

StorageTek Automated Cartridge System Library Software

High Availability Cluster 설치, 구성 및 작업

릴리스 8.4

E69744-01

2015년 12월

StorageTek Automated Cartridge System Library Software

High Availability Cluster 설치, 구성 및 작업

E69744-01

Copyright © 2015, 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 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 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.

차례

머리말	9
대상	9
설명서 접근성	9
규약	9
1. 시작하기	11
시스템 요구 사항	12
클라이언트 옵션	12
서버 옵션	12
스토리지 배열 옵션	12
네트워크 요구 사항	13
소프트웨어 요구 사항	13
ACSL S HA의 설치 전 점검 목록	13
오라클 고객지원센터 담당자	13
고객 지원 담당자	13
ACSL S HA 시스템 하드웨어	14
네트워크 정보	14
2개의 ACSLS HA 서버에 지정된 IP 주소 및 호스트 이름	14
HLI 라이브러리와 통신	15
방화벽	15
광 섬유를 사용한 SCSI 매체 교환기	15
설치 매체	15
ACSL S와 통신하는 클라이언트(백업 또는 ILM) 응용 프로그램	16
ACSL S 사용자 ID 및 그룹	16
상위 레벨 설치 절차	16
2. Solaris 시스템의 ACSLS HA 구성	19
/etc/hosts 구성	19
root에 대한 액세스 구성	19
다중 경로 네트워크 구성	20
공용 인터페이스 및 IPMP	22
라이브러리 인터페이스	25
다중 경로 디스크 구성	25

3. ZFS를 사용한 파일 시스템 구성	27
미러링된 루트 파일 시스템 만들기	27
ACSL S 응용 프로그램에 대한 미러링된 파일 시스템 만들기	29
4. 소프트웨어 패키지 다운로드	33
소프트웨어 패키지 다운로드	33
ACSL S 8.4 다운로드	33
PostgreSQL 8.4 다운로드(선택사항)	34
Oracle Cluster 4.2 다운로드	34
Solaris Cluster 기본 이미지 다운로드	34
Solaris 패치 업데이트가 필요한지 여부 확인	34
ACSL S HA 8.4 다운로드	35
패치 다운로드	36
제품 설명서 다운로드	36
5. ACSL S 8.4 설치	37
첫번째 노드에 설치	37
인접한 노드에 설치	38
6. Solaris Cluster 4.2 설치	41
클러스터 패키지 설치	41
<i>scinstall</i> 루틴	42
<i>scinstall</i> 실행	44
클러스터 구성 확인	44
7. ACSL S HA 8.4 설치 및 시작	47
기본 설치 절차	47
ACSL S HA 구성	48
ACSL S 클러스터 작업 모니터링	49
ha_console.sh 유틸리티	50
클러스터 작업 확인	51
8. ACSL S HA 세부 조정	53
라이브러리 통신을 위한 페일오버 정책 정의	53
RE(중복 전자 부품)를 포함하는 라이브러리	54
페일오버 <i>Pingpong_interval</i> 설정	54
시스템 이벤트의 전자 메일 알림 등록	54

9. ACSLS 클러스터 작업	57
ACSLS의 클러스터 제어 시작	58
acsls-storage에 대한 페일오버 정책 설정	58
클러스터 제어 시 ACSLS 작업 및 유지 관리	58
클러스터 제어 일시 중지	59
ACSLS HA 클러스터 전원 차단	60
일시 중지된 ACSLS 클러스터 시스템 전원 켜기	60
단일 노드 클러스터 만들기	61
10. 소프트웨어 구성 요소 설치, 업그레이드 및 제거	63
ACSLS 패치 설치	63
ACSLS 패키지 제거	64
ACSLS의 업그레이드 릴리스 설치	65
ACSLS HA 다시 설치 또는 업그레이드 설치	65
Solaris Cluster 업그레이드	67
Solaris Cluster 제거	67
11. 클러스터 로깅, 진단 및 테스트	69
전체 클러스터 작업 모니터링	69
클러스터 모니터링 유틸리티	70
복구 및 페일오버 테스트	70
복구 조건	71
복구 모니터링	71
복구 테스트	71
페일오버 조건	73
페일오버 모니터링	73
페일오버 테스트	74
추가 테스트	74
12. 문제 해결 도움말	75
ACSLS가 실행 중인지 확인	75
공유 디스크 리소스에 대한 연결 해결	76
논리 호스트에 ping을 수행할 수 없는 경우	77
노드 간 상호 연결 확인	78
색인	79

그림 목록

2.1. 각 서버 노드의 이더넷 포트 두 개에 연결된 단일 HBCr 라이브러리 인터페이스 카드	21
2.2. 중복 전자 부품을 사용한 라이브러리의 이중 HBC 구성	22
2.3. 외부 공유 스토리지 배열에 대한 서버당 2개의 광 섬유 연결	26
7.1. event_tail.sh의 예	49
7.2. gnome 단말기 창 구성	50

머리말

이 설명서에는 Solaris SPARC 기반 시스템과 x86 기반 시스템에 Oracle StorageTek ACSLS HA(Automated Cartridge System Library Software High Availability) 8.4 Cluster 소프트웨어를 설치 및 구성하기 위한 지침 및 절차가 포함되어 있습니다.

ACSLs HA 8.4는 특히 ZFS 파일 시스템을 사용하는 Solaris 11.2에서 ACSLS 8.4를 지원하도록 설계되었습니다. 이 버전은 모든 사용자 정의된 파일 시스템에서 ACSLS 소프트웨어 설치를 지원합니다.

대상

이 문서는 ZFS를 사용하는 Solaris 11 운영체제에 대한 지식과 Solaris Cluster 4.x에 대한 실무 지식이 있는 숙련된 UNIX 시스템 관리자를 대상으로 합니다.

이 문서에서는 사용된 대부분의 기술에 대한 일반적인 배경 정보를 제공하고 표준 예상 설치 절차에 대한 지침을 제공합니다. 하지만 이 문서만 갖고 UNIX 시스템에 대한 숙련도 및 전문 기술 요구 사항을 대신할 수는 없습니다.

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.

규약

이 문서에 사용된 텍스트 규약은 다음과 같습니다.

규약	의미
굵은체	굵은체 유형은 작업과 연관된 그래픽 사용자 인터페이스 요소, 또는 텍스트나 용어집에 정의된 용어를 나타냅니다.
기울임꼴	기울임꼴 유형은 책 제목, 강조 또는 사용자가 특정 값을 제공할 위치 표시자 변수를 나타냅니다.
고정 폭	고정 폭 유형은 단락 안의 명령, URL, 예제의 코드, 화면에 나타나는 텍스트, 사용자가 입력한 텍스트를 나타냅니다.

1장. 시작하기

ACSLS HA는 구성 요소 또는 부속 시스템 오류가 발생할 경우 무중단 테이프 라이브러리 제어 서비스를 보장하기 위해 이중 중복성, 자동 복구 및 자동 페일오버 복구를 제공하는 하드웨어 및 소프트웨어 구성입니다. 이 문서에서는 ACSLS 소프트웨어에 고가용성을 제공하는 데 필요한 구성, 설치 및 테스트 절차에 대해 설명합니다.

절차를 시작하기 전에 전체 설치 프로세스를 검토하는 것이 좋습니다. 클러스터화된 응용 프로그램을 설치하는 프로세스에는 세부적으로 엄격한 주의가 필요한 여러 단계가 포함되어 있습니다. 이 절차는 일반적으로 UNIX 시스템 통합 전문가가 수행해야 합니다.

여러 개의 하드웨어 및 소프트웨어 구성 요소가 ACSLS HA 시스템과 연관되어 있으며 전체 설치 절차는 며칠이 걸릴 수 있습니다. 기존 운영 라이브러리 환경의 경우에는 ACSLS HA 설치를 진행하는 동안 고객이 간단한 독립형 ACSLS 서버를 설치하여 라이브러리 운용을 처리하는 것이 좋습니다.

구성은 2 노드 클러스터입니다. 여기에는 심각한 시스템 오류를 검색할 수 있는 모니터링 소프트웨어 기능이 포함된 2개의 완전한 부속 시스템(활성 및 대기)이 포함됩니다. 이 구성에서는 복구할 수 없는 모든 부속 시스템 오류에 대해 주 시스템에서 대기 시스템으로 제어를 전환할 수 있습니다. 이 구성은 일반적인 스위치 오버 없이도 부속 시스템 통신 오류를 즉시 복구할 수 있는 중복된 전원 공급 장치, 중복 네트워크 및 I/O 상호 연결을 제공합니다.

ACSLS HA는 Solaris Cluster의 모니터 및 페일오버 기능과 Solaris 운영체제의 다중 경로 기능을 활용해서 작동 중지 시간이 최소화된 탄력적인 라이브러리 제어 작업을 제공합니다. Solaris는 무중단 네트워크 연결을 보장하기 위한 IP 다중 경로 지정과 시스템 데이터에 대한 무중단 액세스를 보장하기 위한 RAID 1 지원 다중 경로 디스크 I/O를 제공합니다. Solaris Cluster는 운영체제, 내부 하드웨어 및 외부 I/O 리소스를 포함하는 시스템 리소스 상태를 감시하며, 필요에 따라 시스템 스위치 오버를 관리할 수 있습니다. 그리고 ACSLS HA 에이전트는 ACSLS 응용 프로그램, 해당 데이터베이스, 해당 파일 시스템 및 StorageTek 라이브러리 리소스에 대한 연결을 모니터링하여 필요에 따라 Solaris Cluster 페일오버 서비스를 호출합니다.

이러한 중복 구성에서 ACSLS 라이브러리 제어 서버는 클러스터 프레임워크 내부 및 외부에 항상 알려져 있는 단일 논리 호스트 ID를 포함합니다. 이 ID는 필요에 따라 변환 중에도 최소한의 작동 중지 시간으로 클러스터 노드 사이에 자동으로 전송됩니다.

프로젝트를 시작하기 전에 여기에 설명된 대로 ACSLS HA 설치 및 구성을 위한 전체 프로세스를 검토하십시오. 필요한 경우 전체 설치에 대한 조언, 지원 또는 처리를 위해 Oracle Advanced Customer Services를 요청할 수 있습니다.

ACSLS 설명서는 다음 위치의 OTN(Oracle Technical Network)을 참조하십시오.

<http://docs.oracle.com/>

시스템 요구 사항

ACSLS HA 서버 구성은 외부 RAID 디스크 배열을 공유하는 2개의 Solaris 서버 노드로 구성됩니다.

클라이언트 옵션

ACSLS HA는 ACSAPI(Automated Cartridge System Application Programming Interface)를 사용하는 모든 ACSLS 클라이언트를 지원합니다. 단일 네트워크 IP 주소는 두 서버 노드 사이에서 공유되어 ACSAPI 클라이언트가 공통 가상 호스트 ID를 사용하여 ACSLS를 처리할 수 있도록 합니다.

SMCE(SCSI Media Changer Emulation)를 사용하여 광 섬유 채널 클라이언트에 제공된 논리적 라이브러리는 이 제품에서 지원되지 않습니다.

서버 옵션

ACSLS HA 8.4는 Solaris Cluster 4.2의 최소 하드웨어 요구 사항을 충족하는 시스템에서 실행해야 합니다. 특정 서버 플랫폼 요구 사항은 *Oracle Solaris Cluster System Requirements* 문서를 참조하십시오.

<http://www.oracle.com/technetwork/server-storage/solaris-cluster/documentation/sysreq-cluster-166689.pdf>

단일 실패 지점을 방지하기 위해서는 각 ACSLS HA 서버 노드를 다음과 같이 구성해야 합니다.

- 이중(중복) 전원 공급 장치
- 10/100/1000 Base-T 이더넷 포트 6개
- 광 섬유 채널 포트 2개(FC 연결 스토리지를 사용하는 경우)
- SAS 포트 2개(SAS 연결 스토리지를 사용하는 경우)

논리적 라이브러리에서 시스템을 사용하려는 경우 SCSI 클라이언트 응용 프로그램을 지원하는 구성에 최소 한 개 이상의 전용 광 섬유 채널 포트가 필요합니다.

광 섬유 연결 라이브러리(예: SL500 또는 SL150)의 경우 한 개의 추가 광 섬유 채널 포트가 필요합니다.

스토리지 배열 옵션

지원되는 디스크 배열 부속 시스템은 *Oracle Solaris Cluster Storage Partner Program* 문서를 참조하십시오.

<http://www.oracle.com/technetwork/server-storage/solaris-cluster/partnerprogram-cluster-168135.pdf>

네트워크 요구 사항

총 7개의 IP 주소를 예약해야 합니다.

- 1 - 논리 호스트(클러스터 VIP(가상 IP))
- 2 - 노드 1 IP 주소
- 3 - 노드 2 IP 주소
- 4 - 라이브러리 인터페이스 1 소스 IP 주소(노드 1)
- 5 - 라이브러리 인터페이스 2 소스 IP 주소(노드 1)
- 6 - 라이브러리 인터페이스 1 소스 IP 주소(노드 2)
- 7 - 라이브러리 인터페이스 2 소스 IP 주소(노드 2)

이상적으로 볼 때, 라이브러리 인터페이스 1은 라이브러리 인터페이스 2와 다른 서브넷에 있어야 합니다([그림 2.1. “각 서버 노드의 이더넷 포트 두 개에 연결된 단일 HBCr 라이브러리 인터페이스 카드” 참조](#)).

소프트웨어 요구 사항

ACSLS HA 8.4에는 다음과 같은 소프트웨어 구성 요소가 필요합니다.

- Oracle Solaris 11.2(SPARC 또는 X86)
- Oracle Solaris Cluster 4.2

ACSLS HA의 설치 전 점검 목록

새 ACSLS HA 시스템을 설치하거나 ACSLS HA 시스템을 새 릴리스로 업그레이드하려면 먼저 ACSLS HA가 설치된 고객 환경에 대한 정보를 확인하고 기록합니다.

이 설치 전 점검 목록을 작성하면 위험을 줄일 수 있습니다. 이 점검 목록은 설치가 문제 없이 진행되고 고객 환경에 대한 다른 세부정보를 확인하느라 설치가 지연되지 않도록 보장합니다.

오라클 고객지원센터 담당자

- 이 고객을 지원하는 지역 오라클 직원은 누구입니까?
- 이들 중 Solaris 시스템 관리 경험이 있는 사람이 있습니까?
- 이들 중 ACSLS 경험이 있는 사람이 있습니까?
- 이들 중 ACSLS HA 경험이 있는 사람이 있습니까?

고객 지원 담당자

- ACSLS HA 서버 및 고객 네트워크와 같은 지원 작업을 수행할 고객의 시스템 관리 직원은 누구입니까?

- 이들 중 Solaris 시스템 관리 경험이 있는 사람이 있습니까?
- 이들 중 ACSLS 경험이 있는 사람이 있습니까?
- 이들 중 ACSLS HA 경험이 있는 사람이 있습니까?
- 네트워크 관리자는 누구입니까?

ACSLS HA 시스템 하드웨어

- Oracle Sun 서버와 모델은 무엇입니까?
- Solaris 릴리스와 업데이트 레벨은 무엇입니까?
- 메모리(최소 10GB)
- 디스크를 미러링하려면 각 서버에 이중 부트 드라이브가 필요합니다.
- 공유 디스크의 모델은 무엇입니까? Solaris Cluster에서 지원됩니까?
- 각 ACSLS 서버를 공유 디스크 배열에 연결하는 데 사용되는 SAS 또는 광 섬유 HBA
- 각 ACSLS 서버에 필요한 이더넷 포트 6개
- ACSLS가 광 섬유 연결 라이브러리(SL500 또는 SL150)를 관리하거나 광 섬유 대상 모드 포트를 사용해서 논리적 라이브러리를 제공하는 경우 각 ACSLS 서버에 광 섬유 HBA가 필요합니다.
- Solaris 서버 및 디스크 배열을 고객의 환경에 연결하기 위해서는 어떤 전원 코드가 필요합니까? 예를 들어, HA 설치 지연을 방지하기 위해서는 플러그가 고객 사이트의 전원 콘센트와 일치해야 합니다.
- HA 설치를 시작하기 전에 HA 서버 및 공유 디스크 배열이 올바르게 구성되었는지 확인합니다.
 - 이더넷 연결을 위한 6포트 요구 사항을 만족하려면 각 HA 서버에 추가 NIC(네트워크 인터페이스 컨트롤러) 카드가 장착되어 있어야 합니다.
 - 외부 공유 디스크 배열에 대한 인터페이스 유형을 확인하고 호환 가능한 HBA가 각 서버에 구성되어 있는지 확인하십시오.
 - ACSLS가 SL500s 또는 SL150s와 같은 광 섬유 연결 라이브러리와 통신하려는 경우 광 섬유 HBA가 필요합니다.
 - ACSLS가 광 섬유 대상 모드를 사용해서 논리적 라이브러리를 제공하는 경우 Qlogic 광 섬유 HBA가 필요합니다.

네트워크 정보

다음 네트워크 정보를 검토합니다.

2개의 ACSLS HA 서버에 지정된 IP 주소 및 호스트 이름

- 설치에는 총 9개의 IP 주소가 필요합니다.
 - ACSLS 서버 노드 1의 로컬 주소
 - ACSLS 서버 노드 2의 로컬 주소
 - 논리 호스트 주소(2개의 HA 노드 사이에 공유되는 가상 IP 주소)
 - 노드 1의 라이브러리 연결 a

- 노드 1의 라이브러리 연결 b(이중 TCP/IP 또는 다중 TCP/IP 중복성)
- 노드 2의 라이브러리 연결 a
- 노드 2의 라이브러리 연결 b(이중 TCP/IP 또는 다중 TCP/IP 중복성)
- 노드 1의 ILOM
- 노드 2의 ILOM
- ACSAPI 클라이언트가 ACSLS와 통신합니까?
 - 페일오버 이벤트 후 활성 노드는 가상 IP 주소를 간주합니다. 어떤 노드가 활성 상태이든지 가상 IP 주소는 ACSLS 클라이언트에 액세스할 수 있습니다.
 - ACSAPI 클라이언트에서 ACSLS HA 가상 IP 주소 또는 호스트 이름을 설정하는 방법을 알고 있습니까? 이 설정은 클라이언트에서 실행되는 ISV 응용 프로그램에 따라 달라집니다.
- ACSLS가 TCP/IP 연결 라이브러리(SL8500s, SL3000s 및 9310s)와 통신하기 위해 사용하는 IP 주소입니다.
- 단일 실패 지점을 방지하려면 고유 별도 서브넷을 통해 각각의 중복 라이브러리 연결을 경로 지정하는 것이 좋습니다. 고용량 네트워크 트래픽으로 인한 문제를 방지하려면 서브넷을 라이브러리 통신에 대해 예약해야 하며 일반적인 네트워크 채터의 방해받지 않아야 합니다.
- 각 ACSLS HA 노드에서 서비스 프로세서(예: ILOM 또는 ALOM)에 액세스하는 데 필요한 IP 주소 및 암호입니다.

