	rw	hw	hw	--	sw, hw, <-11>

참고 - 클라이언트에 지정되는 링 수는 네트워크 장치에 따라 다릅니다. 링 수를 명시적으로 지정할 수 있는 장치(예: nxge 장치)에서는 클라이언트 하나에 링 하나가 지정됩니다. 다른 장치의 경우 클라이언트에 지정되는 링 수는 장치의 구성에 따라 다릅니다. 예 7-3. “ixgbe 장치에서 클라이언트 구성 및 링 할당”을 참조하십시오.

10. 물리적 데이터 링크 net5를 통해 추가 하드웨어 기반 클라이언트를 만들 수 있는지 여부를 확인합니다.

```
# dladm show-linkprop -p rxhwclntavail,txhwclntavail net5
```

LINK	PROPERTY	PERM	VALUE	EFFECTIVE	DEFAULT	POSSIBLE
net5	rxhwclntavail	r-	2	2	--	--
net5	txhwclntavail	r-	2	2	--	--

물리적 데이터 링크 net5를 통해 2개의 하드웨어 기반 Rx 클라이언트와 2개의 하드웨어 기반 Tx 클라이언트를 만들 수 있다는 메시지가 출력됩니다.

11. 소프트웨어 기반 클라이언트인 VNIC vnic4를 만듭니다.

```
# dladm create-vnic -l net5 -p rxrings=sw,txrings=sw vnic4
```

12. vnic4에서 링 사용을 확인합니다.

```
# dladm show-linkprop -p rxrings,txrings vnic4
```

LINK	PROPERTY	PERM	VALUE	EFFECTIVE	DEFAULT	POSSIBLE
vnic4	rxrings	rw	sw	--	--	sw, hw, <1-7>
vnic4	txrings	rw	sw	--	--	sw, hw, <1-11>

13. 물리적 데이터 링크 net5에서 링 사용을 확인합니다.

```
# dladm show-phys -H net5
```

LINK	RINGTYPE	RINGS	CLIENTS
nxge1	RX	0, 4-7	<default,mcast>,vnic4
nxge1	TX	0, 4-11	<default>,vnic4
nxge1	RX	1-2	vnic2
nxge1	RX	3	vnic3
nxge1	TX	1-2	vnic2
nxge1	TX	3	vnic3

vnic4가 물리적 데이터 링크 net5에서 기본 링 세트를 공유하는 소프트웨어 기반 클라이언트라는 메시지가 출력됩니다. VNIC vnic2는 두 개의 링(2-3)을 배타적으로 사용하는 하드웨어 기반 클라이언트이고 vnic3는 한 개의 링(3)을 배타적으로 사용하는 하드웨어 기반 클라이언트입니다.

## 예 7-3 ixgbe 장치에서 클라이언트 구성 및 링 할당

이 예는 ixgbe 장치를 기반으로 하며 물리적 데이터 링크 net4에 대해 클라이언트를 구성하고 링을 할당하는 방법을 보여줍니다.

1. 물리적 데이터 링크 net4를 통해 기존 링 사용을 확인합니다.

```
# dladm show-phys -H net4
LINK      RINGTYPE  RINGS    CLIENTS
net4      RX        0-3      <default,mcast>
net4      RX        4-7      --
net4      RX        8-11     --
net4      RX        12-15    --
net4      TX        0-7      <default>
```

2. 물리적 데이터 링크 net4를 통해 하드웨어 기반 클라이언트를 만들 수 있는지 여부를 확인합니다.

```
# dladm show-linkprop -p rxhwclntavail,txhwclntavail,rxringsavail,txringsavail net4
LINK      PROPERTY  PERM  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
net4      rxhwclntavail  r-    3      3          --      --
net4      txhwclntavail  r-    0      0          --      --
net4      rxringsavail   r-    0      0          --      --
net4      txringsavail   r-    0      0          --      --
```

물리적 데이터 링크 net4를 통해 3개의 하드웨어 기반 Rx 클라이언트를 만들 수 있다는 메시지가 출력됩니다.

3. 하드웨어 기반 Rx 클라이언트인 VNIC vnic3를 만듭니다.

```
# dladm create-vnic -l net4 -p rxrings=hw vnic3
```

사용 가능한 하드웨어 기반 Tx 클라이언트 수(txhwclntavail)는 0이므로 vnic3에 대한 txrings 등록 정보를 구성할 수 없습니다.

4. VNIC vnic3에 지정되는 링을 확인합니다.

```
# dladm show-linkprop -p rxrings,txrings vnic3
LINK      PROPERTY  PERM  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
vnic3     rxrings   rw    hw     hw          --      sw,hw
vnic3     txrings   rw    --     8          --      --
```

5. 물리적 데이터 링크 net4를 통해 추가 하드웨어 기반 클라이언트를 만들 수 있는지 여부를 확인합니다.

```
# dladm show-linkprop -p rxhwclntavail,txhwclntavail,rxringsavail,txringsavail net5
LINK      PROPERTY  PERM  VALUE  EFFECTIVE  DEFAULT  POSSIBLE
net4      rxhwclntavail  r-    2      2          --      --
net4      txhwclntavail  r-    0      0          --      --
net4      rxringsavail   r-    0      0          --      --
net4      txringsavail   r-    0      0          --      --
```

물리적 데이터 링크 net4를 통해 2개의 하드웨어 기반 Rx 클라이언트를 만들 수 있다는 메시지가 출력됩니다.

6. 물리적 데이터 링크 net4에서 링 사용을 확인합니다.

```
# dladm show-phys -H net4
LINK      RINGTYPE  RINGS    CLIENTS
net4      RX        0-3      <default,mcast>
net4      RX        4-7      vnic3
net4      RX        8-11
net4      RX        12-15    --
net4      TX        0-7      <default>,vnic3
```

vnic3은 네 개의 링을 배타적으로 사용하는 하드웨어 기반 Rx 클라이언트라는 메시지가 출력됩니다. Tx 링의 경우 vnic3은 기본 링 세트를 사용하며, 물리적 데이터 링크 net4에서 생성되는 다른 데이터 링크와 링을 공유합니다.

## 풀 및 CPU 관리

Oracle Solaris에서 영역 관리에는 zonecfg 또는 poolcfg 명령을 사용하여 비네트워킹 프로세스에 대한 CPU 리소스 풀을 지정하는 과정이 포함됩니다. 동일한 리소스 풀을 전용으로 지정하여 네트워크 프로세스도 관리하려면 dladm set-linkprop 명령을 사용하여 링크의 pool 등록 정보를 구성합니다. pool 링크 등록 정보를 사용하여 네트워킹 프로세스에 대한 CPU 풀을 지정할 수 있습니다. 이 등록 정보를 사용하면 네트워크 리소스 관리와 영역의 CPU 할당 및 관리를 보다 효율적으로 통합할 수 있습니다.

링크에 pool 등록 정보를 설정하고 링크를 영역의 네트워크 인터페이스로 지정하면 해당 링크가 영역의 풀에 바인딩됩니다. 해당 영역이 배타적 영역이 되도록 설정된 경우 이 영역에 지정되지 않은 다른 링크가 풀의 CPU 리소스를 더 이상 사용할 수 없습니다.

---

**참고** - 등록 정보 cpus를 별도로 설정하여 데이터 링크에 특정 CPU를 지정할 수 있습니다. cpus 및 pool 등록 정보는 상호 배타적입니다. 두 등록 정보를 지정된 데이터 링크에 모두 설정할 수는 없습니다. cpus 등록 정보를 사용하여 데이터 링크에 CPU 리소스를 지정하려면 [데이터 링크에 CPU를 할당하는 방법 \[163\]](#)을 참조하십시오.

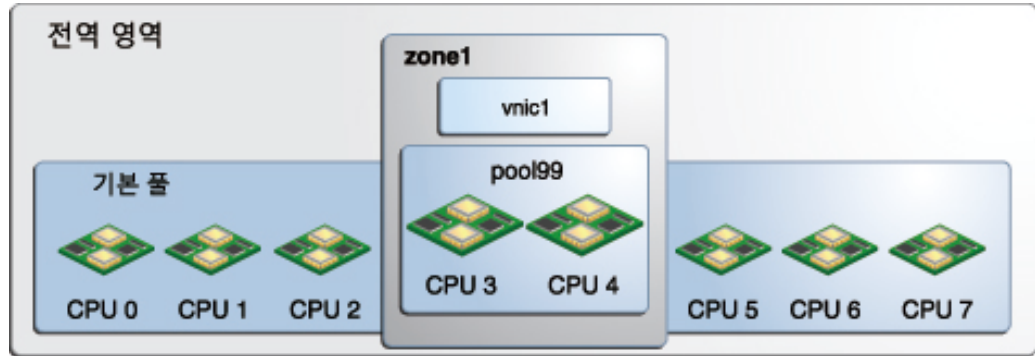
---

영역 내 풀에 대한 자세한 내용은 [“Oracle Solaris 11.2의 리소스 관리”의 13 장, “리소스 풀 만들기 및 관리 작업”](#)를 참조하십시오. 풀을 만들고 풀에 CPU 세트를 지정하는 방법에 대한 자세한 내용은 [poolcfg\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## 풀 및 CPU 작업

다음 그림에서는 pool 등록 정보가 데이터 링크에 지정된 경우 풀 작동 방식을 보여줍니다.

그림 7-1 영역에 지정된 VNIC의 pool 등록 정보



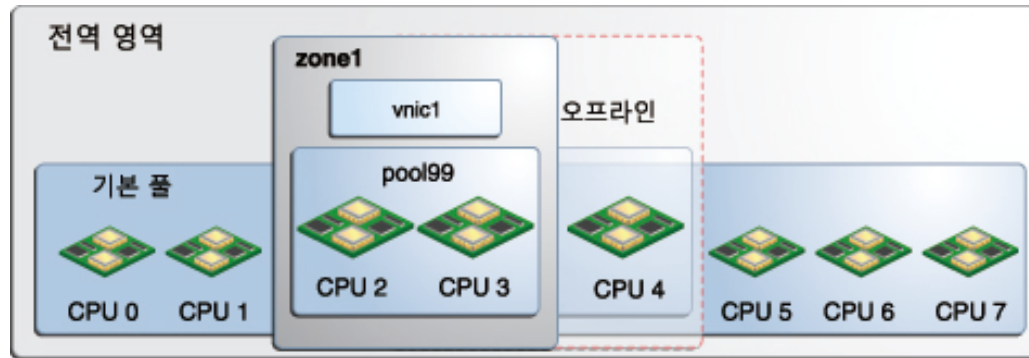
이 그림의 시스템에는 CPU가 8개 있습니다. 시스템에 구성된 풀이 없을 경우 모든 CPU가 기본 풀에 속하며 전역 영역에서 사용됩니다. 하지만 이 예에서는 pool99 풀이 생성되었으며 CPU 3과 CPU 4로 구성됩니다. 이 풀은 배타적 영역인 zone1과 연결됩니다. pool99를 vnic1의 등록 정보로 설정하면 pool99는 vnic1의 네트워킹 프로세스 관리 전용이 됩니다. vnic1이 zone1의 네트워킹 인터페이스에 지정된 후 pool99의 CPU가 zone1의 네트워킹 및 비네트워킹 프로세스를 모두 관리하도록 예약됩니다.

pool 등록 정보는 기본적으로 동적입니다. 일정 범위의 CPU로 영역 풀을 구성할 수 있으며 커널에 따라 풀의 CPU 세트에 지정되는 CPU가 결정됩니다. 데이터 링크에 대한 풀 변경 사항은 자동으로 구현되므로 해당 링크의 풀 관리가 간소화됩니다. 반면, cpu 등록 정보를 사용하여 링크에 특정 CPU를 지정하려면 지정할 CPU를 지정해야 합니다. 풀의 CPU 구성 요소를 변경할 때마다 cpu 등록 정보를 설정해야 합니다.

예를 들어, 그림 7-1. “영역에 지정된 VNIC의 pool 등록 정보”의 시스템 CPU 4는 오프라인 상태로 전환되었습니다. pool 등록 정보는 동적이기 때문에 소프트웨어가 추가 CPU를 풀과 자동으로 연결합니다. 따라서 CPU 두 개로 이루어진 풀의 원래 구성이 보존됩니다. vnic1의 경우 변경 작업이 투명합니다. 다음 그림에서는 업데이트된 구성을 보여줍니다.



