

Oracle® Solaris에서 OpenStack(Havana) 설 치 및 구성

ORACLE®

부품 번호: E56872
2016년 5월

부품 번호: E56872

Copyright © 2014, 2016, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

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

설명서 접근성

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

오라클 고객지원센터 액세스

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

목차

이 설명서 사용	7
1 이 설명서에 대한 정보	9
이 설명서 사용	9
OpenStack 용어집	10
2 평가 구성 설치	11
OpenStack 통합 아카이브 배치	11
이미지 파일 다운로드	12
단일 시스템 설치	12
탄력적 가상 스위치 구성	17
OpenStack 대시보드 사용	18
▼ OpenStack 대시보드에 액세스하는 방법	18
대시보드 살펴보기	19
VM 인스턴스 만들기 및 부팅	22
3 다중 노드 Havana OpenStack 구성을 위해 여러 시스템에 걸쳐 설치	29
3노드 아키텍처 개요	29
컨트롤러 노드 구성	32
▼ 컨트롤러 노드를 구성하는 방법	33
Network Time Protocol 설치	33
MySQL 설치	34
Keystone 설치	36
Heat 설치 및 구성	37
Cinder 설치	38
Glance 설치	42
컨트롤러 노드에서 Neutron 설치 및 구성	43
Nova 설치	44
▼ Horizon을 구성하는 방법	45
Compute 노드 구성	46

▼ Compute 노드를 구성하는 방법	47
네트워크 노드 구성	48
▼ 네트워크 노드를 구성하는 방법	50
Neutron L3 에이전트 구성	52
4 가상 시스템 인스턴스 만들기	63
Flavor 관리	63
Flavor에 대한 정보 표시	63
Flavor 사양 수정	65
이미지 관리	66
이미지에 대한 정보 표시	66
이미지 만들기	67
이미지 저장소에 이미지 추가	68
VM 인스턴스 만들기	69
▼ 명령줄 인터페이스를 사용하여 VM 인스턴스를 만드는 방법	69
5 OpenStack 문제 해결	71
알려진 제한 사항	71
로그 파일 검사	73
문제 조사 및 해결	74
OpenStack 설치 및 구성	75
VM 인스턴스 설치 및 구성	75
색인	79

이 설명서 사용

- **개요** - Oracle Solaris 11.2 시스템에서 OpenStack을 설치하고 OpenStack 가상 시스템을 배치하는 방법에 대해 설명합니다.
- **대상** - 대규모 설치 시스템 관리자
- **필요한 지식** - Solaris 네트워크 및 대규모 시스템 관리 OpenStack에 익숙한 것이 좋습니다.

제품 설명서 라이브러리

이 제품과 관련 제품들에 대한 설명서 및 리소스는 <http://www.oracle.com/pls/topic/lookup?ctx=E69402>에서 사용할 수 있습니다.

피드백

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

이 설명서에 대한 정보

이 장에서는 현재 문서의 구조를 설명하고 내용을 탐색해 봅니다. OpenStack 및 Oracle Solaris 기술 개요는 [Overview: OpenStack Components and Oracle Solaris Technologies](#)을 참조하십시오. Oracle Solaris에 OpenStack을 설치하기 위한 계획 지침 및 요구 사항은 [Planning for an OpenStack Configuration](#)을 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

- “이 설명서 사용” [9]
- “OpenStack 용어집” [10]

이 설명서 사용

이 설명서에서는 주로 Solaris와 다른 플랫폼에서 차이가 있는 OpenStack 정보를 다룹니다. Solaris와 다른 플랫폼에서 작동 방식이 동일한 기능 및 Solaris와 다른 플랫폼에서 수행 방법이 동일한 작업은 일반적으로 이 설명서에서 다루지지 않습니다.

2장. 평가 구성 설치에서는 평가용으로 단일 Solaris 시스템에 OpenStack을 빠르게 설치하는 방법에 대해 설명합니다. 통합 아카이브로 전체 설치가 제공되며 대부분의 구성이 미리 완료되어 있습니다.

3장. 다중 노드 Havana OpenStack 구성을 위해 여러 시스템에 걸쳐 설치에서는 세 Solaris 시스템(컨트롤러 노드, 네트워크 노드, 계산 노드)에서 OpenStack을 설치 및 구성하는 방법에 대해 설명합니다.

4장. 가상 시스템 인스턴스 만들기에서는 VM 인스턴스를 만들고 사용하는 것과 관련된 Solaris 특정 정보를 제공합니다. 테넌트 및 관련 사용자를 만드는 등의 작업은 설명되지 않습니다. 해당 작업은 Solaris와 다른 플랫폼에서 동일하기 때문입니다.

이 Oracle Solaris 릴리스의 기반이 되는 OpenStack에 따라 OpenStack Havana에 대한 일반적인 정보는 [OpenStack 설명서 사이트](#)에서 다음 자료 등을 참조하십시오.

- *OpenStack Training Guides*
- *End User Guide*(OpenStack command-line interface cheat sheet 포함)
- *Admin User Guide*

- *Command-Line Interface Reference*
- *Configuration Reference*
- *Cloud Administrator Guide*

OpenStack 설치에 영향을 끼칠 수 있는 추가 정보는 “알려진 제한 사항” [71]을 참조하십시오.

Solaris에 대한 자세한 내용은 [Oracle Solaris 11.2 Information Library](#)를 참조하십시오. Solaris의 OpenStack에 대한 자세한 내용은 [OpenStack for Oracle Solaris 11](#)을 참조하십시오.

OpenStack 용어집

OpenStack 커뮤니티에서 다른 용어가 동일한 의미로 사용되는 경우가 있습니다. 예를 들어, 클라우드의 가상 시스템을 서버, 인스턴스 또는 Compute VM이라고 할 수 있습니다. OpenStack 기능 요소(예: Compute 또는 네트워킹)를 모듈, 구성 요소 또는 서비스라고 할 수도 있습니다. OpenStack에서는 프로젝트와 테넌트라는 용어를 서로 바꿔서 사용할 수 있습니다. 이 문서에서는 다음 용어가 사용됩니다.

서비스	Nova 또는 Compute 서비스와 같은 OpenStack 서비스입니다.
SMF 서비스	<code>svc:/application/openstack/nova/nova-compute:default</code> 와 같은 Solaris 서비스입니다. "서비스 사용"과 같은 구문은 SMF 서비스를 참조하십시오.
노드	OpenStack 서비스를 호스트하는 시스템입니다. 예를 들어, 컨트롤러 노드는 Keystone, Glance 및 Horizon 서비스를 호스트합니다.
프로젝트	Oracle Solaris 영역에서 프로젝트는 관련 작업에 대한 네트워크 차원의 관리 식별자입니다. 그러나 본 문서에서 이 용어는 OpenStack 정의에 따라 Compute 모듈 내의 논리적 사용자 그룹이라는 의미로 사용됩니다. 프로젝트는 VM 이미지에 대한 액세스와 쿼터를 정의합니다.
VM 인스턴스	클라우드의 가상 시스템입니다. VM 인스턴스는 실행 중인 VM 또는 일시 중지됨과 같은 알려진 상태의 VM으로, 하드웨어 서버처럼 사용될 수 있습니다.
영역	운영체제를 가상화하고 응용 프로그램을 실행할 격리된 보안 환경을 제공하기 위한 Oracle Solaris 기술입니다. 이 용어는 가상화된 환경 자체를 가리킬 수도 있습니다. Oracle Solaris에서 OpenStack의 Compute 가상화는 영역 기술을 기반으로 합니다.

OpenStack 용어에 대한 추가 설명은 <http://docs.openstack.org/glossary/content/glossary.html>을 참조하십시오.

◆◆◆ 2 장

평가 구성 설치

평가를 위해 OpenStack을 단일 Oracle Solaris 시스템에 설치할 수 있습니다. 이 유형의 OpenStack 설치를 단일 노드 설치라고도 합니다.

다중 노드 OpenStack 설치에 대한 설명은 [3장. 다중 노드 Havana OpenStack 구성을 위해 여러 시스템에 걸쳐 설치를 참조하십시오.](#)

이 장에서는 통합 아카이브를 사용하여 단일 노드를 설치하고 브라우저 인터페이스를 통해 간편하게 VM 인스턴스를 만드는 방법에 대해 설명합니다. 이 장에서는 다음 내용을 다룹니다.

- “OpenStack 통합 아카이브 배치” [11]
- “OpenStack 대시보드 사용” [18]

OpenStack 통합 아카이브 배치

[Unified Archives 다운로드 페이지](#)에서는 Oracle Solaris 아카이브만 다운로드하거나 OpenStack 아카이브가 포함된 Oracle Solaris를 다운로드할 수 있습니다.

OpenStack이 포함된 Oracle Solaris 아카이브는 운영체제와 모든 OpenStack 서비스를 설치합니다.

Oracle Solaris OpenStack 통합 아카이브는 다음 기능을 제공합니다.

- OpenStack 서비스: 사용자가 수행해야 할 구성 작업을 줄일 수 있도록 대부분 사전 구성되어 있습니다.
- 스크립트: 수행해야 할 추가 구성을 자동화합니다.
- Solaris 시스템 템플릿
- Glance 이미지 저장소: 2개의 이미지(비전역 영역 및 커널 영역)가 사전 로드되어 있습니다.

베어 메탈 시스템 또는 커널 영역에 OpenStack 통합 아카이브를 배치할 수 있습니다. 옵션은 [“단일 시스템 설치” \[12\]](#)를 참조하십시오.

이미지 파일 다운로드

다음 절차에 따라 Oracle Solaris OS와 OpenStack을 제공하는 단일 이미지 파일을 다운로드할 수 있습니다.

▼ 이미지 파일을 다운로드하는 방법

1. 인터넷 브라우저에서 [Unified Archives](#) 다운로드 페이지로 이동합니다.
2. 라이선스 계약을 읽고 **Accept(동의)** 버튼을 누릅니다.
3. 수행하려는 설치 유형에 해당하는 파일을 다운로드합니다.
 - **사용 중인 시스템 아키텍처에 적절한 통합 아카이브 파일을 다운로드합니다.**
이 파일을 사용하여 다음 유형의 설치 중 하나를 수행합니다.
 - 이 .uar 파일을 `zoneadm install` 명령에 대한 인수로 사용하여 커널 영역을 직접 설치합니다.
 - AI 설치 매니페스트에서 이 .uar 파일을 참조하여 AI 설치 서비스를 통해 AI 설치를 수행합니다.
 - 이 .uar 파일을 사용하여 AI 부팅 가능 매체를 만듭니다.
 - **사용 중인 시스템 아키텍처에 적절한 USB 부팅 이미지 파일을 다운로드합니다.**
이 usb 파일을 사용하여 부팅 가능 매체에서 통합 아카이브를 설치합니다.
4. 다운로드한 파일의 무결성을 확인합니다.
MD5 체크섬 링크를 누릅니다. 다음 `digest` 명령을 실행하여 출력을 체크섬 파일의 해당 체크섬과 비교합니다.

```
$ digest -a md5 file
```

단일 시스템 설치

이 절에서는 단일 시스템 OpenStack 평가 구성을 설치하는 방법에 대해 설명합니다. 설명되는 처음 세 가지 방법에서는 베어 메탈에 설치합니다. 네번째 방법에서는 커널 영역에 설치합니다.

- [다운로드한 USB 파일을 사용하여 설치하는 방법 \[13\]](#)
- [통합 아카이브 파일 및 AI 설치 서비스를 사용하여 설치하는 방법 \[14\]](#)
- [통합 아카이브 파일에서 만들어진 AI 부팅 가능 매체를 사용하여 설치하는 방법 \[15\]](#)

- Havana 통합 아카이브 파일을 사용하여 커널 영역에 설치하는 방법 [16]

▼ 다운로드한 USB 파일을 사용하여 설치하는 방법

이 절차에서는 다운로드한 USB 파일을 사용하여 베어 메탈에 Oracle Solaris 및 OpenStack을 설치하는 방법에 대해 설명합니다. 이 방법의 경우 AI 서버가 필요하지 않으며 베어 메탈 설치에 대해 가장 직접적인 방법입니다.

1. 사용 중인 대상 시스템 아키텍처에 적절한 USB 파일을 다운로드합니다.
이미지 파일을 다운로드하는 방법 [12]을 참조하십시오.
2. USB 플래시 드라이브로 USB 파일을 전송합니다.

- **usbcopy 명령을 사용합니다.**

Oracle Solaris 11.2 시스템에 대한 액세스 권한이 있을 경우 `usbcopy` 유틸리티를 사용합니다. [usbcopy\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

Oracle Solaris 11.2에서 `usbcopy` 명령을 사용해야 합니다. Solaris의 이전 버전에서는 `usbcopy`를 사용할 수 없습니다.

- **dd 명령을 사용합니다.**

Oracle Solaris 11.2 시스템에 대한 액세스 권한이 없는 경우 `dd` 명령을 사용할 수 있습니다.

`dd`를 사용하는 경우 적절한 디스크(플래시 드라이브)를 올바르게 식별해야 합니다.

- **Oracle Solaris 11:**

- a HAL 서비스를 사용 안함으로 설정합니다.

```
# svcadm disable -t hal
```

- b 플래시 드라이브를 넣고 적절한 장치를 찾습니다.

```
# rmformat
```

- c 이미지를 복사합니다.

```
# dd if=/path/image.usb of=/dev/rdsk/device bs=16k
```

- d HAL 서비스를 사용으로 설정합니다.

```
# svcadm enable hal
```

- **Linux:**

- a 플래시 드라이브를 넣고 적절한 장치를 찾습니다.

```
# dmesg | tail
```

- b 이미지를 복사합니다.

```
# dd if=/path/image.usb of=/dev/diskN bs=16k
```

■ MacOSX:

- a 드라이브 /dev/diskN을 식별합니다. 여기서 N은 디스크 번호입니다.

```
# diskutil list  
# diskutil unmountDisk /dev/diskN
```

- b 이미지를 복사합니다.

```
# dd if=/path/image.usb of=/dev/diskN bs=16k
```

- 3. 시스템에 플래시 드라이브를 넣고 USB로 부팅합니다.
SCI(시스템 구성 대화식) 도구가 표시됩니다. SCI 도구가 표시되지 않을 경우 Enter 키를 누르거나 Ctrl-L을 눌러 화면을 다시 표시합니다.

▼ 통합 아카이브 파일 및 AI 설치 서비스를 사용하여 설치하는 방법

이 절차에서는 다운로드한 통합 아카이브 파일 및 AI를 사용하여 베어 메탈에 Oracle Solaris 및 OpenStack을 설치하는 방법에 대해 설명합니다.

1. **사용 중인 대상 시스템 아키텍처에 적절한 통합 아카이브 파일을 다운로드합니다.**
[이미지 파일을 다운로드하는 방법 \[12\]](#)을 참조하십시오.
2. **AI 매니페스트를 만듭니다.**
Oracle Solaris AI 설치 서버에서 AI 설정에 따라 /usr/share/auto_install/manifest/default_archive.xml 파일을 복사하고 수정합니다. ARCHIVE software 섹션에서 다운로드한 .uar 파일의 위치를 지정합니다.
3. **AI 설치 서비스를 설정합니다.**
이전 단계의 AI 매니페스트를 사용하여 AI 설치 서비스를 설정합니다. [Oracle Solaris 11.2 시스템 설치 의 제 III 부, “설치 서버를 사용하여 설치,”](#)를 참조하십시오.
4. **네트워크를 통해 시스템을 부트합니다.**

```
ok boot net -install
```
5. **설치가 완료되면 대상 시스템을 재부트합니다.**

시스템을 구성하도록 SCI 도구가 표시됩니다. SCI 도구가 표시되지 않을 경우 Enter 키를 누르거나 Ctrl-L을 눌러 화면을 다시 표시합니다.

▼ 통합 아카이브 파일에서 만들어진 AI 부팅 가능 매체를 사용하여 설치하는 방법

이 절차에서는 부팅 가능 AI 이미지를 만들어 베어 메탈에 Oracle Solaris 11.2 및 OpenStack을 설치하는 방법에 대해 설명합니다. 부팅 가능 USB 이미지는 다운로드한 통합 아카이브 파일에서 만들어집니다. 이 방법에 대한 자세한 내용은 [Oracle Solaris 11.2 시스템 설치 의 5 장, “매체에서 부트하는 자동 설치”](#)를 참조하십시오.

1. **사용 중인 대상 시스템 아키텍처에 적절한 통합 아카이브 파일을 다운로드합니다.**
[이미지 파일을 다운로드하는 방법 \[12\]](#)을 참조하십시오.

2. **통합 아카이브 파일에서 AI USB를 만듭니다.**

```
# archiveadm create-media -s http://pkg.oracle.com/solaris/release \
-f usb -o /workdir/usb-filename \
/workdir/uar-file
```

여기서 *workdir*은 통합 아카이브 파일을 다운로드한 위치입니다. AI USB 파일이 동일한 디렉토리에 만들어집니다.

3. **USB 플래시 드라이브로 USB 파일을 전송합니다.**

- Oracle Solaris 11.2 시스템으로 USB 파일을 다운로드한 경우 `usbcopy` 명령을 사용합니다.
- Oracle Solaris 11.2 이전 버전의 Oracle Solaris OS 시스템으로 USB 파일을 다운로드한 경우 다음과 같이 `dd` 명령을 사용합니다.

1. HAL 서비스를 사용 안함으로 설정합니다.

```
# svcadm disable -t hal
```

2. 플래시 드라이브를 넣고 적절한 장치를 찾습니다.

```
# rmformat
```

3. 플래시 드라이브로 이미지를 복사합니다.

```
# dd if=/path/image.usb of=/dev/rdisk/device bs=16k
```

4. HAL 서비스를 사용으로 설정합니다.

```
# svcadm enable hal
```

4. **기본 AI 매니페스트를 검토합니다.**

기본 매니페스트를 사용하거나 사용자 정의 매니페스트를 만듭니다. 사용자 정의 매니페스트를 만드는 경우 설치하려는 시스템에서 연결할 수 있는 위치에 사용자 정의 매니페스트를 저장합니다.

5. 시스템에 USB 플래시 드라이브를 넣고 USB로 부팅합니다.

기본 AI 매니페스트를 사용하거나 사용자 정의 매니페스트의 위치를 제공하라는 메시지가 표시됩니다.

시스템을 구성하도록 SCI 도구가 표시됩니다. SCI 도구가 표시되지 않을 경우 Enter 키를 누르거나 Ctrl-L을 눌러 화면을 다시 표시합니다.

▼ Havana 통합 아카이브 파일을 사용하여 커널 영역에 설치하는 방법

이 절차에서는 다운로드한 통합 아카이브 파일을 사용하여 커널 영역에 직접 Oracle Solaris 11.2 및 Havana OpenStack을 설치하는 방법에 대해 설명합니다.

시작하기 전에 커널 영역을 호스트할 시스템이 가상화 요구 사항을 충족하는지 확인하십시오. [Planning for an OpenStack Configuration](#)을 참조하십시오.

1. 사용 중인 대상 시스템 아키텍처에 적절한 통합 아카이브 파일을 다운로드합니다.

[이미지 파일을 다운로드하는 방법 \[12\]](#)을 참조하십시오.

2. 커널 영역을 만듭니다.

```
# zonecfg -z OpenStackKZ create -t SYSsolaris-kz
```

3. 커널 영역을 구성합니다.

커널 영역에 가상 CPU, RAM, 저장소 및 MAC 주소가 충분한지 확인합니다. 커널 영역에 만들어진 비전역 영역은 자동으로 이러한 추가 MAC 주소를 사용할 수 있습니다.