HNI 라이브러리와 통신

서브넷에서 ACSLS와 TCP/IP 연결 라이브러리 사이의 통신이 브로드캐스트 트래픽으로부터 보호됩니까?

방화벽

- ACSAPI 클라이언트와 ACSLS HA 시스템 사이에 방화벽이 있습니까?
- ACSLS HA 시스템과 이 시스템이 관리하는 라이브러리 사이에 방화벽이 있습니까?

방화벽이 존재할 경우 방화벽을 통해 통신하도록 ACSLS와 ACSAPI 클라이언트를 구성하는 방법에 대한 자세한 내용은 *ACSLs 8.4 Administrator's Guide*의 "Firewall Security Option" 부록을 참조하십시오.

광 섬유를 사용한 SCSI 매체 교환기

- 이 ACSLS 시스템이 광 섬유 채널 대상 모드 포트를 사용해서 클라이언트에 논리적 라이브러리를 제공합니까? 그렇다면 QLogic 광 섬유 HBA가 필요합니다.

설치 매체

- Oracle Solaris Cluster, ACSLS, ACSLS HA 및 기타 필수 패키지 또는 패치를 포함하여 Oracle edelivery 웹 사이트에서 소프트웨어 다운로드를 허용하기 위해 HA 서버에서 인터넷으로의 직접 또는 간접 액세스가 사용됩니까?

액세스가 간접일 경우 인터넷에서 HA 서버로 파일을 전송할 수 있는지 확인합니다.

- 소프트웨어를 인터넷에서 로컬 시스템으로 직접 다운로드한다고 가정할 경우 브라우저 구성에 대한 전체 프록시 정보를 설치 시에 사용할 수 있는지 확인합니다.

ACSLs와 통신하는 클라이언트(백업 또는 ILM) 응용 프로그램

- ACSAPI 클라이언트(예: 백업 또는 정보 수명 주기 관리 응용 프로그램)가 ACSLS와 통신합니까?
 - ACSAPI 클라이언트가 ACSLS와 통신할 경우 클라이언트 응용 프로그램은 무엇입니까(예: NetBackup, Oracle SAM)?
 - 사용되는 이러한 클라이언트의 버전은 무엇입니까?
 - 클라이언트가 Windows에서 실행되고 LibAttach를 사용할 경우 실행되는 LibAttach 버전은 무엇입니까?
- 클라이언트 응용 프로그램이 광 섬유 채널 대상 모드 포트를 사용해서 ACSLS 논리적 라이브러리와 통신합니까?
 - 클라이언트 응용 프로그램은 무엇입니까(예: NetBackup, Oracle SAM)?
 - 사용되는 이러한 클라이언트의 버전은 무엇입니까?

ACSLs 사용자 ID 및 그룹

ACSLs는 *acsls* 그룹에서 *acsss*, *acsdb* 및 *acssa*와 같은 사용자 ID가 필요합니다.

이러한 사용자 ID 및 *acsls* 그룹을 ACSLS HA 서버에서 로컬로 설정할 수 있습니까? 아니면 사이트의 중앙 사용자 및 암호 관리 시스템에 통합되어야 합니까?

상위 레벨 설치 절차

전체 ACSLS HA 설치에는 다음 단계가 포함됩니다.

1. 일반적인 외부 광 섬유 채널 또는 SAS2 디스크 배열에 연결된 2개의 Solaris 플랫폼 서버를 설치합니다. 각 서버에 Solaris 11.2를 설치합니다.

Oracle Technology Network 라이브러리에서 제공되는 *Oracle Solaris 11* 시스템 설치 문서를 참조하십시오.

<http://www.oracle.com/technetwork/documentation/solaris-11-192991.html>

2. 기본 Solaris 시스템을 구성합니다.
 - 사용자 액세스 권한
 - 다중 경로 네트워크 액세스 및 디스크 I/O

[19]2장. *Solaris* 시스템의 ACSLS HA 구성 을 참조하십시오

3. ZFS 파일 시스템을 구성합니다.

- 루트 스토리지 풀
- acsls 스토리지 풀

[3장. ZFS를 사용한 파일 시스템 구성](#) 을 참조하십시오

4. 소프트웨어 패키지를 다운로드합니다. [\[33\]4장. 소프트웨어 패키지 다운로드](#) 를 참조하십시오.
 - ACSLS 8.4.0
 - Solaris Cluster 4.2
 - ACSLS HA 8.4.0

[\[33\]4장. 소프트웨어 패키지 다운로드](#) 를 참조하십시오.

5. ACSLS 8.4.0을 설치하고 업데이트를 패치합니다(있는 경우). [\[37\]5장. ACSLS 8.4 설치](#) 를 참조하십시오.
6. Solaris Cluster 4.2를 설치하고 업데이트를 패치합니다(있는 경우). [6장. Solaris Cluster 4.2 설치](#) 를 참조하십시오.
7. ACSLS HA 8.4.0을 설치합니다. [7장. ACSLS HA 8.4 설치 및 시작](#) 을 참조하십시오.
8. ACSLS HA에 대한 클러스터 작업을 세부 조정합니다. [\[53\]8장. ACSLS HA 세부 조정](#) 을 참조하십시오.

2장. Solaris 시스템의 ACSLS HA 구성

기본 Solaris 11.2 시스템이 설치되었으면 다음과 같은 세 가지 컨텍스트에 따라 Solaris Cluster 및 ACSLS HA에 대한 특정 구성 요구 사항이 발생합니다.

1. `/etc/hosts`를 구성합니다.
2. `root` 사용자에게 대한 액세스 권한을 구성합니다.
3. 다중 경로 인터넷 액세스를 위한 네트워크 인터페이스를 구성합니다.
4. 다중 경로 디스크 I/O를 구성합니다.

`/etc/hosts` 구성

각 노드의 `/etc/hosts` 파일에 `localhost`에 대한 항목, 두 클러스터 노드의 이름 및 IP 주소, 논리 호스트가 포함되어 있어야 합니다.

```
127.0.0.1          localhost  loghost
129.99.99.101     thisNode.domain.com  thisNode
129.99.99.102     sisterNode
129.99.99.100     logicalHost.domain.com  logicalHost
```

`root`에 대한 액세스 구성

Solaris Cluster 모니터링 및 페일오버 작업은 어느 노드에서든 전체 클러스터에 대한 제어 권한을 가져야 하는 `root` 사용자에게 의해 처리됩니다. `root` 사용자는 신뢰할 수 있는 사용자로 설정되어 보안 셸(`ssh`) 인증으로 노드 간 개인 네트워크 액세스를 제공합니다.

1. 외부 시스템에서 각 노드에 대한 직접 루트 액세스를 사용으로 설정합니다. `/etc/user_attr` 파일을 편집하고 `root`에 대한 역할을 지정하는 라인을 주석 처리(또는 제거)합니다.

```
# root:::::type=role
```

2. `root`가 시스템 콘솔 이외의 지점에서 시스템에 로그인할 수 있도록 허용합니다.

`/etc/default/login` 파일을 편집하고 콘솔 전용 액세스를 지정하는 라인을 주석 처리합니다.

```
# CONSOLE=/dev/console
```

3. `root`에 대한 보안 셸 로그인 액세스를 허용합니다.

`/etc/ssh/sshd_config` 파일을 편집하고 `PermitRootLogin`에 **yes**를 지정합니다.

```
PermitRootLogin=yes
```

인접한 노드에서 1~3단계를 반복합니다.

4. *root*를 신뢰할 수 있는 사용자로 설정합니다. 그러면 암호를 요구하지 않고도 인증이 설정되는 자매 노드에서 각 노드에 대해 루트의 로그인 프로토콜이 설정됩니다.
 - a. 공용/개인 RSA 키 쌍을 만듭니다. 암호를 사용하지 않고도 한 노드에서 다른 노드로 로그인할 수 있도록 하려면 문장암호를 입력하지 않습니다.

```
# cd /root/.ssh
# ssh-keygen -t rsa
Enter file in which to save the key (//.ssh/id_rsa): ./id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ./id_rsa.
Your public key has been saved in ./id_rsa.pub.
The key fingerprint is:
1a:1b:1c:1d:1e:1f:2a:2b:2c:2d:2e:2f:ea:3b:3c:3d root@node1
```

그러면 */root/.ssh* 디렉토리에 *id_rsa* 및 *id_rsa.pub*의 두 파일이 생성됩니다.

- b. *id_rsa.pub*를 자매 노드의 */root/.ssh* 디렉토리에 복사합니다.

```
# cat id_rsa.pub | ssh root@node2 /
'cat >> /root/.ssh/authorized_keys'
Password:
```

- c. 인증 키가 배치되면 암호 없이 원격으로 명령을 명제화하는 기능을 테스트합니다.

```
# hostname
node1
# ssh root@node2 hostname
node2
```

다중 경로 네트워크 구성

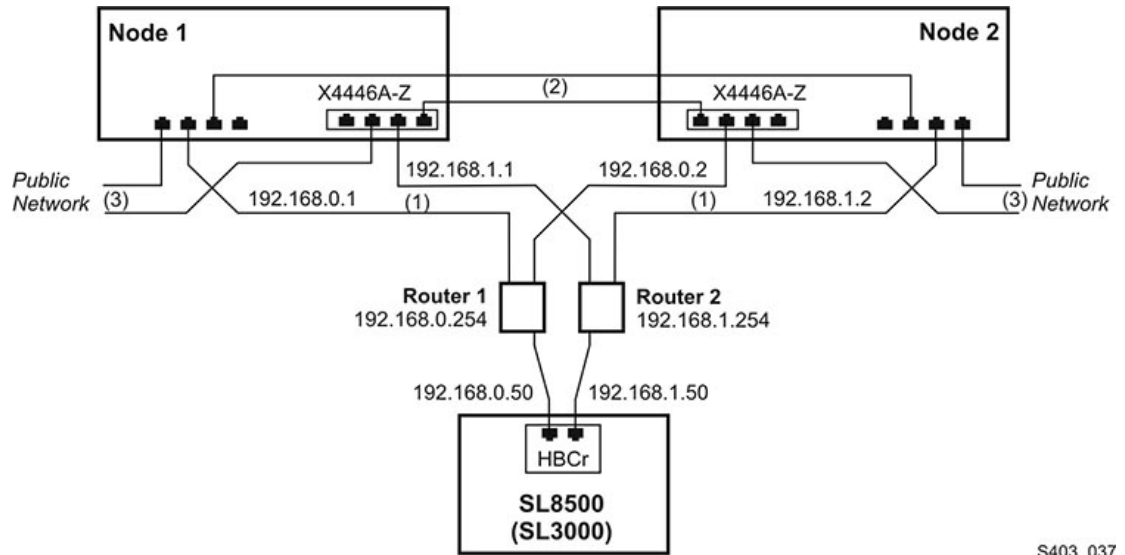
중복성은 고가용성 컴퓨팅을 위한 전반적인 체계입니다. 중복성은 서버에만 적용되지 않고 각 서버의 각 통신 인터페이스에도 적용됩니다. 공용 인터페이스의 경우 Solaris에서 IPMP(인터넷 프로토콜 다중 경로 지정)를 사용합니다. 인터넷 프로토콜 다중 경로 지정은 일반적인 시스템 페일오버를 필요로 하지 않고 네트워크 통신 오류에 대한 즉각적인 NIC 복구를 제공합니다. 라이브러리 인터페이스의 경우, 이는 두 개의 독립 경로에서 두 개의 네트워크 인터페이스가 포함되는 이중 TCP/IP 연결 사용을 의미합니다. 한 경로의 요소가 실패할 경우, ACSLS는 대체 인터페이스를 통해 계속 통신을 수행합니다.

ACSL HA에는 다음에 대한 중복 네트워크 연결이 필요합니다.

- 공용 및 클라이언트 통신

- 라이브러리 통신
- 개인 노드 간 클러스터 통신

그림 2.1. 각 서버 노드의 이더넷 포트 두 개에 연결된 단일 HBCr 라이브러리 인터페이스 카드

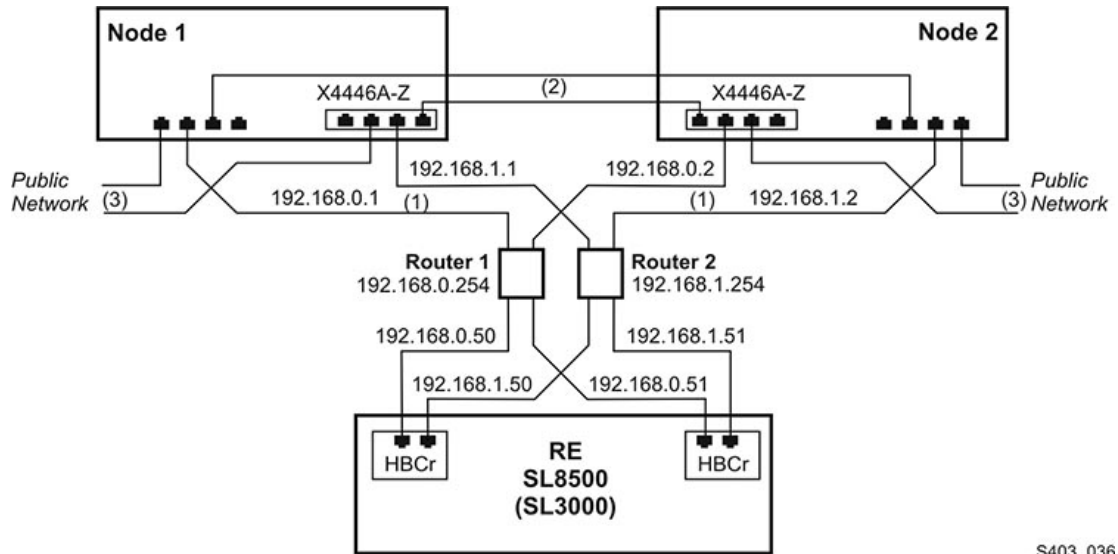


S403_037

이 절의 그림에서는 각 서버에서 2개의 개별 컨트롤러로 액세스할 수 있는 8개의 이더넷 포트를 보여줍니다. 여기에서는 6개의 포트를 사용해서 3개의 중복 연결을 제공합니다. 이 구성에서 2개 포트는 사용되지 않는 상태로 유지됩니다. 겉으로는 복잡해 보이지만 각 서버에서 단 3개의 이중 경로 이더넷 연결만 있습니다.

- 서버-라이브러리 통신
- 개인 네트워크를 통한 서버-서버 하트비트 교환
- 공용 네트워크를 통한 서버-클라이언트 통신

그림 2.2. 중복 전자 부품을 사용한 라이브러리의 이중 HBC 구성



S403_036

중복 전자 부품이 포함된 라이브러리에선 각 서버 노드에서 각 HBCr 라이브러리 컨트롤러의 2개의 독립 경로가 존재합니다. 하나의 HBCr 인터페이스에서 두 포트 모두에 대한 통신이 실패할 경우 ACSLS HA는 대체 HBCr 카드에 대한 자동 전환을 호출합니다. 이 모든 작업은 대체 서버 노드에 대한 페일오버 없이 수행됩니다.

공용 인터페이스 및 IPMP

Solaris IPMP는 NIC, 케이블, 스위치 또는 기타 네트워킹 하드웨어 오류에 대비하여 중복 네트워크 인터페이스를 구축하기 위한 메커니즘을 제공합니다. Solaris 호스트에서 IPMP를 구성할 때는 두 개 이상의 물리적 네트워크 인터페이스를 단일 IPMP 그룹으로 조합합니다.

물리적 장치에 대한 네트워크 인터페이스 이름 매핑을 보려면 `dladm show-phys` 명령을 사용합니다.

예:

```
# dladm show-phys
LINK      MEDIA      STATE     SPEED  DUPLEX    DEVICE
net2      Ethernet  up        100    full     ixgbe1
net3      Ethernet  up        10000  full     ixgbe3
net0      Ethernet  up        10000  full     ixgbe2
net1      Ethernet  up        1000   full     ixgbe0
```

구성된 네트워크 인터페이스의 상태를 보려면 `ipadm`을 사용합니다.

예:

```
# 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      --      129.99.99.99/24
    net1/v6     addrconf ok      --      fe99::999:999:ff23:ee02/10
net4          ip        ok      --      --
    net4/v4     static    ok      --      129.999.99.99/24

```

2개의 인터페이스를 구성하고 ACSLS HA의 공통 그룹 ID에 지정해야 합니다. 이러한 인터페이스 중에는 서버의 기본 인터페이스가 있을 수 있습니다. 이 경우, *ip* 주소가 이미 인터페이스에 지정(그룹에는 지정되지 않음)되었기 때문에 이 인터페이스를 구성 해제한 후 *ipmp* 그룹에서 다시 구성해야 합니다.

이 작업 중에는 네트워크 통신이 중단되기 때문에 서버 콘솔에서 다음 단계를 수행해야 합니다.

기존 기본 인터페이스를 구성 해제하려면 다음을 수행합니다.

```
ipadm delete-addr <primary interface>
```

예:

```
# ipadm delete-addr net0/v4
```

```
ipadm delete-ip <primary interface>
```

예:

```
# ipadm delete-ip net0
```

기본 인터페이스를 다시 구성하려면 다음을 수행합니다.

```
ipadm create-ip <primary interface>
```

예:

```
# ipadm create-ip net0
```

보조 기본 인터페이스를 만들려면 다음을 수행합니다.

```
ipadm create-ip <primary_interface>
```

예:

```
# ipadm create-ip net5
```

ipmp 그룹을 만들려면 다음을 수행합니다.

```
ipadm create-ipmp <group_name>
```

예:

```
# ipadm create-ipmp ipmp0
```

주:

그룹 이름은 영숫자여야 합니다.

그룹에 호스트 *ip-address*를 지정하려면 다음을 수행합니다.

```
ipadm create-addr -T static -a <ip-address> <group_name>
```

예:

```
# ipadm create-addr -T static -a 129.99.99.9 ipmp0
```

그룹에 기본 인터페이스를 추가하려면 다음을 수행합니다.

```
ipadm add-ipmp -i <primary_interface> <group_name>
```

예:

```
# ipadm add-ipmp -i net0 ipmp0
```

그룹에 보조 인터페이스를 추가하려면 다음을 수행합니다.

```
ipadm add-ipmp -i <second_primary_interface> <group_name>
```

예:

```
# ipadm add-ipmp -i net5 ipmp0
```

*ipmp*를 사용하여 *ipmp* 구성을 확인합니다.

```
# ipadm
NAME          CLASS/TYPE STATE   UNDER   ADDR
ipmp0         ipmp      ok      --      --
  ipmp0/v4    static   ok      --      123.45.67.89/8
lo0          loopback  ok      --      --
  lo0/v4      static   ok      --      127.0.0.1/8
  lo0/v6      static   ok      --      ::1/128
net0         ip        ok      ipmp0   --
net5         ip        ok      ipmp0   --
```

ipmp0 그룹 아래에 2개의 네트워크 인터페이스가 구성된 것을 확인합니다. 버전 4 IP 주소가 ipmp0 그룹에 지정된 것을 확인합니다.