그림 7-2 pool 등록 정보의 자동 재구성



dladm show-linkprop 명령을 사용하여 데이터 링크에 대한 정보를 표시할 경우 pool 및 cpus 데이터 링크 등록 정보의 EFFECTIVE 열에 있는 값은 현재 시스템에서 선택된 등록 정보 값을 나타냅니다.

다음은 pool 및 cpus 등록 정보에 대해 표시되는 읽기 전용 값입니다.

- pool 데이터 링크 등록 정보의 경우 EFFECTIVE 열의 값은 네트워크 프로세스에 사용되는 풀을 나타냅니다.
- cpus 데이터 링크 등록 정보의 경우 EFFECTIVE 열의 값은 네트워크 프로세스에 사용되는 CPU를 나타냅니다. 데이터 링크에 대한 cpus 등록 정보를 표시하는 방법을 보여주는 예는 예 7-5. “데이터 링크에 CPU 할당”을 참조하십시오.

영역의 CPU 리소스를 관리하려면 데이터 링크의 풀 등록 정보를 설정할 필요가 없습니다. zonecfg 및 poolcfg와 같은 명령을 사용하여 리소스 풀을 사용할 영역을 구성할 수 있습니다. 데이터 링크에 대해 cpus 및 pool 링크 등록 정보를 설정하지 않은 경우 데이터 링크에 대한 pool 및 cpus 등록 정보의 EFFECTIVE 열에 있는 값은 영역이 부트될 때 영역 구성에 따라 자동으로 설정됩니다. 기본 풀은 pool 등록 정보의 EFFECTIVE 열에 표시되고, 시스템은 cpus 등록 정보의 EFFECTIVE 열에 있는 값을 선택합니다. 따라서 dladm show-linkprop 명령을 사용할 경우 pool 및 cpus 등록 정보 값은 비어 있지만 pool 및 cpus 등록 정보의 EFFECTIVE 열에는 값이 표시됩니다.

영역의 CPU 풀을 네트워킹 프로세스에 지정하기 위해 데이터 링크의 pool 및 cpu 등록 정보를 직접 설정할 수도 있습니다. 이러한 등록 정보를 구성하고 나면 해당 값이 pool 및 cpus 등록 정보의 EFFECTIVE 열에 반영됩니다. 하지만 이 대체 방법은 영역의 네트워크 리소스 관리에 자주 사용되지 않습니다.

## 데이터 링크에 대한 CPU 풀 구성

이 절에서는 링크를 생성할 때나 이후에 링크에 추가 구성이 필요한 경우에 데이터 링크에 대한 pool 등록 정보를 설정하는 방법을 설명합니다.

### ▼ 데이터 링크에 대한 CPU 풀을 구성하는 방법

시작하기 전에 다음 작업을 완료한 상태여야 합니다.

- 지정된 개수의 CPU를 사용하여 프로세서 세트를 생성했습니다.
- 프로세서 세트가 연결될 풀을 생성했습니다.
- 프로세서 세트와 풀을 연결했습니다.

---

참고 - 이러한 필수 요건을 완료하는 방법에 대한 자세한 내용은 “Oracle Solaris 11.2의 리소스 관리”의 “구성을 수정하는 방법”을 참조하십시오.

---

#### 1. 링크의 pool 등록 정보를 영역에 대해 만든 CPU 풀로 설정합니다.

- VNIC를 아직 생성하지 않은 경우 다음 구문을 사용합니다.

```
# dladm create-vnic -l link -p pool=pool VNIC
```

- VNIC가 있는 경우 다음 구문을 사용합니다.

```
# dladm set-linkprop -p pool=pool VNIC
```

#### 2. VNIC를 사용하도록 영역을 설정합니다.

```
global# zonecfg -z zone
zonecfg:zone> add net
zonecfg:zone:net> set physical=VNIC
zonecfg:zone:net> end
```

#### 3. 구현한 변경 사항을 확인하고 커밋한 다음 영역을 종료합니다.

```
zonecfg:zone> verify
zonecfg:zone> commit
zonecfg:zone> exit
```

#### 예 7-4 영역에 링크의 CPU 풀 지정

이 예에서는 영역의 데이터 링크에 풀이 지정되는 방식을 보여줍니다. 이 시나리오는 그림 7-1. “영역에 지정된 VNIC의 pool 등록 정보”의 구성을 기반으로 합니다. 이 예에서는

pool99라는 CPU 풀이 영역에 대해 이미 구성되었다고 가정합니다. 그런 다음 VNIC에 풀이 지정됩니다. 최종적으로, 비전역 영역 zone1이 VNIC를 네트워크 인터페이스로 사용하도록 설정됩니다.

```
# dladm create-vnic -l net1 -p pool=pool99 vnic1

# zonecfg -z zone1
zonecfg:zone1> add net
zonecfg:zone1:net> set physical=vnic1
zonecfg:zone1:net> end
zonecfg:zone1> verify
zonecfg:zone1> commit
zonecfg:zone1> exit
```

## 데이터 링크에 CPU 할당

이 절에서는 cpu 등록 정보를 구성하여 데이터 링크에 CPU 리소스를 지정하는 방법을 설명합니다. 링크와 달리, 데이터 링크에 대해 CPU를 배타적으로 할당할 수 없습니다. 여러 데이터 링크에 대해 동일한 CPU 세트를 할당할 수 있습니다.

### ▼ 데이터 링크에 CPU를 할당하는 방법

1. 관리자로 로그인합니다.  
자세한 내용은 [“Oracle Solaris 11.2의 사용자 및 프로세스 보안”](#)의 [“지정된 관리 권한 사용”](#)을 참조하십시오.
2. 인터페이스에 대한 CPU 지정을 확인합니다.  

```
# dladm show-linkprop -p cpus link
```
3. 링크에 CPU를 지정합니다.  
데이터 링크에 대한 패킷을 처리하는 CPU 목록입니다. 데이터 링크에 대한 인터럽트는 목록에 있는 CPU 중 하나를 대상으로 해야 합니다.  

```
# dladm set-linkprop -p cpus=cpu1,cpu2,... link
```

  
*cpu1,cpu2,...* 링크에 지정할 CPU 번호를 나타냅니다. 링크 전용으로 여러 CPU를 지정할 수 있습니다.
4. (옵션) 링크와 연결된 CPU를 표시합니다.  

```
# dladm show-linkprop -p cpus link
```

#### 예 7-5 데이터 링크에 CPU 할당

이 예제에서는 특정 CPU를 데이터 링크 net0에 전용으로 지정하는 방법을 보여줍니다.

```
# dladm show-linkprop -p cpus net0
```

LINK	PROPERTY	PERM	VALUE	EFFECTIVE	DEFAULT	POSSIBLE
net0	cpus	rw	--	0-2	--	--

시스템에서 데이터 링크 net0에 세 개의 CPU(0-2)를 암시적으로 지정했다는 메시지가 출력됩니다. 하지만 CPU는 데이터 링크 net0에 배타적으로 할당되지 않습니다.

```
# dladm set-linkprop -p cpus=0,1 net0
```

```
# dladm show-linkprop -p cpus net0
```

LINK	PROPERTY	PERM	VALUE	EFFECTIVE	DEFAULT	POSSIBLE
net0	cpus	rw	0-1	0-1	--	--

데이터 링크 net0에 두 개의 CPU(0-1)를 명시적으로 지정했다는 메시지가 출력됩니다. 할당된 CPU는 데이터 링크 net0에 대한 패킷을 처리합니다.

## 플로우를 사용하여 네트워크 리소스 관리

플로우는 단일 속성이나 속성 조합을 기반으로 네트워크 패킷을 분류하는 사용자 정의 방법입니다. 플로우를 사용하여 네트워크 리소스를 더 할당할 수 있습니다. 플로우 개요는 [“플로우를 사용한 네트워크 리소스 관리” \[20\]](#)를 참조하십시오.

네트워크 리소스 관리 플로우 사용 시에는 다음과 같은 단계가 포함됩니다.

### 1. 플로우 만들기

플로우는 패킷 헤더의 정보에서 파생되는 단일 속성 또는 속성 조합을 기반으로 생성됩니다.

다음 속성 중 하나를 사용하여 플로우에 대한 패킷 트래픽을 구성할 수 있습니다.

- 로컬 IP 주소.
- 원격 IP 주소.
- 전송 프로토콜 이름(UDP, TCP 또는 SCTP).
- IPv6 패킷에서만 QoS를 위해 사용되는 DS 필드(차별화된 서비스 필드) 속성입니다. 자세한 내용은 [“Oracle Solaris 11.2의 IP Quality of Service 관리”](#)를 참조하십시오.

다음 속성 조합 중 하나를 사용하여 플로우에 대한 패킷 트래픽을 구성할 수 있습니다.

- 로컬 응용 프로그램 포트 번호를 포함한 전송 프로토콜 이름(UDP, TCP 또는 SCTP).
- 원격 응용 프로그램 포트 번호를 포함한 전송 프로토콜 이름(UDP, TCP 또는 SCTP).
- 로컬 IP 주소 및 로컬 응용 프로그램 포트 번호를 포함하는 전송 프로토콜 이름(UDP, TCP 또는 SCTP). 이 속성 조합은 원격 IP 주소와 원격 응용 프로그램 포트 번호를 더 포함할 수 있습니다(Transport protocol name (UDP, TCP or SCTP) + local IP address + local application port number [+ remote IP address [+ remote application port number]]).

플로우는 이러한 속성 조합 중 하나만을 기반으로 할 수 있습니다. 예를 들어, 사용 중인 전송 프로토콜과 포트(예: FTP의 경우 TCP 포트 21) 또는 IP 주소(예: 특정 소스 IP 주소

의 패킷)에 따라 플로우를 만들 수 있습니다. 대부분의 경우 전송 프로토콜, 로컬 또는 원격 IP 주소 및 로컬 또는 원격 포트를 지정하여 플로우를 만들 수 있습니다. 또한 동일한 링크에 속하는 모든 플로우는 동일한 속성 조합을 사용해야 합니다.

2. 네트워크 리소스와 관련된 등록 정보를 설정하여 플로우의 리소스 사용을 사용자 정의합니다. 현재는 대역폭 및 우선 순위 등록 정보를 플로우에 연결할 수 있습니다.

자세한 내용은 [“플로우 구성” \[165\]](#)을 참조하십시오.

## 플로우에서 리소스를 할당하는 명령

플로우에서 네트워크 리소스를 할당하는 데 사용되는 명령은 다음과 같습니다.

- 동시에 플로우를 만들고 리소스를 추가하려면 다음 명령 구문을 사용합니다.

```
# flowadm add-flow -l link -a attribute=value[,attribute=value] -p prop=value[,...] flow
```

플로우의 특성을 결정하는 정의된 속성 세트가 시스템의 플로우 제어 정책을 구성합니다. 플로우에 대한 패킷 트래픽을 구성하는 데 사용할 수 있는 다른 속성 목록은 [“플로우를 사용하여 네트워크 리소스 관리” \[164\]](#)를 참조하십시오.

- 기존 플로우의 등록 정보를 설정하려면 다음 명령 구문을 사용합니다.

```
# flowadm set-flowprop -p prop=value[,...] flow
```

여기서 *prop*은 플로우에 지정될 수 있는 플로우 등록 정보를 나타냅니다. 플로우 등록 정보는 링크에 직접 지정되는 등록 정보와 같습니다. 하지만 대역폭 및 우선 순위 등록 정보만 플로우와 연결될 수 있습니다. 이러한 등록 정보를 구성하려면 [플로우 구성 방법 \[165\]](#)을 참조하십시오.

자세한 내용은 [flowadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## 플로우 구성

이 절에서는 플로우를 만들고 플로우 등록 정보를 설정하는 방법을 설명합니다.

### ▼ 플로우 구성 방법

1. 관리자로 로그인합니다.