다음 예에서는 가상 CPU 8개, 8GB 한도의 물리적 메모리 및 자동 MAC 주소 지정을 사용하여 영역을 구성합니다. 구성 가능한 리소스에 대한 설명은 [zonecfg\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

```
# zonecfg -z OpenStackKZ
zonecfg:OpenStackKZ> add virtual-cpu
zonecfg:OpenStackKZ:virtual-cpu> set ncpus=8
zonecfg:OpenStackKZ:virtual-cpu> end
zonecfg:OpenStackKZ> select capped-memory
zonecfg:OpenStackKZ:capped-memory> set physical=8g
zonecfg:OpenStackKZ:capped-memory> end
zonecfg:OpenStackKZ> select anet id=0
zonecfg:OpenStackKZ:anet> add mac
zonecfg:OpenStackKZ:anet:mac> set mac-address=auto
zonecfg:OpenStackKZ:anet:mac> end
zonecfg:OpenStackKZ:anet> end
```

```
zonecfg:OpenStackKZ> exit
```

4. 구성을 확인합니다.

```
# zonecfg -z OpenStackKZ info
```

5. 커널 영역을 설치합니다.

다음 예에서는 .uar 파일의 x86 버전을 설치하는 방법을 보여줍니다.

```
# zoneadm -z OpenStackKZ install -a /path/uar-file
```

여기서 *uar-file*은 Havana OpenStack 아카이브가 포함된 Oracle Solaris 11.2입니다.

6. 영역을 부팅합니다.

```
# zoneadm -z OpenStackKZ boot
```

7. 영역 콘솔에 로그인하여 구성을 완료합니다.

```
# zlogin -C OpenStackKZ
```

시스템을 구성하도록 SCI 도구가 표시됩니다. SCI 도구가 표시되지 않을 경우 Enter 키를 누르거나 Ctrl-L을 눌러 화면을 다시 표시합니다.

8. 커널 영역에 IP 주소가 있는지 확인합니다.

통합 아카이브를 사용하는 경우 DHCP 서버가 커널 영역에 IP 주소를 지정해야 합니다. DHCP를 사용하고 있는 경우 커널 영역에 MAC 주소가 지정되었는지 확인합니다. 이전 단계에서 SCI 도구의 네트워크 페이지에서 Automatic(자동)을 선택한 경우 MAC 주소가 지정됩니다. DHCP를 사용하고 있지 않은 경우 커널 영역에 IPv4 주소를 지정해야 합니다.

현재 커널 영역에 대한 시스템 저장소가 없기 때문에 IP 주소가 매우 중요합니다. IP 주소를 통해 커널 영역은 필요한 경우 IPS 패키지 저장소에 연결할 수 있습니다.

탄력적 가상 스위치 구성

설치가 완료되고 시스템이 재부팅되면 EVS(탄력적 가상 스위치)를 구성해야 합니다. EVS를 사용하면 여러 물리적 서버에 걸쳐 가상 스위치를 만들고 구성하며 모니터링할 수 있고 동일한 EVS에 속한 VM을 연결할 수 있습니다. EVS에 대한 자세한 내용은 [Oracle Solaris 11.2의 네트워크 가상화 및 네트워크 리소스 관리의 5 장, “탄력적 가상 스위치 정보”](#)를 참조하십시오.

EVS 구성을 자동화하는 스크립트가 제공됩니다. 다음 명령을 사용하여 스크립트를 실행하십시오.

```
# /usr/demo/openstack/configure_evs.py
```

이 스크립트는 다음 구성을 수행합니다.

- 모든 필수 사용자(예: root, evsuser 및 neutron UNIX 사용자)에 대한 SSH(보안 셸) 키를 만들고 evsuser 사용자에게 대한 /var/user/evsuser/.ssh/authorized_keys 파일에 공개 키를 추가합니다.
- EVS를 구성합니다.
- neutron-server:default 및 neutron-dhcp-agent:default SMF 서비스를 사용하여 설정합니다.
- 사용할 가상 LAN 기술(VLAN 또는 VXLAN) 및 해당하는 ID 또는 세그먼트를 지정합니다.

OpenStack 대시보드 사용

OpenStack에 대한 설치 및 설치 후 구성 작업을 완료한 후 OpenStack 대시보드에 로그인하여 사용 가능한 리소스를 확인하고 VM 인스턴스를 만들어 부팅하십시오.

▼ OpenStack 대시보드에 액세스하는 방법

1. OpenStack 시스템에 연결할 수 있는 시스템에 로그인합니다.
2. 사용 중인 브라우저를 구성합니다.
 - a. JavaScript를 사용하여 설정합니다.
 - b. 쿠키를 보관합니다.
3. 브라우저의 위치 또는 주소 필드에 다음 위치를 입력합니다.

`http://system/horizon/`

`system`은 OpenStack 통합 아카이브가 설치되고 Apache 웹 서버에서 Horizon OpenStack 서비스를 실행 중인 OpenStack 시스템의 이름 또는 IP 주소입니다.

커널 영역에 통합 아카이브를 설치한 경우 OpenStack 시스템은 커널 영역이며 `system`은 커널 영역의 이름 또는 IP 주소입니다.

4. 로그인 화면에서 다음 정보를 제공합니다.
 - 사용자 이름: admin
 - 암호: secrete

대시보드에서 사용 가능한 기능은 로그인한 사용자의 액세스 권한 또는 역할에 따라 달라집니다.

대시보드 살펴보기

클라우드 관리자로 로그인하는 경우 창에는 화면 왼쪽에 2개의 탭인 Project(프로젝트) 및 Admin(관리자)이 있는 패널이 표시됩니다. Admin(관리자) 패널은 기본 클라우드 관리자 뷰입니다. Admin(관리자) 패널의 선택 항목은 다음 기능을 제공합니다.

- 클라우드에서 사용 중인 Nova 인스턴스 및 Cinder 볼륨의 전체 뷰
- 다음과 같은 VM 인스턴스 특성을 정의하는 flavor 정의를 확인 및 편집할 수 있는 기능
 - 가상 CPU 수
 - 메모리 양
 - 지정된 디스크 공간
 - 기본 Solaris 영역의 브랜드: solaris(비전역 영역의 경우) 및 solaris-kz(커널 영역의 경우)
- 클라우드 관리자용 가상 네트워크 및 라우터를 만들 수 있는 기능
- 가상 컴퓨팅 리소스의 소유권을 그룹화하고 격리시켜서 프로젝트를 확인 및 편집할 수 있는 기능
- 클라우드 리소스를 사용하는 개인 또는 서비스인 사용자를 확인 및 편집할 수 있는 기능

그림 1 OpenStack 대시보드 관리자 개요 창

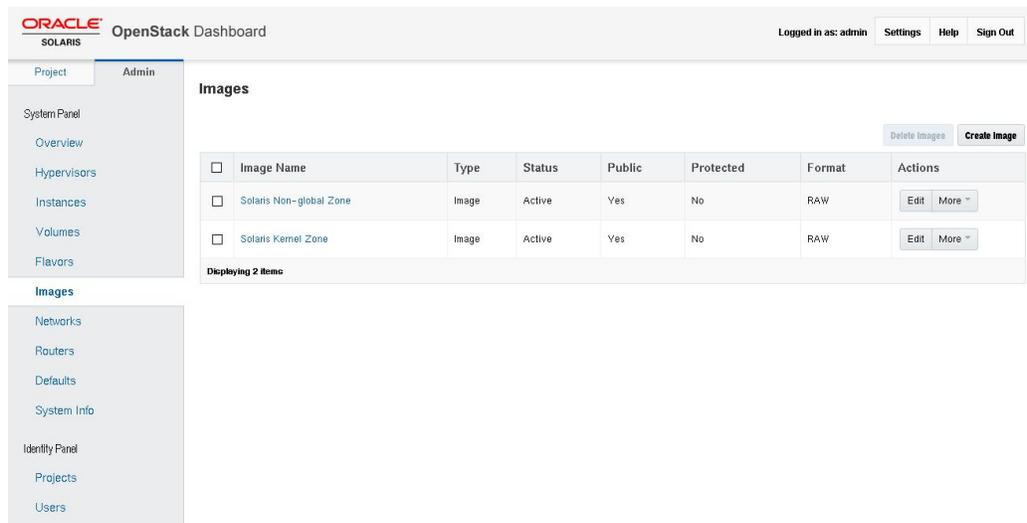
The screenshot shows the OpenStack Dashboard Admin Overview page. The page has a header with the Oracle Solaris logo and 'OpenStack Dashboard' text. On the right, it says 'Logged in as: admin' with links for 'Settings', 'Help', and 'Sign Out'. The left sidebar contains a navigation menu with 'Project' and 'Admin' tabs, and a 'System Panel' section. The main content area is titled 'Overview' and includes a date range selector (From: 2014-08-01, To: 2014-08-25) and a 'Submit' button. Below this, there are summary statistics for Active Instances, Active RAM, VCPU-Hours, and GB-Hours. A table with columns for Project Name, VCPUs, Disk, RAM, VCPU Hours, and Disk GB Hours is shown, but it contains no data and displays 'No items to display.' and 'Displaying 0 items'.

“[단일 시스템 설치](#)” [12]에 설명된 대로 OpenStack을 설치한 경우 OpenStack 시스템이 다음 리소스로 사전 구성되어 있습니다.

- 이미지 2개: Solaris 비전역 영역 및 Solaris 커널 영역
- 프로젝트 2개: demo 및 service
- flavor 10개

다음 그림은 이미지를 보여줍니다.

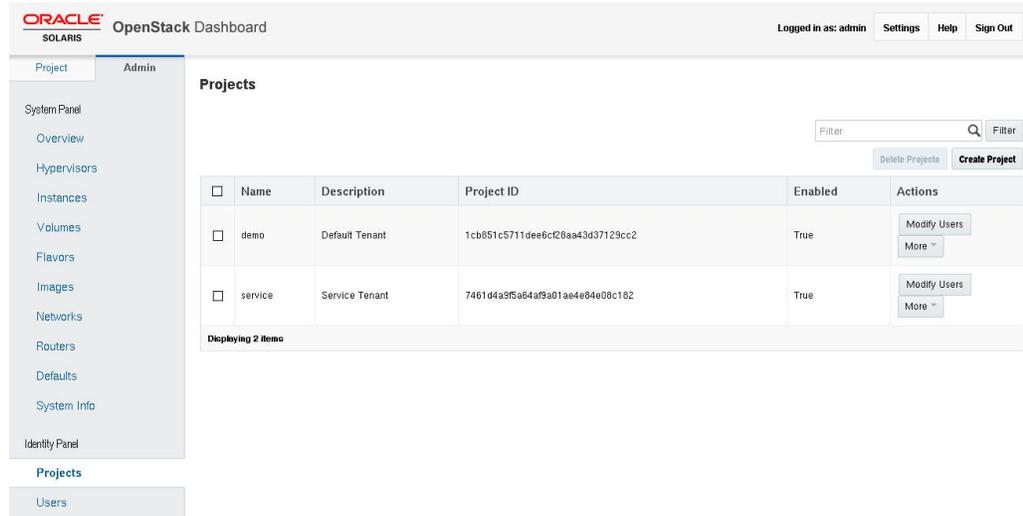
그림 2 OpenStack 대시보드의 이미지 화면



다음 그림은 테넌트라고도 하는 프로젝트를 보여줍니다.

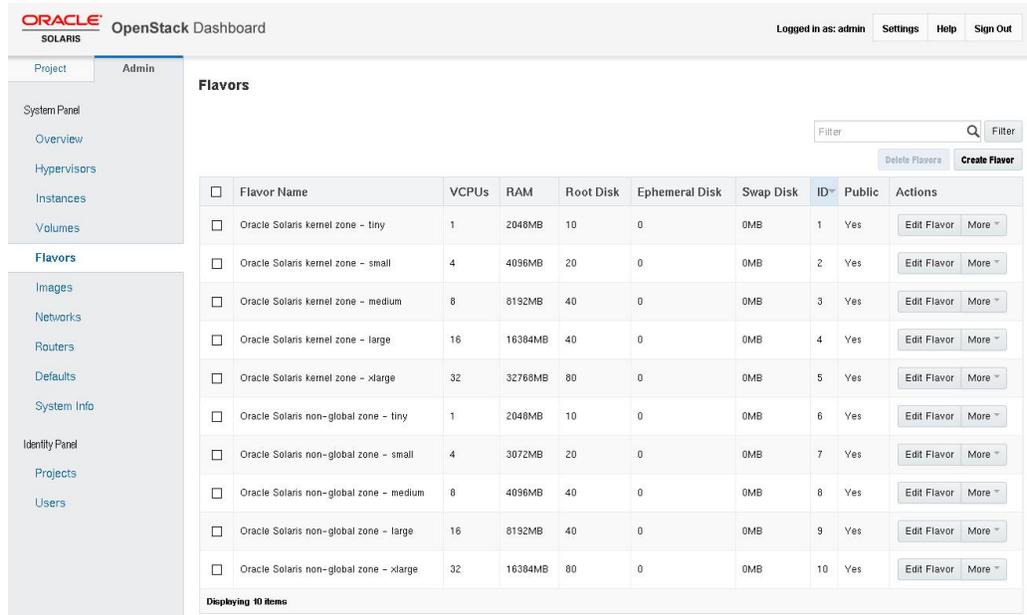
- demo 테넌트가 기본 테넌트입니다. 기본적으로 단일 사용자인 admin을 구성원으로 demo 테넌트가 만들어집니다.
- service 테넌트는 클라우드 관리자가 여러 테넌트에 걸쳐 공유되는 리소스를 만들 때 사용하는 테넌트입니다. 예를 들어, 이 문서의 예제와 시나리오에서는 Neutron 라우터를 service 테넌트에 만들어서 이 라우터를 모든 테넌트에서 공유하게 됩니다. OpenStack 설정에서 다른 목적으로 service 테넌트를 사용하면 안됩니다. OpenStack 서비스는 서비스별 사용자를 매개로 서로 통신하는데, 이들은 모두 admin 역할을 보유하고 service 테넌트에 속합니다.

그림 3 OpenStack 대시보드의 프로젝트 화면



다음 그림은 flavor를 보여줍니다. flavor의 텍스트 목록은 “[Flavor에 대한 정보 표시](#)” [63]를 참조하십시오.

그림 4 OpenStack 대시보드의 Flavors 화면



VM 인스턴스 만들기 및 부팅

대시보드에서 Project(프로젝트) 패널을 사용하여 VM 인스턴스를 만들 수 있습니다.

▼ 대시보드에서 VM 인스턴스를 만드는 방법

시작하기 전에 SSH Keypair가 있는지 확인하십시오. [SSH Keypair를 만드는 방법 \[26\]](#)을 참조하십시오.

외부 네트워크가 정의되었는지 확인합니다. [“Neutron L3 에이전트 구성” \[52\]](#)을 참조하십시오.

1. 대시보드 왼쪽에 있는 패널에서 Project(프로젝트) 탭을 누릅니다.
사용자가 현재 사용 중인 프로젝트가 해당 패널 상단에 표시됩니다. 기본적으로 이 OpenStack 구성의 admin 사용자는 demo 프로젝트를 사용합니다.
2. 패널의 Manage Compute(Compute 관리) 섹션에서 Instances(인스턴스)를 누릅니다.

3. Instances(인스턴스) 패널 오른쪽에서 Launch Instance(인스턴스 구동) 버튼을 누릅니다. 새 VM 인스턴스의 이름을 지정하고 인스턴스의 flavor 및 이미지 유형을 선택할 수 있는 다음 Launch Instance(인스턴스 구동) 대화 상자가 표시됩니다.

그림 5 인스턴스 구동 대화 상자

Launch Instance
✕

Details

Access & Security

Networking

Availability Zone

nova ▾

Instance Name

Flavor

Oracle Solaris kernel zone - tiny ▾

Instance Count

1

Instance Boot Source

--- Select source --- ▾

Specify the details for launching an instance.

The chart below shows the resources used by this project.

Flavor Details

Name	Oracle Sola...
VCPUs	1
Root Disk	10 GB
Ephemeral Disk	0 GB
Total Disk	10 GB
RAM	2,048 MB

Project Limits

Number of Instances

Number of VCPUs

Total RAM

Cancel

Launch

4. Instance Name(인스턴스 이름) 필드에 새 VM 인스턴스의 이름을 입력합니다.

5. Flavor 드롭다운 목록에서 flavor를 선택합니다.

이 OpenStack 시스템이 커널 영역이며 베어 메탈 시스템이 아닌 경우 비전역 영역 flavor를 선택해야 합니다.

6. Instance Boot Source(인스턴스 부팅 소스)에서 Boot from image(이미지로 부팅)를 선택합니다.

Image Name(이미지 이름) 선택 항목이 표시됩니다. 선택한 flavor와 영역 유형이 동일한 이미지를 선택합니다. flavor와 이미지는 solaris 비전역 영역 또는 solaris-kz 커널 영역에 해당해야 합니다. 이 OpenStack 시스템이 커널 영역이며 베어 메탈 시스템이 아닌 경우 비전역 영역 이미지를 선택해야 합니다.

그림 6 선택 항목이 채워진 인스턴스 구동 대화 상자

Launch Instance
✕

Details

Access & Security

Networking

Availability Zone

nova ▾

Instance Name

VM1

Flavor

Oracle Solaris non-global zone - tiny ▾

Instance Count

1

Instance Boot Source

Boot from image ▾

Image Name

Solaris Non-global Zone (1.2 GB) ▾

Specify the details for launching an instance.
The chart below shows the resources used by this project.

Flavor Details

Name	Oracle Sola...
VCPUs	1
Root Disk	10 GB
Ephemeral Disk	0 GB
Total Disk	10 GB
RAM	2,048 MB

Project Limits

Number of Instances

Number of VCPUs

Total RAM

Cancel

Launch

7. 대화 상자에서 **Access & Security(접근 & 보안)** 탭을 누릅니다.
새 VM 인스턴스에서 설치할 SSH Keypair를 선택합니다.
8. 대화 상자에서 **Networking(네트워킹)** 탭을 누릅니다.
새 VM 인스턴스가 연결될 네트워크를 선택합니다.

9. 대화 상자 하단에 있는 Launch(구동) 버튼을 누릅니다.

새 VM 인스턴스가 만들어지고 설치 및 부팅됩니다.

새 인스턴스를 사용 가능한 상태로 설정하는 데 필요한 시간은 이미지 크기, flavor에서 제공된 리소스, OpenStack이 새 VM 인스턴스의 루트 파일 시스템을 배치한 위치 등 여러 요소에 따라 달라집니다.

10. 유동 IP 주소를 새 VM 인스턴스와 연결합니다.

새 VM 인스턴스를 설치하는 동안 이러한 단계를 수행할 수 있습니다. VM 인스턴스에서는 연결된 유동 IP 주소가 있어야만 사용자가 로그인할 수 있습니다.

a. Actions(작업) 열에서 Associate Floating IP(유동 IP 연결) 버튼을 누릅니다.

Manage Floating IP Associations(유동 IP 연관 관리) 대화 상자가 열립니다.

b. IP Address(IP 주소) 드롭다운 메뉴에서 주소를 선택합니다.

IP Address(IP 주소) 필드에 사용 가능한 IP 주소가 없음을 알리는 메시지가 표시되면 + 버튼을 누릅니다. [유동 IP 주소를 프로젝트와 연결하는 방법 \[27\]](#)을 참조하십시오.

c. 연결된 포트를 선택합니다.

포트 목록에 VM 인스턴스의 고정 IP 주소가 표시됩니다.

d. 대화 상자 하단에 있는 Associate(연결) 버튼을 누릅니다.