이러한 변경 사항을 커밋하고 새 구성에 따라 네트워크 통신을 설정하려면 시스템을 재부트해야 할 수 있습니다.

자매 노드에서 네트워크 구성을 반복합니다.

ipmp 그룹과 클러스터 공용 IP 주소 사이의 매핑은 *start_acslsha.sh*로 클러스터를 시작할 때 설정됩니다. 9장. ACSL S 클러스터 작업을 참조하십시오.

라이브러리 인터페이스

남은 네트워크 인터페이스 두 개는 라이브러리 구성에 필요합니다. 이 예에서는 *net1* 및 *net6*이 사용됩니다. 그림 2.1. “각 서버 노드의 이더넷 포트 두 개에 연결된 단일 HBCr 라이브러리 인터페이스 카드” 및 그림 2.2. “중복 전자 부품을 사용한 라이브러리의 이중 HBC 구성”의 경우, 이러한 2개 연결은 클러스터화된 서버 및 라이브러리 사이에 라우터가 단일 실패 지점이 되지 않도록 하기 위해 개별 서브넷을 통해 경로 지정됩니다.

1. 각 노드에서 2개의 네트워크 인터페이스를 만듭니다.

```
# ipadm create-ip net1
# ipadm create-ip net6
```

2. 각 인터페이스에 대해 *ip-address*를 지정합니다.

```
# ipadm create-addr -T static -a <ip-address> net1/v4
# ipadm create-addr -T static -a <ip-address> net6/v4
```

주소 객체는 일반적으로 인터페이스 및 프로토콜 버전을 사용해서 이름이 지정됩니다 (예: *net1/v4*).

3. *ipmp* 구성을 확인합니다.

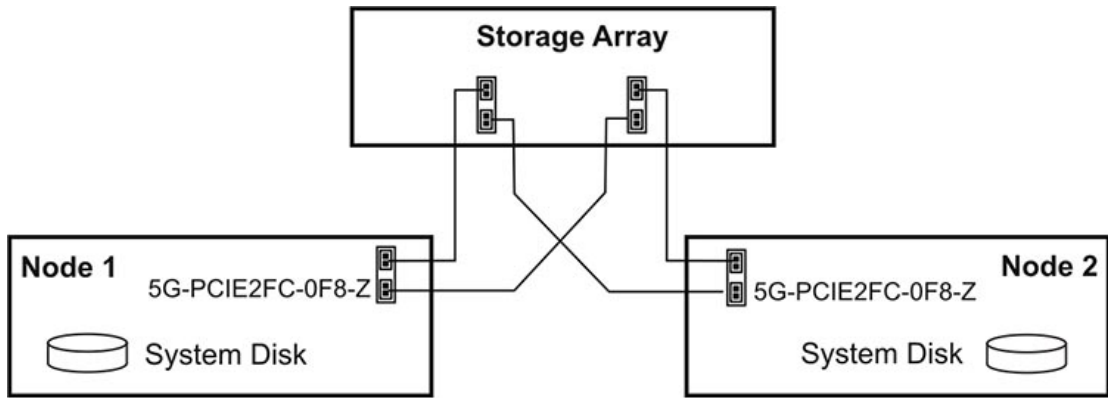
```
# ipadm
NAME                CLASS/TYPE STATE      UNDER    ADDR
ipmp0                ipmp      ok         --        --
  ipmp0/v4           static    ok         --        123.45.67.89/8
lo0                  loopback  ok         --        --
  lo0/v4             static    ok         --        127.0.0.1/8
  lo0/v6             static    ok         --        ::1/128
net0                 ip        ok         ipmp0     --
net1                 ip        ok         --        --
  net1/v4            static    ok         --        192.168.0.1/8
net5                 ip        ok         ipmp0     --
net6                 ip        ok         --        --
  net6/v4            ip        ok         --        192.168.1.1/8
```

4. 이러한 설정을 적용하려면 각 노드를 재부트합니다.

다중 경로 디스크 구성

외부 공유 디스크 배열은 각각 서버와 디스크 배열 사이에 중복 광 섬유 또는 SAS 연결을 포함하는 호스트 두 개에 연결됩니다.

그림 2.3. 외부 공유 스토리지 배열에 대한 서버당 2개의 광 섬유 연결



S403_038

배열은 연결된 호스트에 대해 2개의 가상 드라이브를 제공하도록 설정되어야 합니다.

Solaris 11.2는 디스크 장치에 대해 두 개 이상의 경로가 감지될 경우 다중 경로(MPXIO) 설정을 자동으로 수행합니다. Solaris 시스템이 배열의 각 가상 디스크에 대한 중복 연결을 사용해서 올바르게 구성되었는지 확인합니다.

```
# mpathadm list lu
/dev/rdsk/c0t600A0B800049EE1A0000840552D3E2F9d0s
    Total Path Count: 2
    Operational Path Count: 2
/dev/rdsk/c0t600A0B800049EDD600000DAD52D3DA90d0s2
    Total Path Count: 2
    Operational Path Count: 2
```

mpathadm 표시는 각각 작동되는 경로 수가 두 개인 디스크 장치 두 개를 보여줍니다.

SPARC 시스템에서 ACSLS HA를 실행하는 경우 사용자가 MPXIO를 구성할 때까지 *mpathadm*은 아무 것도 표시하지 않습니다. 화면에 아무 것도 표시되지 않거나 잘못된 구성이 표시되는 경우 Solaris 11.2 시스템에서 *stmsboot -e*를 사용하여 MPXIO를 구성을 진행합니다. Oracle Technology Network 라이브러리(http://docs.oracle.com/cd/E23824_01/html/E23097)에서 *Oracle Solaris* 관리: SAN 구성 및 다중 경로 문서를 참조하십시오.

Solaris I/O 다중 경로 사용 장치에서 Solaris I/O 다중 경로 제어가 적용되고 있음을 나타내는 새 이름을 수신하는지 확인합니다. 장치는 원본 이름과 다른 이름을 갖습니다.

```
Original device name:    c1t0d0
Name After stms boot:   c0t600A0B800049EDD600000C9952CAA03Ed0
```

3장. ZFS를 사용한 파일 시스템 구성

Solaris 11.2는 ZFS 파일 시스템을 기반으로 합니다. 디스크 I/O, 디스크 분할 및 디스크 미러링(또는 RAID)은 전적으로 ZFS에 의해 처리됩니다. 따라서 UFS 파일 시스템에서 일반적으로 수행되었으므로 디스크를 분할할 필요가 없습니다. 전체 시스템 디스크는 단일 파티션으로 제공되어야 합니다.

스토리지 배열은 이미 RAID로 구성되었으므로 ACSLs 파일 시스템에 대해 ZFS를 사용해서 추가 RAID 레벨을 구성할 필요가 없습니다. ZFS RAID는 간단한 JBOD 디스크를 사용하는 경우에 필수지만 정규화된 디스크 배열을 사용할 경우 추가 RAID는 선택사항입니다. 아래 예제에서는 각 접근 방법을 보여줍니다.

미러링된 루트 파일 시스템 만들기

1. Solaris 플랫폼은 2개의 물리적 디스크 드라이브로 구성되어야 합니다. 최적의 ZFS 성능을 위해서는 시스템 디스크 및 해당 미러 드라이브를 분할합니다.

운영체제 설치 이전에 새 시스템에서 파티션 0에 전체 디스크 공간 중 대부분(아니면 전체)이 포함되도록 각 시스템 디스크 드라이브를 분할할 수 있습니다. ZFS는 전체 디스크에 액세스할 수 있는 경우 더 빠르고 안정적으로 작동합니다. 두번째 디스크에서 ZFS에 대해 정의된 파티션 크기가 주 디스크에 정의된 크기와 동일하지 확인합니다.

Solaris 11.2가 이미 설치된 시스템에서 *root* 파티션 크기를 보려면 주 시스템 디스크에서 *format* 또는 *fdisk*를 사용합니다. 같은 크기의 파티션으로 두번째 시스템 디스크를 포맷합니다. 포맷이 완료되면 디스크 레이블을 지정합니다.

2. 시스템이 작동되면 *zpool status* 명령으로 *rpool*을 확인합니다.

```
# zpool status
pool: rpool
state: ONLINE
scan: none requested
config:
    NAME                                STATE      READ  WRITE CKSUM
    rpool                                ONLINE    0     0     0
    c0t5000C5000EA48903d0s0             ONLINE    0     0     0
```

3. 두번째 시스템 디스크를 식별하고 해당 장치 ID를 확인합니다.

```
# echo | format
```

```
AVAILABLE DISK SELECTIONS:
```

- ```
0. c0t5000C5000EA48893d0 <SUN146G cyl 14087 alt 2 hd 24 sec 848>
 /scsi_vhci/disk@g5000c5000ea48893
1. c0t5000C5000EA48903d0 <SUN146G cyl 14087 alt 2 hd 24 sec 848>
 /scsi_vhci/disk@g5000c5000ea48903
```

2단계에 나와 있는 장치와 거의 비슷한 크기의 대체 장치를 선택합니다. 이 예제에서 두 번째 디스크 ID는 c0t5000C5000EA48893d0s입니다.

4. 두 번째 디스크를 *rpool*에 추가합니다.

```
zpool attach -f rpool /
 c0t5000C5000EA48903d0 /
 c0t5000C5000EA48893d0
```

시스템이 미러링된 드라이브에 대한 리실버링을 시작하여 부트 드라이브의 내용을 두 번째 드라이브로 복사합니다. 이 작업은 몇 분 정도 걸리며 중간에 재부트를 통해 중단되지 않아야 합니다.

다음을 사용해서 진행 상황을 모니터링할 수 있습니다.

```
zpool status -v
```

주 1: 리실버링이 완료될 때까지 모든 상태 표시에는 해당 디스크가 성능 저하 모드로 표시됩니다. 주 디스크에서 미러로 정보를 복사하는 중에는 디스크가 성능 저하 상태로 유지됩니다.

주 2: 디스크 레이블이 EFI 디스크로 표시되어 *zpool attach*가 실패하면 *Solaris* 관리: 장치 및 파일 시스템([http://docs.oracle.com/cd/E23824\\_01/pdf/821-1459.pdf](http://docs.oracle.com/cd/E23824_01/pdf/821-1459.pdf)) 문서에 설명된 프로세스를 따릅니다. 이 프로세스는 다음과 같이 EFI 디스크를 SMI로 전환합니다.

```
format -e
(select the drive to serve as the rpool mirror).
format> partition
partition> print
partition> label
 (specify label type "0")
 Ready to label? y
partition> modify
 (select "1" All free Hog)
 Do you wish to continue ... yes
 Free Hog Partition[6]? (specify partition "0")
 (Specify a size of "0" to the remaining partitions)
 Okay to make this current partition table? yes
```

```

Enter table name: "c1t1d0"
Ready to label disk? y
partition> quit
format> quit

```

## 5. 미러링된 *rpool* 구성을 확인합니다.

```

zpool status
pool: rpool
state: ONLINE
scan: resilvered 6.89G in 0h3m with 0 errors
config:

```

| NAME                  | STATE  | READ | WRITE | CKSUM |
|-----------------------|--------|------|-------|-------|
| rpool                 | ONLINE | 0    | 0     | 0     |
| mirror-0              | ONLINE | 0    | 0     | 0     |
| c0t5000C5000EA48903d0 | ONLINE | 0    | 0     | 0     |
| c0t5000C5000EA48893d0 | ONLINE | 0    | 0     | 0     |

인접한 노드에서 이 작업을 반복합니다.

## ACSLS 응용 프로그램에 대한 미러링된 파일 시스템 만들기

ACSLS 파일 시스템은 외부 공유 스토리지 배열의 *zpool*에 상주합니다. 아래 예제에서는 2개의 디스크만 사용하는 간단한 미러링된 배열(RAID 1)이 사용됩니다. 이러한 드라이브는 실제 드라이브일 수 있지만 대부분의 경우 연결된 스토리지 배열과 별개의 드라이브로 표시되는 가상 장치입니다.

스토리지 배열은 이미 RAID로 구성되었으므로 ACSLS 파일 시스템에 대해 ZFS를 사용해서 추가 RAID 레벨을 구성할 필요가 없습니다. ZFS RAID는 간단한 JBOD 디스크를 사용하는 경우에 필수지만 정규화된 디스크 배열을 사용할 경우 추가 RAID는 선택사항입니다. 아래 예제에서는 각 접근 방법을 보여줍니다.

### 1. 공유 스토리지 배열을 준비합니다.

표준 구성에서는 디스크 배열에서 단일 가상 드라이브를 사용합니다. 그렇지 않으면 ZFS RAID 미러링 구성에 같은 크기의 가상 드라이브 2개가 사용됩니다. 디스크 배열 또는 Solaris 포맷 유틸리티와 함께 관리 도구를 사용해서 2개의 가상 드라이브를 같은 크기가 되도록 분할할 수 있습니다.

### 2. ACSLS 설치의 기본 디렉토리를 결정합니다.

ACSLS 8.4는 모든 파일 시스템에 설치할 수 있습니다. 선택한 기본 파일 시스템은 시스템 *rpool*에 존재하지 않아야 합니다. 이미 존재하는 경우 기존 파일 시스템을 먼저 삭제해야 새 *zpool* 아래에 파일 시스템을 만들 수 있습니다.

ACSLS에 대해 기본값인 */export/home* 기본 디렉토리를 사용하려는 경우 Solaris 11.2의 기본 *root* 풀에서 */export* 파일 시스템을 삭제해야 합니다.

`/export/home`이 `rpool`에 연결되었는지 확인하려면 다음 명령을 실행합니다.

```
zfs list
```

`rpool`에서 `/export/home`을 분리하려면 먼저 보존하려는 파일 또는 디렉토리를 저장합니다. 사용자의 홈 디렉토리가 현재 `/export/home`에서 활성 상태가 아닌지 확인합니다. 그런 후 `zfs destroy`를 사용해서 `/export` 아래의 모든 항목을 제거합니다.

```
zfs destroy -r rpool/export
```

이 단계를 반복해서 인접한 노드에서 `rpool/export`를 분리합니다.

3. `format`을 사용해서 연결된 디스크 배열에 있는 드라이브의 장치 이름을 식별합니다.

```
echo | format
AVAILABLE DISK SELECTIONS:
 0. c0t5000C5000EA48893d0 <FUJITSU-MAY2073RCSUN72G-0501-68.37GB>
 /pci@0,0/pci1022,7450@2/pci1000,3060@3/sd@0,0
 /dev/chassis/SYS/HD0/disk
 1. c0t5000C5000EA48893d0 <FUJITSU-MAY2073RCSUN72G-0501-68.37GB>
 /pci@0,0/pci1022,7450@2/pci1000,3060@3/sd@1,0
 /dev/chassis/SYS/HD1/disk
 3. c0t600A0B800049EDD600000C9952CAA03Ed0 <SUN-LCSM100_F-50.00GB>
 /scsi_vhci/disk@g600a0b800049edd600000c9952caa03e
 4. c0t600A0B800049EE1A0000832652CAA899d0 <SUN-LCSM100_F-50.00GB>
 /scsi_vhci/disk@g600a0b800049ee1a0000832652caa899
```

이 예제에서는 장치 이름이 `c0t600A...`로 시작하는 디스크 배열에서 2개의 시스템 디스크와 2개의 가상 디스크가 제공되어 있습니다.

4. `acslspool`을 만듭니다.

정규화된 디스크 배열을 사용하는 표준 구성의 경우 다음과 같이 `acslspool`을 만듭니다.

```
zpool create -m /export/home acslspool /
/dev/dsk/c0t600A0B800049EDD600000C9952CAA03Ed0
```

1단계에서 제안한 대로 ZFS RAID를 추가한 경우 다음과 같이 미러링된 구성을 만듭니다.

```
zpool create -m /export/home acslspool mirror /
/dev/dsk/c0t600A0B800049EDD600000C9952CAA03Ed0 /
/dev/dsk/c0t600A0B800049EE1A0000832652CAA899d0
```

5. 새 *acslspool*을 확인합니다.

```
zpool status acslspool
pool: acslspool
state: ONLINE
scan: none requested
config:
NAME STATE READ WRITE CKSUM
acslspool ONLINE 0 0 0
 mirror-0
 c0t600A0B800049EDD600000C9952CAA03Ed0 ONLINE 0 0 0
 c0t600A0B800049EE1A0000832652CAA899d0 ONLINE 0 0 0
```

---

주:

RAID 디스크 배열을 사용할 경우, 미러링된 ZFS 구성은 선택사항입니다.

---

## 6. 새 풀에서 테스트 파일을 만들고 확인합니다.

```
cd /export/home
date > test
ls
test
cat test
Tue Jan 7 11:48:05 MST 2015
```

## 7. 풀을 내보냅니다.

```
cd /
zpool export acslspool
```

## 8. 인접 노드에 로그인합니다(이하 '새 현재 노드'라 함).

9. 새 현재 노드에서 */export/home*(또는 ACSLS에 대한 파일 시스템)이 *root* 풀의 어디에도 마운트되지 않았는지 확인합니다.

```
zfs list
```

파일 시스템이 *rpool*에 존재할 경우 이 현재 노드에서 위의 2단계를 반복합니다.

10. 새 현재 노드에서 *acslspool*을 가져오고 *acslspool*이 이 노드에 있는지 확인합니다.

```
zpool import acslspool
zpool status
pool: acslspool
state: ONLINE
```

```
scan: none requested
config:
NAME STATE READ WRITE CKSUM
acslspool ONLINE 0 0 0
 mirror-0 ONLINE 0 0 0
 c0t600A0B800049EDD600000C9952CAA03Ed0 ONLINE 0 0 0
 c0t600A0B800049EE1A0000832652CAA899d0 ONLINE 0 0 0
```

*zpool import*가 실패했으면 *zpool import -f*로 작업을 시도할 수 있습니다.

---

주:

RAID 디스크 배열을 사용할 경우, 미러링된 ZFS 구성은 선택사항입니다.

---

#### 11. 테스트 파일이 새 현재 노드에 있는지 확인합니다.

```
cd /export/home
ls
test
cat test
Tue Jan 7 11:48:05 MST 2015
```



---

---

## 4장. 소프트웨어 패키지 다운로드

이 장에서는 ACSLS HA에 필요한 주요 소프트웨어 패키지 및 패치와 제품 설명서를 다운로드하기 위한 절차에 대해 설명합니다.