자세한 내용은 [“Oracle Solaris 11.2의 사용자 및 프로세스 보안”](#)의 [“지정된 관리 권한 사용”](#)을 참조하십시오.

2. (옵션) 사용 가능한 링크를 나열하여 플로우를 구성할 링크를 확인합니다.

```
# dladm show-link
```

3. 선택한 링크의 IP 인터페이스가 IP 주소로 올바르게 구성되었는지 확인합니다.

```
# ipadm show-addr
```

4. 각 플로우에 대해 결정한 속성에 따라 플로우를 만듭니다.

```
# flowadm add-flow -l link -a attribute=value[,attribute=value] flow
```

*link* 플로우를 구성 중인 링크를 나타냅니다.

*attribute* 플로우에 대한 네트워크 패킷을 구성하는 데 사용할 수 있는 단일 속성 또는 속성 조합을 나타냅니다. 속성에 대한 자세한 내용은 “[플로우를 사용하여 네트워크 리소스 관리](#)” [164]를 참조하십시오.

*flow* 플로우에 지정하는 이름을 나타냅니다.

플로우 및 플로우 속성에 대한 자세한 내용은 [flowadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

5. (옵션) 링크의 대역폭에 대한 사용 가능한 값 범위를 표시합니다.

```
# dladm show-linkprop -p maxbw link
```

*link* 플로우를 구성할 데이터 링크를 나타냅니다.

값 범위는 명령 출력의 POSSIBLE 필드 아래에 나열됩니다.

6. 적절한 플로우 등록 정보를 설정하여 플로우에 대한 리소스 제어를 구현합니다.

```
# flowadm set-flowprop -p prop=value[,...] flow
```

리소스를 제어하는 다음 플로우 등록 정보를 지정할 수 있습니다.

*maxbw* 플로우로 식별된 패킷이 사용할 수 있는 링크 대역폭의 최대 크기입니다. 설정한 값이 링크 대역폭에 허용되는 값의 범위 내에 있어야 합니다.

*priority* 지정된 플로우에 속하는 패킷이 처리되는 우선 순위입니다. *priority* 등록 정보에 허용되는 값은 *high*, *medium* 및 *low*입니다. 플로우의 우선 순위가 *high*로 설정된 경우 해당 플로우에 속하는 모든 패킷이 동일한 링크의 다른 패킷보다 먼저 처리됩니다. 이 등록 정보는 대기 시간에 민감한 응용 프로그램에 대한 플로우를 만드는 데 사용됩니다. 이 등록 정보의 기본값은 *medium*입니다.

---

참고 - 현재는 *priority* 등록 정보를 *medium*에서 *low*로 변경할 수 없습니다.

---

7. (옵션) 데이터 링크를 통해 만든 플로우를 표시합니다.

```
# flowadm
```

---

참고 - flowadm 명령을 하위 명령 없이 사용하면 flowadm show-flow 명령과 동일한 정보를 제공합니다.

---

8. (옵션) 지정된 플로우에 대한 등록 정보 값을 표시합니다.

```
# flowadm show-flowprop flow
```

이 명령은 maxbw 및 priority 플로우 등록 정보와 읽기 전용 hwflow 등록 정보를 표시합니다.

hwflow                   플로우의 패킷 분류를 이해하는 데 도움이 되는 읽기 전용 등록 정보입니다. 이 등록 정보의 가능한 값은 on 및 off입니다. on 값은 플로우가 NIC에 대해 오프로드되고 플로우에 대한 패킷이 하드웨어 레벨로 분류됨을 나타냅니다. 이 등록 정보는 flowadm add-flow, flowadm set-flowprop 또는 flowadm reset-flowprop 명령에서 -p 옵션과 함께 사용할 수 없습니다.

---

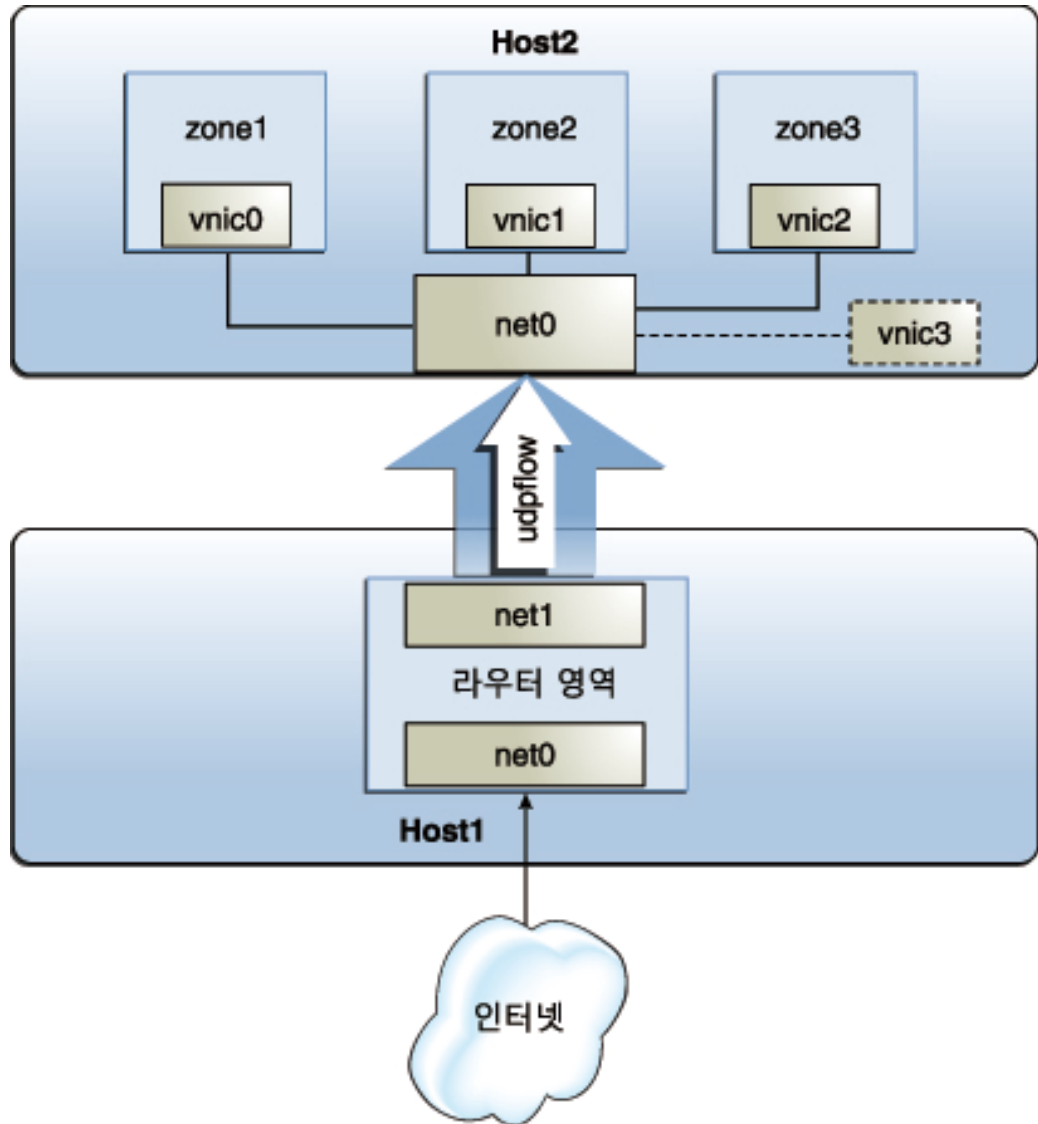
참고 - 현재는 모든 전송 프로토콜, 로컬 또는 원격 IP 주소 및 로컬 또는 원격 포트를 지정하여 정의한 플로우에만 hwflow에 대한 on 값을 지정할 수 있습니다. 또한 일부 NIC는 hwflow 등록 정보를 지원하지 않습니다.

---

## 사용 사례: 데이터 링크 및 플로우 등록 정보를 설정하여 네트워크 리소스 관리

다음 사용 사례는 데이터 링크 등록 정보와 플로우 등록 정보를 모두 설정하여 시스템의 효율성을 높이는 시나리오를 기반으로 합니다. 이 사용 사례는 다음 그림에 표시된 구성을 기반으로 합니다.

그림 7-3 데이터 링크와 플로우의 리소스를 관리하기 위한 시스템 구성



이 그림에서는 서로 연결된 다음과 같은 두 개의 물리적 호스트를 보여줍니다.

- Host1의 구성은 다음과 같습니다.



- 서버 및 라우터 역할을 하는 단일의 비전역 영역입니다. 두 개의 인터페이스가 영역에 지정됩니다. 즉, net0 인터페이스는 인터넷에 연결되고 net1 인터페이스는 Host2를 포함하는 내부 네트워크에 연결됩니다.
- net1을 통해 플로우를 구성하여 트래픽을 격리하고 플로우에 속하는 패킷에서 리소스를 사용하는 방법을 제어합니다. 플로우 구성에 대한 자세한 내용은 [“플로우를 사용하여 네트워크 리소스 관리” \[164\]](#)를 참조하십시오.
- Host2의 구성은 다음과 같습니다.
  - 비전역 영역 세 개와 해당 VNIC입니다. VNIC는 NIC 카드가 링 할당을 지원하는 net0을 통해 구성됩니다. 링 할당에 대한 자세한 내용은 [“NIC 링 관리” \[150\]](#)를 참조하십시오.
  - 각 영역의 네트워크 처리 부하는 서로 다릅니다. 이 예에서 zone1은 HTTP 클라이언트 역할을 합니다. 나머지 영역 zone2 및 zone3은 SSH(보안 셸) 프로토콜을 통해 Host1에 액세스하는 SSH 클라이언트로 사용됩니다. zone1에 대한 네트워크 트래픽은 zone2 및 zone3보다 더 높고 시간에 민감하지 않습니다. 하지만 zone2 및 zone3에 대한 트래픽은 낮고 시간에 민감합니다. 따라서 zone2 및 zone3에 대한 네트워크 트래픽을 더 빠르게 처리하려면 zone1에 대한 네트워크 트래픽에 할당되는 대역폭을 제한해야 합니다. zone1에 할당되는 대역폭을 제한하지 않을 경우 이 영역에서 사용 가능한 대역폭을 모두 사용합니다. 그러면 나머지 zone2 및 zone3 영역에 대한 대역폭이 거부됩니다.
  - 별도의 VNIC가 소프트웨어 기반 클라이언트로 구성됩니다. MAC 클라이언트에 대한 개요는 [“MAC 클라이언트에서 링 할당” \[150\]](#)을 참조하십시오.

이 사용 사례의 작업은 다음 작업을 포함합니다.

- 플로우 만들기 및 플로우 제어 구성 - net1을 통해 플로우가 생성되어 Host1의 net1에 수신되는 플로우에 속하는 패킷에 대해 별도의 리소스 제어를 만듭니다.
- Host2에서 VNIC에 대한 네트워크 리소스 등록 정보 구성 - 처리 부하를 기준으로 각 영역의 VNIC가 전용 링 세트로 구성됩니다. 또한 소프트웨어 기반 클라이언트의 예로 전용 링 없이 별도의 VNIC가 구성됩니다.

참고 - 사용 사례에 영역 구성에 대한 절차는 포함되지 않습니다. 영역을 구성하려면 [“Oracle Solaris 영역 만들기 및 사용”의 1 장](#), [“비전역 영역을 계획 및 구성하는 방법”](#)을 참조하십시오.

### 1. Host1에서 링크 및 IP 인터페이스에 대한 정보를 봅니다.

```
# ipadm
NAME                CLASS/TYPE    STATE    UNDER    ADDR
lo0                 loopback     ok       --       --
  lo0/v4             static       ok       --       127.0.0.1/8
  lo0/v6             static       ok       --       ::1/128
net1                ip           ok       --       --
  net1/v4            static       ok       --       192.168.200.103/24
```

```
net0          ip      ok      --      --
net0/v4      static ok      --      10.134.76.129/24
```

2. Host1에서 net1을 통해 다음 플로우를 만듭니다.

- httpflow - zone1과 net1 사이의 모든 HTTP 트래픽을 포함합니다.

```
# flowadm add-flow -l net1 -a transport=tcp,local_ip=192.168.200.103,\
local_port=80,remote_ip=192.168.200.110 httpflow
```

- sshflow - net1에서 주고 받는 모든 SSH 트래픽을 포함합니다.

```
# flowadm add-flow -l net1 -a transport=tcp,local_ip=192.168.200.103,\
local_port=22 sshflow
```

3. 플로우에 대한 리소스 제어를 구현합니다.