- 다음 순서
- Instances(인스턴스)를 누르고 인스턴스 이름을 누르면 인스턴스에 대한 세부정보가 표시되고 인스턴스의 콘솔 로그가 표시됩니다. 페이지를 다시 로드하여 로그 업데이트를 확인합니다.
 - Volumes(볼륨)를 눌러 만들어진 Cinder 볼륨을 확인합니다.
 - Network Topology(네트워크 토폴로지)를 눌러 모든 서브넷 세그먼트, 가상 라우터, 활성 인스턴스 등 클라우드 네트워크의 표현을 확인합니다.
 - Images & Snapshots(이미지 & 스냅샷)를 눌러 Glance 이미지 저장소로 업로드된 통합 아카이브를 확인합니다.
 - 새 VM 인스턴스 설치가 완료되고 인스턴스가 Active(활성) 상태에 도달하면 인스턴스에 로그인합니다. 다음 명령은 단계 7의 키와 단계 10의 유동 IP 주소를 사용하여 루트로 영역에 로그인합니다.

```
# ssh root@floating-ip-address
```

▼ SSH Keypair를 만드는 방법

1. 대시보드 왼쪽에 있는 패널에서 Project(프로젝트) 탭을 누릅니다.
2. 패널의 Manage Compute(Compute 관리) 섹션에서 Access & Security(접근 & 보안)를 누릅니다.

3. **Keypairs** 탭을 누릅니다.
4. **Create Keypair(Keypair 생성) 버튼**을 누릅니다.
Create Keypair(Keypair 생성) 대화 상자가 열립니다.
5. 대화 상자의 **Keypair Name(Keypair 이름) 필드**에 키 쌍 이름을 입력합니다.
6. 대화 상자에서 **Create Keypair(Keypair 생성) 버튼**을 누릅니다.
새 Keypair가 자동으로 다운로드됩니다.
새 Keypair가 자동으로 다운로드되지 않을 경우 제공되는 **Download keypair(Keypair 다운로드) 링크**를 누릅니다.
Access & Security(접근 & 보안) 패널의 Keypairs 탭에 새 Keypair가 나열됩니다.

▼ 테넌트에 대한 네트워크를 만드는 방법

1. 대시보드 왼쪽에 있는 패널에서 **Project(프로젝트) 탭**을 누릅니다.
2. 패널의 **Manage Network(네트워크 관리) 섹션**에서 **Networks(네트워크)**를 누릅니다.
3. **Create Network(네트워크 생성) 버튼**을 누릅니다.
Create Network(네트워크 생성) 대화 상자가 열립니다.
4. **Network(네트워크) 탭의 Name(이름) 필드**에 네트워크 이름을 입력합니다.
5. **Subnet(서브넷) 및 Subnet Details(서브넷 상세 정보) 탭**에서 요청되는 정보를 제공합니다.
6. 대화 상자 하단에 있는 **Create(생성) 버튼**을 누릅니다.
Networks(네트워크) 패널에 새 네트워크 및 서브넷이 나열됩니다.

▼ 유동 IP 주소를 프로젝트와 연결하는 방법

1. 대시보드 왼쪽에 있는 패널에서 **Project(프로젝트) 탭**을 누릅니다.
2. 패널의 **Manage Compute(Compute 관리) 섹션**에서 **Access & Security(접근 & 보안)**를 누릅니다.
3. **Floating IPs(유동 IP) 탭**을 누릅니다.
4. **Allocate IP To Project(프로젝트에 IP 할당) 버튼**을 누릅니다.
Allocate Floating IP(유동 IP 할당) 대화 상자가 열립니다.
5. 대화 상자의 드롭다운 메뉴에서 유동 IP를 할당할 풀을 선택합니다.

6. 대화 상자에서 Allocate IP(IP 할당) 버튼을 누릅니다.

다중 노드 Havana OpenStack 구성을 위해 여러 시스템에 걸쳐 설치

이 장에서는 다중 노드 OpenStack 구성을 설치하는 방법에 대해 설명합니다. 각 클라우드에는 대시보드 인스턴스, 이미지 저장소 및 ID 서비스가 하나씩만 필요하며, 저장소와 Compute 인스턴스가 여러 개 포함될 수 있습니다. 운용 환경에서 해당 서비스는 다중 노드에 걸쳐 구성됩니다. 특정 클라우드 배치에 대한 요구 사항과 관련하여 각 구성 요소를 평가한 후 구성 요소를 별도의 노드에 설치해야 할지 여부와 필요한 해당 유형의 노드 수를 결정하십시오.

이 장에서는 별도의 물리적 시스템에 배치된 아키텍처에 대해 설명합니다. OVM Server for SPARC(LDom)가 실행되는 서버에서 단일 Oracle SPARC 서버를 분할하고 다중 노드 OpenStack을 구성하려면 [Multi-node Solaris 11.2 OpenStack on SPARC Servers](#)를 참조하십시오.

3노드 아키텍처 개요

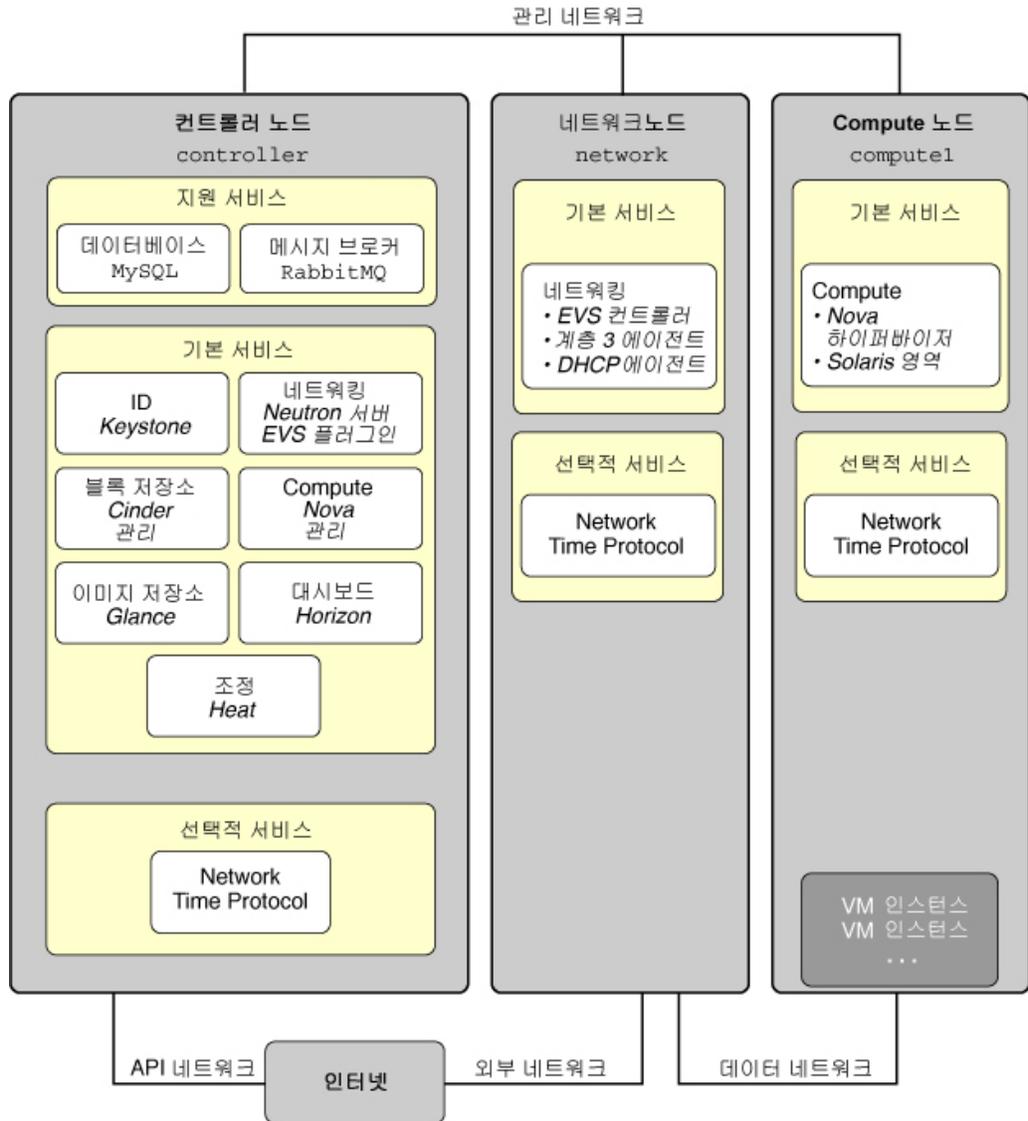
이 장에서 설명되는 아키텍처는 다음과 같은 세 시스템에 배치됩니다.

- 컨트롤러 노드. 컨트롤러 노드는 대부분의 공유 OpenStack 서비스 및 기타 도구가 실행되는 위치입니다. 컨트롤러 노드는 API, 예약 및 기타 클라우드용 공유 서비스를 제공합니다. 컨트롤러 노드에는 대시보드, 이미지 저장소 및 ID 서비스가 있습니다. 추가적으로 Nova Compute 관리 서비스와 Neutron 서버도 이 노드에 구성됩니다.
- 네트워크 노드. 네트워크 노드는 Neutron 계층 3 및 DHCP 네트워크 서비스를 사용하는 Nova 인스턴스에 가상 네트워킹과 네트워킹 서비스를 제공합니다.
- Compute 노드. Compute 노드는 VM 인스턴스(Nova Compute 인스턴스)가 설치되는 위치입니다. VM 인스턴스는 Cinder 볼륨 서비스가 프로비전하는 iSCSI 대상을 사용합니다.

이 아키텍처에서 세 노드는 관리 서브넷이라는 공통 서브넷을 공유합니다. 컨트롤러 노드와 각 Compute 노드는 데이터 서브넷이라는 별도의 공통 서브넷을 공유합니다. 각 시스템은 net0 물리적 인터페이스를 통해 관리 네트워크에 연결됩니다. 네트워크 노드와 Compute 노드는 net1 물리적 인터페이스를 통해 데이터 네트워크에 연결됩니다.

다음 그림은 이 장에서 설명되는 높은 레벨의 아키텍처 뷰를 보여줍니다.

그림 7 3노드 구성 참조 아키텍처



다음 표에서는 각 노드에 설치된 OpenStack 관련 SMF 서비스를 보여줍니다. 목록에는 명령(예: svcadm 명령)에서 사용할 수 있는 각 SMF 서비스 이름이 자세히 표시됩니다. 인스턴스 이름 없이는 이름이 모호한 경우에만 SMF 서비스의 인스턴스 이름이 나열됩니다.

표 1 컨트롤러, 네트워크 및 Compute 노드에 설치된 SMF 서비스

컨트롤러 노드	네트워크 노드	Compute 노드
mysql	neutron-dhcp-agent	nova-compute
rabbitmq	neutron-l3-agent	ntp
keystone	evs-controller	
cinder-api	ntp	
cinder-db		
cinder-db		
cinder-scheduler		
cinder-volume:default		
cinder-volume:setup		
glance-api		
glance-db		
glance-registry		
glance-scrubber		
neutron-server		
evs		
nova-api-ec2		
nova-api-osapi-compute		
nova-cert		
nova-conductor		
nova-objectstore		
nova-scheduler		
http		
ntp		
heat-api		
heat-db		
heat-api-cfn		
heat-api-cloudwatch		
heat-engine		

이 아키텍처 예에서는 Swift 객체 저장소 서비스가 명시되지 않습니다. Swift 구성에 대한 일반적인 정보는 OpenStack 커뮤니티 사이트의 정보(예: [OpenStack Configuration Reference](#))를 참조하십시오. Solaris 시스템에서의 Swift 서비스 구성 방법에 대한 자세한 내용과 Solaris의 OpenStack에 대한 기타 정보는 [OpenStack for Oracle Solaris 11](#)을 참조하십시오.

Oracle Solaris 시스템에 OpenStack을 배치하는 데 유용한 OpenStack 구성 매개변수 목록은 [Getting Started with OpenStack on Oracle Solaris 11.2](#)의 "Common Configuration Parameters for OpenStack"을 참조하십시오.

샘플 3노드 OpenStack 구성의 구현을 준비하려면 다음 정보를 가지고 있는지 확인하십시오.

- 컨트롤러 노드의 IP 주소와 호스트 이름
- 네트워크 노드의 IP 주소와 호스트 이름
- Compute 노드의 IP 주소와 호스트 이름
- 필요에 따라 다른 서비스 사용자의 암호

샘플 구성에서 세 노드의 이름은 controller, network, compute1입니다.

컨트롤러 노드 구성

컨트롤러 노드에는 대시보드 서비스, 이미지 저장소 및 ID 서비스가 하나씩 필요하며, MySQL, RabbitMQ와 Compute, 블록 저장소 및 네트워킹 서비스도 포함됩니다.

Oracle Solaris 11에서 ZFS와 응용 프로그램 간 메모리 사용을 보다 잘 관리하려면 다음 예에 표시된 것과 같이, 노드에 대해 `usr_reserve_hint_pct` 매개변수를 설정합니다.

```
# echo "set user_reserve_hint_pct=80" >>/etc/system.d/site:kernel-zones-reserve
# reboot
```

여기서 `site`는 회사를 나타낼 수 있습니다.

이 매개변수를 다른 OpenStack 노드에도 설정해야 합니다.

이 매개변수에 대한 자세한 내용은 <https://support.oracle.com>에서 MOS 계정에 로그인한 다음 Document 1663862.1, *Memory Management Between ZFS and Applications in Oracle Solaris 11.2*를 검토하십시오.

OpenStack 서비스 간의 통신은 AMQP(Advanced Message Queuing Protocol)를 통해 수행됩니다. Solaris에서 AMQP는 RabbitMQ를 통해 구현됩니다. 따라서 RabbitMQ는 필수 서비스입니다. 일반적으로 클라우드의 단일 노드는 RabbitMQ를 실행하도록 구성됩니다. 이 아키텍처에서 RabbitMQ는 컨트롤러 노드에서 실행되도록 구성됩니다.

▼ 컨트롤러 노드를 구성하는 방법

1. (선택사항) NTP를 설치 및 구성합니다.
“[Network Time Protocol 설치](#)” [33]를 참조하십시오.
2. (선택사항) MySQL을 설치 및 구성합니다.
“[MySQL 설치](#)” [34]를 참조하십시오.
3. RabbitMQ를 설치합니다.
 - a. RabbitMQ 패키지를 설치합니다.

```
controller# pkg install rabbitmq
```
 - b. RabbitMQ SMF 서비스를 사용으로 설정합니다.

```
controller# svcadm enable rabbitmq
```
4. Keystone을 설치 및 구성합니다.
“[Keystone 설치](#)” [36]를 참조하십시오.
5. Cinder를 설치 및 구성합니다.
“[Cinder 설치](#)” [38]를 참조하십시오.
6. Glance를 설치 및 구성합니다.
“[Glance 설치](#)” [42]를 참조하십시오.
7. Neutron을 설치 및 구성합니다.
“[컨트롤러 노드에서 Neutron 설치 및 구성](#)” [43]을 참조하십시오.
8. Nova를 설치 및 구성합니다.
“[Nova 설치](#)” [44]를 참조하십시오.
9. Horizon을 구성합니다.
[Horizon을 구성하는 방법](#) [45]을 참조하십시오.

Network Time Protocol 설치

NTP(Network Time Protocol) 설치 는 선택사항이지만 강력히 권장됩니다. NTP를 설치하는 경우 클라우드 배치 내 각 서비스 노드에 설치하십시오.

NTP는 클라우드의 모든 서비스 노드에서 시간을 일관성 있게 유지할 수 있도록 도와줍니다. 네트워크에서 NTP를 사용으로 설정하는 경우 네트워크를 통해 시간을 가져오도록 서비스 노드를 구성하십시오.

- 서비스 노드가 상주하는 IP 서브넷에서 IP 멀티캐스트가 사용으로 설정된 경우 IP 멀티캐스트를 활용하여 NTP를 구성할 수 있습니다.
- 서비스 노드가 상주하는 IP 서브넷에서 IP 멀티캐스트가 사용으로 설정되지 않은 경우 NTP를 수동으로 구성하십시오.

▼ Network Time Protocol을 설치 및 구성하는 방법

1. NTP 패키지를 설치합니다.

```
controller# pkg install ntp
```

2. 구성 파일을 설치합니다.

```
controller# cp /etc/inet/ntp.client /etc/inet/ntp.conf
```

3. NTP를 구성합니다.

필요한 구성은 노드의 서브넷에서 IP 멀티캐스트가 사용으로 설정되었는지 여부에 따라 달라집니다.

- IP 멀티캐스트가 사용으로 설정된 경우 추가 구성이 필요하지 않습니다.
- IP 멀티캐스트가 사용으로 설정되지 않은 경우 기존 NTP 서버의 호스트 이름 또는 IP 주소를 구성합니다.

a. `/etc/inet/ntp.conf` 파일에서 `multicastclient` 옵션을 주석 처리합니다.

```
# multicastclient 224.0.1.1
```

b. `/etc/inet/ntp.conf` 파일에서 서버 옵션 중 하나 이상의 주석 처리를 해제합니다.

```
server ntp_server_1 iburst
server ntp_server_2 iburst
```

4. NTP 서버 SMF 서비스를 사용으로 설정합니다.

```
controller# svcadm enable ntp
```

MySQL 설치

여러 OpenStack 서비스는 데이터베이스 유지 관리를 통해 중요한 리소스, 사용량 및 기타 정보를 추적합니다. 기본적으로 개별 SQLite 데이터베이스가 이 용도로 지정되며 단일 노드

구성에 유용합니다. 다중 노드 구성의 경우 이 정보를 저장하는 데 MySQL 데이터베이스가 권장됩니다.

▼ MySQL 데이터베이스를 설치하는 방법

1. 컨트롤러 노드와 연결된 기본 이름을 확인합니다.
컨트롤러 노드의 기본 IP 주소를 사용하여 해당 노드와 연결된 기본 이름을 확인합니다.

```
controller# getent hosts controller-IP
controller-IP controller-name
```

2. MySQL 서버 패키지를 설치합니다.

```
controller# pkg install mysql-55
```

3. MySQL 클라이언트 패키지를 설치합니다.

```
controller# pkg install mysql-55/client
```

4. MySQL 서버 SMF 서비스를 사용으로 설정합니다.

```
controller# svcadm enable mysql:version_55
```

5. MySQL 서버 root 암호를 설정합니다.

```
controller# mysqladmin -u root password MySQL-root-password
```

6. MySQL 서버를 구성합니다.

CreateOpenStack에서 사용될 테이블을 만듭니다. 해당 데이터베이스에 대해 배타적 액세스를 제공하기 위해 컨트롤러 노드의 서비스에 권한을 부여합니다. 이전 `getent hosts` 명령으로 출력된 `controller-name`을 사용합니다.

```
controller# mysql -u root -p
Enter password: MySQL-root-password
mysql> create database cinder;
mysql> grant all privileges on cinder.* \
-> to 'cinder'@'controller-name' \
-> identified by 'cinder';
mysql> create database glance;
mysql> grant all privileges on glance.* \
-> to 'glance'@'controller-name' \
-> identified by 'glance';
mysql> create database keystone;
mysql> grant all privileges on keystone.* \
-> to 'keystone'@'controller-name' \
-> identified by 'keystone';
mysql> create database nova;
mysql> grant all privileges on nova.* \
-> to 'nova'@'controller-name' \
-> identified by 'nova';
```

```
mysql> flush privileges;
mysql> quit
```

7. MySQL Python 클라이언트 라이브러리 패키지를 설치합니다.

```
controller# pkg install python-mysql
```

Keystone 설치

Keystone 서비스는 컨트롤러 노드에서 설치 및 구성해야 합니다.