### 소프트웨어 패키지 다운로드

ACSLS HA에는 3개의 주요 소프트웨어 패키지가 필요합니다.

- ACSLS 8.4
- Oracle Cluster 4.2
- ACSLS HA 8.4

이러한 소프트웨어 패키지를 각 서버 노드에 다운로드해야 합니다. 패키지를 `/opt` 디렉토리에 넣는 것이 좋습니다.

Oracle Software Delivery Cloud에서 패키지를 사용할 수 있습니다.

<https://edelivery.oracle.com/>

### ACSLS 8.4 다운로드

1. 시스템에서 웹 브라우저를 시작하여 다음 URL의 Oracle Software Delivery Cloud 웹 사이트로 이동합니다.

<https://edelivery.oracle.com>

2. 오라클 고객지원센터 담당자가 제공한 사용자 이름 및 암호로 사인인합니다.
3. Export Restrictions(수출 제한)를 읽고 동의합니다.
4. Filter Products By Programs(프로그램별 제품 필터링) 상자를 선택합니다.
5. Product(제품)에 **acsls**를 입력하고 StorageTek Automated Cartridge System Library Software (ACSL)를 선택합니다.
6. **Select Platform**(플랫폼 선택)을 누른 다음 Solaris 플랫폼(Solaris 또는 X86)을 누릅니다. **Select**(선택)를 누릅니다.
7. Selected Products(선택한 제품) 화면에서 **Continue**(계속)를 누릅니다.
8. Available Releases(사용가능 릴리스)에서 Solaris 플랫폼의 ACSLS 8.4.0.0.0 릴리스에 대한 상자를 선택하고 **Continue**(계속)를 누릅니다.
9. Oracle Terms and Restrictions(오라클 표준 조건 및 제한 사항) 화면에서 라이선스의 조건을 검토하고 이에 동의합니다. **Continue**(계속)를 누릅니다.

10. V77685-xx 패키지를 선택하고 선택한 위치에 zip 파일을 저장합니다.

절차는 5장. [ACSL S 8.4 설치](#) 를 참조하십시오.

## PostgreSQL 8.4 다운로드(선택사항)

PostgreSQL 8.3용 패키지는 ACSLS 8.4를 다운로드할 때 포함됩니다. 패키지는 제대로 작동되며 ACSLS 8.4가 설치될 때 자동으로 설치됩니다. 하지만 PostgreSQL 8.4를 설치하려는 경우 <http://www.postgresql.org/>에서 *postgres 8.4 b2* 파일을 각 노드의 */opt* 디렉토리로 다운로드합니다. ACSLS가 설치되면 */opt*에서 *.bz* 파일을 찾아 PostgreSQL 8.4를 자동으로 설치합니다. 자세한 지침은 ACSLS 8.4 패키지와 함께 포함되어 있는 README.txt 파일의 "INSTALLING POSTGRESQL" 절을 참조하십시오.

## Oracle Cluster 4.2 다운로드

Oracle Cluster 4.2를 다운로드할 때 두 단계를 수행해야 합니다.

- Solaris Cluster 기본 이미지를 다운로드합니다.
- Solaris Cluster 패치 업데이트가 필요한지 확인합니다.

절차는 6장. [Solaris Cluster 4.2 설치](#) 를 참조하십시오.

## Solaris Cluster 기본 이미지 다운로드

1. 시스템에서 웹 브라우저를 시작하여 다음 URL의 Oracle Software Delivery Cloud 웹 사이트로 이동합니다.

<https://edelivery.oracle.com>

2. 오라클 고객지원센터 담당자가 제공한 사용자 이름 및 암호로 사인인합니다.
3. Export Restrictions(수출 제한)를 읽고 동의합니다.
4. Filter Products By Programs(프로그램별 제품 필터링) 상자를 선택합니다.
5. **Oracle Solaris Cluster**를 입력하고 **Enterprise Edition**을 선택합니다.
6. **Select Platform**(플랫폼 선택)(SPARC 또는 x86)을 누른 다음 **Select**(선택)를 누릅니다.
7. Selected Products(선택한 제품) 화면에서 **Continue**(계속)를 누릅니다.
8. Available Releases(사용가능 릴리스) 아래에서 **Alternate Release**(대체 릴리스) 및 **4.2.0.0 Enterprise Edition**(Solaris 플랫폼의 경우)을 선택합니다. **Continue**(계속)를 누릅니다.
9. Copyright License(저작권 라이선스)를 읽고 동의하고 **Continue**(계속)를 누릅니다.
10. V46190-xx 패키지를 선택하고 선택한 위치에 zip 파일을 저장합니다.

## Solaris 패치 업데이트가 필요한지 여부 확인

현재 설치된 Solaris 버전을 확인합니다.

```
pkg info entire | grep Version
```

- Solaris 버전이 11.2.12 이하인 경우 Oracle Solaris Cluster 패치 업데이트가 필요하지 않습니다.
- Solaris 버전이 11.2.13 이상인 경우 Oracle Solaris Cluster 패치 업데이트가 필요합니다.

1. 오라클 고객지원센터 웹 사이트로 이동합니다.

<http://support.oracle.com>

2. **Sign In**(사인인)을 누르고 오라클 고객지원센터 담당자가 제공한 사용자 이름 및 암호를 입력합니다.
3. **Patches & Updates**(패치 및 업데이트)를 누릅니다.
4. Patch Search(패치 검색) 아래에서 **Product or Family (Advanced)**(제품 또는 제품군(고급))를 누릅니다.
5. Product is(제품) 상자에 **Solaris Cluster**를 입력합니다.
6. Release is(릴리스) 상자에서 아래쪽 화살표를 누르고 **Solaris Cluster 4.2.0**을 선택합니다. **Search**(검색)를 누릅니다.
7. 사용 중인 플랫폼(SPARC 또는 X86)과 일치하는 ORACLE SOLARIS CLUSTER 4.2.5.x.x REPO ISO 이미지를 검색하고 해당 패치를 선택합니다. **Download**(다운로드)를 누릅니다.
8. Patch Search(패치 검색) 아래에서 정보를 확인하고 **Download**(다운로드)를 누릅니다.
9. 패치 패키지를 선택하고 선택한 위치에 zip 파일을 저장합니다.

## ACSLs HA 8.4 다운로드

1. 시스템에서 웹 브라우저를 시작하여 다음 URL의 Oracle Software Delivery Cloud 웹 사이트로 이동합니다.

<https://edelivery.oracle.com>

2. 오라클 고객지원센터 담당자가 제공한 사용자 이름 및 암호로 사인인합니다.
3. Export Restrictions(수출 제한)를 읽고 동의합니다.
4. Filter Products By **Programs**(프로그램별 제품 필터링)를 선택합니다.
5. Product(제품)에 **acsls**를 입력하고 StorageTek Automated Cartridge System Library Software (ACSLs) High-Availability Agent (HA)를 선택합니다.
6. 플랫폼(SPARC 또는 x86)을 선택한 다음 **Select**(선택)을 누릅니다.
7. Selected Products(선택한 제품) 화면에서 **Continue**(계속)를 누릅니다.
8. 플랫폼의 ACSLS HA를 확인하고 **Continue**(계속)를 누릅니다.
9. Copyright License(저작권 라이선스)를 읽고 동의하고 **Continue**(계속)를 누릅니다.
10. zip 파일을 선택하고 **Download**(다운로드)를 누릅니다.
11. Copyright License(저작권 라이선스)를 읽고 동의하고 **Continue**(계속)를 누릅니다.
12. V75269-xx 패키지를 선택하고 선택한 위치에 zip 파일을 저장합니다.

절차는 [7장. ACSLS HA 8.4 설치 및 시작](#) 를 참조하십시오.

## 패치 다운로드

패치는 ACSLS 8.4, Solaris Cluster 4,2 및 ACSLS HA 8.4에 사용할 수 있습니다. 오라클 고객지원센터 사이트에서 패치 업데이트를 확인합니다.

<https://support.oracle.com>

1. Oracle ID 및 암호를 사용해서 사인인합니다.
2. **Patches and Updates**(패치 및 업데이트) 탭을 선택합니다.
3. **Search**(검색) 탭에서 **Product or Family (Advanced)**(제품 또는 제품군(고급))를 누릅니다.
4. 소프트웨어 패키지에 대해 위에 설명되어 있는 절차를 따릅니다.

## 제품 설명서 다운로드

관련된 제품 설명서를 다운로드하려면 다음을 수행합니다.

<http://docs.oracle.com>

### ACSL의 경우:

1. 스토리지를 찾고 Storage Software(스토리지 소프트웨어)를 선택합니다.
2. StorageTek ACSLS Manager 설명서를 선택한 다음 Automated Cartridge System Library Software 8.4를 선택합니다.
3. View Library(라이브러리 보기)를 선택합니다.

### Solaris Cluster의 경우:

1. 운영체제를 찾고 Operating Systems(운영체제)를 선택합니다.
2. Oracle Solaris Cluster 아래에서 Oracle Solaris Cluster 4.2를 선택합니다.
3. Cluster 4.2 및 설명서에 대해 원하는 언어를 선택합니다.

## 5장. ACSLS 8.4 설치

ACSLs 8.4 설치는 *StorageTek Automated Cartridge System Library Software 8.4 Installation Guide*에 자세히 설명되어 있습니다. ACSLS 8.4를 설치하려면 다음 상위 레벨 절차를 따릅니다.

### 첫번째 노드에 설치

1. ACSLS 8.4를 각 서버의 `/opt` 디렉토리에 다운로드합니다.
2. 다운로드한 zip 파일의 압축을 풉니다.
3. `acslspool`이 현재 노드에 마운트되었는지 확인합니다.

```
zfs list
NAME USED AVAIL REFER MOUNTPOINT
acslspool 1.60G 47.4G 1.60G /export/home
rpool 6.97G 60.0G 4.58M /rpool
rpool/ROOT 4.39G 60.0G 31K legacy
rpool/ROOT/solaris 4.39G 60.0G 3.17G /
rpool/ROOT/solaris/var 1.22G 60.0G 1.21G /var
rpool/VARSHARE 95.5K 60.0G 95.5K /var/share
rpool/dump 1.55G 60.0G 1.50G -
rpool/swap 1.03G 60.0G 1.00G -
node2:# clrg resume acsls-rg
```

4. ACSLS 설치 디렉토리로 이동하고 패키지 설치 스크립트를 실행합니다.

```
cd /opt/ACSLs_8.4.0
./pkg_install.sh
```

5. 이 노드에서 `/etc/passwd`를 확인합니다. 사용자를 기록하고 `acsss`, `acssa` 및 `acsdb` 사용자에게 지정된 그룹 ID 번호를 기록해둡니다.

```
tail -3 /etc/passwd
grep acsls /etc/group
```

인접한 노드에 설치할 때는 두번째 노드에 지정된 사용자 ID 번호가 여기에 나열된 해당 ID 번호와 일치하는지 확인합니다.

6. ACSLS 환경을 준비하고 패키지 설치 스크립트를 실행합니다.

```
. /var/tmp/acsls/.acsls_env
cd $ACS_HOME/install
./install.sh
```

주:

설치 스크립트에 데이터베이스 백업 디렉토리가 프롬프트되면 공유 디스크 배열에 마운트된 디렉토리를 지정합니다. ACSLS 설치 디렉토리(*\$installDir*) 아래의 경로를 사용합니다. 예를 들어, ACSLS를 */export/home*에 설치할 경우 데이터베이스 백업 파일에 대해 */export/home/backup*을 사용합니다.

7. 연결된 라이브러리가 연결되어 있는지 확인합니다.

```
su - acsss
$ testlmutcp <library ip address>
```

8. 라이브러리 구성 루틴을 실행합니다.

```
$ acsss_config
```

9. ACSLS 8.4에 대한 패치 업데이트를 확인합니다. 패치가 존재하는 경우 지시에 따라 다운로드 및 설치합니다.

10. 현재 노드에서 *acslspool*을 내보냅니다.

```
$ exit
cd /
zpool export acslspool
```

이 작업은 사용자 또는 작업이 현재 ACSLS 파일 시스템에 활성 상태인 경우 실패합니다.

## 인접한 노드에 설치

*STKacsls* 패키지에 있는 대부분의 파일은 공유 디스크 배열(이 파일이 이미 설치되어 있는 경우)에 추출되지만 ACSLS 사용자가 각 노드에 추가되고 여러 시스템 파일(예: SMF 시작/중지 스크립트, cron 작업)이 각 노드에 설치되어 있어야 하므로 인접한 노드에도 ACSLS 8.4를 설치해야 합니다.

1. 인접한 노드에 로그인하고 *acslspool*을 가져옵니다.

```
zpool import acslspool
```

이 작업이 실패하면 *zpool import -f acslspool*을 시도합니다.

2. ACSLS 패키지 설치 디렉토리로 이동하고 패키지를 설치합니다.

```
cd /opt/ACSL_8.4.0
./pkg_install.sh
```

설치 루틴은 일부 ACSLS 파일이 공유 드라이브의 설치 디렉토리에 존재함을 알려줍니다. 이를 덮어쓸지 여부를 선택할 수 있습니다.

3. 사용자를 확인하고 이 노드에서 *acsss*, *acssa* 및 *acsdb* 사용자에 대해 지정된 그룹 ID를 확인합니다.

```
tail -3 /etc/passwd
grep acsls /etc/group
```

이러한 번호가 첫번째 노드에 지정된 것과 일치하는지 확인합니다. 기본적으로 GID는 100이어야 하지만 GID 100이 이미 사용 중인 경우 이 번호가 변경됩니다.

UID와 GID 번호는 두 노드 사이에 일치해야 합니다. 이러한 번호가 두번째 노드에서 서로 다를 경우 두번째 노드의 UID 및 GID 번호가 첫번째 노드의 번호와 일치하도록 */etc/passwd* 파일을 편집합니다.

이제 *\$ACS\_HOME*에서 파일의 그룹 ID가 *acsls*의 그룹 소유권을 갖는지 확인합니다. 소유권이 없는 경우 *STKacsls* 패키지를 제거하고 다시 설치해야 합니다.

```
pkgrm STKacsls
pkgadd STKacsls
```

4. ACSLS 환경을 상속하고 설치 셸 스크립트를 실행합니다.

```
. /var/tmp/acsls/.acsls_env
cd $ACS_HOME/install
./install.sh
```

주 1: 설치 스크립트에서 데이터베이스 백업 디렉토리를 확인하라고 프롬프트를 표시하면 공유 디스크 배열에 마운트된 디렉토리를 지정합니다. ACSLS 설치 디렉토리 (*\$installDir*) 아래의 경로를 사용합니다. 예를 들어, ACSLS를 */export/home*에 설치할 경우 데이터베이스 백업 파일에 대해 */export/home/backup*를 사용합니다.

주 2: 두번째 노드에서 *install.sh*를 실행할 때는 ACSLS GUI를 다시 설치할 필요가 없습니다. 설치 스크립트에서 GUI에 대해 *Acsls GUI* 도메인이 존재하는 것으로 표시되면 GUI 다시 설치 프롬프트에 대해 **no**를 선택하고 GUI 제거 프롬프트에 대해 **no**를 선택합니다.

5. ACSLS 8.4에 대한 패치가 첫번째 노드에 추가되어 있는 경우 이 노드에 패치 설치를 되풀이합니다.
6. 연결된 라이브러리가 연결되어 있는지 확인합니다.

```
su - acsss
$ testlmutcp <library ip address>
```

7. 라이브러리 구성 루틴을 실행합니다.

```
$ acsss_config
```

HA 서버와 라이브러리 사이의 중복성을 사용으로 설정하려면 [그림 2.1. “각 서버 노드의 이더넷 포트 두 개에 연결된 단일 HBCr 라이브러리 인터페이스 카드”](#) 또는 [그림 2.2. “중복 전자 부품을 사용한 라이브러리의 이중 HBC 구성”](#)을 참조하십시오. `acsss_config(option-8)`를 실행할 때는 각 ACS에 대해 두 개의 연결을 정의하고 각 연결에 대해 라이브러리 IP 주소를 입력합니다.



## 6장. Solaris Cluster 4.2 설치

Solaris Cluster 설치는 Oracle technetwork 사이트에서 제공되는 *Oracle Solaris Cluster Software Installation Guide*에 자세히 설명되어 있습니다(이 문서의 [\[33\]4장. 소프트웨어 패키지 다운로드](#) 참조).

ACSLSHA 8.4는 Oracle Solaris Cluster 4.2를 사용하는 Solaris 11.2에서 지원됩니다.

### 클러스터 패키지 설치

이 절차를 따라 클러스터 소프트웨어를 설치합니다.

1. `/opt/OSC` 디렉토리를 만듭니다.

```
mkdir /opt/OSC
```

2. "[Oracle Cluster 4.2 다운로드](#)"에서 확인한 운영체제 버전에 따라 Solaris Cluster 패키지의 ISO 이미지 하나 또는 두 개를 다운로드했을 수 있습니다. 다운로드한 각 클러스터 패키지를 `/opt/OSC` 디렉토리로 이동합니다.
3. 패키지의 압축을 풉니다. 압축을 푼 각 패키지의 ISO 이미지를 식별합니다.
4. 각 ISO 이미지로부터 의사 장치를 만듭니다.

```
/usr/sbin/lofiadm -a /opt/OSC/V46190-01.iso
```

```
/usr/sbin/lofiadm -a /opt/OSC/osc-4_2_5_1_0-repo-incr.iso
```

각 의사 장치에 대해 만든 `/dev/lofi` 인스턴스 번호를 추적합니다.

5. `/opt/OSC` 디렉토리에서 각 의사 장치에 대한 마운트 지점을 만듭니다.

```
mkdir mnt
```

```
mkdir mnt1
```

6. 이 마운트 지점에 의사 장치를 마운트합니다.

```
mount -F hsfs -o ro /dev/lofi/1 /opt/OSC/mnt
```

```
mount -F hsfs -o ro /dev/lofi/2 /opt/OSC/mnt1
```

7. 다음 중 하나를 선택합니다.

- Solaris 버전이 11.2.13 이상인 경우 8단계를 진행합니다.
- Solaris 버전이 11.2.12 이하이고 기본 클러스터 이미지만 다운로드한 경우 이제 해당 기본 이미지에 대한 저장소를 게시할 수 있습니다.

```
pkg set publisher -G '*' -g file:/opt/OSC/mnt/repo ha-cluster
```

이 절의 13단계를 진행하여 패키지를 설치합니다.

8. 다음 단계에서 OSC 기본 패키지를 읽기/쓰기 파일 시스템으로 복사하여 기본 패키지로 패치 업데이트를 병합합니다.