- httpflow의 경우 최대 대역폭을 500M으로 설정합니다.

```
# flowadm set-flowprop -p maxbw=500M httpflow
```

- sshflow의 경우 우선 순위를 high로 설정합니다.

```
# flowadm set-flowprop -p priority=high sshflow
```

4. 생성된 플로우에 대한 정보를 확인합니다.

```
# flowadm
FLOW      LINK      PROTO  LADDR          LPORT  RADDR          RPORT  DSFLD
httpflow  net1      tcp    192.168.200.103  80     192.168.200.110 --     --
sshflow   net1      tcp    192.168.200.103  22     --             --     --
```

```
# flowadm show-flowprop
```

FLOW	PROPERTY	PERM	VALUE	DEFAULT	POSSIBLE
httpflow	maxbw	rw	500	--	--
httpflow	priority	rw	medium	medium	low,medium,high
httpflow	hwflow	r-	off	--	on,off
sshflow	maxbw	rw	--	--	--
sshflow	priority	rw	high	medium	low,medium,high
sshflow	hwflow	r-	off	--	on,off

출력에 대한 자세한 내용은 [flowadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

5. Host2에서 net0에 각 영역에 대한 VNIC를 구성합니다.

```
# dladm create-vnic -l net0 vnic0
# dladm create-vnic -l net0 vnic1
# dladm create-vnic -l net0 vnic2
```

6. 각 VNIC에 대한 리소스 제어를 구현합니다.

```
# dladm set-linkprop -p rxrings=4,txrings=4 vnic0
# dladm set-linkprop -p rxrings=2,txrings=2 vnic1
# dladm set-linkprop -p rxrings=1,txrings=1 vnic2
```

7. 해당 영역에 VNIC를 지정합니다.

```
# zonecfg -z zone1
# zonecfg:zone1> add net
# zonecfg:zone1:net> set physical=vnic0
# zonecfg:zone1:net> end
# zonecfg:zone1> commit
# zonecfg:zone1> exit
# zoneadm -z zone1 reboot
```

```
# zonecfg -z zone2
# zonecfg:zone2> add net
# zonecfg:zone2:net> set physical=vnic1
# zonecfg:zone2:net> end
# zonecfg:zone2> commit
# zonecfg:zone2> exit
# zoneadm -z zone2 reboot
```

```
# zonecfg -z zone3
# zonecfg:zone3> add net
# zonecfg:zone3:net> set physical=vnic2
# zonecfg:zone3:net> end
# zonecfg:zone3> commit
# zonecfg:zone3> exit
# zoneadm -z zone3 reboot
```

8. 주 인터페이스인 net0과 링을 공유하는 소프트웨어 기반 클라이언트를 만듭니다.

```
# dladm create-vnic -p rxrings=sw,txrings=sw -l net0 vnic3
```

9. Host2의 CPU 세트인 pool1이 zone1에 지정된다고 가정합니다. 또한 CPU의 동일한 pool1을 지정하여 zone1에 대한 네트워크 프로세스를 관리합니다.

```
# dladm set-linkprop -p pool=pool1 vnic0
```



## 네트워크 트래픽 및 리소스 사용 모니터링

---

이 장에서는 데이터 링크 및 플로우에서 네트워크 리소스의 사용에 대한 네트워크 통계를 모니터링하는 작업에 대해 설명합니다. 로그 파일에 네트워크 트래픽 통계를 기록하도록 시스템의 네트워크 계정을 구성합니다. 이 통계 정보는 프로비저닝, 통합 및 청구 용도에 대한 리소스 할당을 분석하는 데 도움이 됩니다. 이 장에서는 통계를 표시하는 데 사용할 수 있는 두 가지 명령인 `dlstat` 및 `flowstat`를 소개합니다.

이 장의 내용:

- “데이터 링크 및 플로우의 네트워크 트래픽 통계 모니터링 개요” [173]
- “네트워크 트래픽 통계를 모니터링하는 명령” [175]
- “Displaying Network Traffic Statistics of Links” [175]
- “Displaying Network Traffic Statistics of Flows” [181]
- “네트워크 트래픽에 대한 네트워크 정산 구성” [183]

### 데이터 링크 및 플로우의 네트워크 트래픽 통계 모니터링 개요

패킷은 시스템에 들어오고 나갈 때 경로를 순회합니다. 세부적인 레벨에서 패킷은 NIC의 수신(Rx) 링과 전송(Tx) 링을 통해 수신 및 전송됩니다. 이러한 링의 인바운드 패킷은 추가 처리를 위해 네트워크 스택의 위로 전달되고 아웃바운드 패킷은 네트워크로 전송됩니다.

시스템 리소스를 결합하고 할당하여 네트워크 트래픽을 관리할 수 있습니다. 데이터 링크 및 플로우 모두에 대한 수신측과 전송측 네트워크 트래픽 통계를 모니터링할 수 있습니다. 이 장에서는 데이터 링크 및 플로우에서 주로 수신측 네트워크 트래픽 통계를 중심으로 설명합니다.

데이터 링크 등록 정보를 설정하여 데이터 링크의 수신 링, 전송 링 및 기타 리소스를 구성할 수 있습니다. 데이터 링크의 네트워크 트래픽에 따라 전용 하드웨어 링을 데이터 링크에 지정함으로써 패킷을 처리하는 시스템의 효율성을 높일 수 있습니다. 예를 들어, 네트워크 트래픽이 가장 많은 데이터 링크에 링을 더 할당할 수 있습니다. 하드웨어 링을 데이터 링크에 할당하는 방법에 대한 자세한 내용은 “클라이언트 구성 및 링 할당” [154]을 참조하십시오.

다음과 같은 이유로 데이터 링크에는 전용 하드웨어 링이 없을 수 있습니다.

- 하드웨어 리소스 부족. 예를 들어, 데이터 링크에 독점적으로 지정할 수 있는 사용 가능한 링이 없을 수 있습니다.
- 하드웨어 기능 부족. 예를 들어, NIC가 하드웨어 링을 노출하지 않습니다.

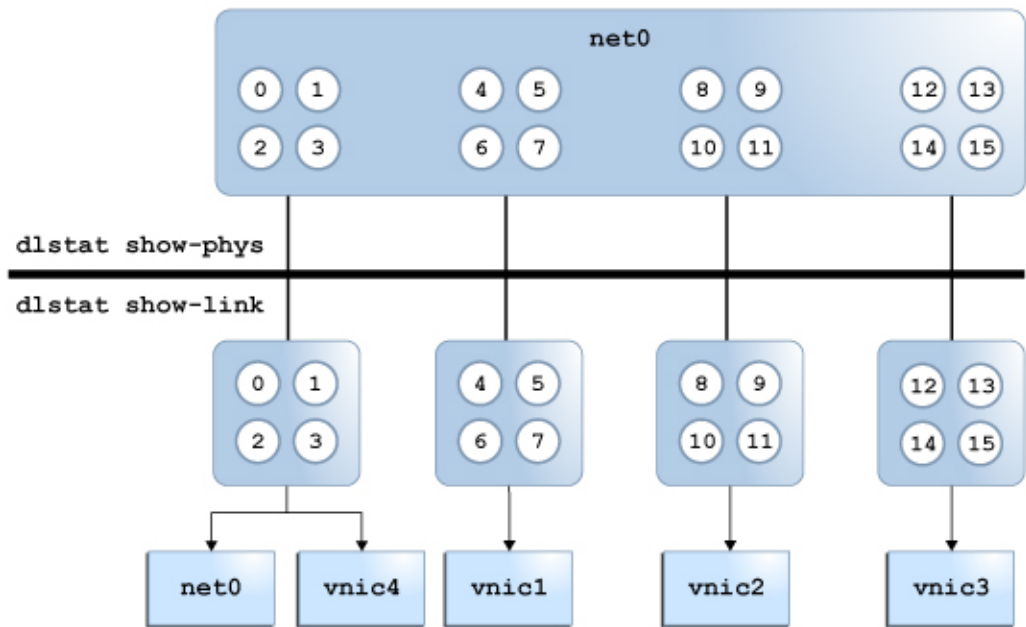
- 데이터 링크가 하위 하드웨어 데이터 링크에 연결되지 않을 수 있습니다. 예를 들어, etherstub을 통해 vNIC를 만드는 경우입니다.

일부 데이터 링크는 다음과 같은 이유로 링을 공유하도록 구성될 수 있습니다.

- 데이터 링크가 전용 링이 필요한 집중적 프로세스를 수행하지 않을 수도 있습니다.
- NIC가 링 할당을 지원하지 않을 수도 있습니다.
- 데이터 링크는 링 할당을 지원하지만 배타적 사용을 위해 링을 더 이상 지정할 수 없습니다.

다음 그림은 데이터 링크 사이에서 하드웨어 링 할당을 보여줍니다.

그림 8-1 데이터 링크의 링 할당



이 그림에서는 다음 구성을 보여줍니다.

- net0 데이터 링크에는 다른 데이터 링크에 할당할 수 있는 16개의 하드웨어 링(0-15)이 있습니다.
- vNIC vnic1, vnic2, vnic3 및 vnic4는 데이터 링크 net0을 통해 구성됩니다.
- vNIC vnic1, vnic2 및 vnic3에는 각각 4개의 전용 하드웨어 링이 지정됩니다.
- 하드웨어 링(0-3)은 데이터 링크 net0과 vNIC vnic4 사이에 공유됩니다. 다음 예에서는 물리적 데이터 링크 net0에 대한 링 할당을 보여줍니다.

```
# dladm show-phys -H net0
LINK          RINGTYPE RINGS          CLIENTS
net0          RX        0-3            <default,mcast>,vnic4
net0          RX        4-7            vnic1
net0          RX        8-11          vnic2
net0          RX        12-15         vnic3
net0          TX        0-7            <default>,vnic4,vnic3,vnic2,vnic1
```

- dladm show-phys 명령을 사용하여 물리적 데이터 링크 net0에 대한 네트워크 트래픽 통계를 표시합니다. 예 8-1. “시스템의 물리적 링크에 대한 트래픽 통계 표시”을 참조하십시오.
- dladm show-link 명령을 사용하여 데이터 링크 net0, vnic1, vnic2, vnic3 및 vnic4에 대한 네트워크 트래픽 통계를 표시합니다. 예 8-7. “전용 하드웨어 링이 있는 데이터 링크에 대한 네트워크 트래픽 통계 표시”을 참조하십시오.

## 네트워크 트래픽 통계를 모니터링하는 명령

dlstat 및 flowstat 명령을 사용하면 각각 데이터 링크 및 플로우의 네트워크 트래픽 통계를 모니터링할 수 있습니다. 이러한 명령은 dladm 및 flowadm 명령과 동일합니다. 다음 표에서는 관리 명령 쌍의 기능을 모니터링 명령 쌍과 비교합니다.

관리 명령		모니터링 명령	
명령	기능	명령	기능
dladm	데이터 링크 구성 및 관리	dlstat	데이터 링크에 대한 트래픽 통계 표시
flowadm	플로우 구성 및 관리	flowstat	플로우에 대한 트래픽 통계 표시

## Displaying Network Traffic Statistics of Links

dlstat 명령의 다음 변형을 사용하여 네트워크 트래픽 정보를 표시할 수 있습니다.