▼ Keystone을 설치 및 구성하는 방법

1. Keystone 패키지를 설치합니다.

```
controller# pkg install keystone
```

2. Keystone 구성 파일을 수정합니다.

/etc/keystone/keystone.conf 파일에서 다음 두 매개변수의 주석 처리를 해제하고 해당 매개변수를 설정합니다.

a. **admin_token** 매개변수를 설정합니다.

admin_token 매개변수는 Keystone 서비스와 기타 OpenStack 서비스 간의 "공유 보안"입니다. 이 매개변수의 값은 모든 문자열일 수 있지만 이 값은 표시되거나 배포되지 않아야 합니다. 해당 문자열을 만드는 방법 중 하나는 다음 명령과 같이 OpenSSL을 사용하는 것입니다.

```
controller# openssl rand -hex 10
random_string
```

이 출력 값을 사용하여 /etc/keystone/keystone.conf 파일에서 admin_token 매개변수를 설정합니다.

```
admin_token = random_string
```

b. **connection** 매개변수를 설정합니다.

connection 매개변수는 Keystone 데이터베이스의 위치와 사용되는 데이터베이스의 종류를 나타내는 URI입니다.

이전 `getent hosts` 명령으로 출력된 *controller-name*을 사용하여 /etc/keystone/keystone.conf 파일에서 connection 매개변수를 설정합니다.

```
connection = mysql://keystone:keystone@controller-name/keystone
```

3. PKI(공개 키 기반구조) 토큰을 생성합니다.

```
controller# su - keystone -c "keystone-manage pki_setup"
```

4. Keystone SMF 서비스를 사용으로 설정합니다.

```
controller# svcadm enable keystone
```

5. Keystone 데이터베이스를 채웁니다.

이 단계는 수동으로 완료할 수도 있고, 다음 예와 같이 `sample_data.sh` 스크립트를 사용할 수도 있습니다. 이전 `getent hosts` 명령으로 출력된 `controller-name`을 사용합니다.

```
controller# su - keystone -c "env \
CONTROLLER_ADMIN_ADDRESS=controller-name \
CONTROLLER_INTERNAL_ADDRESS=controller-name \
CONTROLLER_PUBLIC_ADDRESS=controller-name \
/usr/demo/openstack/keystone/sample_data.sh"
```

`sample_data.sh` 스크립트는 각 API 서비스가 상주하는 노드와 각 서비스에 대한 암호를 정의하는 환경 변수를 지원합니다. 스크립트에서 환경으로부터 설정할 수 있는 매개변수에 대한 자세한 내용을 검토합니다. 기본적으로 `service` 테넌트에 속하는 각 서비스에 대해 Keystone 사용자가 만들어지며 사용자 이름과 동일한 암호가 지정됩니다. 예를 들어, `nova` 사용자가 만들어지는 경우 암호는 `nova`입니다.

Heat 설치 및 구성

Heat는 직접 만든 템플릿을 기반으로 클라우드 응용 프로그램을 배치할 수 있는 OpenStack 조정 엔진입니다. Keystone과 동일한 노드에 Heat를 설치합니다.

▼ Heat를 구성하는 방법

시작하기 전에 이 작업을 수행하기 전에 먼저 [Keystone을 설치 및 구성하는 방법](#)에 설명된 대로 Keystone을 구성해야 합니다.

1. Heat 패키지를 설치합니다.

```
controller# pkg install heat
```

2. Heat 설정 스크립트를 실행합니다.

```
# /usr/demo/openstack/keystone/heat-keystone-setup
```

3. `/etc/heat/api-past.ini`에서 다음 정보를 업데이트하여 파일을 편집합니다.

```
# Auth middleware that validates token against keystone
[filter:authtoken]
paste.filter_factory = heat.common.auth_token:filter_factory
```

```
auth_uri = http://controller-IP:5000/v2.0
identity_uri = http://controller-IP:35357
admin_tenant_name = keystone
admin_user = heat
admin_password = heat-password
```

4. Heat 서비스를 사용으로 설정합니다.

```
# svcadm enable -rs heat-api heat-db heat-engine heat-api-cfn heat-api-cloudwatch
```

Cinder 설치

Cinder 구성에서는 최소한 다음 정보를 지정해야 합니다.

- Keystone에 대해 인증할 권한 부여 정보
- 만들 볼륨의 클래스

▼ Cinder를 설치 및 구성하는 방법

1. Cinder 패키지를 설치합니다.

```
controller# pkg install cinder
```

2. 인증 구성 정보를 지정합니다.

/etc/cinder/api-paste.ini 파일에서 다음 매개변수의 주석 처리를 해제하고 해당 매개변수를 설정합니다. 해당 매개변수는 Keystone API 서비스의 위치 및 Cinder 인증 정보를 지정합니다.

```
auth_uri = http://controller-name:5000/v2.0
identity_uri = http://controller-name:35357
admin_tenant_name = service
admin_user = cinder
admin_password = cinder-password
```

3. Cinder 볼륨 서비스가 만들어야 할 볼륨의 클래스를 지정합니다.

/etc/cinder/cinder.conf 파일에서 해당하는 volume_driver 매개변수의 주석 처리를 해제합니다. 다음과 같은 네 가지 볼륨 클래스가 지원됩니다.

ZFSVolumeDriver

Cinder 볼륨 서비스와 동일한 노드에서 Nova용 로컬 볼륨을 만들 수 있도록 지원합니다.

ZFISCSIDriver

원격 Nova Compute 노드용 iSCSI 대상을 만들고 내보낼 수 있도록 지원합니다.

ZFSFCDriver

원격 Nova Compute 노드용 광 섬유 채널 LUN을 만들고 내보낼 수 있도록 지원합니다.

ZFSSAISCSDriver

원격 Nova Compute 노드용 iSCSI 대상을 만들고 원격 Oracle ZFS Storage Appliance에서 내보낼 수 있도록 지원합니다. 이 드라이버에 대한 추가 매개변수는 `/etc/cinder/cinder.conf` 파일에서 설정해야 합니다.

이 장에서 설명되는 예에서 iSCSI는 Nova 인스턴스용 볼륨을 제공하는 데 사용됩니다. ZFSVolumeDriver의 기본 선택 항목을 주석 처리하고 ZFSISCSIDriver 선택 항목의 주석 처리를 해제합니다.

```
# Driver to use for volume creation (string value)
# The local ZFS driver provides direct access to ZFS volumes that it
# creates. The other listed drivers provide access to ZFS volumes via
# iSCSI or Fibre Channel and are suitable for cases where block storage
# for Nova compute instances is shared.
#volume_driver=cinder.volume.drivers.solaris.zfs.ZFSVolumeDriver
volume_driver=cinder.volume.drivers.solaris.zfs.ZFSISCSIDriver
#volume_driver=cinder.volume.drivers.solaris.zfs.ZFSFCDriver
#volume_driver=cinder.volume.drivers.zfssa.zfssaiscsi.ZFSSAISCSDriver
```

4. 추가 구성 매개변수를 설정합니다.

`/etc/cinder/cinder.conf` 파일에서 다음 매개변수의 주석 처리를 해제하고 해당 매개변수를 설정합니다. 이러한 매개변수는 Glance API 서비스, 해당하는 Cinder용 데이터베이스 및 RabbitMQ 서비스의 위치를 지정합니다.

```
glance_host=controller-name
sql_connection=mysql://cinder:cinder@controller-name/cinder
rabbit_host=controller-name
volume_driver=cinder.volume.drivers.solaris.zfs.ZFSISCSIDriver
```

5. iSCSI 대상이 구성되면 해당하는 SMF 서비스를 사용으로 설정합니다.

```
controller# svcadm enable iscsi/target stmf
```

6. Cinder SMF 서비스를 사용으로 설정합니다.

```
controller# svcadm enable cinder-db
controller# svcadm enable cinder-api cinder-scheduler
controller# svcadm enable cinder-volume:default cinder-volume:setup
```

참조 [How to Build OpenStack Block Storage on ZFS](#)를 참조하십시오.

▼ ZFS Storage Appliance iSCSI Cinder 드라이버를 구성하는 방법

Oracle ZFS Storage Appliance iSCSI Cinder 드라이버를 통해 Oracle ZFSSA(ZFS Storage Appliance)를 Cinder용 블록 저장소 리소스로 원활하게 사용할 수 있습니다. 이 드

라이버는 Cinder 서버가 Nova 서비스에 의해 인스턴스화된 가상 시스템에 할당할 수 있는 iSCSI 볼륨을 만들 수 있도록 해줍니다. 드라이버는 cloud/openstack/cinder 패키지로 제공 됩니다. 사용 중인 어플라이언스에서 ZFSSA 소프트웨어 릴리스 2013.1.2.0 이상이 실행되고 있어야 합니다.

시작하기 전에 Oracle ZFS Storage Appliance에서 풀을 구성하십시오. 기존 풀을 사용하도록 선택할 수 있습니다.

1. 워크플로우 `cinder.akwf`를 실행합니다.

기존 사용자를 사용하거나 Cinder 드라이버 작업을 수행할 수 있는 역할 권한이 부여된 새 사용자를 만들 수 있습니다.

`cinder.akwf` 워크플로우는 다음 작업을 수행합니다.

- 사용자가 없을 경우 사용자를 만듭니다.
- Cinder 드라이버 작업을 수행할 수 있는 역할 권한 부여를 설정합니다.
- 현재 서비스가 사용 안함으로 설정된 경우 RESTful 서비스를 사용으로 설정합니다.

사용 중인 어플라이언스의 CLI(명령줄 인터페이스) 또는 BUI(브라우저 사용자 인터페이스)에서 워크플로우를 실행할 수 있습니다.

■ CLI에서 워크플로우를 실행합니다.

```
zfssa:maintenance workflows> download
zfssa:maintenance workflows download (uncommitted)> show
Properties:
    url = (unset)
    user = (unset)
    password = (unset)

zfssa:maintenance workflows download (uncommitted)> set url="url to the cinder.akwf file"
    url = "url to the cinder.akwf file"
zfssa:maintenance workflows download (uncommitted)> commit
Transferred 2.64K of 2.64K (100%) ... done

zfssa:maintenance workflows> ls
Properties:
    showhidden = false

Workflows:

WORKFLOW   NAME                                     OWNER SETID ORIGIN
VERSION
workflow-000 Clear locks                 root  false Oracle Corporation
1.0.0
workflow-001 Configuration for OpenStack Cinder Driver root  false Oracle Corporation
1.0.0

zfssa:maintenance workflows> select workflow-001
```

```

zfssa:maintenance workflow-001 execute (uncommitted)> set name=openstack
                    name = openstack
zfssa:maintenance workflow-001 execute (uncommitted)> set password=openstack-password
                    password = *****
zfssa:maintenance workflow-001 execute (uncommitted)> commit
User openstack created.

```

■ BUI에서 워크플로우를 실행합니다.

- a. Maintenance(유지 관리) -> Workflows(워크플로우)를 선택하고 더하기 아이콘을 사용하여 새 워크플로우를 업로드합니다.
- b. Browse(찾아보기) 버튼을 누르고 `cinder.akwf` 파일을 선택합니다.
- c. UPLOAD(업로드) 버튼을 눌러 워크플로우 업로드를 완료합니다.
- d. BUI Workflows(BUI 워크플로우) 페이지에 나타나는 새 행을 눌러 Cinder 드라이버 워크플로우를 실행합니다.
워크플로우에서 사용자 이름 및 암호를 입력하라는 메시지를 표시합니다. 이 사용자 이름 및 암호는 `cinder.conf` 파일에서 `zfssa_auth_user` 및 `zfssa_auth_password`로도 사용됩니다.

2. `cinder.conf` 파일에서 매개변수를 설정합니다.

`cinder.conf` 파일에서 다음과 같은 필수 등록 정보를 지정합니다.

- `volume_driver` - `cinder.volume.drivers.zfssa.zfssaiscsi.ZFSSAISCSIDriver`의 주석 처리가 해제되었는지 확인합니다. 다른 세 가지 선택 항목이 주석 처리되었는지 확인합니다.
- `zfssa_host` - ZFSSA 관리 호스트의 이름 또는 IP 주소입니다.
- `zfssa_auth_user` - ZFSSA에서 Cinder 사용자의 사용자 이름입니다.
- `zfssa_auth_password` - ZFSSA에서 Cinder 사용자의 암호입니다.
- `zfssa_pool` - 볼륨 할당에 사용할 풀입니다.
- `zfssa_target_portal` - ZFSSA iSCSI 대상 포탈(`data-ip:port`)입니다. 기본 포트는 3260입니다.
- `zfssa_project` - ZFSSA 프로젝트의 이름입니다. 어플라이언스에 프로젝트가 없을 경우 시작 시 드라이버가 해당 이름으로 프로젝트를 만듭니다. 이 프로젝트에는 드라이버가 만든 모든 볼륨이 포함됩니다. 볼륨 특성(예: 블록 크기) 및 액세스(예: 개시자, 대상, 보안) 설정을 위해 추가 ZFSSA 등록 정보가 제공됩니다.
- `zfssa_initiator_group` - 개시자 그룹의 이름입니다. 어플라이언스에 개시자 그룹이 없을 경우 시작 시 드라이버가 해당 이름으로 개시자 그룹을 만듭니다. `default` 개시자 그룹을 사용하려면 이 매개변수의 값을 `default`로 설정합니다. `default` 개시자 그룹은 평가 용도로 유용할 수 있습니다. 원치 않거나 충돌하는 개시자에게 볼륨이 노출될 수 있으므로 `default` 개시자 그룹은 일반적으로 사용되지 않아야 합니다.

- `zfssa_target_interfaces` - ZFSSA iSCSI 대상 네트워크 인터페이스입니다. 다음 명령을 사용하여 인터페이스를 확인할 수 있습니다.

```
zfssa:configuration net interfaces> show
Interfaces:

INTERFACE STATE CLASS LINKS   ADDRS           LABEL
e1000g0  up   ip   e1000g0  1.10.20.30/24  Untitled Interface
```

- `connection` - `connection`을 `sql_connection`으로 변경합니다. 다음 행을 찾습니다.

```
connection=mysql://cinder:cinder...
```

이 행을 아래와 같이 수정합니다.

```
sql_connection=mysql://cinder:cinder...
```

3. ZFSSA iSCSI 서비스가 온라인 상태인지 확인합니다.

ZFSSA iSCSI 서비스가 온라인 상태가 아니면 어플라이언스에서 BUI 또는 CLI를 통해 사용으로 설정합니다. 다음 예에서는 어플라이언스에서 CLI를 사용하는 방법을 보여줍니다.

```
zfssa:> configuration services iscsi
zfssa:configuration services iscsi> enable
zfssa:configuration services iscsi> show
Properties:
<status> = online
...
```

4. Cinder 볼륨 SMF 서비스를 사용으로 설정합니다.

```
controller# svcadm enable cinder-volume:default cinder-volume:setup
```

Glance 설치

Cinder 구성과 마찬가지로 Glance를 설정하려면 몇 가지 인증 정보와 MySQL 및 RabbitMQ 서비스의 위치에 대한 정보를 구성해야 합니다.

▼ Glance를 설치 및 구성하는 방법

1. Glance 패키지를 설치합니다.

```
controller# pkg install glance
```

2. Glance 구성 매개변수를 설정합니다.

다음 파일 각각에서 다음 매개변수의 주석 처리를 해제하고 해당 매개변수를 설정합니다.

- /etc/glance/glance-api.conf
- /etc/glance/glance-cache.conf
- /etc/glance/glance-registry.conf
- /etc/glance/glance-scrubber.conf

```
auth_uri = http://controller-name:5000/v2.0
identity_uri = http://controller-name:35357
admin_tenant_name = service
admin_user = glance-password
admin_password = glance
```

3. MySQL 데이터베이스의 URI를 지정합니다.

/etc/glance/glance-api.conf 파일과 /etc/glance/glance-registry.conf 파일에서 MySQL 데이터베이스의 URI를 지정합니다.

```
connection=mysql://glance:glance@controller-name/glance
```

4. RabbitMQ 서비스의 위치를 지정합니다.

/etc/glance/glance-api.conf에서 RabbitMQ 서비스의 위치를 설정합니다.

```
rabbit_host = controller-name
```

5. Glance SMF 서비스를 사용으로 설정합니다.

```
controller# svcadm enable glance-db
controller# svcadm enable glance-api glance-registry glance-scrubber
```

컨트롤러 노드에서 Neutron 설치 및 구성

이 장에서 설명되는 아키텍처에서는 Neutron API 서비스가 컨트롤러 노드에서 실행됩니다. 이 서비스가 네트워크 노드에 설치된 EVS 컨트롤러와 통신할 수 있도록 하려면 컨트롤러의 neutron 사용자에게 대한 공개 SSH 키가 네트워크 노드의 evsuser 사용자의 authorized_keys 파일에 있어야 합니다.

▼ Neutron을 설치 및 구성하는 방법

1. Neutron 패키지를 설치합니다.

```
controller# pkg install neutron
```

2. neutron 사용자에게 대한 컨트롤러 노드에서 SSH 공개 키를 만듭니다.

이 키는 Neutron API 서비스가 EVS 컨트롤러에 액세스할 수 있도록 해줍니다.

ssh-keygen 명령을 neutron 사용자로 사용하고 neutron 사용자에게 대한 키를 만듭니다.

```
controller# su - neutron \
-c "ssh-keygen -N '' -f /var/lib/neutron/.ssh/id_rsa -t rsa"
```

3. 키를 EVS 컨트롤러 노드에 복사합니다.

이전 단계에서 생성한 SSH 공개 키인 /var/lib/neutron/.ssh/id_rsa.pub를 EVS 컨트롤러가 실행 중인 노드에 복사합니다. 이 키는 EVS 컨트롤러를 구성할 때 참조됩니다.

4. Neutron 구성 파일에서 매개변수를 설정합니다.

Keystone 인증 정보 및 RabbitMQ 서비스의 위치를 지정합니다. /etc/neutron/neutron.conf 파일에서 다음 매개변수의 주석 처리를 해제하고 해당 매개변수를 설정합니다.

```
rabbit_host = controller-name

auth_uri = http://controller-name:5000/v2.0
identity_uri = http://controller-name:35357
admin_tenant_name = service
admin_user = neutron
admin_password = neutron-password
```

5. EVS 컨트롤러의 위치를 지정합니다.

a. EVS 컨트롤러의 위치를 가져옵니다.

getent hosts 명령에 네트워크 노드의 IP 주소를 사용하여 네트워크 노드의 이름을 가져옵니다.

```
network# getent hosts network-IP
network-IP network-name
```

b. EVS 컨트롤러의 위치를 설정합니다.

/etc/neutron/plugins/evs/evs_plugin.ini 파일에서 다음 매개변수의 주석 처리를 해제하거나 해당 매개변수를 설정합니다. getent hosts 명령의 출력을 사용하여 EVS 컨트롤러의 위치를 설정합니다.

```
evs_controller = ssh://evsuser@network-name
sql_connection = path-to-database
```

6. Neutron 서버 서비스를 사용으로 설정합니다.

```
controller# svcadm enable neutron-server
```

Nova 설치

컨트롤러 노드에서 Nova를 구성할 때도 일반적인 인증 및 기타 서비스 정보가 필요합니다.

▼ Nova를 설치 및 구성하는 방법

1. Nova 패키지를 설치합니다.

```
controller# pkg install nova
```

2. 인증 구성 정보를 지정합니다.

/etc/nova/api-paste.ini 파일에서 다음 매개변수의 주석 처리를 해제하고 해당 매개변수를 설정합니다. 해당 매개변수는 Keystone API 서비스의 위치 및 Nova 인증 정보를 지정합니다.

```
auth_uri = http://controller-name:5000/v2.0
identity_uri = http://controller-name:35357
admin_tenant_name = service
admin_user = nova
admin_password = nova-password
```

3. 추가 구성 매개변수를 설정합니다.