OSC 패키지 병합을 위해 읽기/쓰기 파일 시스템을 만듭니다.

```
cd /opt/OSC
```

```
mkdir merged_iso
```

9. 기본 OSC 이미지 저장소를 만든 디렉토리로 복사합니다.

```
cp -r mnt/repo merged_iso
```

10. 이미지 두 개를 병합한 디렉토리로 동기화합니다.

```
rsync -aP mnt1/repo merged_iso
```

11. 저장소의 검색 인덱스를 재구성합니다.

```
pkgrepo rebuild -s merged_iso/repo
```

12. 병합한 패키지의 *ha-cluster* 저장소를 게시합니다.

```
pkg set-publisher -g file:/opt/OSC/merged_iso/repo ha-cluster
```

13. Oracle Solaris Cluster 패키지를 설치합니다.

```
pkg install --accept ha-cluster-full
```

인접한 노드에서 1~13단계를 반복합니다.

## scinstall 루틴

Solaris Cluster 설치 루틴은 두 노드 사이에 일련의 검사를 수행해서 두 서버 모두에서 시스템 작업을 모니터할 수 있고 시작 및 페일오버 작업을 제어할 수 있는지 확인합니다.

예비 단계:

1. *scinstall*을 실행하기 전에 바로 전에 설치한 클러스터 유틸리티에 대한 경로를 포함하는 *root*에 대한 환경을 설정하면 도움이 됩니다. */root/.profile* 파일을 편집합니다. */usr/cluster/bin*을 포함하도록 경로 문을 변경합니다.

```
export PATH=/usr/cluster/bin:/usr/bin:/usr/sbin
```

각 노드에서 이 항목을 변경해야 합니다. 새 경로를 상속하려면 로그아웃하고 다시 로그인하거나 단순히 *su -*를 실행합니다.

2. *rpc/bind*에 대한 *config/local\_only* 등록 정보가 *false*인지 확인합니다.

```
svccfg -s network/rpc/bind listprop config/local_only
```

이 등록 정보가 *true*를 반환할 경우에는 이를 *false*로 설정해야 합니다.

```
svccfg -s network/rpc/bind setprop config/local_only=false
```

그런 후 다음을 확인합니다.

```
svccfg -s network/rpc/bind listprop config/local_only
```

3. Cluster 소프트웨어의 필수 하드웨어 설정 요구 사항은 개인 네트워크 연결 2개가 두 노드 사이에서 클러스터 작업의 무중단 통신을 수행할 수 있도록 예약되어 있는 것입니다.

**그림 2.1. “각 서버 노드의 이더넷 포트 두 개에 연결된 단일 HBCr 라이브러리 인터페이스 카드”**에는 (2)로 레이블이 지정되어 있는 물리적 연결이 나와 있습니다. 각 연결은 단일 실패 지점이 클러스터의 내부 통신을 중단하지 않도록 보장하기 위해 별도 네트워크 어댑터로부터 시작됩니다. *scinstall* 루틴은 이러한 두 연결을 각각 검사하여 다른 네트워크 트래픽이 전송 중이 아닌지 확인합니다. 마지막으로 *scinstall*은 통신이 두 라인 사이에 작동하는지 확인합니다. 물리적 연결이 확인된 다음에는 루틴이 172.16으로 시작하는 개인 내부 주소에 각 인터페이스를 연결합니다.

*scinstall*을 실행하려면 먼저 이 개인 연결에 대해 설정한 각 서버에서 2개의 네트워크 포트에 대해 지정된 네트워크 장치 ID를 확인합니다. *dladm show-phys*를 실행하여 인터페이스 지정을 확인합니다.

```
dladm show-phys
```

4. 한쪽 노드의 클러스터를 나타내도록 논리 호스트 이름 및 IP 주소를 설정해야 합니다. 이 논리 호스트는 활성 호스트가 *node1* 또는 *node2*에서 실행되는지 여부에 관계없이 네트워크 통신에 안정적으로 응답합니다.

논리 호스트 이름과 논리 IP 주소를 포함하도록 두 노드 모두에서 */etc/hosts* 파일을 업데이트합니다. 이 호스트는 ACSL HA(“**ACSL HA 구성**”)를 시작할 때 활성화됩니다.

- 성공적인 클러스터 설치를 위해서는 Solaris 공통 에이전트 컨테이너를 사용으로 설정합니다. 에이전트 컨테이너가 사용으로 설정되었는지 확인합니다.

```
cacaoadm status
```

상태 응답에 에이전트 컨테이너가 시스템 시작 시 DISABLED로 표시되는 경우 다음과 같이 사용으로 설정합니다.

```
cacaoadm enable
```

## scinstall 실행

두 노드 중 하나에서 `scinstall` 명령을 실행한 후 다음 절차를 따릅니다.

- 기본 메뉴에서 **Create a new cluster**를 선택합니다.
- 하위 메뉴에서 **Create a new cluster**를 선택합니다.
- 초기 기본값을 수락합니다.
- Typical** 설치를 선택합니다.
- 클러스터의 이름(예: `acs1s_cluster`)을 지정합니다.
- 클러스터 노드 프롬프트에서 인접한 노드의 호스트 이름을 입력합니다. 노드 목록이 올바르게 바르면 수락합니다.
- 이 용도로 식별한 2개의 개인 노드 상호 연결을 정의합니다. 설치 루틴이 물리적 연결에 대한 TCP 링크를 구성하도록 허용합니다.
- 프롬프트에 따라 클러스터를 만듭니다. 쿼럼 장치로 작동할 특정 장치가 식별되지 않은 한 `scinstall` 루틴이 쿼럼 장치를 선택하도록 허용합니다.
- 유틸리티에서 클러스터 검사가 두 노드 모두에서 실패한 것으로 보고될 경우 걱정할 필요가 없습니다. 사소한 경고의 경우에도 오류가 보고됩니다. 각 노드에 대한 보고서를 검토하고 심각한 오류 또는 위반이 반환되지 않았는지 확인합니다. 루틴은 작업 중 발생한 오류 또는 경고와 관련된 세부정보를 보고하는 로그 파일 경로를 표시합니다. 로그 파일을 검토하고 식별된 심각하거나 약간 심각한 문제를 해결합니다.

`scinstall` 루틴은 노드 하나에서 실행되고 노드 두 개에 Solaris Cluster를 설치합니다. 루틴이 노드 하나를 구성하고 해당 노드를 재부트한 다음 두번째 노드를 구성하고 이 노드를 재부트하는 동작을 확인합니다.

## 클러스터 구성 확인

- 두 노드가 모두 클러스터에 포함되었는지 확인합니다.

```
clnode list -v
Node Type
---- ----
node1 cluster
node2 cluster
```

## 2. Solaris Cluster에서 사용할 수 있는 장치 목록을 확인합니다.

```
cldevice list -v
DID Device Full Device Path
d1 node1:/dev/rdisk/c0t600A0B800049EDD600000C9952CAA03Ed0
d1 node2:/dev/rdisk/c0t600A0B800049EDD600000C9952CAA03Ed0
d2 node1:/dev/rdisk/c0t600A0B800049EE1A0000832652CAA899d0
d2 node2:/dev/rdisk/c0t600A0B800049EE1A0000832652CAA899d0
d3 node1:/dev/rdisk/c1t0d0
d4 node1:/dev/rdisk/c1t1d0
d5 node2:/dev/rdisk/c1t0d0
d6 node2:/dev/rdisk/c1t1d0
```

이 예제에서 공유 디스크 장치는 d1 및 d2이고 d3 및 d4는 node1 부트 장치이며, d5 및 d6은 node2 부트 장치입니다. d1 및 d2는 어느 노드에서도 액세스할 수 있습니다.

3. 쿼럼은 3개 이상의 장치로 구성됩니다. 쿼럼은 시작 이벤트 중에 활성 노드가 되는 특정 노드를 확인하기 위해 사용됩니다.

전체 쿼럼이 구성되었는지 확인합니다.

```
clquorum list -v
Quorum Type

d1 shared_disk
node1 node
node2 node
```

두번째 shared\_disk를 선택적으로 쿼럼 장치 목록에 추가할 수 있습니다.

```
clquorum add d2
clquorum list -v
Quorum Type

d1 shared_disk
d2 shared_disk
node1 node
node2 node
```

공유 디스크 장치가 나열되지 않았으면 해당 장치의 ID를 확인하고 이를 쿼럼에 추가합니다.

- a. 각 공유 디스크에 대한 장치 ID를 식별합니다.

```
cldevice list -v
```

- b. *clsetup*을 실행하여 퀴럼 장치를 추가합니다.

```
clsetup

Select '1' for quorum.
Select '1' to dd a quorum device.
Select 'yes' to continue.
Select 'Directly attached shared disk'
Select 'yes' to continue.
Enter the device id (d<n>) for the first shared drive.
Answer 'yes' to add another quorum device.
Enter the device id for the second shared drive.
```

- c. *clquorum show*를 실행하여 퀴럼 멤버십을 확인합니다.

```
clquorum show
```

4. 전체 클러스터 구성을 검토합니다.

```
cluster check -v | egrep -v "not applicable|passed"
```

목록에서 위반된 인스턴스를 찾습니다.

5. 등록된 리소스 유형의 목록을 확인합니다.

```
clrt list
SUNW.LogicalHostname:4
SUNW.SharedAddress:2
SUNW.gds:6
```

*SUNW.gds*가 나열되지 않았으면 등록합니다.

```
clrt register SUNW.gds
```

*clrt list*를 사용해서 확인합니다.

---

---

## 7장. ACSLS HA 8.4 설치 및 시작

*SUNWscacsls* 패키지에는 Oracle Solaris Cluster와 통신하는 ACSLS 에이전트 소프트웨어가 포함됩니다. 여기에는 ACSLS와 Solaris Cluster 사이의 적절한 작업을 보장하는 특수 구성 파일 및 패치가 포함됩니다.

### 기본 설치 절차

1. 다운로드한 *SUNWscacsls.zip* 파일을 /opt에 압축을 풉니다.

```
cd /opt
unzip SUNWscacsls.zip
```

2. *SUNWscacsls* 패키지를 설치합니다.

```
pkgadd -d .
```

3. 인접한 노드에서 1~2단계를 반복합니다.
4. 두 노드 중 하나에 *acslspool*이 마운트된 상태인지 확인합니다.

```
zpool status acslspool
```

*acslspool*이 마운트되지 않았으면 다른 노드를 선택합니다.

*acslspool*이 어느 노드에도 마운트되지 않았으면 다음과 같이 현재 노드에 가져옵니다.

```
zpool import -f acslspool
```

*zpool status*를 사용하여 확인합니다.

5. *acslspool*을 소유하는 노드의 /opt/ACSLSHA/util 디렉토리로 이동하고 *copyUtils.sh* 스크립트를 실행합니다. 이 작업을 수행하면 필수 파일이 업데이트되거나 두 노드의 적합한 위치로 복사됩니다. 인접한 노드에서는 이 작업을 반복할 필요가 없습니다.

```
cd /opt/ACSLSHA/util
./copyUtils.sh
```

6. *acs1spool*이 활성 상태인 노드에서 사용자 *acsss*로 ACSLS 응용 프로그램을 시작 (*acsss enable*)하고 작동 중임을 확인합니다. 발생한 모든 문제를 해결합니다. 주요 문제는 노드에서 *STKacsls* 패키지를 제거하고 다시 설치하여 해결할 수 있습니다.

STKacsls 패키지를 다시 설치해야 하는 경우 패키지를 설치한 후 */opt/ACSLSHA/util/copyUtils.sh* 스크립트를 실행합니다.

7. *acsls*를 종료합니다.

```
su - acsss
$ acsss shutdown
$ exit
#
```

8. 활성 노드에서 *acs1spool*을 내보냅니다.

```
zpool export acslspool
```

---

주:

사용자 *acsss*가 로그인되어 있거나, 사용자 셸이 *acs1spool*의 모든 위치에서 활성 상태이거나 *acsss* 서비스가 사용으로 설정된 상태로 있는 경우 이 작업이 실패합니다.

---

9. 인접한 노드에서 *acs1spool*을 가져옵니다.

```
zpool import acslspool
```

10. 이 노드에서 ACSLS 응용 프로그램을 시작하고 라이브러리 작업이 성공적인지 확인합니다. 발생한 모든 문제를 해결합니다. 주요 문제는 노드에서 *STKacsls* 패키지를 제거하고 다시 설치하여 해결할 수 있습니다.

STKacsls 패키지를 다시 설치해야 하는 경우 패키지를 설치한 후 */opt/ACSLSHA/util/copyUtils.sh* 스크립트를 실행합니다.

## ACSL S HA 구성

이 단계는 Solaris Cluster에서 관리 및 제어하는 ACSLS 리소스를 만듭니다.

- *acsls-rs*는 ACSLS 응용 프로그램 자체입니다.
- *acsls-storage*는 ACSLS가 상주하는 ZFS 파일 시스템입니다.
- *<logical host>*는 가상 IP(두 노드에 공통적인 네트워크 ID)입니다. "[/etc/hosts 구성](#)"을 참조하십시오.

이 리소스 핸들이 생성되면 이름 *acsls-rg* 아래에 공통 리소스 그룹이 지정됩니다.

이 리소스를 구성하려면 *acs1spool*이 마운트되었는지 먼저 확인하고(*zpool list*) */opt/ACSLSHA/util* 디렉토리로 이동하여 *acsAgt configure*를 실행합니다.



```
cd /opt/ACSLSHA/util
./acsAgt configure
```

이 유틸리티는 논리 호스트 이름을 묻는 프롬프트를 표시합니다. 논리 호스트가 `/etc/hosts` 파일에 정의되어 있고 해당 IP 주소가 [2장. Solaris 시스템의 ACSLS HA 구성](#) 장에서 정의된 `imp` 그룹으로 매핑되는지 확인합니다. 또한 `acsAgt configure`를 실행하기 전에 `zpool list`를 사용하여 `acs1spool`이 현재 서버 노드에 마운트되었는지 확인합니다.

이 구성 단계를 완료하는 데 약간의 시간이 걸릴 수 있습니다. 리소스 핸들이 생성되면 작업에서 ACSLS 응용 프로그램을 시작하려고 시도합니다.

## ACSLS 클러스터 작업 모니터링

ACSLS 클러스터 작업을 확인할 수 있는 여러 우세 지점이 있습니다. Solaris Cluster가 1분마다 ACSLS 응용 프로그램을 프로브하는 경우 프로브 결과가 발생할 때마다 이 결과를 확인하는 것이 좋습니다. 프로브가 노드 스위치 오버 이벤트를 트리거하도록 상태를 반환하는 경우 한 노드의 종료 작업과 인접 노드의 시작 작업을 확인하는 것이 좋습니다. 일반적으로 시간에 따른 ACSLS 응용 프로그램의 작업 상태를 확인하는 것이 좋습니다.

ACSLS 관점의 기본 작동 우세 지점입니다. `acsss_event.log`의 끝 부분에는 경우에 따라 전체 시스템 상태의 중요한 부분이 나와 있을 수 있습니다.

`/opt/ACSLSHA/util/` 디렉토리의 `event_tail.sh` 도구는 두 노드 중 하나의 `acsss_event.log`에 대한 직접 액세스를 제공합니다. 이 도구에서 제공한 보기는 한 노드에서 다른 노드로 제어가 전환되는 경우에도 활성 상태로 유지됩니다. 이 도구는 일반 ACSLS 작업 이외에도 ACSLS 클러스터 리소스 그룹(`acs1s-rg`)의 각 상태 변경을 동적으로 추적하여 한 노드가 오프라인으로 전환되고 다른 노드가 온라인으로 전환되는 경우 실시간으로 볼 수 있습니다. 다음과 같이 셸에서 이 도구를 명제화합니다.

```
/opt/ACSLSHA/util/event_tail.sh
```

### 그림 7.1. event\_tail.sh의 예

```
event_tail.sh
2015-07-31 12:47:04 ACSLS[0]:
1426 N sa_demux.c 1 296
Server system recovery complete.

2015-07-31 12:47:04 ACSLS[0]:
1419 N sa_demux.c 1 296
Server system running.

--- Cluster Resource Groups ---
Group Name Mode Name Suspended Status

acs1s-rg acs1s-ha1 Yes Online
 acs1s-ha2 Yes Offline
```

단일 노드 관점에서 시작 및 중지 작업을 보려면 다음과 같이 해당 노드에서 `start_stop_log`를 확인합니다.

```
tail -f /opt/ACSLSHA/log/start_stop_log
```

작업 노드에서 각각의 주기적인 프로브 결과를 확인하려면 다음을 수행합니다.

```
tail -f /opt/ACSLSHA/log/probe_log
```

Solaris Cluster 및 ACSLS 클러스터 에이전트가 중요 이벤트의 세부정보를 Solaris 시스템 로그(*var/adm/messages*)로 전송합니다. 제공된 노드에서 시스템 로그를 보기 위해 */opt/ACSLSHA/log* 디렉토리에 링크가 제공됩니다.

```
tail -f /opt/ACSLSHA/log/messages
```

## ha\_console.sh 유틸리티

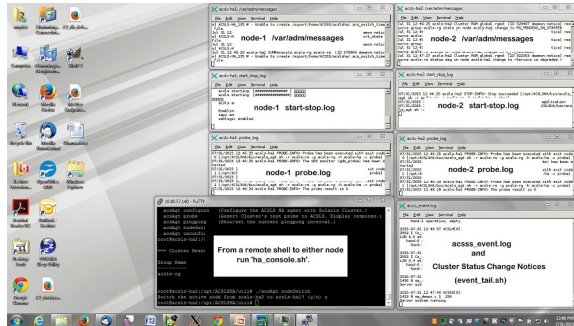
클러스터화된 구성의 여러 우세 지점 및 시간에 따른 노드 간 클러스터 제어 마이그레이션을 사용하는 경우 단일 관점에서 특정 시간 동안의 시스템 작동 작업을 쫓아가기 어려울 수 있습니다. *ha\_console.sh* 유틸리티는 광범위한 보기를 더욱 쉽게 제공할 수 있게 해줍니다.

원격 데스크탑에서 ACSLS HA 시스템의 노드 중 하나에 로그인하고 *ha\_console.sh*를 실행합니다. 이 유틸리티는 *DISPLAY*를 보낼 위치를 결정하기 위해 로그인 ID(*who am i*)를 확인합니다. 로컬 콘솔 또는 데스크탑 시스템에서 HA 노드로 직접 로그인하여 표시 내용을 확인합니다. 문제가 발생하는 경우 */opt/ACSLSHA/log* 디렉토리의 *gnome-terminal.log*에서 메시지를 찾아봅니다.

```
/opt/ACSLSHA/util/ha_console.sh
```

이 유틸리티는 이 섹션에 설명한 두 노드의 모든 로그를 모니터합니다. 로컬 콘솔 화면의 *gnome* 단말기 창 7개를 실행합니다. 다음 방식으로 화면에서 창을 구성하는 데 도움이 될 수 있습니다.