명령	제공되는 정보
dlstat [link]	데이터 링크당 인바운드 및 아웃바운드 트래픽 통계 표시
dlstat -rt [link]	
dlstat show-link [link]	
dlstat show-link -rt [link]	데이터 링크당 링별 인바운드 및 아웃바운드 트래픽 통계 표시

명령	제공되는 정보
<code>dlstat show-phys [link]</code>	네트워크 물리 장치당 인바운드 및 아웃바운드 트래픽 통계 표시
<code>dlstat show-phys -rt [link]</code>	네트워크 물리 장치당 링별 인바운드 및 아웃바운드 트래픽 통계 표시
<code>dlstat show-aggr [link]</code> <code>dlstat show-aggr -rt [link]</code>	집계당 포트별 인바운드 및 아웃바운드 트래픽 통계 표시
<code>dlstat show-bridge [bridge]</code> <code>dlstat show-bridge -rt [bridge]</code>	브리지당 인바운드 및 아웃바운드 트래픽 통계 표시

`dlstat` 명령에 `-r` 옵션을 사용하여 수신측 통계 정보를 표시하거나 `-t` 옵션을 사용하여 전송측 통계 정보를 표시할 수 있습니다. 다른 옵션에 대한 자세한 내용은 [dlstat\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## 네트워크 장치의 네트워크 트래픽 통계 표시

`dlstat show-phys` 명령은 물리적 네트워크 장치를 나타내는 통계를 제공합니다. [그림 8-1. "데이터 링크의 링 할당"](#)에 표시된 것처럼 `dlstat show-phys` 명령은 네트워크 스택의 장치 계층에 있는 하드웨어 링에서 작동합니다.

다음 명령 구문을 사용하여 네트워크 장치에서 네트워크 트래픽 통계를 표시할 수 있습니다.

```
# dlstat show-phys [-r|-t] [-Tu | -Td] [link] [interval [count]]
```

<code>-r</code>	수신측 네트워크 트래픽 통계만 표시합니다. <code>-t</code> 옵션을 이 옵션과 함께 지정해서는 안 됩니다.  <code>-r</code> 옵션 또는 <code>-t</code> 옵션을 지정하지 않은 경우 전송측 네트워크 통계와 수신측 네트워크 통계가 모두 표시됩니다.
<code>-t</code>	전송측 네트워크 트래픽 통계만 표시합니다. <code>-r</code> 옵션을 이 옵션과 함께 지정해서는 안 됩니다.  <code>-r</code> 또는 <code>-t</code> 옵션을 지정하지 않은 경우 전송측 네트워크 통계와 수신측 네트워크 통계가 모두 표시됩니다.
<code>-Tu</code>	내부 표현에 현재 시간을 표시합니다.
<code>-Td</code>	현재 시간을 표준 날짜 형식으로 표시합니다.
<code>link</code>	네트워크 통계를 모니터링할 데이터 링크의 이름입니다. 데이터 링크를 지정하지 않은 경우 시스템에 구성된 모든 데이터 링크에 대한 정보가 표시됩니다.
<code>interval</code>	네트워크 통계를 새로 고칠 간격(초)을 지정합니다.



`count` 표시된 네트워크 트래픽 통계를 새로 고칠 횟수를 지정합니다. 개수 값을 지정하지 않은 경우 통계를 무기한 새로 고칩니다.

**예 8-1** 시스템의 물리적 링크에 대한 트래픽 통계 표시

이 예에서는 시스템에 있는 각 링크의 수신 네트워크 트래픽과 송신 네트워크 트래픽이 모두 표시됩니다. 패킷 수 및 바이트 크기가 표시됩니다.

```
# dlstat show-phys
LINK  IPKTS  RBYTES  OPKTS  OBYTES
net5      0      0      0      0
net6      0      0      0      0
net0  25.57K  5.10M   1.93K  226.05K
net0      179   26.63K   161   22.75K
net3      0      0      0      0
net4      0      0      0      0
net2      0      0      0      0
net8      238  137.16K   191   8.41K
net1      0      0      0      0
...
```

출력에는 다음 정보가 표시됩니다.

- LINK                   이름으로 식별된 물리적 데이터 링크 또는 가상 데이터 링크
- IPKTS                  링크의 인바운드 패킷 수
- RBYTES                링크에서 수신된 바이트 수
- OPKTS                  링크의 아웃바운드 패킷 수
- OBYTES                이 링크에서 전송된 바이트 수

**예 8-2** 네트워크 장치에 대한 수신측 트래픽 통계 표시

이 예에서는 수신 중인 네트워크 트래픽 통계가 간격 값 2초와 개수 값 3을 사용하여 표시됩니다.

```
# dlstat show-phys -r 2 3
LINK  TYPE  INDEX  IPKTS  RBYTES
net0  rx    0      8.03M  12.09G
net1  rx    0      0      0
net0  rx    0      8.79K  13.28M
net1  rx    0      0      0
net0  rx    0      8.50K  12.83M
net1  rx    0      0      0
```

`net0` 및 `net1` 데이터 링크를 세트로 고려합니다. 첫번째 데이터 링크 세트인 `net0` 및 `net1`은 수신된 총 패킷 수와 바이트 수를 표시합니다. 이 예에서 수신된 총 패킷 수는 8.03M이고, `net0`에 수신된 총 바이트 수는 12.09G입니다. 두번째 데이터 링크 세트인 `net0` 및 `net1`은 네

트래픽 통계를 초당 속도(정규화된 값이라고도 함) 단위로 표시합니다. 즉, 8.79K는 net0에서 2초 간격으로 수신되는 패킷의 정규화된 값입니다. 마찬가지로 세번째 데이터 링크 세트인 net0 및 net1도 네트워크 트래픽 통계에 대한 정규화된 값을 2초 간격으로 표시합니다.

**예 8-3**            네트워크 장치에 대한 수신측 트래픽 통계 표시

이 예에서는 데이터 링크 net0에 대한 수신 트래픽 통계가 표시됩니다.

```
# dlstat show-phys -r net0
LINK  TYPE  ID   INDEX  IPKTS   RBYTES
net0   rx    local --      0        0
net0   rx     hw   1        0        0
net0   rx     hw   2   1.73M   2.61G
net0   rx     hw   3        0        0
net0   rx     hw   4   8.44M   12.71G
net0   rx     hw   5   5.68M   8.56G
net0   rx     hw   6   4.99M   7.38G
net0   rx     hw   7        0        0
```

이 예에서 net0 데이터 링크에는 INDEX 필드 아래에 식별된 8개의 수신 링이 있습니다. 링당 패킷의 균일한 배포는 링이 링크의 부하에 따라 링크에 올바르게 할당되었음을 나타내는 이상적인 구성입니다. 균일하지 않은 배포는 링크당 링의 부적절한 배포를 나타냅니다. 불균등 배포는 NIC가 동적 링 할당을 지원하는지 여부에 따라 확인할 수 있습니다. 이 경우 패킷을 보다 고르게 처리할 수 있도록 링크당 링을 재배포할 수 있습니다. 자세한 내용은 [“NIC 링 관리” \[150\]](#)를 참조하십시오.

**예 8-4**            네트워크 장치에 대한 전송측 트래픽 통계 표시

이 예에서는 네트워크 장치로서 net0의 전송 링 사용량이 표시됩니다.

```
# dlstat show-phys -t net0
LINK  TYPE  INDEX  OPKTS  OBYTES
net0   tx    0       93    4.63K
net0   tx    1        0        0
net0   tx    2        0        0
net0   tx    3        0        0
net0   tx    4        0        0
net0   tx    5       47    11.02K
net0   tx    6       23     7.13K
net0   tx    7        0        0
```

**예 8-5**            네트워크 장치에 대한 트래픽 통계를 시간과 함께 표시

다음 예에서는 네트워크 장치로서 net0에 대한 네트워크 트래픽 관련 통계가 현재 시간의 내부 표현으로 표시됩니다.

```
# dlstat show-phys -Tu net0
1401652481
          LINK  IPKTS  RBYTES  OPKTS  OBYTES
```

```
net0      184  27.14K    165  22.91K
```

다음 예에서는 네트워크 장치로서 net0에 대한 네트워크 트래픽 관련 통계가 표준 날짜 형식으로 된 현재 시간과 함께 표시됩니다.

```
# dlstat show-phys -Td net0
Sun Jun 1 12:54:47 PDT 2014
      LINK  IPKTS  RBYTES  OPKTS  OBYTES
net0    184   27.14K   165   22.91K
```

## 데이터 링크의 네트워크 트래픽 통계 표시

dlstat show-link 명령을 사용하여 데이터 링크에 대한 네트워크 트래픽 통계를 표시할 수 있습니다.

**예 8-6**            데이터 링크에 대한 네트워크 트래픽 통계 표시

이 예에서는 데이터 링크 vnic0에 대한 네트워크 트래픽 통계를 보여줍니다.

```
# dlstat show-link vnic0
LINK  IPKTS  RBYTES  OPKTS  OBYTES
vnic0  3      180     0      0
```

**예 8-7**            전용 하드웨어 링이 있는 데이터 링크에 대한 네트워크 트래픽 통계 표시

이 예에서는 4개의 전용 Rx 링이 있는 데이터 링크 vnic0에 대한 수신측 네트워크 트래픽 통계를 보여줍니다. 출력에서 ID 열의 hw 값은 데이터 링크 vnic0에 전용 하드웨어 링이 있음을 나타냅니다.

```
# dlstat show-link -r vnic0
LINK  TYPE  ID  INDEX  IPKTS  RBYTES  INTRS  POLLS  IDROPS
vnic0  rx    local  --      0      0      0      0      0
vnic0  rx    other  --      64    2.94K   0      0      0
vnic0  rx    hw     8       0      0      0      0      0
vnic0  rx    hw     9      53    7.97K   53     0      0
vnic0  rx    hw    10      4     392     4      0      0
vnic0  rx    hw    11  153.65K  220.68M  153.65K  0      0
```

**예 8-8**            데이터 링크에 대한 전송측 네트워크 트래픽 통계 표시

이 예에서는 데이터 링크 vnic0에 대한 전송측 네트워크 트래픽 통계를 보여줍니다.

```
# dlstat show-link -t vnic0
LINK  TYPE  ID  INDEX  OPKTS  OBYTES  ODROPS
vnic0  tx    local  --      0      0      0
vnic0  tx    other  --      19     798     0
vnic0  tx    sw     --      0      0      0
```

**예 8-9** 전용 하드웨어 링이 없는 데이터 링크에 대한 네트워크 트래픽 통계 표시

이 예에서는 전용 Rx 링이 없는 데이터 링크 net6에 대한 네트워크 트래픽 통계를 보여줍니다. 출력에서 ID 열의 sw 값은 데이터 링크 net6이 전용 하드웨어 링으로 구성되지 않았음을 나타냅니다.

```
# dlstat show-link -r net6
LINK TYPE ID INDEX IPKTS RBYTES INTRSPOLLS IDROPS
net6 rx local -- 0 0 0 0 0
net6 rx other -- 0 0 0 0 0
net6 rx sw -- 0 0 0 0 0
```

## 링크 집계의 네트워크 트래픽 통계 표시

dlstat show-aggr 명령은 트래픽이 시스템의 집계를 통과할 때 각 집계의 포트에 대한 네트워크 패킷 통계를 보여줍니다.

**예 8-10** 링크 집계에 대한 네트워크 트래픽 통계 표시

```
# dlstat show-aggr
LINK PORT IPKTS RBYTES OPKTS OBYTES
aggr0 -- 13 832 13 780
aggr0 net0 0 0 13 780
aggr0 net3 13 832 0 0
```

이 예에서는 두 기본 링크 net0 및 net3을 가진 링크 집계 aggr0의 구성을 보여줍니다. 네트워크 트래픽이 수신되거나 집계를 통해 시스템에서 전송되면 수신 및 송신 패킷과 해당 크기에 대한 정보가 모든 포트에 대해 보고됩니다. 포트는 집계의 기본 링크로 식별됩니다.

링크 집계에 대한 자세한 내용은 [“Oracle Solaris 11.2의 네트워크 데이터 링크 관리”의 2장, “링크 집계를 사용하여 고가용성 구성”](#)을 참조하십시오.

## 브리지의 네트워크 트래픽 통계 표시

dlstat show-bridge 명령은 각 브리지에 대한 네트워크 통계를 표시하고 각 브리지에 연결된 링크에 대한 통계를 나열합니다.