/etc/nova/nova.conf 파일에서 다음 매개변수의 주석 처리를 해제하고 해당 매개변수를 설정합니다. 이러한 매개변수는 추가 Keystone 인증 서비스 끝점, Glance API 서비스, Neutron API 서비스, RabbitMQ 서비스 및 Nova 관련 데이터베이스에 대한 데이터베이스 URI를 지정합니다.

```
keystone_ec2_url=http://controller-name:5000/v2.0/ec2tokens
glance_host=controller-name
neutron_url=http://controller-name:9696
neutron_admin_username=neutron
neutron_admin_password=neutron-password
neutron_admin_tenant_name=service
neutron_admin_auth_url=http://controller-name:5000/v2.0
rabbit_host=controller-name
connection=mysql://nova:nova@controller-name/nova
```

4. Nova SMF 서비스를 사용으로 설정합니다.

```
controller# svcadm enable nova-conductor
controller# svcadm enable nova-api-ec2 nova-api-osapi-compute \
nova-cert nova-conductor nova-objectstore nova-scheduler
```

▼ Horizon을 구성하는 방법

1. Horizon 패키지를 설치합니다.

```
horizon# pkg install horizon
```

2. Horizon용 인증서를 생성합니다.

다음 명령은 Horizon용 자체 서명된 인증서를 생성하고 OpenStack 대시보드 구성 파일을 Apache 구성 파일 디렉토리에 복사합니다. 자체 서명된 인증서를 만드는 방법은 [Apache SSL/TLS Strong Encryption: FAQ](#)를 참조하십시오.

```
controller# DASHBOARD=/etc/openstack_dashboard
controller# openssl req -new -x509 -nodes \
-out horizon.crt -keyout horizon.key
controller# mv horizon.crt horizon.key ${DASHBOARD}
controller# chmod 0600 ${DASHBOARD}/horizon.*
controller# sed \
-e "/SSLCertificateFile/s:/path.*:${DASHBOARD}/horizon.crt:" \
-e "/SSLCertificateKeyFile/d" \
-e "/SSLCertificateKeyFile/s:/path.*:${DASHBOARD}/horizon.key:" \
< /etc/apache2/2.2/samples-conf.d/openstack-dashboard-tls.conf \
> /etc/apache2/2.2/conf.d/openstack-dashboard-tls.conf
```

3. `~/conf.d/openstack-dashboard-tls.conf` 파일에서 다음 매개변수로 Horizon 패키지의 사이트 주소와 서버 이름을 지정합니다.

```
RedirectPermanent=site-address
ServerName=server-name
```

주 - 현재 샘플 3노드 구성에서는 두 매개변수가 컨트롤러 노드 시스템을 지정합니다.

4. 다음 작업 중 하나를 수행하여 새 구성 파일을 읽습니다.

- Apache 서비스가 사용 안함으로 설정된 경우 사용으로 설정합니다.

```
controller# svcadm enable apache22
```

- Apache 서비스가 온라인 상태인 경우 다시 시작합니다.

```
controller# svcs apache22
STATE      STIME    FMRI
online     Jul_07   svc:/network/http:apache22
controller# svcadm restart apache22
```

Compute 노드 구성

Compute 노드는 VM 인스턴스가 설치되는 위치입니다. 클라우드에 대한 Compute 노드가 여러 개 필요할 수도 있습니다.

Oracle Solaris 11에서 ZFS와 응용 프로그램 간 메모리 사용을 보다 잘 관리하려면 다음 예에 표시된 것과 같이, 노드에 대해 `usr_reserve_hint_pct` 매개변수를 설정합니다.

```
# echo "set user_reserve_hint_pct=80" >>/etc/system.d/site:kernel-zones-reserve
```

```
# reboot
```

여기서 *site*는 회사를 나타낼 수 있습니다.

이 매개변수를 다른 OpenStack 노드에도 설정해야 합니다.

이 매개변수에 대한 자세한 내용은 <https://support.oracle.com>에서 MOS 계정에 로그인한 다음 Document 1663862.1, *Memory Management Between ZFS and Applications in Oracle Solaris 11.2*를 검토하십시오.

▼ Compute 노드를 구성하는 방법

1. (선택사항) NTP를 설치 및 구성합니다.

“Network Time Protocol 설치” [33]를 참조하십시오.

2. Nova 패키지를 설치합니다.

```
compute1# pkg install nova
```

3. RAD(Remote Access Daemon)를 다시 시작합니다.

Nova는 RAD를 사용하여 Oracle Solaris 영역 프레임워크와 통신합니다.

```
compute1# svcadm restart rad:local
```

4. 인증 구성 정보를 지정합니다.

/etc/nova/api-paste.ini 파일에서 다음 매개변수의 주석 처리를 해제하고 해당 매개변수를 설정합니다. 해당 매개변수는 Keystone API 서비스의 위치 및 Nova 인증 정보를 지정합니다.

```
auth_uri = http://controller-name:5000/v2.0
identity_uri = http://controller-name:35357
admin_tenant_name = service
admin_user = nova
admin_password = nova-password
```

5. 인증 및 데이터베이스 매개변수와 관련 서비스를 구성합니다.

/etc/nova/nova.conf 파일에서 다음 매개변수의 주석 처리를 해제하고 해당 매개변수를 설정합니다. 이러한 매개변수는 추가 Keystone 인증 서비스 끝점, Glance API 서비스, Neutron API 서비스, RabbitMQ 서비스 및 Nova 관련 데이터베이스에 대한 데이터베이스 URI를 지정합니다.

```
keystone_ec2_url=http://controller-name:5000/v2.0/ec2tokens
glance_host=controller-name
neutron_url=http://controller-name:9696
neutron_admin_username=neutron
```

```
neutron_admin_password=neutron-password
neutron_admin_tenant_name=service
neutron_admin_auth_url=http://controller-name:5000/v2.0
rabbit_host=controller-name
connection=mysql://nova:nova@controller-name/nova
```

6. Compute 노드에 EVS 패키지를 설치합니다.

```
# pkg install evs
```

7. root 사용자에게 대한 Compute 노드에서 SSH 공개 키를 만듭니다.

이 키는 Solaris 영역 프레임워크가 EVS 컨트롤러에 액세스할 수 있도록 해줍니다.

ssh-keygen 명령을 root 사용자로 사용하고 root 사용자에게 대한 키를 만듭니다.

```
compute1# su - root -c "ssh-keygen -N '' -f /root/.ssh/id_rsa -t rsa"
```

8. 키를 EVS 컨트롤러 노드에 복사합니다.

이전 단계에서 생성한 SSH 공개 키인 /root/.ssh/id_rsa.pub를 EVS 컨트롤러가 실행 중인 노드(이 구성의 경우 네트워크 노드)에 복사합니다. 네트워크 노드에서 EVS 컨트롤러를 구성하는 경우 이 키가 참조됩니다.

이 단계 후에 Glance 이미지를 등록할 수 있습니다. [“이미지 만들기” \[67\]](#) 및 [“이미지 저장소에 이미지 추가” \[68\]](#)를 참조하십시오.

9. Nova Compute 서비스를 사용으로 설정합니다.

```
compute1# svcadm enable nova-compute
```

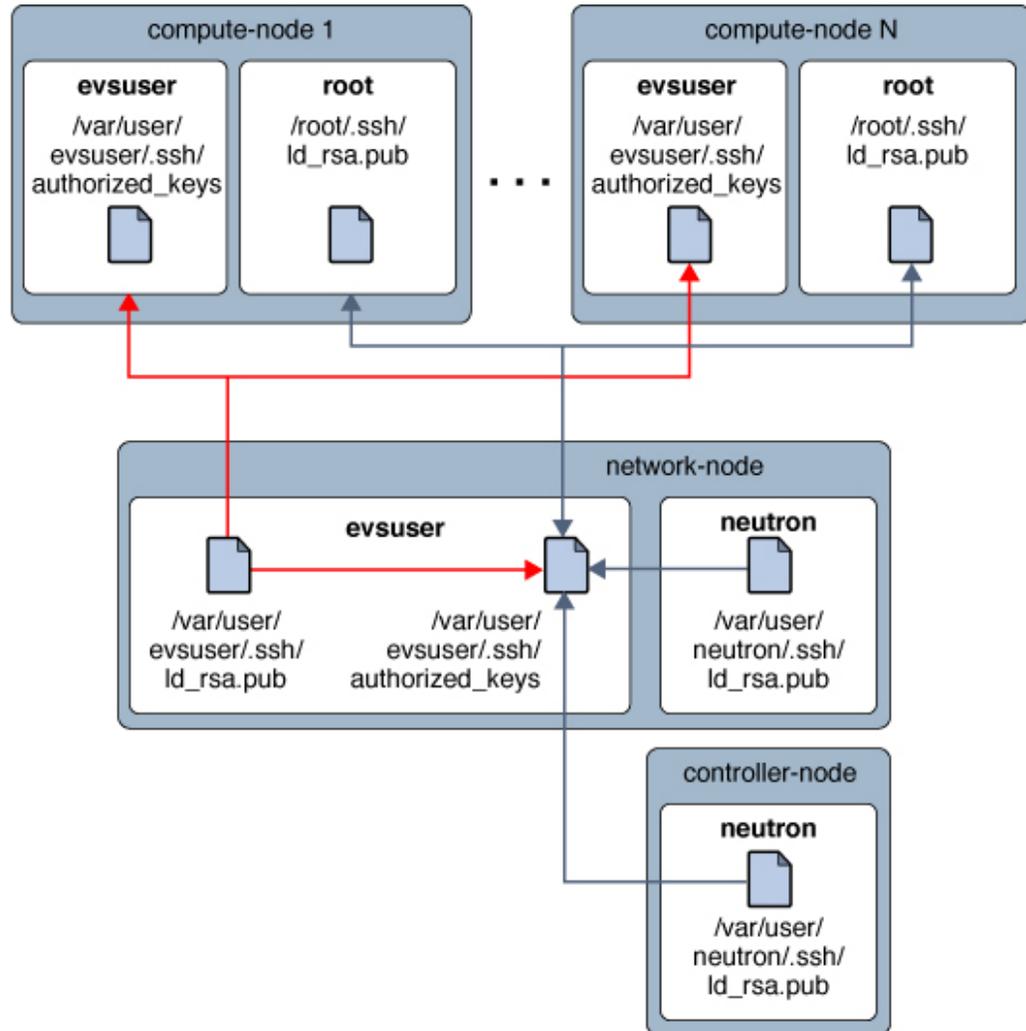
네트워크 노드 구성

네트워크 노드를 구성하려면 EVS(탄력적 가상 스위치)와 Neutron DHCP 에이전트를 구성해야 합니다. 선택적으로 Neutron L3(계층 3) 에이전트를 구성할 수 있습니다.

EVS는 OpenStack 네트워킹에 대한 백엔드를 구성하고 VLAN 또는 VXLAN을 사용하는 VM 인스턴스 간의 통신을 지원합니다. VM 인스턴스는 하나의 Compute 노드에 있거나 여러 Compute 노드에 걸쳐 있을 수 있습니다. EVS에 대한 자세한 내용은 [Oracle Solaris 11.2의 네트워크 가상화 및 네트워크 리소스 관리의 5장, “탄력적 가상 스위치 정보”](#)를 참조하십시오.

네트워크 노드를 구성할 때 evsuser의 SSH 공개 키를 Compute 노드와 네트워크 노드에 있는 각 evsuser의 authorized_keys 파일에 복사해야 합니다. 다음 그림에서 SSH 공개 키 배포를 참조하십시오. 그림에는 여러 Compute 노드가 구성된 것으로 가정합니다.

그림 8 EVS 컨트롤러 SSH 키 배포



Oracle Solaris 11에서 ZFS와 응용 프로그램 간 메모리 사용을 보다 잘 관리하려면 다음 예에 표시된 것과 같이, 노드에 대해 `usr_reserve_hint_pct` 매개변수를 설정합니다.

```
# echo "set user_reserve_hint_pct=80" >>/etc/system.d/site:kernel-zones-reserve
# reboot
```

여기서 `site`는 회사를 나타낼 수 있습니다.

이 매개변수를 다른 OpenStack 노드에도 설정해야 합니다.

이 매개변수에 대한 자세한 내용은 <https://support.oracle.com>에서 MOS 계정에 로그인한 다음 Document 1663862.1, *Memory Management Between ZFS and Applications in Oracle Solaris 11.2*를 검토하십시오.

▼ 네트워크 노드를 구성하는 방법

1. (선택사항) NTP를 설치 및 구성합니다.

“Network Time Protocol 설치” [33]를 참조하십시오.

2. Neutron 패키지를 설치합니다.

```
network# pkg install neutron
```

3. neutron 및 evsuser 사용자에게 대한 네트워크 노드에서 SSH 공개 키를 만듭니다.

```
network# su - neutron \
-c "ssh-keygen -N '' -f /var/lib/neutron/.ssh/id_rsa -t rsa"

network# su - neutron \
-c "ssh-keygen -N '' -f /var/lib/neutron/.ssh/id_rsa -t rsa"
```

4. EVS 권한이 부여된 키 파일에 키 파일을 결합합니다.

이전에 컨트롤러 및 Compute 노드에서 만들어진 SSH 공개 키를 2개의 새 키에 결합하고 evsuser에 대해 권한이 부여된 키 파일에 해당 조합을 연결합니다.

```
network# cat \
/path-to-neutron@controller/id_rsa.pub \
/path-to-root@compute1/id_rsa.pub \
/var/lib/neutron/.ssh/id_rsa.pub \
/var/user/evsuser/.ssh/id_rsa.pub \
>> /var/user/evsuser/.ssh/authorized_keys
```

5. evsuser의 SSH 공개 키인 /var/user/evsuser/.ssh/id_rsa.pub를 Compute 노드와 네트워크 노드에 있는 각 evsuser의 authorized_keys 파일에 복사합니다.

전체 클라우드 구성 컨텍스트에서 evsuser의 SSH 공개 키 배포에 대한 개요는 [그림 8](#)를 참조하십시오.

6. SSH 연결이 제대로 작동 중인지 확인합니다.

ssh 명령을 사용하여 EVS 컨트롤러에 연결하고, 각 연결에 대한 메시지가 표시되는 경우 yes를 입력합니다. OpenStack 아키텍처의 각 노드에서 이 단계를 수행합니다.

```
controller# su - neutron -c "ssh evsuser@network-name whoami"
compute1# su - root -c "ssh evsuser@network-name whoami"
network# su - neutron -c "ssh evsuser@network-name whoami"
network# su - root -c "ssh evsuser@network-name whoami"
```

7. EVS 컨트롤러 패키지를 설치합니다.

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

8. RAD를 다시 시작합니다.

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

9. EVS 컨트롤러를 구성합니다.

Nova 인스턴스가 통신할 가상 네트워크가 VLAN인지 아니면 VXLAN인지 결정합니다. 아래 예에서는 다음 VLAN 구성을 수행합니다.

- ID가 13인 VLAN은 클라우드 외부 연결에 사용되는 외부 네트워크에 대해 구성됩니다.
- ID 범위가 1000-2000인 VLAN은 EVS로 만들어진 서브넷에 대해 구성됩니다.

```
network# evsadm set-prop -p controller=ssh://evsuser@network-name
network# evsadm
network# evsadm set-controlprop -p l2-type=vlan
network# evsadm set-controlprop -p uplink-port=net1
network# evsadm set-controlprop -p vlan-range=13,1000-2000
```

10. 인증 매개변수 및 RabbitMQ 서비스의 위치를 구성합니다.

/etc/neutron/neutron.conf 파일에서 다음 매개변수의 주석 처리를 해제하고 해당 매개변수를 설정합니다. 이러한 매개변수는 추가 Keystone 인증 서비스 끝점, Glance API 서비스, Neutron API 서비스, RabbitMQ 서비스 및 Neutron 관련 데이터베이스에 대한 데이터베이스 URI를 지정합니다.

```
[keystone_authtoken]
signing_dir = /var/lib/neutron/keystone-signing
auth_host = 127.0.0.1
auth_port = 35357
auth_protocol = http
auth_uri = http://controller-IP:5000/v2.0
admin_tenant_name = service
admin_user = neutron
admin_password = neutron-password
identity_uri = http://controller-IP:35357
.
[DEFAULT]
core_plugin = neutron.plugins.evs.plugin.EVSNeutronPluginV2
allow_overlapping_ips = False

quotas]
quota_driver = neutron.plugins.evs.db.quotas_db.EVSDbQuotaDriver
```

11. Neutron DHCP 에이전트를 구성합니다.

/etc/neutron/dhcp_agent.ini 파일에서 다음 매개변수의 주석 처리를 해제하고 해당 매개변수를 설정하여 EVS 컨트롤러의 위치를 지정합니다.

```
evs_controller = ssh://evsuser@network-name
```

12. **(선택사항) Nova 인스턴스의 DNS 요청 분석에 사용할 기본 도메인을 지정합니다.**
 /etc/neutron/dhcp_agent.ini 파일에서 dhcp_domain 매개변수의 주석 처리를 해제하고 해당 매개변수를 Nova 인스턴스의 DNS 요청 분석에 사용할 기본 도메인으로 설정합니다.
13. **DHCP 에이전트를 사용으로 설정합니다.**

```
network# svcadm enable neutron-dhcp-agent
```
14. **Neutron L3 에이전트를 구성합니다.**
 이 단계는 선택사항이지만 강력히 권장됩니다.

Neutron L3 에이전트 구성

이 절에서는 외부 네트워크를 나타내는 가상 네트워크를 만드는 방법을 보여줍니다. 이 가상 네트워크에서 DHCP는 사용되지 않습니다. 대신 유동 IP 주소가 만들어집니다. 이러한 유동 IP 주소는 특정 테넌트에 지정되며 해당 테넌트에 속하는 사용자용으로 Nova VM 인스턴스에 지정될 수 있습니다. Neutron L3 에이전트는 Nova 인스턴스에 지정된 주소와 유동 IP 주소 간에 자동으로 일대일 NAT 매핑을 만듭니다.

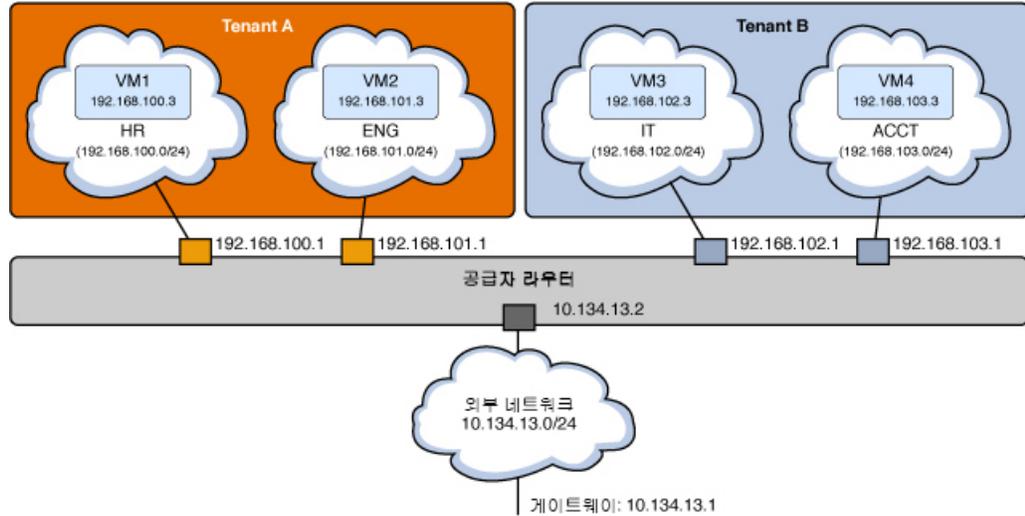
OpenStack Neutron의 Oracle Solaris 11.2 구현은 개인 네트워크 배치 모델과 함께 공급자 라우터를 지원합니다. 이 배치 모델에서 각 테넌트에는 하나 이상의 개인 네트워크가 포함될 수 있으며 모든 테넌트 네트워크는 동일한 라우터를 공유합니다. 데이터 관리자가 라우터를 만들고 소유하며 관리합니다. 테넌트의 네트워크 토폴로지 뷰에는 라우터가 표시되지 않습니다. 라우터가 한 개뿐이므로 테넌트 네트워크는 겹치는 IP 주소를 사용할 수 없습니다. 관리자는 테넌트 대신 개인 네트워크를 만듭니다.