그림 7.2. gnome 단말기 창 구성



단일 단말기 화면에서 전체 ACSLS 클러스터 컴플렉스의 광범위한 보기가 표시됩니다.

원격 시스템이 로컬 화면으로 표시 데이터를 전송하므로 *yopen X-11*은 로컬 시스템에 액세스합니다. UNIX 시스템에서 이를 수행하기 위한 명령은 *xhost +*입니다. Windows 시스템에 *xming* 또는 *exceed*와 같은 X-11 클라이언트 소프트웨어가 설치되어 있어야 합니다.

*ha\_console.sh*를 사용하는 데 어려움이 있는 경우 로컬 시스템에서 각 노드까지 다중 로그인 세션을 열어 이 절에서 설명한 다양한 로그를 확인합니다.

## 클러스터 작업 확인

1. `acs1sha`가 시작되고 Solaris Cluster에 등록되었으면 클러스터 명령을 사용해서 ACSLS 리소스 그룹 및 연관된 리소스의 상태를 확인합니다.

```
clrg status
=== Cluster Resource Groups ===
Group Name Node Name Suspended Status

acs1s-rg node1 No Online
 node2 No Offline

clrs status
=== Cluster Resources ===
Resource Name Node Name State Status Message

acs1s-rs node1 Online Online
 node2 Offline Offline
acs1s-storage node1 Online Online
 node2 Offline Offline
<logical host> node1 Online Online
 node2 Offline Offline
```

2. 초기 테스트를 시작하기 위해 클러스터 페일오버 준비를 일시적으로 중단합니다.

```
clrg suspend acs1s-rg
clrg status
```

3. 활성 노드에서 대기 노드로 클러스터 전환 작업을 테스트합니다.

```
cd /opt/ACSLSHA/util
./acsAgt nodeSwitch
```

스위치 오버 작업은 이전 절에서 설명한 절차를 사용하여 다양한 측면에서 모니터링할 수 있습니다.

4. ACSLS 서버의 논리 호스트 이름을 사용해서 ACSLS 클라이언트 시스템에서 네트워크 연결을 확인합니다.

```
ping acs1s_logical_host
ssh root@acs1s_logical_host hostname
passwd:
```

이 작업은 활성 노드의 호스트 이름을 반환합니다.

5. ACSLS 작업을 확인합니다.

```
su acsss
$ acsss status
```

6. 반대 노드에서 3~5단계를 반복합니다.
7. 클러스터 페일오버 준비를 재개합니다.

```
clrg resume acsls-rg
clrg status
```

8. 다음 테스트는 노드 페일오버 동작 확인과 관련됩니다.

다중 페일오버 시나리오를 차례대로 수행하려면 기본 pingpong 간격을 20분에서 5분으로 줄입니다. 자세한 내용은 [8장. ACSLA HA 세부 조정](#) 장을 참조하십시오. 테스트용인 경우 기본 설정을 낮추는 것이 좋습니다.

pingpong 간격을 변경하려면 `/opt/ACSLSHA/util` 디렉토리로 이동하고 `acsAgt pingpong`을 실행합니다.

```
./acsAgt pingpong
Pingpong_interval
 current value: 1200 seconds.
 desired value: [1200] 300
Pingpong_interval : 300 seconds
```

9. 활성 노드를 재부트하고 [“ACSLA 클러스터 작업 모니터링”](#)에서 제안된 관점에 따라 두 시스템 콘솔에서 작업을 모니터링합니다. 대기 노드에 대한 자동 페일오버 작업을 확인합니다.
10. 4단계에서 제안된 클러스터 시스템에서 논리 호스트에 대한 네트워크 액세스를 확인합니다.
11. ACSLS 작업이 새 노드에서 활성 상태이면 이 노드를 재부트하고 반대 노드에 대한 페일오버 작업을 확인합니다.

`ha_console.sh`를 사용하여 작업을 모니터링하는 경우 재부트 노드와 연관된 창이 사라진 것으로 보일 수 있습니다. 노드가 다시 작동되면 한 노드에서 `ha_console.sh` 명령을 다시 한 번 실행하여 새로 재부트된 노드의 창을 복원합니다.

12. 4단계에서 제안된 네트워크 확인을 반복합니다.

[9장. ACSLS 클러스터 작업](#)에는 페일오버 시나리오의 전체 세트가 나와 있습니다. ACSLS HA 시스템을 운용 환경에 배치하기 전에 여러 시나리오를 테스트할 수 있습니다. 시스템을 운용 환경으로 되돌리기 전에 권장되는 pingpong 간격 설정을 복원하여 지속적으로 페일오버가 반복되지 않게 합니다.

## 8장. ACSLS HA 세부 조정

이 장에서는 라이브러리 컴플렉스에 최적의 페일오버 정책을 설정하는 방법, 원치 않는 페일백 이벤트를 방지하기 위해 기본 pingpong 간격을 조정하는 방법 및 페일오버 이벤트에 대해 전자 메일 알림을 등록하는 방법에 대해 설명합니다.

### 라이브러리 통신을 위한 페일오버 정책 정의

ACSLs HA 에이전트는 ACSLS 및 연결된 라이브러리 사이의 통신을 지속적으로 모니터링합니다. 이러한 통신은 지속적인 ACSLS 운영에 핵심적입니다. 하지만 실패한 라이브러리 통신에서 수행해야 할 작업(있는 경우)은 로컬 ACSLS HA 관리자에 의해 결정되는 정책에 따라 달라집니다.

정책 테이블 `$ACS_HOME/acs1sha/ha_acs_list.txt`를 사용해서 로컬 관리자는 HA 복구가 필요한 모든 ACS에 대해 원하는 페일오버 작업을 정의할 수 있습니다. 라이브러리 통신 오류 시, 그리고 관리자의 지시에 따라 ACSLS HA 에이전트는 대체 노드에서 ACS 통신이 성공적으로 확인된 경우 해당 노드로 페일오버를 수행합니다.

다중 ACS 환경에서는 단일 ACS와의 통신이 실패할 때 ACSLS HA 시스템이 페일오버를 수행하는 것이 바람직할 수 있습니다. 페일오버 작업이 연결된 모든 라이브러리의 운영을 중단하기 때문에 관리자는 데이터 센터에서 더 중요한 ACS에 대해 일반적인 페일오버 작업을 제한하는 방법을 선호할 수 있습니다. 정책 레코드는 라이브러리 통신이 손실될 때 클러스터 페일오버 작업이 필요한 각 ACS에 대한 `ha_acs_list.txt`에 생성됩니다. 각 레코드에는 다음 2개 필드가 포함됩니다.

ACS Number    Fail-over Action (true or false)

첫번째 필드는 ACS ID이고 두번째 필드는 `true` 또는 `false`의 부울 값입니다. 정책 설정의 논리는 다음과 같습니다.

- 두번째 필드가 `false`이면 ACS에 대한 통신이 실패하고 복원할 수 없는 경우에도 ACSLS HA 에이전트가 대체 노드에 대한 클러스터 페일오버 작업을 시작하지 않습니다.
- 두번째 필드가 `true`이면 주 노드로부터의 통신을 다시 설정하려는 시도가 실패할 때마다 ACSLS HA 에이전트는 클러스터 페일오버 작업을 명제화합니다. 시스템은 대체 노드에서 라이브러리 연결이 확인된 경우에만 페일오버를 수행합니다.

이 파일에 나열되지 않은 ACS의 경우 기본 작업은 `false`입니다.

## RE(중복 전자 부품)를 포함하는 라이브러리

RE(중복 전자 부품)가 포함된 라이브러리의 경우 ACSLS HA 에이전트는 클러스터 페일오버 작업을 선택하기 전에 통신을 대체 RE 경로로 전환하려고 시도합니다. 이러한 RE 전환 작업은 단일 SL8500 또는 SL3000이나 이중 LMU를 포함하는 오래된 9310에만 적용됩니다. 분할된 라이브러리에서는 자동 RE 전환이 시도되지 않습니다.

## 페일오버 *Pingpong\_interval* 설정

Solaris Cluster *Pingpong\_interval*은 첫번째 클러스터 페일오버 이벤트 이후 전체 복구를 복원할 수 없는 경우 페일오버 작업이 반복되지 않도록 방지하는 시간 초과 등록 정보입니다.

이 등록 정보는 ACSLS 리소스 그룹에 대한 사용자 수정 가능한 등록 정보입니다. 기본값은 20분으로 설정됩니다. 이 설정의 경우 첫번째 페일오버 이벤트는 ACSLS HA 에이전트에서 페일오버 작업을 요청할 때 즉시 발생합니다. 하지만 페일오버 작업을 트리거할 수 있는 조건이 새 클러스터 노드에서 지워지지 않은 경우에는 정의된 pingpong 간격이 만료될 때까지 이후의 페일오버 작업이 지연됩니다. 그 결과 근본 문제가 해결될 때까지 하나의 클러스터 노드와 다른 노드 사이의 불필요한 제어 충돌이 방지됩니다.

이 등록 정보의 기본 설정을 변경하려면 `$ACS_HOME/acslsha/pingpong_interval` 파일에 정의된 기본 숫자를 수정하면 됩니다. 해당 숫자는 초로 표현됩니다.

기본 설정인 1200초는 대부분의 중간 라이브러리부터 큰 라이브러리 구성까지 적합한 설정입니다. 이 등록 정보에 대한 최적의 시간 초과 값은 라이브러리 구성에 존재하는 실제 LSM 및 테이프 드라이브 수에 따라 달라집니다. 더 큰 라이브러리 구성에서는 페일오버 이벤트 후 복구하는 데 시간이 이보다 오래 걸리므로 LSM이 10개 이상이거나 드라이브가 40개 이상인 시스템 구성에서는 이 숫자 간격을 더 크게 설정해야 합니다.

40개 LSM 구성의 경우에는 1800(30분)으로 설정하는 것이 좋으며, 1~4개 LSM으로 구성된 작은 라이브러리의 경우에는 900(15분)으로 설정하는 것이 좋습니다.

여기에서 수행한 변경 사항은 `acsAgt configure` 명령을 사용하여 ACSLS HA를 재구성할 때까지 적용됩니다.

```
cd /opt/ACSLSHA/util
./acsAgt configure
```

이 명령은 `acsls-rg` 리소스 그룹이 이미 활성화되어 있는 경우에도 명제화될 수 있습니다. 일반적인 HA 작업에 영향을 주지 않고 새로운 기본 설정을 등록합니다.

*pingpong\_interval* 설정은 `acsAgt pingpong`을 사용하여 테스트를 위해 동적으로 변경할 수 있습니다. 이 명령을 사용하는 값 세트는 `acsAgt configure`로 리소스 그룹을 다시 시작할 때까지 유효합니다.

## 시스템 이벤트의 전자 메일 알림 등록

관리 책임이 있는 사용자는 시스템 부트 이벤트 및 ACSLS HA 클러스터 페일오버 이벤트를 포함해서 시스템 이벤트의 자동 전자 메일 알림을 등록할 수 있습니다.

이러한 이벤트를 등록하려면 사용자가 자신의 전자 메일 주소를 다음 디렉토리 아래의 해당 파일에 추가해야 합니다.

```
$ACS_HOME/data/external/email_notification/
 boot_notification
 ha_failover_notification
```

제목 표시 아래에 각 받는 사람에 대한 전자 메일 주소를 한 라인에 입력합니다. 그런 다음에는 시스템이 부트되거나 HA 클러스터가 대기 노드로 페일오버를 수행할 때마다 등록된 각 사용자에게 전자 메일 알림이 전송됩니다.

이 기능은 메일 전송 서비스가 ACSLS 서버에 사용으로 설정되었고 네트워크 방화벽 제약 조건에서 데이터 센터로부터의 전자 메일 통신이 허용된다고 가정합니다.





## 9장. ACSLS 클러스터 작업

Solaris Cluster는 작업 제어를 한 서버 노드에서 다음 서버 노드로 이전하여 심각한 오류 시나리오에서도 자동으로 시스템 복구를 수행할 수 있도록 설계되었습니다. 하지만 Solaris 시스템에서 발생하는 대부분의 오류에서는 복구를 위해 전체 시스템에 대한 스위치 오버 작업이 필요하지 않습니다.

- 네트워크 통신과 관련된 오류는 Solaris IPMP에서 신속하고 조용하게 처리됩니다.
- 시스템 디스크 오류는 Solaris ZFS에서 조용히 자동으로 처리됩니다.
- 연결된 스토리지 배열의 단일 디스크 드라이브와 관련된 오류는 스토리지 배열 펌웨어에 의해 자동으로 복구됩니다. 그리고 스토리지 배열에 디스크 오류 복구 기능이 부족할 경우에는 Solaris ZFS가 미러링된 구성의 대체 드라이브에 대한 무중단 디스크 I/O를 제공합니다.
- 공유 배열에 대한 HBA 포트가 실패할 경우 Solaris는 대체 포트로 자동으로 전환합니다. 마찬가지로, 공유 배열의 컨트롤러 모듈이 실패하거나 상호 연결 케이블이 분리된 경우 Solaris는 즉시 디스크 리소스에 연결되는 대체 경로로 돌아갑니다.
- 라이브러리 통신 경로에서 발생하는 오류는 ACSLS의 이중 TCP/IP 논리에 의해 자동으로 복구됩니다. 실패한 라이브러리 컨트롤러 카드의 작업은 라이브러리 RE(중복 전자 부품)와 연관된 ACSLS HA 논리에 의해 자동으로 복구됩니다.
- ACSLS에서 실행 중인 여러 프로세스가 실패하면 ACSLS 데몬이 실패한 프로세스를 즉시 재부트합니다.
- ACSLS 데몬 자체가 실패하거나 남은 ACSLS 서비스의 실행이 중지될 경우에는 Solaris SMF(서비스 관리 기능)를 사용해서 실패한 서비스를 즉시 다시 시작할 수 있습니다.

이러한 모든 시나리오는 Solaris Cluster의 개입 없이 자동으로 신속하게 처리됩니다. 하지만 다른 심각한 결함으로 인해 활성 서버 노드의 ACSLS 작업에 영향을 줄 경우, ACSLS HA는 Solaris Cluster가 대체 노드로 제어를 전환하도록 지시합니다.

ACSLs HA는 시작된 다음 1분마다 시스템을 프로브하여 다음과 같은 이벤트가 발생하는지 확인합니다.

- 연결된 라이브러리에 대한 통신 손실
- ACSLS 논리 호스트에 대한 네트워크 연결 손실
- 클라이언트 호출에 대한 RPC 수신기 포트 연결 손실
- ACSLS 파일 시스템에 대한 액세스 손실
- acsls SMF 서비스의 복구 불가능한 유지 관리 상태

이러한 이벤트는 모두 클러스터 페일오버를 트리거합니다. Solaris Cluster는 또한 활성 서버 노드에서 치명적인 시스템 조건이 발생할 때 페일오버를 수행합니다.

## ACSLs의 클러스터 제어 시작

클러스터 페일오버 제어를 활성화하려면 다음을 수행합니다.

```
cd /opt/ACSLSHA/util
./acsAgt configure
```

유틸리티에서 논리 호스트 이름을 묻는 프롬프트를 표시합니다. 논리 호스트가 */etc/hosts* 파일에 정의되어 있고 해당 IP 주소가 [2장. Solaris 시스템의 ACSLS HA 구성](#) 장에서 정의된 *ipmp* 그룹으로 매핑되는지 확인합니다. *acsAgt configure*를 실행하기 전에 *zpool list*를 사용하여 *acs1spool*이 현재 서버 노드에 마운트되었는지 확인합니다.

이 작업은 ACSLS의 클러스터 제어를 시작합니다. Solaris Cluster는 시스템을 모니터링하면서 1분마다 프로브하여 특히 ACSLS의 상태와 일반적인 Solaris 시스템을 확인합니다. 치명적인 것으로 간주되는 모든 조건에서는 대체 노드에서 작업이 시작됩니다.

ACSLs 리소스 그룹의 클러스터 상태를 확인하려면 다음을 수행합니다.

```
clrg status
```

화면에서 다음이 수행됩니다.

- 각 노드의 상태를 표시합니다.
- 어느 노드가 활성 노드인지 식별합니다.
- 페일오버 작업이 일시 중지되었는지 여부를 확인합니다.

## acsls-storage에 대한 페일오버 정책 설정

해당 노드 및 공유 RAID 디스크 장치 사이에서 통신이 끊어질 때마다 활성 노드를 재부트하도록 *acsls-storage* 리소스에 정책을 설정하는 것이 좋습니다. 이 작업으로 인해 활성 노드가 디스크에 연결할 수 없는 경우 활성 노드에서 제어를 포기하고 Solaris Cluster에서 대체 노드에 제어권을 전달하도록 합니다. *Failover\_mode*를 SOFT에서 HARD로 설정하면 공유 스토리지 장치에 대한 통신이 끊길 때마다 활성 노드를 재부트합니다.

기존 *Failover\_mode*를 확인하려면 다음 명령을 실행합니다.

```
clrs show -v acsls-storage | grep Failover
```

*Failover\_mode*는 다음과 같이 HARD로 설정해야 합니다.

```
clrs set -p Failover_mode=HARD acsls-storage
```

## 클러스터 제어 시 ACSLS 작업 및 유지 관리

클러스터 제어가 활성화된 다음에는 일반적인 방식으로 ACSLS를 작동할 수 있습니다. 표준 *acsss* 제어 유틸리티를 사용해서 ACSLS를 시작 및 중지합니다. 클러스터 제어 시에는 독

립형 ACSLS 서버에서 응용 프로그램을 시작 및 중지할 때와 마찬가지로 ACSLS 서비스를 시작 및 중지할 수 있습니다. 작업은 다음과 같은 표준 *acsss* 명령을 사용해서 관리됩니다.

```
acsss enable
acsss disable
acsss db
```

이러한 명령을 사용해서 *acsss* 서비스를 수동으로 시작하거나 중지하면 Solaris Cluster가 페일오버 작업에 영향을 주지 않습니다. Solaris SMF 명령(예: *svcadm*)을 사용해도 클러스터가 영향을 주지 않습니다. *acsss* 서비스가 중단될 때마다 기본적으로 이러한 서비스를 다시 시작해야 하는 주체는 클러스터가 아니라 SMF입니다.

Solaris Cluster는 다음과 같은 환경에서만 인접한 노드에서 제어를 복원하기 위해 작업에 영향을 줍니다.

- ACSLS 파일 시스템과의 통신 손실
- 모든 중복 공용 이더넷 포트와의 통신 손실
- 지정된 라이브러리와 통신 손실 및 복구할 수 없는 통신 상태

## 클러스터 제어 일시 중지

유지 관리 작업에서 원하지 않는 클러스터 페일오버 이벤트를 트리거할 것으로 의심되는 경우 *acs1s* 리소스 그룹의 클러스터 제어를 일시 중지합니다.