**예 8-11** 브리지에 대한 네트워크 트래픽 통계 표시

이 예에서는 rbbblue0 및 stbred0 브리지에 대한 네트워크 통계를 표시합니다.

```
# dlstat show-bridge
BRIDGE LINK IPKTS RBYTES OPKTS OBYTES DROPS FORWARDS
rbbblue0 -- 1.93K 587.29K 2.47K 3.30M 0 0
```

	simblue1	72	4.32K	2.12K	2.83M	0	--
	simblue2	1.86K	582.97K	348	474.04K	0	--
stbred0	--	975	976.69K	3.44K	1.13M	0	38
	simred3	347	472.54K	1.86K	583.03K	0	--
	simred4	628	504.15K	1.58K	551.51K	0	--

## Displaying Network Traffic Statistics of Flows

플로우에 대한 통계를 사용하여 시스템에 정의된 모든 플로우의 패킷 트래픽을 평가할 수 있습니다. 플로우에 대한 통계를 표시하려면 `flowstat` 명령을 사용합니다. 자세한 내용은 [flowstat\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

다음 명령 구문을 사용하여 플로우에 대한 네트워크 트래픽 통계를 표시합니다.

```
# flowstat [-r|-t] [-l link] [-Tu | -Td] [flow] [interval [count]]
```

- r                   수신측 네트워크 트래픽 통계만 표시합니다. -t 옵션을 이 옵션과 함께 지정해서는 안됩니다.  
-r 옵션 또는 -t 옵션을 지정하지 않은 경우 전송측 네트워크 통계와 수신측 네트워크 통계가 모두 표시됩니다.
- t                   전송측 네트워크 트래픽 통계만 표시합니다. -r 옵션을 이 옵션과 함께 지정해서는 안됩니다.  
-r 옵션 또는 -t 옵션을 지정하지 않은 경우 전송측 네트워크 통계와 수신측 네트워크 통계가 모두 표시됩니다.
- l *link*            네트워크 통계를 모니터링할 데이터 링크의 이름입니다. 데이터 링크를 지정하지 않은 경우 시스템에 구성된 모든 플로우에 대한 정보가 표시됩니다.
- Tu                  내부 표현에 현재 시간을 표시합니다.
- Td                  현재 시간을 표준 날짜 형식으로 표시합니다.
- flow*                네트워크 통계를 모니터링할 플로우의 이름입니다. 플로우를 지정하지 않은 경우 지정된 링크에 따라 모든 플로우 통계가 표시됩니다.
- interval*          네트워크 통계를 새로 고칠 간격(초)을 지정합니다. 간격 값을 지정하지 않은 경우 패킷 및 바이트의 총 수가 표시됩니다.
- count*             표시된 네트워크 트래픽 통계를 새로 고칠 횟수를 지정합니다. 개수 값을 지정하지 않은 경우 통계를 무기한 새로 고칩니다.

다음 예제에서는 시스템에서 구성된 플로우에 대한 정보를 표시하는 여러 방법을 보여줍니다.

**예 8-12**            플로우에 대한 네트워크 트래픽 통계 표시

이 예에서는 시스템에 구성된 모든 플로우에 대한 네트워크 트래픽 통계가 간격 값 1초와 개수 값 2를 사용하여 표시됩니다.

```
# flowstat 1 2
FLOW      IPKTS  RBYTES  IDROPS   OPKTS   OBYTES  ODROPS
flow1     1.78M  2.68G   443     889.57K 58.72M   0
flow2      0      0       0       0       0       0
flow1     8.31K  12.51M  243     4.22K   280.45K 0
flow2      0      0       0       0       0       0
```

flow1 및 flow2 플로우를 세트로 고려합니다. 첫번째 플로우 세트인 flow1 및 flow2는 플로우에서 주고 받은 총 네트워크 트래픽 통계 수를 표시합니다. 이 예에서 1.78M은 flow1에서 수신한 총 패킷 수입니다. 두번째 플로우 세트인 flow1 및 flow2는 네트워크 통계를 초당 속도(정규화된 값이라고도 함)로 표시합니다. 이 예에서 8.31K는 flow1에서 1초 간격으로 수신하는 패킷의 정규화된 값입니다.

**예 8-13**            플로우에 대한 전송측 트래픽 통계 표시

이 예에서는 시스템에 구성된 모든 플로우의 송신 트래픽에 대한 네트워크 트래픽 통계를 표시합니다.

```
# flowstat -t
FLOW      OPKTS  OBYTES  ODROPS
flow1    24.37M  1.61G   0
flow2      0      0       0
```

**예 8-14**            데이터 링크의 플로우에 대한 수신측 트래픽 통계 표시

이 예에서는 데이터 링크 net0에 구성된 모든 플로우에 대한 수신 네트워크 트래픽이 간격 값 2초와 개수 값 5를 사용하여 표시됩니다.

```
# flowstat -r -l net0 2 5
FLOW      IPKTS  RBYTES  IDROPS
flow1     2.38M  3.59G   14.89K
flow2      0      0       0
flow1     8.24K  12.40M  180
flow2      0      0       0
flow1     8.94K  13.47M  206
flow2      0      0       0
flow1     7.43K  11.19M  161
flow2      0      0       0
flow1     8.38K  12.62M  213
flow2      0      0       0
```

flow1 및 flow2 플로우를 세트로 고려합니다. 첫번째 플로우 세트인 flow1 및 flow2는 플로우에 수신된 패킷 및 바이트의 총 수를 표시합니다. 이 예에서 수신된 총 패킷 수는 2.38M이고, flow1에 수신된 총 바이트 수는 3.59G입니다. 두번째 플로우 세트인 flow1 및 flow2는 네트워크 통계를 초당 속도(정규화된 값이라고도 함)로 표시합니다. 이 예에서 8.24K는

flow1에서 2초 간격으로 수신하는 패킷의 정규화된 값입니다. 마찬가지로 다른 플로우 세트도 네트워크 트래픽 통계에 대한 정규화된 값을 2초 간격으로 표시합니다.

예 8-15 플로우에 대한 트래픽 통계를 시간과 함께 표시

다음 예에서는 데이터 링크 net0을 통해 생성되는 모든 플로우의 수신 트래픽에 대한 통계를 현재 시간의 내부 표현을 사용하여 표시합니다.

```
# flowstat -r -l net0 -Tu
1364380279
      FLOW      IPKTS  RBYTES  IDROPS
tcp-flow 183.11K 270.24M      0
udp-flow      0      0      0
```

다음 예에서는 데이터 링크 net0을 통해 생성되는 모든 플로우의 수신 트래픽에 대한 통계를 표준 날짜 형식으로 된 현재 시간과 함께 표시합니다.

```
# flowstat -r -l net0 -Td
Wednesday, March 27, 2013 04:01:01 PM IST
      FLOW      IPKTS  RBYTES  IDROPS
tcp-flow 183.11K 270.24M      0
udp-flow      0      0      0
```

## 네트워크 트래픽에 대한 네트워크 정산 구성

확장 정산 기능을 사용하여 시스템에서 네트워크 정산을 설정할 수 있습니다. 네트워크 정산에는 로그 파일에서 네트워크 트래픽에 대한 통계 수집이 포함됩니다. 트래픽 레코드를 추적, 프로비저닝, 통합 및 청구 용도로 유지 관리할 수 있습니다. 나중에 로그 파일을 참조하여 일정 기간의 네트워크 사용에 대한 기록 정보를 확인할 수 있습니다.

네트워크 정산을 설정하려면 확장 정산 기능의 `acctadm` 명령을 사용합니다. 자세한 내용은 [acctadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오. 네트워크 정산 설정을 마친 후에는 `flowstat` 명령을 사용하여 트래픽 통계를 기록합니다.

### ▼ 네트워크 정산 설정 방법

1. 관리자로 로그인합니다.  
자세한 내용은 “Oracle Solaris 11.2의 사용자 및 프로세스 보안”의 “지정된 관리 권한 사용”을 참조하십시오.
2. 확장 정산 기능으로 사용으로 설정될 수 있는 정산 유형의 상태를 확인합니다.

```
# acctadm [process | task | flow | net]
```

확장 정산 기능은 4가지 유형의 정산을 사용으로 설정할 수 있습니다. `acctadm` 명령의 선택적 피연산자는 다음 계정 유형에 따라 달라집니다.

- `process` - 프로세스 계정
- `task` - 작업 계정
- `flow` - 플로우 계정
- `net` - 네트워크 계정

---

참고 - 네트워크 계정은 “[플로우를 사용하여 네트워크 리소스 관리](#)” [164]에 설명된 대로 `flowadm` 및 `flowstat` 명령을 통해 관리되는 플로우에도 적용됩니다. 따라서 이러한 플로우의 계정을 설정하려면 `acctadm` 명령에 `net` 옵션을 사용합니다. IPQoS 구성에 대한 플로우 정산을 사용으로 설정하는 `flow` 옵션을 사용하지 마십시오.

`net`을 지정하면 네트워크 계정의 상태가 표시됩니다. `net`을 사용하지 않으면 네 가지 계정 유형의 상태가 모두 표시됩니다.

---

### 3. 네트워크 트래픽에 대해 확장 계정을 사용으로 설정합니다.

```
# acctadm -e extended -f filename net
```

여기서 `filename`에는 네트워크 트래픽 통계를 캡처하는 로그 파일의 전체 경로가 포함됩니다. 지정한 임의 디렉토리에 로그 파일을 만들 수 있습니다.

### 4. 확장 네트워크 계정이 활성화되었는지 확인합니다.

```
# acctadm net
```

#### 예 8-16 시스템에서 네트워크 정산 설정

이 예제에서는 시스템에서 기록 트래픽 정보를 캡처하고 표시하기 위해 네트워크 계정을 구성하는 방법을 보여줍니다.

다음과 같이 모든 계정 유형의 상태를 확인합니다.

```
# acctadm
Task accounting: inactive
Task accounting file: none
Tracked task resources: none
Untracked task resources: extended
Process accounting: inactive
Process accounting file: none
Tracked process resources: none
Untracked process resources: extended,host
Flow accounting: inactive
Flow accounting file: none
Tracked flow resources: none
Untracked flow resources: extended
Net accounting: inactive
```



```

Network accounting file: none
Tracked Network resources: none
Untracked Network resources: extended

```

네트워크 계정이 활성화되지 않았다고 출력 결과에 표시됩니다. 따라서 확장 네트워크 계정을 사용으로 설정해야 합니다.

```

# acctadm -e extended -f /var/log/net.log net
# acctadm net
    Net accounting: active
    Net accounting file: /var/log/net.log
    Tracked net resources: extended
    Untracked net resources: none

```

## 네트워크 트래픽에 대한 기록 통계 표시

네트워크 계정을 사용으로 설정한 후 `dlstat` 및 `flowstat` 명령을 사용하여 로그 파일에서 정보를 추출할 수 있습니다.

네트워크의 기록 데이터를 표시하려면 먼저 네트워크에 대해 확장 계정을 사용으로 설정해야 합니다. 또한 플로우의 트래픽에 대한 기록 데이터를 표시하려면 먼저 [“플로우를 사용하여 네트워크 리소스 관리” \[164\]](#)에 설명된 대로 시스템에서 플로우를 구성해야 합니다.

## 데이터 링크에 대한 기록 네트워크 트래픽 통계 표시

다음 명령 구문을 사용하여 데이터 링크에 대한 기록 네트워크 트래픽 통계를 표시할 수 있습니다.

```

# dlstat show-link -h [-a] -f filename [-d date] [-F format] [-s start-time] [-e end-time]
  [link]

```

- h                    데이터 링크의 수신 및 송신 패킷별로 리소스 사용에 대한 기록 정보의 요약을 표시합니다.
- a                    데이터 캡처 후에 이미 삭제된 데이터 링크를 포함하여 모든 데이터 링크의 리소스 사용을 표시합니다.
- f *filename*        `acctadm` 명령을 사용하여 네트워크 계정을 사용으로 설정할 때 정의된 로그 파일을 지정합니다.
- d *date*            지정된 날짜에 대해 기록된 정보를 표시합니다.
- F *format*          분석용으로 구성할 수 있는 특정 형식으로 데이터를 표시합니다. 현재 지원되는 유일한 형식은 `gnuplot`입니다.