기본적으로 이 모델의 라우터는 동일한 테넌트에 속하는 개인 네트워크 간의 경로 지정을 금지합니다. 즉, 특정 개인 네트워크의 VM 인스턴스는 다른 개인 네트워크의 VM 인스턴스와 통신할 수 없습니다. 해당하는 모든 인스턴스가 동일한 테넌트에 속하는 경우에도 마찬가지입니다. 이 동작을 변경하려면 /etc/neutron/l3_agent.ini 구성 파일에서 allow_forwarding_between_networks를 True로 설정하고 neutron-l3-agent SMF 서비스를 다시 시작하십시오.

이 모델의 라우터는 테넌트 VM 인스턴스 외부에 대한 연결을 가능하게 해줍니다. 해당 라우터는 인터페이스에서 라우터를 외부 네트워크에 연결하는 양방향 NAT를 수행합니다. 테넌트는 필요에 따라 또는 유동 IP 쿼터에 의해 허용되는 한도에 따라 여러 개의 유동 IP(공용 IP)를 만들어 외부 연결에 필요한 VM 인스턴스와 연결합니다.

다음 그림은 Oracle Solaris 11.2 Neutron 배치 모델을 보여줍니다. 그림에 뒤이어 그림에 대한 설명이 나열됩니다.

그림 9 개인 네트워크 모델을 사용하는 공급자 라우터



앞선 그림에 표시되는 모델에서 각 테넌트에는 2개의 내부 네트워크와 2개의 VM 인스턴스가 있습니다.

- | | |
|----------|--|
| Tenant A | <ul style="list-style-type: none"> ■ 서브넷이 192.168.100.0/24이며 게이트웨이가 192.168.100.1인 HR 네트워크 ■ 서브넷이 192.168.101.0/24이며 게이트웨이가 192.168.101.1인 ENG 네트워크 ■ HR에 연결되며 고정 IP 주소가 192.168.100.3인 VM1 ■ ENG에 연결되며 고정 IP 주소가 192.168.101.3인 VM2 |
| Tenant B | <ul style="list-style-type: none"> ■ 서브넷이 192.168.102.0/24이며 게이트웨이가 192.168.102.1인 IT 네트워크 ■ 서브넷이 192.168.103.0/24이며 게이트웨이가 192.168.103.1인 ACCT 네트워크 ■ IT에 연결되며 고정 IP 주소가 192.168.102.3인 VM3 ■ ACCT에 연결되며 고정 IP 주소가 192.168.103.3인 VM4 |

양방향 NAT 테이블은 다음 표와 같이 구성됩니다.

내부 IP	유동 IP
192.168.100.3	10.134.13.40

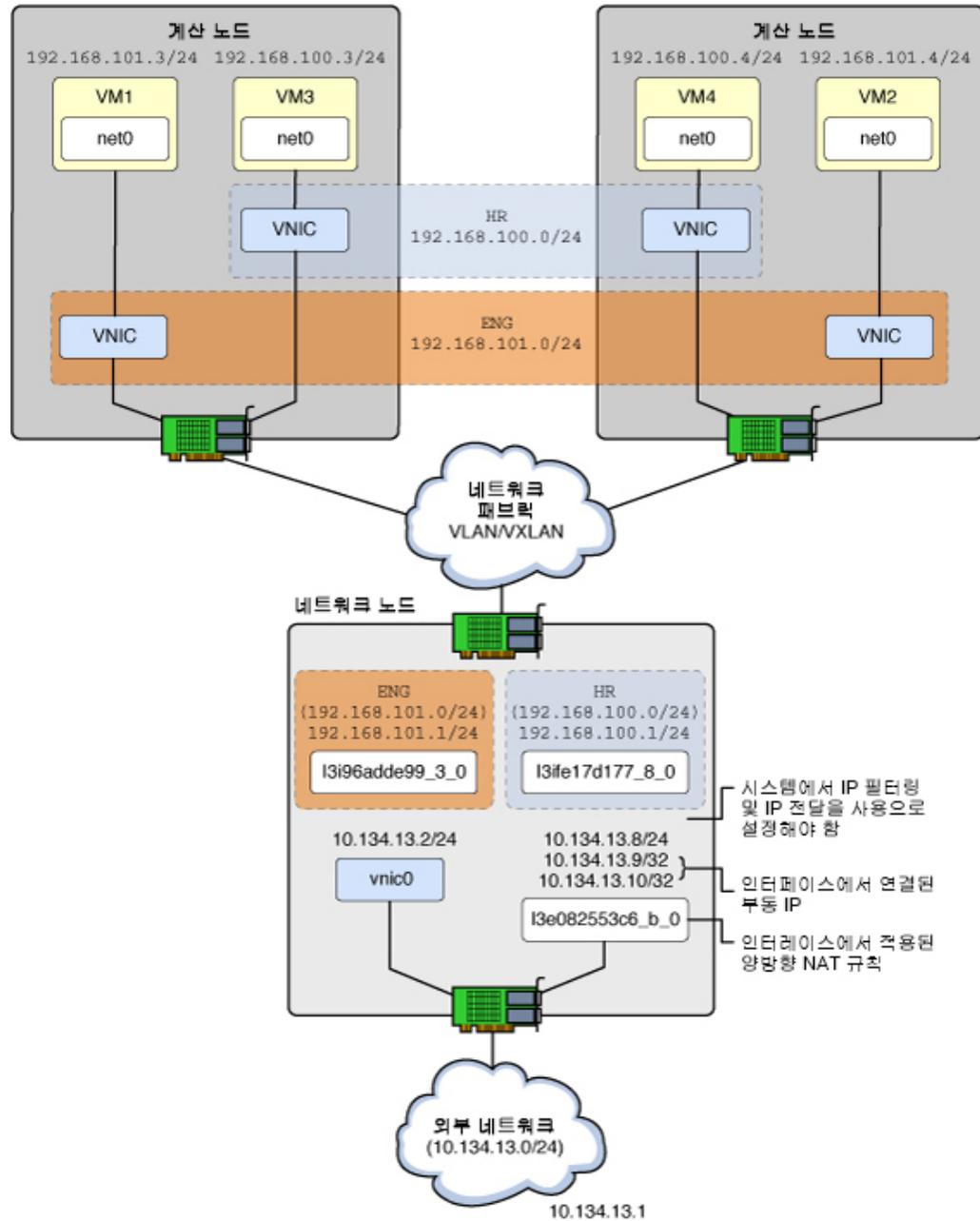
내부 IP	유동 IP
192.168.101.3	10.134.13.9

모든 게이트웨이 인터페이스는 neutron-l3-agent SMF 서비스가 실행되고 있는 노드에서 인스턴스화됩니다. 클라우드에 여러 개의 Neutron 인스턴스가 포함될 수 있지만 클라우드 당 하나의 neutron-l3-agent만 필요합니다.

외부 네트워크는 외부에서 연결할 수 있는 서브넷 10.134.13.0/24와 연결된 공급자 네트워크입니다. 테넌트는 이 네트워크에서 유동 IP 주소를 만들어 해당 VM 인스턴스와 연결합니다. VM1과 VM2에는 각각 유동 IP 10.134.13.40과 10.134.13.9가 연결됩니다. 해당 IP 주소를 통해 외부에서 VM1 및 VM2에 연결할 수 있습니다.

다음 다이어그램은 해당 테넌트 리소스가 네트워크 노드 및 2개의 Compute 노드에서 구성되는 방법을 보여줍니다. 그림에 뒤이어 그림에 표시된 일부 기능에 대한 설명이 나열됩니다.

그림 10 Neutron L3 에이전트 구성



VNIC	가상 네트워크 인터페이스입니다.
l3e...	L3 에이전트가 양방향 NAT가 수행되는 외부("e") 네트워크에서 VNIC를 만들었습니다.
l3i...	L3 에이전트가 기본 게이트웨이 IP 주소를 사용하는 내부("i") 네트워크에서 VNIC를 만들었습니다.

다음 목록은 이 구성 예에서 IP 주소가 사용될 방법을 보여줍니다.

10.134.13.1

기본 게이트웨이

10.134.13.2 - 10.134.13.7

테넌트에 OpenStack API(예: Nova, Cinder, Glance)를 노출시키기 위해 별도로 설정된 IP 주소

10.134.13.9 - 10.134.13.254

테넌트 VM 인스턴스에 대한 유동 IP 주소

▼ Neutron L3 에이전트를 구성하는 방법

이 절차에서는 service 테넌트를 사용하여 데이터 센터의 모든 테넌트에 사용될 라우터, 외부 네트워크 및 외부 서브넷을 만드는 방법을 보여줍니다.

데이터 센터 관리자가 이 절차를 수행합니다. OpenStack 대시보드에서는 한 번에 하나씩 테넌트의 리소스를 관리할 수 있기 때문에 사용자는 명령줄을 사용하여 공유된 단일 라우터를 구성하고 여러 테넌트의 네트워크 및 서브넷을 연결해야 합니다.

다음 단계를 수행할 때는 [그림 10](#)을 참조하십시오.

시작하기 전에 이 절차를 수행하기 전에 내부 네트워크 구성을 완료해야 합니다.

1. Solaris IP 필터 기능을 사용으로 설정합니다.

```
network# svcadm enable ipfilter
```

2. 전체 호스트에서 IP 전달을 사용으로 설정합니다.

```
network# ipadm set-prop -p forwarding=on ipv4
network# ipadm set-prop -p forwarding=on ipv6
```

3. EVS가 올바르게 구성되고 외부 네트워크에 필요한 VLAN ID를 사용하는지 확인합니다.

다음 예제에서 VLAN ID 및 범위 값은 [9단계](#)에서 완성한 EVS 구성을 기반으로 합니다.

```
network# evsadm show-controlprop -p vlan-range,l2-type
PROPERTY  PERM  VALUE          DEFAULT  HOST
l2-type   rw    vlan          vlan     --
```

```
vlan-range rw 13,1000-2000 -- --
```

4. service 테넌트가 있는지 확인합니다.

```
network# keystone tenant-list
```

5. 공급자 라우터를 만듭니다.

공급자 라우터가 service 테넌트에 OpenStack neutron 사용자로 만들어집니다. 새 라우터의 UUID(id)를 기록해 둡니다. 이 UUID는 다음 단계에서 사용합니다.

```
network# export OS_USERNAME=neutron
network# export OS_PASSWORD=neutron-password
network# export OS_TENANT_NAME=service
network# export OS_AUTH_URL=http://controller-name:5000/v2.0
network# neutron router-create provider_router
Created a new router:
+-----+-----+
| Field          | Value          |
+-----+-----+
| admin_state_up | True           |
| external_gateway_info |              |
| id             | 181543df-40d1-4514-ea77-fddd78c389ff |
| name           | provider_router |
| status         | ACTIVE        |
| tenant_id      | f164220cb02465db929ce520869895fa |
+-----+-----+
```

6. L3 에이전트 구성 파일을 업데이트합니다.

이전 단계의 라우터 UUID(id)를 사용하여 /etc/neutron/l3_agent.ini 파일에서 router_id 값을 업데이트합니다.

```
router_id = 181543df-40d1-4514-ea77-fddd78c389ff
```

7. neutron-l3-agent SMF 서비스를 사용으로 설정합니다.

```
network# svcadm enable neutron-l3-agent
```

8. 외부 네트워크를 만듭니다.

가상 네트워크가 service 테넌트에 OpenStack neutron 사용자로 만들어집니다.

```
network# neutron net-create --provider:network_type=vlan \
--provider:segmentation_id=13 --router:external=true external_network
Created a new network:
+-----+-----+
| Field          | Value          |
+-----+-----+
| admin_state_up | True           |
| id             | f67f0d72-0ddf-11e4-9d95-e1f29f417e2f |
| name           | external_network |
| provider:network_type | vlan          |
| provider:segmentation_id | 13           |
+-----+-----+
```

router:external	True	
shared	False	
status	ACTIVE	
subnets		
tenant_id	f164220cb02465db929ce520869895fa	

9. 서브넷을 외부 네트워크와 연결합니다.

외부 네트워크에 대해 연결된 서브넷 네트워크를 만들어 DHCP가 사용 안함으로 설정되었으며 서브넷이 외부 경로 지정에 사용될 것임을 지정합니다. 유동 IP 주소가 지정될 위치에서 경로 지정이 가능한 IP 서브넷을 지정합니다. 이 예에서는 서브넷이 VLAN ID 13과 연관됩니다.

```
network# neutron subnet-create --enable-dhcp=False \
--allocation-pool start=10.134.13.8,end=10.134.13.254 \
--name external_subnet external_network 10.134.13.0/24
Created a new subnet:
```

Field	Value
allocation_pools	{"start": "10.134.13.8", "end": "10.134.13.254"}
cidr	10.134.13.0/24
dns_nameservers	
enable_dhcp	False
gateway_ip	10.134.13.1
host_routes	
id	5d9c8958-0de0-11e4-9d96-e1f29f417e2f
ip_version	4
name	external_subnet
network_id	f67f0d72-0ddf-11e4-9d95-e1f29f417e2f
tenant_id	f164220cb02465db929ce520869895fa

10. 라우터에 외부 네트워크를 추가합니다.

다음 명령에서 첫번째 UUID는 provider_router UUID이며 두번째 UUID는 external_network UUID입니다.

```
network# neutron router-gateway-set \
181543df-40d1-4514-ea77-fddd78c389ff \
f67f0d72-0ddf-11e4-9d95-e1f29f417e2f
Set gateway for router 181543df-40d1-4514-ea77-fddd78c389ff
network# neutron router-list -c name -c external_gateway_info
```

name	external_gateway_info
provider_router	{"network_id": "f67f0d72-0ddf-11e4-9d95-e1f29f417e2f"}

11. 라우터에 테넌트의 개인 네트워크를 추가합니다.

neutron net-list 명령으로 표시되는 네트워크가 이전에 구성된 것입니다.

```
network# keystone tenant-list
+-----+-----+-----+
|          id          | name | enabled |
+-----+-----+-----+
| 511d4cb9ef6c40beadc3a664c20dc354 | demo | True |
| f164220cb02465db929ce520869895fa | service | True |
+-----+-----+-----+
network# neutron net-list --tenant-id=511d4cb9ef6c40beadc3a664c20dc354
+-----+-----+-----+
| id                  | name | subnets |
+-----+-----+-----+
| c0c15e0a-0def-11e4-9d9f- | HR | c0c53066-0def-11e4-9da0- |
| e1f29f417e2f          |    | e1f29f417e2f 192.168.100.0/24 |
| ce64b430-0def-11e4-9da2- | ENG | ce693ac8-0def-11e4-9da3- |
| e1f29f417e2f          |    | e1f29f417e2f 192.168.101.0/24 |
+-----+-----+-----+
```

다음 명령에서 첫번째 UUID는 provider_router UUID이며 두번째 UUID는 HR 서브넷 UUID입니다.

```
network# neutron router-interface-add \
181543df-40d1-4514-ea77-fddd78c389ff \
c0c53066-0def-11e4-9da0-e1f29f417e2f (HR subnet UUID)
Added interface 7843841e-0e08-11e4-9da5-e1f29f417e2f to router 181543df-40d1-4514-ea77-fddd78c389ff.
```

다음 명령에서 첫번째 UUID는 provider_router UUID이며 두번째 UUID는 ENG 서브넷 UUID입니다.

```
network# neutron router-interface-add \
181543df-40d1-4514-ea77-fddd78c389ff \
ce693ac8-0def-11e4-9da3-e1f29f417e2f
Added interface 89289b8e-0e08-11e4-9da6-e1f29f417e2f to router 181543df-40d1-4514-ea77-fddd78c389ff.
```

- 참조
- [L3 에이전트 구성을 관찰하는 방법 \[60\]](#)
 - [“알려진 제한 사항” \[71\]](#)

▼ 테넌트 사용자로 유동 IP 주소를 만들고 연관시키는 방법

이 절차는 OpenStack Horizon 대시보드를 사용하는 테넌트 사용자가 수행합니다.

1. **OpenStack Dashboard(OpenStack 대시보드)에 로그인합니다.**
테넌트 사용자의 자격 증명을 사용하여 [OpenStack 대시보드에 액세스하는 방법 \[18\]](#)에 설명된 대로 로그인합니다.
2. **Project(프로젝트) -> Access & Security(접근 & 보안) -> Floating IPs(유동 IP)를 선택합니다.**

3. **external_network**를 선택합니다.
4. **Allocate IP(IP 할당) 버튼**을 누릅니다.
Floating IPs(유동 IP) 탭은 유동 IP 주소 10.134.13.9가 할당되었음을 보여줍니다.
5. **Associate(연결) 버튼**을 누릅니다.
6. **플다운 메뉴에서 VM 인스턴스의 포트**를 선택합니다.
Project(프로젝트) -> Instances(인스턴스) 창은 유동 IP 주소가 VM 인스턴스와 연결되었음을 보여줍니다.
VM 인스턴스를 구동 중인 동안 Keypair(SSH 공개 키)를 선택한 경우 VM 인스턴스의 root 사용자에게 대한 authorized_keys 파일에 해당 SSH 키가 추가됩니다.
7. **실행 중인 VM 인스턴스에 로그인**합니다.

```
global# ssh root@10.134.13.9
Last login: Fri Jul 18 00:37:39 2014 from 10.132.146.13
Oracle Corporation      SunOS 5.11      11.2      June 2014
root@host-192-168-101-3:~# uname -a
SunOS host-192-168-101-3 5.11 11.2 i86pc i386 i86pc
root@host-192-168-101-3:~# zoneadm list -cv
ID NAME          STATUS  PATH          BRAND  IP
2 instance-00000001 running /            solaris excl
root@host-192-168-101-3:~# 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/dhcp  inherited ok     --      192.168.101.3/24
```

▼ L3 에이전트 구성을 관찰하는 방법

IP 필터 명령(예: ipf, ippool 및 ipnat) 및 네트워킹 명령(예: dladm 및 ipadm)을 사용하여 neturon-l3-agent로 완료된 구성을 관찰하고 문제를 해결할 수 있습니다.