클러스터 제어를 일시 중지하려면 다음을 수행합니다.

```
clrg suspend acs1s-rg
```

리소스 그룹이 일시 중지되면 이러한 작업을 트리거할 수 있는 조건에 관계없이 Solaris Cluster가 인접한 노드로 제어를 전환하려고 시도하지 않습니다.

따라서 라이브러리 운용이 완전히 작동 중인 상태라도 시스템 복구를 보다 적극적으로 수행할 수 있습니다.

일시 중지된 모드에서 활성 노드가 재부트될 경우에는 재부트 후에 *acs1spool*을 마운트하지 않으며, ACSLS 작업이 중단됩니다. 이 상황을 해결하기 위해서는 클러스터 제어를 재개합니다.

클러스터 제어를 재개하려면 다음을 수행합니다.

```
clrg resume acs1s-rg
```

공유 디스크 리소스가 현재 노드에 마운트되었으면 정상 작업이 재개됩니다. 하지만 Solaris Cluster에서 활성화 시 *zpool*이 마운트되지 않은 것을 발견하면 제어를 인접한 노드로 즉시

전환합니다. 인접 노드에 액세스할 수 없는 경우 제어가 현재 노드로 다시 전환됩니다. 클러스터에서 `acs1spool`을 마운트하고 이 노드에서 ACSLS 서비스를 시작하려고 합니다.

## ACSLs HA 클러스터 전원 차단

다음 절차에서는 ACSLS HA 시스템 전원을 차단해야 할 경우 안전한 전원 차단 순서를 제공합니다.

1. 클러스터에서 활성 노드를 확인합니다.

```
clrg status
```

온라인 노드를 찾습니다.

2. 활성 노드에 `root`로 로그인하고 ACSLS 리소스 그룹의 Solaris Cluster 제어를 중단합니다.

```
clrg suspend acsls-rg
```

3. 사용자 `acsss`로 전환하고 `acsss` 서비스를 종료합니다.

```
su - acsss
$ acsss shutdown
```

4. `acsss`로 로그아웃하고 정상적으로 노드 전원을 차단합니다.

```
$ exit
init 5
```

5. 대체 노드에 로그인하고 `init 5`를 사용해서 전원을 차단합니다.
6. 물리적 전원 스위치를 사용해서 공유 디스크 배열의 전원을 차단합니다.

## 일시 중지된 ACSLS 클러스터 시스템 전원 켜기

제어된 종료 이전에 활성 상태였던 노드에서 ACSLS 작업을 복원하려면 다음을 수행합니다.

1. 물리적 전원 스위치를 사용해서 두 노드 전원을 로컬로 켜거나 Sun Integrated Lights Out Manager를 사용해서 원격으로 전원을 켭니다.
2. 공유 디스크 배열 전원을 켭니다.
3. `root`로 어느 한 노드에 로그인합니다.
4. `acsss`로 로그인하려고 시도하거나 `$ACS_HOME` 디렉토리를 나열하려고 시도하면 공유 디스크 리소스가 어느 노드에도 마운트되지 않은 것을 확인할 수 있습니다. 클러스터 모니터링을 재개하려면 다음 명령을 실행합니다.

```
clrg resume acsls-rg
```

이 작업에서는 시스템이 작동 중지될 때 활성 상태였던 노드에 Solaris Cluster가 공유 디스크를 마운트합니다. 이 작업에서는 또한 *acsss* 서비스를 자동으로 재부트하고 정상 작업이 재개됩니다.

## 단일 노드 클러스터 만들기

일부 경우에는 다른 노드가 서비스 중인 상태에서 ACSLS가 한 노드의 독립형 서버 환경에서 작업을 계속해야 할 수 있습니다. 이 경우는 하드웨어 유지 관리, 운영 체제 업그레이드 또는 Solaris Cluster 업그레이드의 경우에 해당합니다.

다음 절차에 따라 독립형 ACSLS 서버를 만듭니다.

1. 비클러스터 모드에서 원하는 노드를 재부트합니다.

```
reboot -- -x
```

SPARC 서버의 OBP(Open Boot Prom)에서 비클러스터 모드로 부트하려면 다음을 수행합니다.

```
ok: boot -x
```

X86 서버에서는 GRUB 부트 메뉴를 편집해야 합니다.

- a. 시스템의 전원을 켭니다.
  - b. GRUB 부트 메뉴가 나타나면 **e**(edit)를 누릅니다.
  - c. 하위 메뉴에서 화살표 키를 사용해서 **kernel /platform/i86pc/multiboot**를 선택합니다. 이 항목을 선택한 상태로 **e**를 누릅니다.
  - d. 편집 모드에서 **-x**를 다중 부트 옵션 **kernel /platform/i86pc/multiboot -x**에 추가하고 **return**을 누릅니다.
  - e. 다중 부트 **-x** 옵션이 선택된 상태로 **b**를 눌러서 해당 옵션으로 부트합니다.
2. 부트 주기가 완료되면 루트로 로그인하고 ACSLS Z-pool을 가져옵니다.

```
zpool import acslspool
```

디스크 리소스가 다른 노드에 연결된 상태로 남아 있다면 필요한 경우 **-f**(force) 옵션을 사용합니다.

```
zpool import -f acslspool
```

3. *acsss* 서비스를 실행합니다.

```
su - acsss
$ acsss enable
```

---

## 10장. 소프트웨어 구성 요소 설치, 업그레이드 및 제거

이 장에서는 ACSLS HA와 연관된 다양한 구성 요소를 업그레이드하거나 제거할 경우 따라야 하는 절차에 대해 설명합니다.

### ACSLs 패치 설치

STKacsls 패키지에 대한 패치를 설치하려면 다음을 수행합니다.

1. 클러스터 제어를 일시 중지합니다.

```
clrg suspend acsls-rg
```

2. 패치를 `/opt` 디렉토리에 다운로드하고 패키지 압축을 풉니다.
3. `/opt/ACSLs_8.x.x` 디렉토리로 이동하고 패치 `README.txt` 파일의 지침을 따릅니다.
4. ACSLS 작업을 사용 안함으로 설정하고 인접 노드로 제어를 전환합니다.

```
su - acsss
$ acsss shutdown
$ exit
cd /opt/ACSLSHA/util
acsAgt nodeSwitch
```

5. 새 노드에 ACSLS 패치를 설치합니다.
6. 업데이트된 ACSLS 구조에 ACSLS HA 변경 사항을 적용합니다.

```
cd /opt/ACSLSHA/util
./copyUtils.sh
```

7. ACSLS 서비스를 사용으로 설정합니다.

```
su - acsss
acsss enable
```

8. `acsls-rg` 리소스 그룹의 클러스터 제어를 재개합니다.

```
clrg resume acsls-rg
```

## ACSLs 패키지 제거

ACSLs 패키지 제거는 ACSLS를 업그레이드하는 경우 필요할 수 있습니다. 이렇게 하려면 클러스터 제어를 사용 안함으로 설정하고, 두 노드에서 ACSLS 서비스를 중단한 후 각 노드에서 패키지를 제거해야 합니다. 다음 절차를 따르십시오.

1. 클러스터 제어를 일시 중지합니다.

```
node1:# clrg suspend acsls-rg
```

2. 활성 노드에서 ACSLS를 종료합니다.

```
node1:# su - acsss
node1:$ acsss shutdown
node1:$ exit
node1:#
```

3. 공유 디스크 배열에서 파일 시스템을 내보냅니다.

```
node1:# cd /
node1:# zpool export acslspool
```

사용자 `acsss`로 로그인되어 있으면 이 작업이 실패합니다.

4. 대체 노드에 로그인하고 공유 디스크 배열을 가져옵니다.

```
node1:# ssh <alternate node>
node2:# zpool import acslspool
```

5. ACSLS를 종료합니다.

```
node2:# su - acsss
node2:$ acsss shutdown
node2:$ exit
node2:#
```

6. `STKacsls` 패키지를 제거합니다.

```
node2:# pkgrm STKacsls
```

7. 원본 노드로 돌아가고 `STKacsls` 패키지를 제거합니다.

```
node2:# exit
node1:# pkgrm STKacsls
```



## ACSLs의 업그레이드 릴리스 설치

ACSLs의 새 릴리스를 설치하려면 먼저 두 노드에서 *STKacsls* 패키지를 제거해야 합니다. 위 절에 설명된 절차를 참조하십시오. 새 패키지를 설치하려면 다음 절차를 따릅니다.

1. *STKacsls* 패키지를 */opt* 디렉토리에 다운로드하고 패키지 압축을 풉니다. 대체 노드에서 이 단계를 반복합니다.
2. Solaris Cluster가 일시 중지되면 공유 디스크 배열(*acslspool*)이 현재 노드에 마운트되었는지 확인합니다.

```
node1:# zpool list
```

*acslspool*이 마운트되지 않았으면 대체 노드에 로그인합니다. 어느 노드에도 마운트되지 않았으면 *acslspool*을 가져옵니다.

3. */opt/ACSLs\_8.x.x* 디렉토리로 이동하고 *README.txt* 파일의 지침을 따릅니다.
4. *acslspool*을 내보냅니다.

```
node1:# zpool export acslspool
```

사용자 *acsss*로 로그인되어 있으면 이 작업이 실패합니다.

5. 대체 노드에 로그인하고 1~3단계를 반복합니다.
6. */opt/ACSLSHA/util* 디렉토리로 이동하고 *copyUtils.sh*를 실행합니다.

```
node2:# cd /opt/ACSLSHA/util
node2:# ./copyUtils.sh
```

7. ACSLS 라이브러리 제어를 시작합니다.

```
node2:# su - acsss
node2:$ acsss enable
node2:$ exit
node2:#
```

8. *acsls* 리소스 그룹의 클러스터 제어를 재개합니다.

```
node2:# clrg resume acsls-rg
```

## ACSLs HA 다시 설치 또는 업그레이드 설치

*SUNWscacsls* 패키지를 다시 설치하려면 ACSLS 라이브러리 작업을 중단하고 *acsls-rg* 리소스 그룹을 오프라인으로 전환해야 합니다.

1. ACSLS를 종료합니다.

```
su - acsss
$ acsss shutdown
$ exit
#
```

2. 사용자 정의된 변경 사항을 *ha\_list.txt* 및 *pingpong\_interval*에 저장합니다.

```
cd $ACS_HOME/acslsha
cp ha_list.txt ha_list.save
cp pingpong_interval pingpong_interval.save
```

3. *acsls-rg* 리소스 그룹을 오프라인으로 전환합니다.

```
clrg offline acsls-rg
```

이 작업은 *acslspool*을 ZFS 파일 시스템에서 마운트 해제합니다.

4. *acsls-rg* 리소스 그룹을 구성 해제합니다.

```
cd /opt/ACSLSHA/util
./acsAgt unconfigure
```

5. 업데이트된 *SUNWscacsls* 패키지를 두 노드에 다운로드하고 압축을 풉니다.
6. *acslspool*을 두 노드 중 하나에 다시 마운트합니다.

```
zpool import -f acslspool
```

7. ACSLS를 시작하여 올바르게 작동하는지 확인합니다.

```
su - acsss
$ acsss enable
$ exit
#
```

8. *copyUtils.sh* 유틸리티를 실행합니다.

```
cd /opt/ACSLSHA/uti
./copyUtils.sh
```

9. 2단계에서 저장한 사용자 정의 파일을 복원합니다.

```
cd $ACS_HOME/acslsha
cp ha_list.save ha_list.txt
cp pingpong_interval.save pingpong_interval
```

10. *acsIs-rg* 리소스 그룹을 구성합니다.

```
cd /opt/ACSLSHA/util
./acsAgt configure
```

이 작업은 ACSLS를 중지하고 다시 시작합니다.

몇 분 안에 *c1rg status* 명령으로 *acsIs-rg* 리소스 그룹이 다시 온라인 상태가 되고 ACSLS가 작동 중임이 표시되어야 합니다.

## Solaris Cluster 업그레이드

Solaris Cluster를 제거 또는 업그레이드하기 전에 ACSLS의 작동을 중지하고 *acsIs-rg* 리소스 그룹을 구성 해제합니다.

```
su - acsss
$ acsss shutdown
$ exit
cd /opt/ACSLSHA/util
./acsAgt unconfigure
```

특정 업그레이드 절차는 현재 Solaris Cluster 설명서를 참조하십시오. Solaris Cluster 업그레이드를 위한 일반적인 명령은 다음과 같습니다.

```
scinstall -u
```

Solaris Cluster 설치가 업데이트되면 *acsIs-rg* 리소스 그룹을 구성합니다.

```
cd /opt/ACSLSHA/util
./acsAgt configure
```

## Solaris Cluster 제거

1. ACSLS 및 *acsIs-rg* 리소스 그룹을 작동 중지합니다.

```
su - acsss
$ acsss shutdown
$ exit
cd /opt/ACSLSHA/util
./acsAgt unconfigure
```

2. 두 노드를 모두 비클러스터 모드로 재부트합니다.

```
reboot -- -x
```

3. 두 노드가 모두 작동 중인 경우 노드 중 하나에 로그인하고 Solaris Cluster 패키지를 제거합니다.

```
scinstall -r
```

## 11장. 클러스터 로깅, 진단 및 테스트

이 장에서는 ACSLS HA 설치를 테스트하고 시스템에서 발생할 수 있는 문제 진단 및 해결을 위해 사용할 수 있는 여러 가지 리소스에 대해 설명합니다.

### 전체 클러스터 작업 모니터링

시작 중 또는 스위치 오버 이벤트 중 진행되는 작업은 노드 2개에 널리 배포됩니다. 따라서 테스트 중 전체 작업을 확인하도록 선택한 우세 지점은 작업이 진행되면서 펼쳐지지 않은 이벤트를 확인할 수 있는 기능을 결정할 수 있습니다. “[ha\\_console.sh 유틸리티](#)”에 광범위한 보기 설정에 대한 절차가 설명되어 있습니다.

테스트 중 전체 HA 동작을 관찰하는 데 권장되는 대시보드 구성은 셀 창 8개, 각 노드의 창 4개를 포함하는 것입니다.

1. *root*에 대한 명령 셸은 필요에 따라 다양한 명령을 명제화하도록 각 노드에 대해 예약되어야 합니다.
2. 시스템 */var/adm/messages* 파일의 끝 부분을 표시하도록 각 노드의 창을 설정합니다.

```
tail -f /var/adm/messages
```

Solaris Cluster는 모든 정보 메시지를 이 로그 파일에 인쇄합니다.

3. *acsls-rs* 리소스 *start\_stop log*의 끝 부분을 표시하려면 각 노드에서 다른 창을 설정합니다.

```
tail -f /var/cluster/logs/DS/acsls-rg/acsls-rs/start_stop_log.txt
```

*acsls\_agt.sh* 시작 스크립트가 게시한 모든 메시지는 여기에 표시됩니다.

4. *acsls-rs* 프로브 로그의 끝 부분을 표시하려면 각 노드에서 세번째 창을 설정해야 합니다.

```
tail -f /var/cluster/logs/DS/acsls-rg/acsls-rs/probe_log.txt
```

응용 프로그램이 시작되면 Solaris Cluster는 1분마다 ACSLS 리소스를 프로브합니다. 각 프로브에서 숫자 코드가 클러스터로 반환되며 이 결과는 *probe\_log.txt* 파일에 인쇄됩니다. 각 프로브를 통해 표준 반환 값 5개가 이 로그에 게시됩니다.

```
0 - The probe found that ACSLS is healthy and functioning normally.
```

- 1 - The probe may not have completed due to a functional error.
- 2 - The probe reports that ACSLS is in a transitional state.
- 3 - The ACSLS application has been intentionally placed offline.
- 201 - A condition was detected that requires fail-over action.

이는 Solaris Cluster가 페일오버 작업을 시작하는 201 코드에 대한 응답일 뿐입니다. 해당 작업을 프롬프트하는 조건은 9장 [ACSLs 클러스터 작업](#) 장에 나와 있습니다. 클러스터 프로브의 모든 기타 반환 코드는 정보로 간주되며 클러스터 응답 작업은 명제화되지 않습니다.

테스트용 샘플 프로브는 명령줄에서 언제든지 명제화할 수 있습니다. `acsAgt probe` 명령을 사용합니다.

```
#!/opt/ACSLSHA/util/acsAgt probe
```

위에서 설명한 모든 로그는 Solaris Cluster에 표시되는 대로 시스템 보기를 반영합니다. `$ACS_HOME/log/` 디렉토리의 추가 로그 두 개는 ACSLS 응용 프로그램 레벨의 보기를 제공합니다. `acsss_event.log`는 응용 프로그램이 시작된 순간부터 ACSLS에서 발생하는 모든 중요 이벤트를 보고합니다. SMF에서 발생하는 모든 ACSLS 시작 문제는 `acsIs_start.log`에 기록됩니다.

## 클러스터 모니터링 유틸리티

Solaris Cluster 유틸리티는 `/usr/cluster/bin` 디렉토리에 있습니다.

- ACSLS 리소스 그룹의 현재 상태를 보려면 `clrg list -v`를 실행합니다.
- 두 개의 클러스터 노드에 대한 현재 상태를 보려면 `clrg status`를 실행합니다.
- 리소스 그룹의 상태를 보려면 `clrs status`를 실행합니다.
- 노드, 쿼럼 장치 및 클러스터 리소스에서 상세 정보 상태를 가져오려면 `cluster status`를 실행합니다.
- 클러스터 구성에서 자세한 구성 요소 목록을 보려면 `cluster show`를 실행합니다.
- 리소스 그룹의 각 이더넷 노드에 대한 상태를 보려면 `clnode status -m`를 실행합니다.
- 각 노드의 다양한 `acsIs-rg` 리소스 상태를 보려면 `scstat -g`를 실행합니다.
- 하트비트 네트워크 링크의 상태를 확인하려면 `clintr status`를 실행합니다.
- IPMP 상태를 확인하려면 `scstat -i`를 실행합니다.
- 노드 상태를 보려면 `scstat -n`를 실행합니다.
- 쿼럼 구성 및 상태를 보려면 `scstat -q` 또는 `clq status`를 실행합니다.
- 시간 초과 값을 포함해서 자세한 클러스터 리소스를 보려면 `clresource show -v`를 실행합니다.

## 복구 및 페일오버 테스트

이 절에서는 복구 및 페일오버 테스트를 위한 조건, 모니터링 및 테스트에 대해 설명합니다.

## 복구 조건

치명적인 시스템 조건 중에는 시스템 페일오버 이벤트 없이도 복구할 수 있는 경우가 많습니다. 예를 들어, IPMP의 경우 각 그룹의 이더넷 연결 하나가 어떠한 이유로든 실패할 수 있지만 대체 경로를 통해 무중단으로 통신이 재개됩니다.