- s *start-time*      네트워크 통계에 대해 기록된 정보를 표시할 시작 시간을 지정합니다. MM/DD/YYYY, hh:mm:ss 형식을 사용합니다. hour(hh)는 24시간제 표기법을 사용해야 합니다. 날짜를 포함하지 않으면 시간 범위가 현재 날짜로 지정된 데이터가 표시됩니다.
- e *end-time*      네트워크 통계에 대해 기록된 정보를 표시할 종료 시간을 지정합니다. MM/DD/YYYY, hh:mm:ss 형식을 사용합니다. hour(hh)는 24시간제 표기법을 사용해야 합니다. 날짜를 포함하지 않으면 시간 범위가 현재 날짜로 지정된 데이터가 표시됩니다.
- link*                지정한 데이터 링크에 대한 기록 데이터를 표시합니다. 이 옵션을 사용하지 않으면 구성된 모든 데이터 링크에 대한 기록 네트워크 데이터가 표시됩니다.

**예 8-17**            데이터 링크의 리소스 사용에 대한 기록 통계 표시

이 예에서는 시스템에 있는 모든 데이터 링크에 대한 리소스 사용과 네트워크 트래픽에 대한 기록 통계를 표시합니다.

```
# dlstat show-link -h -f /var/log/net.log
LINK DURATION IPKTS  RBYTES  OPKTS  OBYTES  BANDWIDTH
net0   80      1031   546908   0       0       2.44 Mbps
net1  100     2045  235977   0       0       9.67 Mbps
```

## 플로우에 대한 기록 네트워크 트래픽 통계 표시

다음 명령 구문을 사용하여 플로우에 대한 기록 네트워크 트래픽 통계를 표시할 수 있습니다.

- ```
# flowstat -h [-a] -f filename [-d date] [-F format] [-s start-time] [-e end-time] [flow]
```
- h                    구성된 플로우의 수신 및 송신 패킷별로 리소스 사용에 대한 기록 정보의 요약을 표시합니다.
  - a                    데이터 캡처 후에 이미 삭제된 데이터 링크를 포함하여 모든 구성된 플로우의 리소스 사용을 표시합니다.
  - f *filename*        acctadm 명령을 사용하여 네트워크 계정을 사용으로 설정할 때 정의된 로그 파일을 지정합니다.
  - d                    지정된 날짜에 대해 기록된 정보를 표시합니다.
  - F *format*          데이터를 특정 형식으로 표시합니다. 현재 지원되는 유일한 형식은 gnuplot입니다.
  - s *start-time*      네트워크 통계에 대해 기록된 정보를 표시할 시작 시간을 지정합니다. MM/DD/YYYY, hh:mm:ss 형식을 사용합니다. hour(hh)는 24시간제 표기법

을 사용해야 합니다. 날짜를 포함하지 않으면 시간 범위가 현재 날짜로 지정된 데이터가 표시됩니다.

|                          |                                                                                                                                                 |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-e end-time</code> | 네트워크 통계에 대해 기록된 정보를 표시할 종료 시간을 지정합니다. MM/DD/YYYY, hh:mm:ss 형식을 사용합니다. hour(hh)는 24시간제 표기법을 사용해야 합니다. 날짜를 포함하지 않으면 시간 범위가 현재 날짜로 지정된 데이터가 표시됩니다. |
| <code>flow</code>        | 지정한 플로우에 대한 기록 데이터를 표시합니다. 이 옵션을 사용하지 않으면 구성된 모든 플로우에 대한 기록 네트워크 데이터가 표시됩니다.                                                                    |

#### 예 8-18 플로우의 리소스 사용에 대한 기록 통계 표시

다음 예제에서는 시스템에서 플로우의 트래픽별 리소스 사용에 대한 기록 통계를 표시합니다.

```
# flowstat -h -f /var/log/net.log
FLOW      DURATION  IPACKETS  RBYTES    OPACKETS  OBYTES    BANDWIDTH
flowtcp   100      1031      546908    0          0          43.76Kbps
flowudp   0         0         0          0          0          0.00Mbps
```

다음 예제에서는 지정된 날짜 및 시간 범위에서 flowtcp의 트래픽별 리소스 사용에 대한 기록 통계를 표시합니다.

```
# flowstat -h -s 02/19/2008,10:39:06 -e 02/19/2008,10:40:06 \
-f /var/log/net.log flowtcp
FLOW      START      END          RBYTES  OBYTES    BANDWIDTH
flowtcp   10:39:06  10:39:26    1546    6539      3.23 Kbps
flowtcp   10:39:26  10:39:46    3586    9922      5.40 Kbps
flowtcp   10:39:46  10:40:06    240     216       182.40 bps
flowtcp   10:40:06  10:40:26    0        0          0.00 bps
```

다음 예제에서는 지정된 날짜 및 시간 범위에서 flowtcp의 트래픽별 리소스 사용에 대한 기록 통계를 gnuplot 형식으로 표시합니다.

```
# flowstat -h -s 02/19/2008,10:39:06 -e 02/19/2008,10:40:06 \
-F gnuplot -f /var/log/net.log flowtcp
# Time tcp-flow
10:39:06 3.23
10:39:26 5.40
10:39:46 0.18
10:40:06 0.00
```



## 색인

---

### 번호와 기호

- acctadm 명령, 183
- CPU, 19, 159
  - CPU 할당, 163
- CPU 관리, 159
- CPU 풀, 159
  - 기본 풀, 160
  - 데이터 링크, 19
  - 링크에 지정, 162
  - 작업, 159
- DCBX 프로토콜, 70
- dladm 명령, 97, 100
  - create-etherstub, 22, 32
  - create-vlan, 151
  - create-vnic, 21, 100, 150, 162
  - create-vxlan, 58
  - delete-vnic, 43
  - delete-vxlan, 62
  - modify-vnic, 39, 40, 42
  - set-linkprop, 24, 150
  - show-ether, 81
  - show-link, 62
  - show-linkprop, 46, 151, 153
  - show-phys, 49, 153, 153
  - show-vnic, 35, 36, 49, 100
  - show-vxlan, 58, 62
- dlstat 명령, 22, 175, 175
  - show-aggr, 180
  - show-bridge, 180
  - show-link, 185
  - show-phys, 176
- dlstat show-ether 명령, 81
- DS 필드, 164
- ECP 상태 및 통계 표시 예, 82
- ECP(에지 제어 프로토콜)
  - ECP 상태 및 통계 표시, 82
  - ECP(Edge Control Protocol), 79
- etherstub, 14, 15
  - 개인 가상 네트워크, 32
  - 구성, 22
  - 만들기, 22
- EVB, 18 살퍼볼 내용 에지 가상 브리징
- EVB 관련 데이터 링크 등록 정보 설정 예, 84
- EVS
  - 관리 작업, 105
  - 보안 요구 사항, 103
  - 영역 작업, 102
  - 이름 공간 관리, 93
  - 이점, 91
  - 필수 패키지, 102
- EVS 관리자
  - 설명, 95
- EVS 구성 요소, 94
  - EVS 관리자, 94
  - EVS 노드, 94
  - EVS 컨트롤러, 94
  - EVS 클라이언트, 94
- EVS 노드
  - 정의, 97
- EVS 컨트롤러, 95
  - 계획, 107
  - 구성, 110
  - 만들기 및 관리, 107
  - 설정, 115
  - 표시, 115
  - 필수 패키지, 102, 107
- EVS 컨트롤러 등록 정보, 95
  - 설정, 110, 115
  - 표시, 110, 115
- EVS 클라이언트, 97
- evsadm

- EVS 컨트롤러 등록 정보를 관리하기 위한 하위 명령, 99
- EVS 클라이언트 등록 정보를 관리하기 위한 하위 명령, 99
- IPnet을 관리하기 위한 하위 명령, 98
- VPort를 관리하기 위한 하위 명령, 99
- 탄력적 가상 스위치를 관리하기 위한 하위 명령, 98
- evsadm 명령, 97, 98
  - add-ipnet, 98, 119
  - add-vport, 99, 119
  - create-evs, 98, 119
  - delete-evs, 98, 133
  - remove-ipnet, 98, 127
  - remove-vport, 99, 132
  - reset-vport, 99, 132
  - set-controlprop, 99, 115
  - set-evsprop, 98, 99, 125
  - set-prop, 95, 95, 99, 115
  - set-vportprop, 129
  - show-controlprop, 100, 115
  - show-evs, 98, 123
  - show-evsprop, 98, 125
  - show-ipnet, 99, 127
  - show-prop, 99, 115
  - show-vport, 99, 131
  - show-vportprop, 99, 130
- evsstat 명령, 134
  - 탄력적 가상 스위치에 대한 네트워크 트래픽 통계 표시, 100
- flowadm 명령, 164, 167
  - add-flow, 165, 166, 170
  - help, 165
  - set-flowprop, 165, 166, 170
  - show-flowprop, 167, 170
- flowstat 명령, 22, 175, 181
- GVRP(GARP VLAN 등록 프로토콜), 24
- IP 네트워크 살펴볼 내용 IPnet
- IP 주소, 92
- ipadm 명령, 169
  - create-addr, 23
  - create-ip, 23
  - IP 정보 표시, 169
  - show-addr, 58
- IPnet
  - 관리, 123, 127
  - 제거, 127
  - 탄력적 가상 스위치에 추가, 119
  - 표시, 127
- IPoIB
  - 영역에서 임시로 만들기, 31
- LLDP, 70
  - LLDP를 사용하여 VM 간 통신 관리, 78
  - LLDP를 사용하여 VM 간 통신 관리, 78
- MAC 주소, VNIC, 14
- MAC 클라이언트, 150
  - 구성, 154
  - 기본, 153
  - 링 할당, 154
  - 소프트웨어 기반, 150
  - 하드웨어 기반, 150
- NIC 링, 19, 19
  - 살펴볼 다른 내용 링 할당
  - 수신 및 전송, 150
- Oracle Solaris 커널 영역, 102
- Oracle Solaris 탄력적 가상 스위치, 87, 89
  - 살펴볼 다른 내용 EVS
- Oracle VSI 관리자, 73, 79
- oracle\_v1 인코딩, 73
  - 정의, 79
- ORACLE\_VSIMGR\_V1, 79
- SCTP, 164
- SLA
  - 가상 스위치, 88
- SR-IOV
  - VNIC 사용, 46
  - 가상 기능 살펴볼 내용 VF
  - 데이터 링크 검사, 46
  - 데이터 링크에 대해 사용, 46
- SSH 인증
  - 설정, 110
- TCP, 164
- UDP, 164
- VDP 살펴볼 내용 VSI 검색 및 구성 프로토콜
- VDP 통계, 표시 명령, 81
- VDP TLV, 79
- VDP를 사용하여 VNIC 정보 교환, 79
- VF 정보
  - 표시, 49
  - VF 표시, 49
- VF VNIC
  - 마이그레이션, 48