1. **neutron-l3-agent로 만들어진 VNIC**를 표시합니다.

```
network# dladm show-vnic
LINK          OVER      SPEED  MACADDRESS      MACADDRTYPE  VIDS
l3i7843841e_0_0 net1      1000   2:8:20:42:ed:22 fixed          200
l3i89289b8e_0_0 net1      1000   2:8:20:7d:87:12 fixed          201
l3ed527f842_0_0 net0      100    2:8:20:9:98:3e  fixed          0
```

2. **neutron-l3-agent로 만들어진 IP 주소**를 표시합니다.

```
network# ipadm
NAME          CLASS/TYPE STATE  UNDER  ADDR
```

```

l3ed527f842_0_0      ip      ok      --      --
  l3ed527f842_0_0/v4  static ok      --      10.134.13.8/24
  l3ed527f842_0_0/v4a static ok      --      10.134.13.9/32
l3i7843841e_0_0      ip      ok      --      --
  l3i7843841e_0_0/v4  static ok      --      192.168.100.1/24
l3i89289b8e_0_0      ip      ok      --      --
  l3i89289b8e_0_0/v4  static ok      --      192.168.101.1/24

```

3. IP 필터 규칙을 표시합니다.

```

network# ipfstat -io
empty list for ipfilter(out)
block in quick on l3i7843841e_0_0 from 192.168.100.0/24 to pool/4386082
block in quick on l3i89289b8e_0_0 from 192.168.101.0/24 to pool/8226578
network# ippool -l
table role = ipf type = tree number = 8226578
{ 192.168.100.0/24; };
table role = ipf type = tree number = 4386082
{ 192.168.101.0/24; };

```

4. IP NAT 규칙을 표시합니다.

```

network# ipnat -l
List of active MAP/Redirect filters:
bimap l3ed527f842_0_0 192.168.101.3/32 -> 10.134.13.9/32
List of active sessions:
BIMAP 192.168.101.3 22 <- -> 10.134.13.9 22 [10.132.146.13 36405]

```


◆◆◆ 4 장

가상 시스템 인스턴스 만들기

이 장에서는 OpenStack 클라우드에서 VM(가상 시스템) 인스턴스를 프로비전하는 방법에 대해 설명합니다. 각 VM 인스턴스는 테넌트에 속합니다. 사용자는 하나 이상의 테넌트에서 하나 이상의 VM 인스턴스를 만들어 사용할 수 있습니다. OpenStack VM 인스턴스 생성 및 관리에 대한 일반적인 정보는 [OpenStack End User Guide](#)를 참조하십시오.

VM 인스턴스를 만들려면 flavor 및 이미지가 필요합니다. 이 장은 다음 내용으로 구성되어 있습니다.

- “Flavor 관리” [63]
- “이미지 관리” [66]
- “VM 인스턴스 만들기” [69]

사용할 OpenStack 명령에 대한 환경 변수를 설정하는 파일을 만들 수 있습니다. 환경 변수를 설정하지 않을 경우 각 명령에서 옵션을 지정해야 합니다. 자세한 내용은 [OpenStack Command-Line Interface Reference](#)를 참조하십시오.

Flavor 관리

*flavor*는 VM 인스턴스 유형 또는 가상 하드웨어 템플릿입니다. *flavor*는 일련의 가상 시스템 리소스(예: 가상 CPU 수, 메모리 양 및 VM 인스턴스에 지정된 디스크 공간)를 지정합니다. Solaris에서 *flavor*에는 기본 영역의 브랜드인 *solaris*(비전역 영역의 경우) 및 *solaris-kz*(커널 영역의 경우)도 포함됩니다. 가상 CPU가 16개이며 RAM이 16384MB인 커널 영역을 인스턴스 *flavor*의 예로 들 수 있습니다.

*flavor*에 대한 일반적인 정보는 *OpenStack Cloud Administrator Guide*의 “[Flavors](#)” 섹션을 참조하십시오.

Flavor에 대한 정보 표시

`nova flavor-list` 명령은 사용 가능한 *flavor* 목록을 표시합니다. VM 인스턴스를 만들 때 *flavor*의 이름 또는 ID를 사용하게 됩니다. *flavor* 목록에 `extra-specs`를 포함시키려면 `nova`

flavor-list 명령에 --extra-specs 옵션을 지정하십시오. extra_specs 값에 대한 자세한 내용은 “Flavor 사양 수정” [65]을 참조하십시오.

다음 예에서 RXTX_Factor 및 Is_Public 열은 공간 절약을 위해 출력에서 생략됩니다. 해당 열에 대한 자세한 내용은 [OpenStack Command-Line Interface Reference](#)를 참조하십시오.

```
$ nova flavor-list
```

ID	Name	Memory_MB	Disk	Ephemeral	Swap	VCPUs
1	Oracle Solaris kernel zone - tiny	2048	10	0		1
10	Oracle Solaris non-global zone - xlarge	16384	80	0		32
2	Oracle Solaris kernel zone - small	4096	20	0		4
3	Oracle Solaris kernel zone - medium	8192	40	0		8
4	Oracle Solaris kernel zone - large	16384	40	0		16
5	Oracle Solaris kernel zone - xlarge	32768	80	0		32
6	Oracle Solaris non-global zone - tiny	2048	10	0		1
7	Oracle Solaris non-global zone - small	3072	20	0		4
8	Oracle Solaris non-global zone - medium	4096	40	0		8
9	Oracle Solaris non-global zone - large	8192	40	0		16

다음 명령은 지정된 flavor에 대한 세부정보를 표시합니다. 명령의 첫번째 버전은 flavor 이름을 지정하며, 두번째 버전은 flavor ID를 지정합니다. 두 명령의 출력은 동일합니다.

```
$ nova flavor-show 'Oracle Solaris kernel zone - large'
$ nova flavor-show 4
```

Property	Value
name	Oracle Solaris kernel zone - large
ram	16384
OS-FLV-DISABLED:disabled	False
vcpus	16
extra_specs	{'zonecfg:brand': 'solaris-kz'}
swap	
os-flavor-access:is_public	True
rxtx_factor	1.0
OS-FLV-EXT-DATA:ephemeral	0
disk	40
id	4

Flavor 사양 수정

Flavor 사양을 변경하려면 `flavor-key` 하위 명령을 사용하여 `extra_specs` 값을 수정하십시오.

```
nova flavor-key flavor action key=value [key=value ...]
```

flavor flavor의 이름 또는 ID입니다.

action set 또는 unset입니다.

key=value *key*는 사양 이름입니다. *value*는 해당 사양에 대한 새 값입니다. *action*이 unset인 경우 *key*만 지정합니다.

```
$ nova flavor-key 4 set zonecfg:bootargs=-v
$ nova flavor-show 4
+-----+-----+
| Property                | Value                                |
+-----+-----+
| name                    | Oracle Solaris                      |
|                          | kernel zone - large                 |
| ram                     | 16384                                |
| OS-FLV-DISABLED:disabled | False                                |
| vcpus                   | 16                                    |
| extra_specs             | {u'zonecfg:brand':                  |
|                          |   u'solaris-kz', u'zonecfg:bootargs': u'-v'} |
| swap                    |                                       |
| os-flavor-access:is_public | True                                |
| rxtx_factor             | 1.0                                  |
| OS-FLV-EXT-DATA:ephemeral | 0                                    |
| disk                    | 40                                    |
| id                      | 4                                    |
+-----+-----+
```

다음 등록 정보는 커널 영역과 비전역 영역에서 지원됩니다.

- zonecfg:bootargs
- zonecfg:brand
- zonecfg:hostid

다음 등록 정보는 비전역 영역에서만 지원됩니다.

- zonecfg:file-mac-profile
- zonecfg:fs-allowed
- zonecfg:limitpriv

기타 영역 구성 등록 정보는 OpenStack에서 지원되지 않습니다. 해당 영역 구성 등록 정보에 대한 자세한 내용은 [zonecfg\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

`sc_profile` 키를 사용하여 시스템 구성 프로파일을 지정할 수 있습니다.

```
$ nova flavor-key 4 set sc_profile=/system/volatile/profile/sc_profile.xml
```

nova flavor-key 명령으로는 extra_specs 값만 수정하거나 추가할 수 있습니다. 기존 flavor의 다른 사양(예: RAM)을 수정하려면 flavor를 삭제하고 수정된 flavor를 동일한 이름으로 만들어야 합니다. flavor 삭제 및 생성에 대한 자세한 내용은 [OpenStack Admin User Guide](#)를 참조하십시오.

이미지 관리

가상 시스템 이미지 또는 이미지는 부팅 가능 운영체제가 설치된 가상 디스크를 포함하는 단일 파일입니다. 이미지는 가상 시스템 파일 시스템에 대한 템플리트를 제공합니다.

대시보드 이외에 glance 및 nova 명령줄 클라이언트나 이미지 서비스 및 Compute API를 통해서도 이미지를 관리할 수 있습니다.

이미지에 대한 정보 표시

nova image-list 명령은 사용 가능한 이미지 목록을 표시합니다. VM 인스턴스를 만들 때 이미지의 이름 또는 ID를 사용하게 됩니다.

```
$ nova image-list
+-----+-----+-----+-----+
| ID                | Name                | Status | Server |
+-----+-----+-----+-----+
| e422aae1-b0ba-618c-85d3-a214059800e2 | Solaris Kernel Zone | ACTIVE |        |
| e82aa857-ec92-4859-f530-deb89274863e | Solaris Non-global Zone | ACTIVE |        |
+-----+-----+-----+-----+
```

glance image-list 명령은 디스크 포맷, 컨테이너 형식, 이미지 크기 등 추가 정보를 표시합니다.

```
$ glance image-list --human-readable
```

nova image-show 및 glance image-show 명령은 지정된 이미지에 대한 세부 정보를 표시합니다.

```
$ nova image-show 'Solaris Kernel Zone'
+-----+-----+
| Property                | Value                |
+-----+-----+
| OS-EXT-IMG-SIZE:size    | 1547458560           |
| created                  | 2014-06-29T15:40:49Z |
| id                       | e422aae1-b0ba-618c-85d3-a214059800e2 |
| metadata architecture  | x86_64               |
| metadata hypervisor_type | solariszones         |
| metadata vm_mode        | solariszones         |
| minDisk                  | 0                     |
+-----+-----+
```

```

| minRam          | 0          |
| name           | Solaris Kernel Zone |
| progress       | 100        |
| status         | ACTIVE     |
| updated        | 2014-06-29T15:40:55Z |
+-----+
$ glance image-show 'Solaris Kernel Zone'
+-----+
| Property          | Value          |
+-----+
| Property 'architecture' | x86_64        |
| Property 'hypervisor_type' | solariszones  |
| Property 'vm_mode'      | solariszones  |
| checksum            | b2fc9560c15603c7663326db82d5ddaa |
| container_format     | bare          |
| created_at          | 2014-06-29T15:40:49.108578 |
| deleted             | False         |
| disk_format         | raw           |
| id                  | e422aae1-b0ba-618c-85d3-a214059800e2 |
| is_public           | True          |
| min_disk            | 0             |
| min_ram             | 0             |
| name                | Solaris Kernel Zone |
| owner               | 7461d4a9f5a64af9a01ae4e84e08c182 |
| protected           | False         |
| size                | 1547458560    |
| status              | active        |
| updated_at          | 2014-06-29T15:40:55.769756 |
+-----+

```

이미지 만들기

Solaris에서 OpenStack 이미지는 통합 아카이브이며 Oracle Solaris 11.2를 실행 중이어야 합니다. `archiveadm` 명령을 사용하여 Oracle Solaris 11.2를 실행하는 전역, 비전역 및 커널 영역에서 새 통합 아카이브를 만들 수 있습니다. OpenStack에서 사용할 이미지를 Glance 저장소로 업로드하십시오.

통합 아카이브는 복제 아카이브 또는 복구 아카이브일 수 있습니다. 현재 활성 상태의 부팅 환경을 기반으로 복제 아카이브를 만들거나 모든 부팅 환경과 시스템 구성 정보가 포함된 복구 아카이브를 만드십시오. 복제 아카이브에는 OS 인스턴스의 시스템 구성 정보가 포함되지 않습니다. 복제 아카이브의 경우 설치 프로그램에서 재구성을 강제로 수행할 수도 있고, SC (시스템 구성) 프로파일에서 구성을 수동으로 제공할 수도 있습니다. 복제 아카이브에는 비 활성 BE 등도 포함되지 않습니다. 시스템의 모든 요소가 필요한 경우 복구 통합 아카이브를 사용하십시오. 통합 아카이브에 대한 자세한 내용은 [Oracle Solaris 11.2의 시스템 복구 및 복제용 Unified Archive 사용](#)을 참조하십시오.

다음 명령은 myzone이라는 실행 중인 비전역 영역의 통합 아카이브를 캡처합니다.

```

global# zonecfg -z myzone create
global# zoneadm -z myzone install

```

```
global# zlogin myzone \
'sed /^PermitRootLogin/s/no$/without-password/ \
< /etc/ssh/sshd_config > /system/volatile/sed.$$ ; \
cp /system/volatile/sed.$$ /etc/ssh/sshd_config'
global# archiveadm create -z myzone /var/tmp/myzone.uar
```

기존 VM 인스턴스의 스냅샷을 만들어 OpenStack 이미지를 만들 수도 있습니다. nova image-create 명령을 통해 실행 중인 VM 인스턴스의 스냅샷을 생성하여 이미지를 만드십시오.

VM 인스턴스를 만드는 데 사용할 이미지를 만드는 것 이외에 데이터 백업을 위한 사용자 정의 이미지를 사용하거나 VM 인스턴스를 자동 복구할 수도 있습니다. 자동 복구 이미지는 VM 인스턴스가 rescue 모드로 설정될 때 부팅되는 특수 유형의 이미지입니다. 자동 복구 이미지를 통해 관리자는 VM 인스턴스에 대한 파일 시스템을 마운트하여 문제를 해결할 수 있습니다.

이미지 저장소에 이미지 추가

OpenStack 이미지 서비스인 Glance는 디스크 및 서버 이미지에 대한 저장소, 검색, 등록 및 전달 서비스를 제공합니다. 레지스트리 서버는 클라이언트에 이미지 메타데이터 정보를 제공하는 이미지 서비스입니다. 이미지 캐시는 이미지가 요청될 때마다 이미지 서비스가 이미지 서버에서 이미지를 다시 다운로드하지 않고 로컬 호스트에서 이미지를 가져오는 데 사용됩니다.

다음 명령은 이전 절에서 만들어진 통합 아카이브를 Glance 저장소로 업로드합니다. raw를 포맷 유형으로 사용합니다. architecture 등록 정보를 지정해야 합니다.

```
global# glance image-create --container-format bare --disk-format raw \
--is-public true --name "Oracle Solaris 11.2 x86 NGZ" \
--property architecture=x86_64 \
--property hypervisor_type=solariszones \
--property vm_mode=solariszones < /var/tmp/myzone.uar
```

glance image-create 명령은 이미지를 업로드하고 모든 등록 정보 값을 한 번에 설정할 수 있습니다. 다음 스크립트는 architecture 등록 정보가 현재 호스트의 아키텍처로 설정된 이미지를 업로드하는 방법을 보여줍니다.

```
#!/bin/ksh

# Upload Unified Archive image to glance with proper Solaris decorations

arch=$(archiveadm info -p $1|grep ^archive|cut -d '|' -f 4)

if [[ "$arch" == "i386" ]]; then
    imgarch=x86_64
else
    imgarch=sparc64
fi
```

```

name=$(basename $1 .uar)

export OS_USERNAME=glance
export OS_PASSWORD=glance
export OS_TENANT_NAME=service
export OS_AUTH_URL=http://controller-name:5000/v2.0

glance image-create --name $name --container-format bare --disk-format raw --owner service
--file $1 --is-public True --property architecture=$imgarch --property
hypervisor_type=solariszones
--property vm_mode=solariszones --progress

```

VM 인스턴스 만들기

VM 인스턴스를 만들려면 flavor, 이미지 및 네트워크가 필요합니다.

▼ 명령줄 인터페이스를 사용하여 VM 인스턴스를 만드는 방법

1. 새 VM 인스턴스를 만들 테넌트를 선택합니다.

VM 인스턴스를 만들려면 명령에 테넌트 이름 또는 ID를 지정해야 합니다.

```

$ keystone tenant-list
+-----+-----+-----+
|          id          | name | enabled |
+-----+-----+-----+
| 6ea34f7dafa5ce3c9a1b9de659e59d77 | demo | True |
| 0bda9b63b800ca808031a38637d50f3e | service | True |
+-----+-----+-----+

```

2. 새 VM 인스턴스를 만들 기반으로 사용할 이미지를 선택합니다.

VM 인스턴스를 만들려면 명령에 이미지 이름 또는 ID를 지정해야 합니다. 이미지 ID를 비롯하여 각 이미지의 사양을 표시하는 방법은 “[이미지에 대한 정보 표시](#)” [66]를 참조하십시오.

전역 영역보다 릴리스 레벨이 낮은 비전역 영역 VM 인스턴스를 배치하면 설치 시 VM 인스턴스가 자동으로 전역 영역 릴리스 레벨로 업그레이드됩니다. 전역 영역보다 릴리스 레벨이 높은 비전역 영역 VM 인스턴스를 배치하려는 시도는 실패합니다.

3. 새 VM 인스턴스를 만들 기반으로 사용할 flavor를 선택합니다.

필요한 사양이 지정된 flavor가 있는지 확인합니다. 필요한 사양을 flavor에 추가하거나 필요한 사양이 지정된 새 flavor를 만듭니다. flavor ID 및 extra-specs를 비롯하여 각 flavor의 사양을 표시하는 방법은 “[Flavor에 대한 정보 표시](#)” [63]를 참조하십시오.

4. 사용할 새 VM 인스턴스에 대한 네트워크를 선택합니다.

VM 인스턴스를 만들려면 명령에 네트워크 이름 또는 ID를 지정해야 합니다. `neutron net-list` 명령으로 1단계에서 선택한 테넌트에 대한 네트워크가 표시되지 않을 경우 `neutron net-create` 명령을 사용하여 이 테넌트에 대한 네트워크를 만듭니다. 자세한 내용은 [OpenStack Command-Line Interface Reference](#)를 참조하십시오. 네트워크의 ID를 기록해 둡니다.

5. 인스턴스를 만듭니다.

`nova boot` 명령을 사용하여 Compute 인스턴스를 만들고 부팅합니다. *imageID*는 2단계에서, *flavorID*는 3단계에서, *nicID*는 4단계에서 지정한 것입니다. 자세한 내용은 [OpenStack Command-Line Interface Reference](#)를 참조하십시오.

```
# nova boot --image imageID --flavor flavorID --nic net-id=nicID
```

6. 사용할 새 VM 인스턴스에 대한 유동 IP 주소를 선택합니다.

`neutron floatingip-list` 명령을 사용하여 단계 1에서 선택한 테넌트에 대한 유동 IP 주소를 표시합니다. 필요한 경우 `neutron floatingip-create` 명령을 사용하여 이 테넌트에 대한 유동 IP 주소를 만듭니다. 유동 IP 주소의 ID를 기록해 둡니다.

7. 유동 IP 주소를 새 VM 인스턴스와 연결합니다.

`neutron floatingip-associate` 명령을 사용하여 단계 6의 유동 IP 주소를 새 VM 인스턴스와 연결합니다.

◆◆◆ 5 장

OpenStack 문제 해결

이 장에서는 다음 정보를 제공합니다.

- 이 릴리스의 알려진 문제
- OpenStack과 연관된 로그 파일 사용
- 문제 조사 및 해결

알려진 제한 사항

Oracle Solaris 11.2의 OpenStack(Havana 2013.2.3)과 관련하여 알려진 문제는 다음과 같습니다.

- OpenStack 대시보드에서 VM 인스턴스에 대한 원격 콘솔 액세스가 지원되지 않습니다. 대신 대시보드를 사용하여 SSH Keypair를 업로드하십시오. 이 Keypair는 해당 VM 인스턴스에 대한 root의 authorized_keys 파일에 푸시됩니다.
- Neutron은 네트워크 가상화용 플러그인을 하나만 지원하므로 Solaris를 실행하는 Nova 노드만 완전히 지원됩니다.
- 현재 Cinder 볼륨 첨부이 비전역 영역에서 지원되지 않습니다.
- VM 인스턴스에서 Oracle Solaris 11.2가 실행 중이어야 합니다.
- VM 인스턴스 크기 조정이 지원되지 않습니다.
nova resize 명령이 지원되지 않습니다. nova resize 명령의 출력에는 명령이 완료된 것으로 표시될 수 있지만 nova resize-confirm 명령은 인스턴스 크기 조정을 확인할 수 없다고 보고하며 nova show 명령은 인스턴스의 크기가 조정되지 않았다고 표시합니다.
- VM 인스턴스 라이브 마이그레이션이 지원되지 않습니다.
nova live-migration 명령이 지원되지 않습니다.
- Cinder 백업이 지원되지 않습니다.
cinder-backup 서비스는 cinder 패키지를 설치할 때 설치되지만 사용 안함으로 설정되므로 사용 안함 상태로 유지해야 합니다.
- 대시보드의 Launch Instance(인스턴스 구동) 대화 상자에서 Instance Boot Source(인스턴스 부팅 소스)로 Boot from image(이미지로 부팅)만 지원됩니다. Project(프로젝트) -> Images & Snapshots(이미지 및 스냅샷) -> Actions(작업) 메뉴에서 CreateVolumeFromImage가 지원되지 않습니다. Solaris OpenStack 대시보드의 다른 사

용자 정의에 대한 자세한 내용은 [Solaris OpenStack Horizon customizations](#)를 참조하십시오.