공유 디스크 배열은 각 서버에서 두 개의 고유 포트를 사용해서 서버에 연결되어야 합니다. 하나의 경로가 중단될 경우 대체 경로를 통해 중단 없이 디스크 I/O 작업이 재개됩니다.

ACSLs는 Solaris SMF(서비스 관리 기능)로 모니터되는 여러 개의 소프트웨어 서비스로 구성됩니다. 사용자 `acsss`로 `acsss status` 명령을 사용해서 각 `acsss` 서비스를 나열할 수 있습니다. 이러한 서비스 중에는 PostgreSQL 데이터베이스, WebLogic 웹 응용 프로그램 서버 및 ACSLS 응용 프로그램 소프트웨어가 있습니다. 제공된 서비스가 Solaris 시스템에서 실패할 경우 SMF가 시스템 페일오버 없이 서비스를 자동으로 재부트합니다.

`acsls` 서비스 자체는 상위 `acsss_daemon`으로 모니터되는 여러 하위 프로세스로 구성됩니다. ACSLS 하위 프로세스를 나열하려면 `psacs` 명령을 사용합니다(사용자 `acsss`). 어떠한 이유로든 하위 프로세스 중 하나가 중단되면 상위 프로세스가 하위 프로세스를 즉시 재부트하고 정상 작업을 복구합니다.

## 복구 모니터링

디스크 I/O 및 이더넷 연결과 같은 시스템 리소스 복구를 보기 위한 최적의 위치는 시스템 로그 `/var/adm/messages`입니다.

SMF는 모니터하는 각 소프트웨어 서비스에 대한 특정 로그를 유지 관리합니다. 이 로그에는 시작, 다시 시작 및 종료 이벤트가 표시됩니다. 서비스 로그에 대한 전체 경로를 보려면 `svcs -l service-name` 명령을 실행합니다. ACSLS 서비스는 `acsss` 명령: `$ acsss status`를 사용해서 나열할 수 있습니다. 하위 프로세스는 `$ acsss p-status` 명령을 사용해서 나열할 수 있습니다.

ACSLs 하위 프로세스에 대한 복구를 보려면 `acsss_event.log($ACS_HOME/ACSSS/log/acsss_event.log)`를 모니터하면 됩니다. 이 로그에는 ACSLS 하위 프로세스를 포함하는 모든 복구 이벤트가 표시됩니다.

## 복구 테스트

중복 네트워크 연결은 Solaris IPMP 논리로 자동으로 다시 시작되어야 합니다. 공유 디스크 배열에 대해 중단된 데이터 연결은 중복된 데이터 경로에서 Solaris에 의해 자동으로 다시 시작되어야 합니다. Solaris 서비스 관리 기능으로 제어되는 서비스는 SMF에 의해 자동으로 다시 시작됩니다.

실제 페일오버 이벤트를 포함하는 테스트의 경우에는 `$ACS_HOME/acslsha/pingpong_interval` 파일에 정의된 등록 정보 설정을 확인합니다. 페일오버 이벤트를 트리거할 수 있는 조건에도 불구하고 Solaris Cluster는 이전 페일오버 이벤트가 지정된 `pingpong_interval` 내에 발생한 경우 페일오버 작업을 시작하지 않습니다.

`pingpong` 간격을 보거나 동적으로 변경하려면 `/opt/ACSLSHA/util` 디렉토리로 이동하고 `acsAgt pingpong`을 실행합니다.

```
./acsAgt pingpong
Pingpong_interval
 current value: 1200 seconds.
 desired value: [1200] 300
Pingpong_interval : 300 seconds.
```

다음 기술을 모두 사용하여 HA 설치의 복원성을 평가합니다.

1. ACSLS가 작동 중일 때 활성 노드의 각 IPMP 그룹에서 하나의 이더넷 연결을 분리합니다. # `scstat -i`를 사용해서 상태를 모니터합니다.

`/var/adm/messages`에서 반응을 확인합니다. ACSLS 작업은 이 절차로 중단되지 않습니다.

2. 클러스터 `Failover_mode`가 **HARD**로 설정되었는지 확인합니다. ACSLS가 작동 중일 때 활성 서버에서 공유 디스크 리소스에 대한 하나의 광 섬유 또는 SAS 연결을 분리합니다.

`/var/adm/messages`에서 반응을 확인합니다. ACSLS 작업은 이 절차로 중단되지 않습니다.

중복된 각 I/O 연결에서 이 테스트를 반복합니다.

3. `acsss_daemon`을 종료하여 ACSLS를 갑자기 종료합니다. `pkill acsss_daemon`을 사용합니다.

`svcs -l acsls`를 실행하여 서비스 로그를 찾습니다.

`acsss_daemon`이 중지되면 이 로그의 끝 부분을 확인합니다. SMF에 의해 서비스가 자동으로 다시 시작되었음을 확인합니다. `acsls shutdown`으로 `acsls`를 중지하면 비슷한 작업이 표시됩니다.

4. SMF를 사용하여 `acsls` 서비스를 사용 안함으로 설정합니다.

이 작업은 `svcadm disable acsls`를 사용해서 루트로 수행하거나 `acsss disable`을 사용해서 사용자 `acsss`로 수행할 수 있습니다.

SMF가 이러한 종료 이벤트를 담당하기 때문에 `acsls` 서비스를 다시 시작하려고 시도하지 않습니다. 이는 적절한 동작입니다. SMF 아래에서 `acsls` 서비스를 다시 시작해야 합니다. `root`로 `svcadm enable acsls` 명령을 사용합니다. 또는 사용자 `acsss`로 `acsss-enable` 명령을 사용합니다.

5. `acsdb` 서비스를 작동 중지합니다.

사용자 `acsdb`로 `.acsls_env` 파일을 소스로 지정합니다.

```
$ su acsdb
$. /var/tmp/acsls/.acsls_env
```

이제 다음 명령을 사용하여 PostgreSQL 데이터베이스를 갑자기 사용 안함으로 설정합니다.



```
pg_ctl stop /
-D $installDir/acbdb/ACSDb1.0/data /
-m immediate
```

이 작업을 수행하면 데이터베이스가 작동 중지되므로 *acs1s* 프로세스가 중지됩니다. *svcs -l acbdb*를 실행하여 *acbdb* 서비스 로그를 찾습니다.

데이터베이스를 작동 중지할 때 *acbdb* 서비스 로그 및 *acs1s* 서비스 로그의 끝 부분을 확인합니다. *acbdb* 서비스가 작동 중지될 때 *acs1s* 서비스도 작동 중지되는지 확인합니다. 두 서비스 모두 SMF에 의해 자동으로 다시 시작됩니다.

6. ACSLS가 작동 중일 때 사용자 *acsss*로 *psacs*를 실행하여 *acsss\_daemon* 아래에서 실행 중인 하위 프로세스 목록을 가져옵니다.

이러한 하위 프로세스 중 하나를 중지합니다. *acsss\_event.log*에서 하위 프로세스가 다시 시작되고 복구 절차가 호출되는지 확인합니다.

## 페일오버 조건

Solaris Cluster Software는 Solaris 시스템을 모니터링하면서 시스템 페일오버 이벤트가 필요한 치명적인 조건을 찾습니다. 이러한 조건들 중에는 사용자가 시작한 페일오버(*acsAgt nodeSwitch* 또는 *clrg switch -n*), 활성 노드에 대한 시스템 재부트, 시스템 작동 중단, 치명적인 메모리 결함 또는 활성 노드에서 복구할 수 없는 I/O 통신 등이 포함됩니다. Solaris Cluster는 특정 응용 프로그램에 대해 지정된 HA 에이전트도 모니터링합니다. ACSLS HA 에이전트는 다음과 같은 조건에서 시스템 페일오버 이벤트를 요청합니다.

- 활성 노드와 논리 호스트 사이에서 TCP/IP 통신이 손실된 경우
- *\$ACS\_HOME* 파일 시스템이 마운트되지 않은 경우
- 데이터베이스 백업 파일 시스템(*\$ACS\_HOME/.../backup*)은 마운트되지 않습니다.
- *\$ACS\_HOME/acslsha/ha\_acs\_list.txt* 파일에 지정된 ACS에 해당하는 라이브러리에 대한 통신이 끊긴 경우(이 상태가 온라인이 아니면 *switch lmu*가 불가능하거나 실패함)

## 페일오버 모니터링

일부 경우에는 # *clrg status* 명령을 사용해서 해당 노드의 페일오버 상태를 모니터링할 수 있습니다.

*start\_stop\_log*의 끝 부분을 확인하여 페일오버 작업을 모니터링할 수도 있습니다.

```
tail -f /var/cluster/logs/DS/acsls-rg/acsls-rs/start_stop_log.txt
```

진단 페일오버 작업을 수행할 때는 두 노드 모두에서 */var/adm/messages* 파일을 확인 (*tail -f*)하는 것이 유용할 수 있습니다. “[ACSLs 클러스터 작업 모니터링](#)”을 참조하십시오.

## 페일오버 테스트

1. 클러스터 페일오버 이벤트를 시작하는 간단한 명령은 `acsAgt nodeSwitch`입니다.

```
acsAgt nodeSwitch
```

또는 해당 클러스터 명령을 사용합니다.

```
clrg switch -n <node name> acsls_rg
```

이 작업으로 ACSL S 응용 프로그램이 작동 중지되고 활성 서버에서 대기 시스템으로 작업이 전환됩니다. `-M -e` 옵션은 새 노드에서 SMF 서비스를 사용으로 설정하도록 클러스터 서버에 지시합니다. "[ACSL S 클러스터 작업 모니터링](#)"을 참조하십시오.

2. 활성 노드에서 시스템 재부트를 실행하면 대체 노드로 HA 전환이 즉시 시작됩니다.

이 작업을 수행하면 새로운 활성 노드에서 ACSL S가 실행됩니다. 대기 노드에서 대기 시스템이 새로운 활성 노드 역할을 가장할 때 `/var/adm/messages` 파일의 끝 부분을 감시합니다. `# clrg status` 명령을 주기적으로 실행할 수도 있습니다.

3. `init 5`를 사용해서 활성 서버 노드의 전원을 차단하고 시스템 페일오버를 확인합니다.
4. 활성 서버 노드와 공유 디스크 스토리지 배열 사이의 데이터 라인을 모두 분리하고 대기 노드로의 시스템 전환을 확인합니다.
5. 지정된 라이브러리가 정책 파일 `ha_acs_list.txt`에 나열된다고 가정하고 활성 서버 노드와 해당 라이브러리 사이의 이더넷 통신 연결 라인을 모두 분리합니다.

대기 노드에 대한 시스템 페일오버를 확인합니다.

## 추가 테스트

미러링된 부트 드라이브가 핫 플러그할 수 있는 경우 부트 드라이브 중 하나를 사용 안함으로 설정하고 시스템이 완전히 작동되는 상태인지 확인합니다. 부트 드라이브 하나가 사용 안함으로 설정된 경우 시스템을 재부트해서 노드가 대체 부트 드라이브로부터 작동하는지 확인합니다. 2개의 각 노드에서 각 부트 드라이브에 대해 이 작업을 반복합니다.

활성 노드에서 단일 전원 공급 장치를 제거하면 시스템이 대체 전원 공급 장치를 사용해서 완전한 작동 상태를 유지해야 합니다.

## 12장. 문제 해결 도움말

ACSLs HA 8.4는 Solaris Cluster 4.2 제어에 따라 IPMP 및 ZFS를 포함하는 Solaris 11.2의 2 노드 시스템에서 작동하는 ACSLS 응용 프로그램의 통합입니다.

### ACSLs가 실행 중인지 확인

ACSLs 서비스가 활성 노드에서 실행되는지 확인하려면 사용자 `acsss`로 다음 명령을 사용합니다.

```
su - acsss
$ acsss status
```

하나 이상의 서비스가 사용 안함으로 설정되었으면 `$ acsss enable`을 사용해서 사용으로 설정합니다.

상태 표시에 하나 이상의 ACSLS 서비스가 유지 관리 모드인 것으로 표시되면 `$ acsss l-status` 명령을 실행합니다.

결함이 있는 서비스의 로그 파일에 대한 경로를 조사하고 이 로그에서 서비스가 유지 관리 모드로 설정된 이유를 설명할 수 있는 단서를 찾아봅니다.

하나 이상의 `acsls` 서비스가 유지 관리 모드이면 `acsss` 명령으로 사용 안함으로 설정한 후 다시 사용으로 설정하여 서비스를 지울 수 있습니다.

```
$ acsss shutdown
$ acsss enable
```

`root`로 `# svcadm clear <service name>`을 사용하여 개별 서비스를 지웁니다.

기본 결함이 해결되기 전까지는 서비스가 지워지지 않습니다.

문제의 원인을 확인하기 위해 특정 작업 로그를 검토할 수도 있습니다. 대부분은 `$ACS_HOME/log` 디렉토리에서 찾을 수 있습니다.

검토할 기본 로그는 `acsss_event.log`입니다. 이 로그에는 ACSLS의 전체 작업과 관련된 대부분의 이벤트가 기록됩니다.

문제가 ACSLS GUI 또는 논리적 라이브러리 작업과 관련이 있을 경우에는 `$ACS_HOME/log/sslm` 디렉토리에서 관련 로그를 찾을 수 있습니다.

ACSLs GUI 및 WebLogic의 경우에는 *AcsIsDomain.log*, *AdminServer.log* 및 *gui\_trace.logs*를 찾아보십시오.

WebLogic과 관련된 설치 문제는 *weblogic.log*에서 찾을 수 있습니다.

논리적 라이브러리 문제의 경우 논리적 라이브러리가 구성된 다음 *slim\_event.logs* 및 *smce\_stderr.log*를 참조합니다.

## 공유 디스크 리소스에 대한 연결 해결

1. *acsls-storage* 리소스가 활성 클러스터 노드에 대해 온라인인지 확인합니다.

```
clrs status acsls-storage
```

2. *acsls-storage* 리소스가 온라인이 아니면 활성 노드의 ZFS에 리소스가 마운트되었는지 확인합니다.

```
zpool status
```

*acslspool1*이 활성 노드에 마운트되지 않은 경우 대기 노드에 마운트되었는지 확인합니다.

```
ssh standby hostname zpool status
```

공유 디스크 리소스가 대기 노드에 마운트된 경우 클러스터 제어를 해당 노드로 전환합니다.

```
clrg switch -n standby hostname acsls-rg
```

3. *acslspool1*이 활성 노드에 마운트되지 않았고 *acsls-storage* 리소스가 오프라인 상태인 경우 *acslspool1*이 활성 노드에 표시되는지 확인합니다.

```
zpool import (no argument)
```

---

주:

이 작업은 *acsls-storage*가 오프라인인 경우에만 작동합니다. 이를 오프라인으로 전환하려면 *clrs disable acsls-storage* 명령을 사용합니다.

---

*acslspool1*이 활성 노드에 표시되는 경우 가져오기를 시도합니다.

```
zpool import -f acslspool
```

*import* 작업이 성공하면 *acsls-storage* 리소스를 Solaris Cluster에 대해 온라인 상태로 전환합니다.

```
clrs enable acsls-storage
```

*acslspool*이 활성 노드에 표시되지 않으면 공유 드라이브에 대한 물리적 연결을 해결할 필요가 있습니다.

## 논리 호스트에 ping을 수행할 수 없는 경우

1. 논리 호스트 이름이 Solaris Cluster에 등록되었는지 확인합니다.

```
clrslh list
```

2. 활성 노드를 확인합니다.

```
clrg status | grep -i Online
```

3. 활성 노드에 ping을 수행할 수 있는지 확인합니다.

```
ping <node name>
```

4. *logical-host* 이름 리소스가 활성 노드에 대해 온라인 상태인지 확인합니다.

```
clrslh status
```

논리 호스트가 온라인이 아니면 사용으로 설정합니다.

```
clrs enable <logical host>
```

5. 공용 그룹에 지정된 IP 인터페이스의 상태를 확인합니다.

```
ipadm
```

출력 표시에서 공용 *ipmp* 그룹의 각 멤버에 대해 *ok* 상태를 확인합니다.

6. 공용 그룹(*ipmp0*)의 각 인터페이스에 대해 해당 물리적 상태를 확인합니다.

```
dladm show-phys
```

7. 논리 호스트가 공용 *ipmp* 그룹에서 2개의 인터페이스 중 하나 또는 다른 하나에 연결되었는지 확인합니다(5단계에서 확인됨).

```
arp <logical-hostname>
```

```
ifconfig net0
```

```
ifconfig net4
```

이 예제에서는 net0 및 net4가 공용 *imp* 그룹에 지정되었다고 가정합니다.

두 개의 인터페이스 중 하나의 MAC 주소는 논리 호스트 이름에 지정된 MAC 주소와 일치해야 합니다.

## 노드 간 상호 연결 확인

두 노드 사이에 클러스터에 대한 통신 손실로 인해 클러스터 제어가 실패한다고 의심될 경우에는 다음과 같이 클러스터에 대한 개인 상호 연결을 확인합니다.

```
cluster status -t interconnect
```

---

# 색인

소프트웨어, 13  
스토리지 배열 옵션, 12

## 기호

ACSL 8.3 설치  
첫번째 노드, 37  
ACSL 8.4 설치  
인접한 노드, 38  
ACSL 클러스터 제어  
ACSL HA 전원 차단, 60  
단일 노드 클러스터 만들기, 61  
시작, 58  
일시 중지, 59  
일시 중지된 ACSL 클러스터 시스템 전원 켜기,  
60  
작업 및 유지 관리, 58  
Solaris Cluster 4.1  
scinstall 루틴, 42  
scinstall 실행, 44  
클러스터 구성 확인, 44  
클러스터 패키지 설치, 41  
Solaris Cluster 및 ACSL HA 구성  
root의 액세스 구성, 19  
공용 인터페이스 및 IPMP, 22  
다중 경로 네트워크, 20  
다중 경로 디스크, 25  
라이브러리 인터페이스, 25  
zfs 파일 시스템 구성  
ACSL에 대한 미러링된 파일 시스템 만들기, 29  
미러링된 루트 만들기, 27

## 人

소프트웨어 구성 요소  
ACSL HA 다시 설치 또는 업그레이드 설치, 65  
ACSL 업그레이드 설치, 65  
ACSL 패치 설치, 63  
ACSL 패키지 제거, 64  
Solaris Cluster 업그레이드, 67  
Solaris Cluster 제거, 67  
소프트웨어 패키지  
ACSL 8.4, 33  
ACSL HA, 35  
Oracle Cluster 4.1, 34  
시스템 요구 사항  
네트워크, 13  
서버 옵션, 12

## ㅈ

조정  
pingpong\_interval 설정, 54  
라이브러리 통신을 위한 페일오버 정책, 53  
전자 메일 알림 등록, 54  
중복 전자 부품, 54  
진단 및 테스트  
복구 및 페일오버, 70  
클러스터 모니터링 유틸리티, 70