- 만들기, 47
- 표시, 49
- VF VNIC 만들기, 47
- Virtual eXtensible Local Area Network 살펴볼 내용 VXLAN
- Virtual Extensible Local Area Network 살펴볼 내용 VXLAN
- VLAN, 89, 91
  - VNIC로, 24
  - VNIC의 VLAN ID 수정, 38
  - 영역에서 임시로 만들기, 31
- VM의 통신 제어, 74
- VNI, 52
- VNIC, 13, 14, 14
  - CPU 플 리소스 지정, 162
  - MAC 주소 수정, 40
  - SR-IOV 사용, 46
  - VLAN ID 사용, 24
  - VLAN ID 수정, 38
  - VLAN으로, 21
  - VNIC IP 인터페이스 만들기, 23
  - 개인 가상 네트워크, 32
  - 관리, 34
  - 구성, 22, 26
  - 기본 링크 변경, 41
  - 등록 정보 설정, 150
  - 마이그레이션, 41
  - 만들기, 21
  - 삭제, 43
  - 시스템 생성, 22
  - 여러 MAC 주소 표시, 36
  - 영역에 사용, 28
  - 영역에 지정, 29
  - 영역에서 구성, 29
  - 영역에서 임시로 만들기, 30
  - 정보 표시, 35
  - 정적 IP 주소 지정, 23
- VNIC 삭제, 43
- VNIC anet 리소스, 97, 100
- VNIC의 EVB 관련 등록 정보 예, 85
- VPort
  - 관리, 123, 129
  - 등록 정보 설정, 129
  - 등록 정보 표시, 130
  - 재설정, 132
  - 제거, 132
  - 탄력적 가상 스위치에 추가, 119
  - 표시, 131
- VPort 등록 정보, 92
  - IP 주소, 92
  - MAC 주소, 92
  - SLA
    - 서비스 클래스, 92
    - 우선 순위, 92
    - 최대 대역폭, 92
- VSI 살펴볼 내용 가상 스테이션 인스턴스
- VSI 관리자, 79
  - oracle\_v1, 79
- VSI 관리자 ID
  - ORACLE\_VSIMGR\_V1, 79
- VSI 버전, 79
- VSI 식별자, 79
- VSI 유형 ID, 79
- VSI 프로파일, 79
- VSI VDP(검색 및 구성 프로토콜)
  - VDP TLV, 79
  - VDP 통계 표시, 81
  - 이더넷 링크에 대한 VDP 상태, 81
- VXLAN, 18, 51, 89, 91, 101
  - anet 리소스, 55
  - lower-link, 62
  - VNIC, 55
  - VXLAN 구성, 58
  - VXLAN 끝점, 52
  - VXLAN 생성 예, 60
  - VXLAN 세그먼트, 55
  - 개요, 51
  - 구성 계획, 57
  - 삭제, 62
  - 영역과 함께 사용, 55
  - 영역에 대해 구성, 58
  - 영역에 지정, 62
  - 영역의 anet에 VXLAN을 지정하는 예, 63
  - 요구 사항, 57
  - 이름 지정 규칙, 52
  - 이점, 52
  - 토폴로지, 53
  - 표시, 62
- VXLAN 삭제, 62
- VXLAN 세그먼트 ID, 52, 57 살펴볼 내용 VNI
- VXLAN 표시, 62
- VXLAN을 통해 만든 VNIC를 영역에 지정하는 예, 61

zoneadm 명령, 27  
zonecfg 명령, 27, 97, 162, 170

## ㄱ

가상 네트워크, 13, 16, 26, 52  
  살펴볼 다른 내용 VNIC  
  etherstub, 14  
  etherstub 구성, 22  
  VNIC, 14  
  VNIC 구성, 22  
  가상 스위치, 14  
  개인, 13  
  구성 요소, 14  
  구축, 26  
  구현, 18  
  배타적 IP 유형 영역, 29  
  영역, 14, 26  
  영역 구성, 26  
  영역 재구성, 28  
  외부 및 내부, 13  
가상 네트워크 인터페이스 카드 살펴볼 내용 VNIC  
가상 네트워크의 구성 요소  
  구성, 21  
가상 링크 상태  
  표시, 37  
가상 머신, 88  
가상 스위치, 14, 14, 88  
가상 포트, 88 살펴볼 내용 VPort  
가상 NIC 살펴볼 내용 VNIC  
개인 가상 네트워크, 13, 15  
  etherstub으로 구성, 32  
계층 2 네트워크, 52  
관리  
  탄력적 가상 스위치, evsadm 하위 명령, 98  
  EVS 컨트롤러 등록 정보, evsadm 하위 명령, 99  
  EVS 클라이언트 등록 정보, evsadm 하위 명령, 99  
  IPnet, evsadm 하위 명령, 98  
  VPort, evsadm 하위 명령, 99  
구성  
  VXLAN, 58  
기록 네트워크 트래픽 통계 표시  
  데이터 링크, 185  
  플로우, 185  
기본 클라이언트, 151, 153

## ㄴ

내부 가상 네트워크, 13  
네트워크 가상화, 13  
  etherstub, 15  
  가상 네트워크, 13  
  가상 스위치, 14  
  구현을 위한 dladm 하위 명령, 150  
  네트워크 리소스 관리, 13  
  데이터 센터 통합, 19  
  영역, 16  
네트워크 계정, 183, 184  
네트워크 계정 구성, 183  
네트워크 데이터 링크 관리  
  EVB, 18  
네트워크 레인, 19  
네트워크 리소스  
  CPU, 19  
  CPU 풀, 19  
  NIC 링, 19  
  대역폭, 19  
  우선 순위, 19  
네트워크 리소스 관리, 13, 19, 149, 149  
  CPU, 159  
  CPU 풀, 159  
  NIC 링, 150  
  구현을 위한 dladm 하위 명령, 150  
  데이터 링크 등록 정보 사용, 19, 20, 149  
  데이터 링크 및 플로우 등록 정보 설정, 167  
  이점, 20  
  플로우 사용, 19, 20, 164  
네트워크 통계, 175  
  기록 트래픽 정보, 184  
  네트워크 사용 모니터링, 173  
네트워크 트래픽 통계  
  네트워크 장치에 대한 표시, 176  
  데이터 링크에 대한 표시, 179  
  링크 집계에 대한 표시, 180  
  링크에 대한 표시, 175  
  브리지에 대한 표시, 180  
  플로우에 대한 표시, 181

## ㄷ

단일 루트 I/O 가상화 살펴볼 내용 SR-IOV  
대역폭  
  데이터 링크에 설정, 19



- 플로우에 대해 설정, 166
- 플로우에 설정, 19
- 데이터 링크
  - CPU 풀 할당, 19
  - CPU 할당, 19
  - EVB 등록 정보, 82
  - 기록 네트워크 트래픽 통계 표시, 185
  - 대역폭 할당, 19
  - 리소스 제어 등록 정보, 20, 150
- 데이터 링크 등록 정보
  - cpu, 149
  - maxbw, 149
  - pool, 149
  - rxrings 및 txrings, 149
- 데이터 링크에 CPU 할당, 163
- 데이터 센터 통합, 19
  
- ㄹ**
- 로컬 응용 프로그램 포트 번호, 164
- 로컬 IP 주소, 164
- 링 관리
  - 명령, 151
- 링 할당, 150, 151
  - VLAN, 151
  - 링 사용 및 링 지정, 153
  
- ㄴ**
- 마이그레이션
  - VF VNIC, 48
  - VNIC, 41
- 만들기
  - IP 인터페이스, 23
- 명령
  - EVS 관리, 97
  - 가상 네트워크의 구성 요소 구성, 21
  - 네트워크 트래픽 통계 모니터링, 175
  - 데이터 링크에서 리소스 할당, 150
  - 링 관리, 151
  - 플로우에서 리소스 할당, 165
- 모니터링
  - 네트워크 사용, 173
  - 네트워크 트래픽 통계
    - 명령, 175
- 물리적 링크 상태
  - 표시, 37
- 물리적 링크의 EVB 관련 데이터 링크 등록 정보 표시
  - 예, 85
  
- ㄷ**
- 반사 중계, 14, 69
- 변경
  - 기본 EVB 구성, 82
  
- ㄹ**
- 사용으로 설정
  - VM 간에 외부 스위치를 통해 통신, 74
- 서브넷, 92
- 서비스 단계 계약 살펴볼 내용 SLA
- 서비스 품질(QoS), 19, 149
- 설치
  - 에지 가상 브리징, 73
  - 소프트웨어 기반 클라이언트, 150, 151
  - 수신 링 살펴볼 내용 Rx 링
  - 시스템 생성 VNIC, 22
    - 표시, 36
  
- ㅇ**
- 액세스 제어 목록(ACL), 70
- 업링크 포트
  - 정의, 90
- 에지 가상 브리징, 69 살펴볼 내용 EVB
  - EVB를 사용하는 경우와 사용하지 않는 경우의 서버 효율성, 72
  - EVB를 통해 네트워크 및 서버 효율성을 향상시키는 방법, 71
  - LLDP를 사용하여 VM 간 통신 관리, 78
  - Oracle VSI 관리자, 79
  - oracle\_v1, 79
  - oracle\_v1 인코딩, 79
  - VDP 사용, 79
  - VDP 상태 표시 예, 81
  - VDP 통계 표시 예, 81
  - VDP 프로토콜, 79
  - VM의 통신 제어, 74
  - VNIC 정보 교환, 79
  - VNIC 정보를 교환하는 방법, 80
  - VSI 관리자, 79

- VSI 관리자 ID, 79
  - VSI 버전, 79, 82
  - VSI 식별자, 79
  - VSI 유형 ID, 79, 82
  - VSI 프로파일, 79
  - 개요, 69
  - 구성 요소, 79
  - 기본 EVB 구성 변경, 82
  - 데이터 링크 등록 정보, 82
  - 링크 등록 정보 표시 예, 81
  - 반사 중계, 69, 70
  - 설치, 73
  - 액세스 제어 목록, 70
  - 외부 스위치를 통해 통신하도록 VM을 사용으로 설정, 74
  - 이더넷 링크에 대한 VDP 상태, 81
  - 자동 가상 포트 구성, 70
  - 영역, 14, 97
    - VNIC 지정, 29
    - VXLAN 사용, 55
    - VXLAN 지정, 62
    - 가상 네트워크에 대해 재구성, 28
    - 네트워크 가상화용으로 구성, 26
    - 임시로 VNIC 만들기, 30
  - 영역에 연결된 VNIC 삭제, 45
  - 영역에서 임시로 VNIC 만들기, 31
  - 영역에서 임시로 VNIC, VLAN 및 IPoIB 분할 영역 만들기, 31
  - 외부 가상 네트워크, 13
  - 우선 순위
    - 데이터 링크, 19
    - 플로우, 19, 165
  - 원격 응용 프로그램 포트 번호, 164
  - 원격 IP 주소, 164
  - 의사 이더넷 NIC 살펴볼 내용 etherstub
  - 임시로 VNIC 만들기
    - 영역에서, 30
- ㅈ**
- 자동으로 생성된 VXLAN 데이터 링크, 101
  - 전송 링 살펴볼 내용 Tx 링
- ㅊ**
- 탄력적 가상 스위치, 88, 101
- IP 주소 블록 연결, 90
  - VLAN을 기반으로 구성, 137
  - VNIC anet 리소스 만들기, 122
  - VNIC 만들기, 121
  - 가상 포트 연결, 90
  - 개요, 87
  - 계획, 106
  - 관리, 123
  - 등록 정보 설정, 125
  - 등록 정보 표시, 125
  - 리소스
    - IP 네트워크, 92
    - 가상 포트, 92
  - 만들기, 119
  - 모니터링, 134
  - 삭제, 133
  - 테넌트에 대한 VXLAN을 기반으로 구성, 142
  - 표시, 119, 123
  - 필수 패키지, 117
- 테넌트**
- 정의, 93
- ㅍ**
- 표시**
- EVB 관련 데이터 링크 등록 정보, 85
  - EVS 컨트롤러 등록 정보, 115
  - IPnet, 127
  - VDP와 ECP의 상태 및 통계, 81
  - VPort, 131
  - 가상 링크 상태, 37
  - 물리적 링크 상태, 37
  - 탄력적 가상 스위치 정보, 117, 119, 123
- 프로토콜**
- DCBX, 70
  - ECP, 79
  - VDP, 79
- 플로우**, 20, 22, 164
- 구성, 165
  - 기록 네트워크 트래픽 통계 표시, 186
  - 대역폭 설정, 165, 166
  - 등록 정보 설정, 165
  - 만들기, 165, 166
  - 속성 기반, 164
  - 우선 순위 설정, 165
  - 정보 표시, 167

---

플로우 구성, 165  
명령, 165  
플로우 제어, 164  
플로우를 구성하는 속성  
DS 필드, 164  
로컬 IP 주소, 164  
원격 IP 주소, 164  
응용 프로그램 포트 번호 및 IP 주소를 포함하는 전송 프로토콜, 164  
응용 프로그램 포트 번호를 포함한 전송 프로토콜, 164  
전송 프로토콜, 164

## ㅎ

하드웨어 기반 클라이언트, 150  
하드웨어 링, 151  
확장 계정 기능, 183  
링크 및 플로우에 대한 네트워크 계정, 184  
작업 계정, 184  
프로세스 계정, 184  
플로우 계정, 184