- /etc/neutron/l3_agent.ini 파일의 external_network_dataink 옵션에 대한 값으로 VXLAN 데이터 링크가 지원되지 않습니다. external_network_dataink 옵션에 대한 값으로 VXLAN 데이터 링크를 설정하면 Neutron L3 에이전트가 외부 네트워크에서 VNIC를 만들고 연결하지 못합니다.

- 명령줄을 사용하여 프로젝트에 대한 네트워크 리소스의 쿼터를 수정해야 합니다.

Horizon에서는 네트워크 리소스의 쿼터를 수정할 수 없습니다. Horizon 대시보드에서 프로젝트를 만들거나 기존 프로젝트의 비네트워크 리소스를 수정할 수 있습니다. 프로젝트의 네트워크, 서브넷, 포트, 라우터 또는 유동 IP 주소에 대한 쿼터를 수정하려면 neutron quota-update 명령을 사용해야 합니다.

비네트워크 리소스를 수정하는 경우에도 다음 오류 메시지가 표시됩니다. 이 메시지는 무시해도 됩니다. 이 메시지와 달리 비네트워크 리소스에 대한 쿼터가 적용되었습니다.

```
Error: Modified project information and members, but unable to modify project quotas.
```

- SMF와 OpenStack에서 서비스 상태를 다르게 보고할 수 있습니다.

다음 예에서는 nova-cert 서비스가 OpenStack에서 사용 안함으로 설정되었음을 보여주지만 SMF는 서비스가 online 상태임을 보여줍니다.

```
root@c190-133:~# nova service-disable c190-133 nova-cert
+-----+-----+-----+
| Host   | Binary | Status |
+-----+-----+-----+
| c190-133 | nova-cert | disabled |
+-----+-----+-----+
root@c190-133:~# svcs nova-cert
STATE          STIME      FMRI
online          21:14:11  svc:/application/openstack/nova/nova-cert:default
```

- neutron-l3-agent SMF 서비스가 다시 시작될 때 유지 관리 상태로 전환됩니다.

임시해결책: ipfilter 서비스를 다시 시작하고 neutron-l3-agent를 지우십시오.

```
network# svcadm restart ipfilter:default
network# svcadm clear neutron-l3-agent:default
```

- 네트워크 노드의 기본 게이트웨이가 특정 설정에서 제거됩니다.

네트워크 노드의 IP 주소가 external_network 주소 공간에서 파생되며 neutron router-gateway-clear 명령을 사용하여 provider_router에서 external_network를 제거하는 경우 네트워크 노드의 기본 게이트웨이가 삭제되며 네트워크 노드에 액세스할 수 없습니다.

```
network# neutron router-gateway-clear router_UUID
```

임시해결책: 콘솔을 통해 네트워크 노드에 연결하고 기본 게이트웨이를 다시 추가하십시오.

- 여러 인스턴스가 동시에 만들어질 때 Nova sqlite 데이터베이스가 잠깁니다.

많은 수(예: 10개 이상)의 인스턴스가 동시에 만들어지는 경우 다음 오류 메시지와 함께 특정 기간 동안 nova list 명령의 작동이 중지됩니다.

```
$ nova list
ERROR: The server has either erred or is incapable of performing the
requested operation. (HTTP 500) (Request-ID:
req-0ad63452-6753-c9fc-8275-e80604d42569)
```

또한 Horizon이 인스턴스를 찾을 수 없습니다.

이 문제는 Nova sqlite 데이터베이스 잠김으로 인한 것입니다. 잠시 후에 데이터베이스가 해제되고 nova list 및 Horizon이 예상대로 작동합니다.

임시해결책: sqlite 대신 MySQL 데이터베이스를 사용하십시오. [Compute 노드를 구성하는 방법 \[47\]](#)을 참조하십시오.

로그 파일 검사

SMF 서비스 및 다양한 Solaris 프로세스가 오류 메시지를 검색하거나 화면에 표시된 메시지에 대한 세부 정보를 수집할 수 있는 로그 파일을 생성합니다. SMF 서비스 로그 파일에는 유용한 디버깅 정보가 포함되어 있습니다.

문제에 따라 nova-compute, nova-scheduler 및 neutron-server SMF 서비스에 대한 로그 파일이 유용한 경우도 있습니다. svcs -L 명령을 사용하여 SMF 서비스 로그 파일의 이름을 찾을 수 있습니다.

```
$ svcs -L neutron-server
/var/svc/log/application-openstack-neutron-neutron-server:default.log
```

권한이 있는 사용자로 svcs -Lv 명령을 사용하여 서비스에 대한 로그 파일을 확인할 수 있습니다.

```
# svcs -Lv neutron-server
```

svcs -xv 명령은 서비스의 상태와 로그 파일의 이름을 표시합니다.

```
$ svcs -xv neutron-server
svc:/application/openstack/neutron/neutron-server:default (OpenStack Neutron Network Service)
State: online since Fri Jul 25 12:11:16 2014
See: /var/svc/log/application-openstack-neutron-neutron-server:default.log
Impact: None.
```

SMF 서비스 로그 파일 이외에 OpenStack 서비스도 로그 파일을 생성하며 여러 Solaris 프로세스가 고유 로그 파일을 가집니다. 일부 OpenStack 서비스는 /var/log 디렉토리의 OpenStack 서비스 이름 아래에 정보를 기록합니다. 예를 들어, OpenStack 이미지 저장소 로그 파일은 /var/log/glance에 있습니다. VM 인스턴스를 만들고 부팅하는 동안 문제가 발생하는 경우 /var/log/zones 디렉토리에서 로그 파일을 찾아 보십시오.

대부분의 OpenStack 구성 파일은 /etc 디렉토리의 OpenStack 서비스 이름 아래에 있습니다. 예를 들어, OpenStack 네트워킹 구성 파일은 /etc/neutron에 있습니다. Horizon에 대한 구성 파일은 /etc/openstack_dashboard에 있습니다.

OpenStack 서비스에 대한 로그 파일에서 세부 정보를 확인하려면 해당 서비스에 대한 구성 파일에서 verbose 옵션을 설정하십시오. 구성 파일에서 verbose 옵션이 이미 false로 설정되었거나 주석 처리되었을 수 있습니다. verbose 옵션의 주석 처리를 해제하거나 해당 옵션을 추가하고 verbose=true를 설정하십시오. 마찬가지로 구성 파일에서 debug=true를 설정하여 해당 구성 파일로 영향을 받는 작업의 자세한 출력을 확인할 수 있습니다. [Getting Started with OpenStack on Oracle Solaris 11.2](#)의 "Common Configuration Parameters for OpenStack" 및 [OpenStack documentation site](#)의 *OpenStack Configuration Reference*에 나오는 구성 옵션 표를 참조하십시오.

구성 파일에서 debug=true를 설정하는 것과 마찬가지로 개별 OpenStack 서비스 명령은 --debug 옵션을 사용할 수 있습니다.

적절한 RBAC 프로파일을 사용하여 OpenStack 서비스 로그 파일을 확인하거나 pfedit 명령으로 OpenStack 서비스 구성 파일을 수정하십시오. 다음 프로파일을 지정할 수 있습니다.

- OpenStack 블록 저장소 관리
- OpenStack Compute 관리
- OpenStack ID 관리
- OpenStack 이미지 관리
- OpenStack 네트워크 관리
- OpenStack 객체 저장소 관리
- OpenStack 관리

문제 조사 및 해결

서비스 이름이 없는 svcs -x 명령은 maintenance 상태의 모든 서비스를 표시합니다. 모든 OpenStack 노드에서 svcs 명령을 사용하여 필요한 모든 서비스가 online 상태인지 확인하십시오. 필요한 서비스가 online 상태가 아닌 경우 [“로그 파일 검사” \[73\]](#)에 설명된 대로 로그 파일을 확인하십시오.

SMF 서비스와 관련된 문제 해결에 대한 일반적인 지원은 [Oracle Solaris 11.2의 시스템 서비스 관리](#)의 [“서비스 문제 해결”](#)을 참조하십시오.

문제가 네트워크 문제인 것 같으면 모든 노드에서 evsadm 명령을 실행하십시오.

OpenStack 대시보드를 사용 중인 동안 문제가 발생했지만 작업이 성공적으로 완료되는 경우 명령줄에서 동일한 작업을 시도해 보십시오. 명령줄에 유용한 오류 메시지가 표시될 수 있습니다. 명령에 verbose 출력 옵션이 있는지 여부를 확인하십시오.

컨트롤러 노드 이외의 다른 노드에서 문제가 발생하는 경우 해당 노드의 명령줄에서 nova list와 같은 간단한 명령을 실행하여 해당 노드가 컨트롤러 노드와 통신할 수 있는지 확인하십시오.

OpenStack 설치 및 구성

이 절에서는 OpenStack을 설치하고 구성할 때 발생할 수 있는 오류에 대해 설명합니다.

대시보드 오류

다음 메시지와 같은 "권한이 부여되지 않음" 오류 메시지가 표시되면 RSA 호스트 키가 변경되었는지 여부를 확인하십시오.

```
Error: Unauthorized: Unable to retrieve usage information.
Error: Unauthorized: Unable to retrieve quota information.
Error: Unauthorized: Unable to retrieve project list information.
Error: Unauthorized: Unable to retrieve instance list information.
```

VM 인스턴스 설치 및 구성

이 절에서 설명되는 문제는 특별히 VM 인스턴스와 관련되어 있습니다.

VM 인스턴스가 오류 상태임

호스트 시스템과 다른 아키텍처인 VM 인스턴스를 설치하려고 시도한 경우 VM 인스턴스가 오류 상태일 수 있습니다. 이 경우 구체적으로 아키텍처 불일치를 나타내는 오류 메시지가 표시되지 않을 수도 있습니다. 이 문제가 발생하지 않도록 하려면 glance 이미지 저장소에 이미지를 업로드할 때 이미지의 architecture 등록 정보를 올바르게 설정해야 합니다. Horizon을 사용하여 이미지를 업로드하는 경우 업로드 후 이미지에 대한 등록 정보를 설정해야 합니다. 또는 명령줄을 사용하여 이미지를 업로드하고 하나의 glance image-create 명령에서 등록 정보 값을 설정할 수 있습니다. 예는 [“이미지 저장소에 이미지 추가” \[68\]](#)를 참조하십시오.

VM 인스턴스 등록 정보 값이 영역 등록 정보 값과 일치하지 않음

OpenStack이 VM 인스턴스에 대해 보고하는 일부 정보와 Solaris가 해당하는 영역에 대해 보고하는 정보가 일치하지 않습니다. Horizon에 표시되는 정보 또는 nova 명령으로 표시되는 정보가 zoneadm 명령 또는 기타 Solaris 명령으로 표시되는 정보와 일치하지 않을 수 있습니다.

이름	Horizon에 표시되거나 nova list 명령으로 표시되는 VM 인스턴스의 이름은 인스턴스를 만들 때 지정한 이름(예: example-instance)입니다. zoneadm list 명령으로 표시되는 영역의 이름은 instance-00000001과 유사합니다. nova show 명령을 사용하여 VM 인스턴스와 연관된 영역을 확인할 수 있습니다. nova show 출력에서 OS-EXT-SRV-
----	--

ATTR:instance_name 등록 정보의 값은 영역 이름이며, name 등록 정보의 값은 VM 인스턴스 이름입니다.

UUID	Horizon에 표시되거나 nova show 명령으로 표시되는 VM 인스턴스의 UUID가 zoneadm list -p 명령으로 표시되는 동일한 영역의 UUID와 일치하지 않습니다. zoneadm 명령이 표시하는 UUID는 Nova에 대해 사용된 식별자와 다른 식별자입니다.
CPU	Horizon에 표시되는 VM 인스턴스의 VCPU 수는 상한값이 설정된 CPU 수로, 인스턴스가 사용할 수 있는 부분 CPU 수까지만 가상화됩니다. 이 수는 상한값이 설정된 CPU의 인스턴스 내에서 관찰성을 제공하지 않습니다. psrinfo 명령은 영역에 할당된 전용 CPU를 보고합니다.
메모리	Horizon에 표시되는 VM 인스턴스의 메모리 양이 해당 VM 인스턴스에 로그인할 때 prtconf 명령으로 표시되는 메모리 양과 다를 수 있습니다. Horizon은 VM 인스턴스를 만들 때 사용되는 flavor로 지정된 메모리 양을 보여줍니다. prtconf 명령은 모든 시스템 메모리를 보고합니다.
저장소	VM 인스턴스가 ZOSS(공유 저장소의 영역)를 사용하는 비전역 영역이 아닌 경우 Horizon에 표시되는 VM 인스턴스의 저장소 양이 해당 VM 인스턴스에 로그인할 때 표시되는 저장소 양과 다를 수 있습니다.

네트워크 해체

네트워크 노드에서 Neutron 구성에 문제가 발생할 때 구성을 해체하고 처음부터 다시 시작해야 하는 경우 이 절차를 따르십시오. 구성 취소를 시작해야 하는 시점에 따라 절차에 제공된 순서를 따르십시오.

▼ Neutron에서 네트워크 구성을 제거하는 방법

1. Horizon 대시보드에서 이 단계를 수행합니다.
 - a. 모든 유동 IP 주소의 연관을 해제합니다.
 - b. 모든 유동 IP 주소를 제거합니다.
2. 터미널 창에서 다음 명령을 입력합니다.


```
# neutron router-gateway-clear router-id external-network-id
```

```
# neutron router-interface-delete router-id subnet-id
```

 - a. 라우터 게이트웨이 인터페이스를 제거하려면 다음 명령을 입력합니다.

```
# neutron router-gateway-interface-delete router-id external-network-id
```

- b. 남은 라우터 인터페이스를 제거하려면 다음 명령을 입력합니다.

```
# neutron router-interface-delete router-id subnet-id
```

3. Horizon 대시보드에서 다음을 수행합니다.

- a. 모든 VM 인스턴스를 종료합니다.
- b. 서브넷을 삭제합니다.
서브넷을 삭제하는 중 문제가 발생하면 [Vport를 제거하는 방법 \[77\]](#)을 참조하십시오.
- c. 네트워크를 삭제합니다.

▼ Vport를 제거하는 방법

서브넷 삭제를 방해하는 문제가 발생할 경우 이 절차를 사용하십시오.

1. 어떤 vport가 현재 사용 중인지 확인합니다.

```
# evsadm
```

2. 사용 중인 vport를 재설정합니다.

```
# evsadm reset-vport vport
```

3. vport를 제거합니다.

```
# evsadm delete-vport vport
```


색인

번호와 기호

- AI, 12
 - 부팅 가능 매체, 15
 - 설치 서비스, 14
- AMQP, 32
- AMQP(Advanced Message Queuing Protocol), 32
- archiveadm 명령, 15, 67
- Cinder
 - ZFS Storage Appliance, 39
 - 백업, 71
 - 설치, 38
- Compute 노드, 29
 - 구성, 46
- configure_evs.py EVS 구성 스크립트, 17
- DHCP 에이전트, 51
- EVS
 - configure_evs.py 구성 스크립트, 17
 - evsadm 명령, 51, 56
 - 구성, 17
 - 네트워크 노드에서 구성, 51
- evsadm 명령, 51
- flavor, 21, 63
 - 살펴볼 다른 내용 VM 인스턴스 extra-specs, 63
- Glance, 68
 - 설치, 42
- Heat
 - 설치, 37
- Horizon
 - SSL 액세스 구성, 45
 - 오류, 72
- Keystone
 - 설치, 36
- L3 에이전트, 52
- LDom, 29
- MySQL, 34
- Neutron
 - DHCP 에이전트, 51
 - L3 에이전트, 52
 - 설치, 43
 - 유동 IP 주소, 59
 - neutron-dhcp-agent SMF 서비스, 52
 - neutron-l3-agent SMF 서비스, 57
- Nova
 - 오류, 72
 - 컨트롤러 노드에 설치, 44
- NTP, 33
- NTP(Network Time Protocol), 33
- OpenStack 설치
 - 다중 노드 구성, 29
 - 단일 노드 구성, 11
 - 통합 아카이브 사용, 11
 - 평가 구성, 11
- OpenStack 통합 아카이브 살펴볼 내용 통합 아카이브
- OVM Server for SPARC, 29
- RabbitMQ, 32
- RAD, 47
- RAD(Remote Access Daemon), 47
- SC 프로파일, 65, 67
- SCI 도구, 14
- SPARC, 29
- SQLite, 34
- Swift, 32
- usbcopy 명령, 13
- VM 인스턴스, 10
 - flavor, 19, 21, 63
 - 로그인, 26, 60
 - 만들기, 22, 63, 69
 - 백업, 68
 - 부팅, 22, 69

스냅샷, 68
 이미지, 20, 66
 자동 복구, 68
 VM(가상 시스템) 살피볼 내용 VM 인스턴스
 ZFSSA, 39
 ZFSSA(ZFS Storage Appliance), 39
 zlogin 명령, 17
 zoneadm 명령
 boot 하위 명령, 17
 install 하위 명령, 12
 zonecfg 명령
 add 하위 명령, 16
 create 하위 명령, 16
 select 하위 명령, 16
 set 하위 명령, 16

L

네트워크 노드, 29
 구성, 48

C

다중 노드 OpenStack 설치, 29
 단일 노드 OpenStack 설치
 OVM Server for SPARC 사용, 29
 통합 아카이브 사용, 11
 대시보드
 flavor 보기, 21
 VM 인스턴스 만들기, 22
 VM 인스턴스 부팅, 22
 로그인, 18
 이미지 보기, 20
 프로젝트 보기, 20
 데이터베이스, 34
 디버깅 옵션, 74

R

레지스트리 서버, 68

S

스냅샷, 68
 시스템 구성 대화식 도구 살피볼 내용 SCI 도구
 시스템 구성 프로파일 살피볼 내용 SC 프로파일

O

영역 콘솔 로그인, 17
 영역 프레임워크, 47
 유동 IP 주소, 59
 이미지, 20, 66
 살피볼 다른 내용 VM 인스턴스
 레지스트리 서버, 68
 백업, 68
 스냅샷, 68
 이미지 캐시, 68
 자동 복구, 68
 인스턴스 살피볼 내용 VM 인스턴스
 인스턴스 템플릿 살피볼 내용 flavor

Z

자동 복구 이미지, 68

K

컨트롤러 노드, 29
 구성, 32
 클라우드 가상 시스템 살피볼 내용 VM 인스턴스

E

테넌트, 20
 템플릿
 하드웨어 살피볼 내용 flavor
 통합 아카이브
 OpenStack 이미지 만들기, 67
 다운로드, 12
 단일 노드 OpenStack 설치, 11
 베어 메탈 AI 서비스, 14
 베어 메탈 USB, 13
 베어 메탈 부팅 가능 AI, 15
 커널 영역, 16

P

프로젝트, 20

H

하드웨어 템플릿 살피볼 내용 flavor