Red Hat Enterprise Linux 6

클러스터 관리

고가용성 추가 기능 설정 및 관리

Last Updated: 2017-10-12

Red Hat Enterprise Linux 6 클러스터 관리

고가용성 추가 기능 설정 및 관리

Red Hat 엔지니어링 컨텐츠 서비스
docs-need-a-fix@redhat.com

법적 공지

초록

고가용성 추가 기능 설정 및 관리에서는 **Red Hat Enterprise Linux 6**의 고가용성 추가 기능의 설정 및 관리에 대해 설명합니다.

차례

소개

이 문서는 Red Hat 고가용성 추가 기능 구성 요소의 설치, 설정 및 관리에 관한 내용을 제공합니다. Red Hat 고가용성 추가 기능 구성 요소는 컴퓨터 그룹(*노드* 또는 *멤버*라 부름)을 클러스터로 함께 작동하도록 연결할 수 있습니다. 이 문서에서 *클러스터 (cluster)* 또는 *클러스터군 (clusters)*은 Red Hat 고가용성 추가 기능을 실행하는 컴퓨터의 그룹을 가리키는데 사용됩니다.

이 문서는 Red Hat Enterprise Linux 에 대해 숙련된 작업 지식을 갖고 있고 클러스터, 스토리지, 서버 컴퓨팅에 대한 개념을 이해하고 있는 사용자를 위한 것입니다.

Red Hat Enterprise Linux 6에 관한 보다 자세한 내용은 다음 자료에서 참조하십시오:

- *Red Hat Enterprise Linux 설치 가이드*— Red Hat Enterprise Linux 6 설치 관련 내용을 다루고 있습니다.
- *Red Hat Enterprise Linux 운용 가이드*— Red Hat Enterprise Linux 6 활용, 설정, 관리 관련 내용을 다루고 있습니다.

Red Hat Enterprise Linux 6 용 고가용성 추가 기능 및 관련 제품에 대한 자세한 내용은 다음 자료에서 참조하십시오:

- *고가용성 개요*— Red Hat 고가용성 추가 기능의 높은 수준의 개요를 다루고 있습니다.
- *LVM (Logical Volume Manager) 관리*— 클러스터 환경에서 LVM을 실행하는 방법을 포함하여 LVM 에 대한 설명을 다루고 있습니다.
- *GFS 2 (Global File System 2): 설정 및 관리*— 장애 복구형 스토리지 추가 기능에 들어있는 Red Hat GFS 2 (Red Hat Global File System 2) 설치, 설정, 관리에 관한 내용을 다루고 있습니다.
- *DM Multipath* — Red Hat Enterprise Linux 6의 장치 맵퍼 멀티패스 (Device-Mapper Multipath) 기능 사용에 관한 내용을 다루고 있습니다.
- *로드 밸런서 (Load Balancer) 관리*— 실제 서버 그룹 전역에 걸쳐 IP 부하 균형 유지를 위해 LVS (Linux Virtual Servers)를 제공하는 통합된 소프트웨어 구성 요소 집합인 로드 밸런서 추가 기능을 통한 고성능 시스템 및 서비스 설정에 관한 내용을 다루고 있습니다.
- *릴리즈 노트*— Red Hat 제품의 최신 릴리즈에 관한 내용을 다루고 있습니다.

고가용성 추가 기능 문서 및 기타 다른 Red Hat 문서는 HTML, PDF, RPM 버전으로 Red Hat Enterprise Linux 문서 CD 및 http://docs.redhat.com/docs/en-US/index.html에서 보실 수 있습니다.

1. 피드백

문서 내용 개선을 위한 제안이 있거나 오자를 발견했을 경우 언제든지 알려 주시기 바랍니다. **doc-Cluster_Administration** 구성 요소에 대한 리포트를 버그질라(Bugzilla)에 제출해 주시면 됩니다. (http://bugzilla.redhat.com/bugzilla/)

문서의 식별자를 꼭 기입해 주시기 바랍니다.

```
Cluster_Administration(EN)-6 (2013-2-15T16:26)
```

문서 식별자를 기입하실 때 해당 가이드의 버전을 정확하게 알려주시기 바랍니다.

문서 자료 개선을 위한 제안이 있으시면, 최대한 상세하고 명확히 설명해 주시기 바랍니다. 오류를 발견하셨다면, 저희가 쉽게 식별할 수 있도록 섹션 번호와 주위의 문장들을 함께 보내주시기 바랍니다.

1장. RED HAT 고가용성 추가 기능 설정 및 관리 개요

Red Hat 고가용성 추가 기능을 사용하면 컴퓨터의 집합 (*노드* 또는 *멤버*라고 부름)을 연결하여 함께 클러스터로 작동하게 할 수 있습니다. 사용자의 클러스터 구성 요구 사항에 적합하도록 Red Hat 고가용성 추가 기능을 사용할 수 있습니다. (예를 들어 GFS2 파일 시스템에서 파일을 공유하거나 서비스 페일오버를 설정하기 위해 클러스터를 구성할 수 있습니다.)

참고

고가용성 추가 기능 및 Red Hat Global File System 2 (GFS2)를 사용하여 Red Hat Enterprise Linux 클러스터를 배포 및 업그레이드하기 위한 성공 사례에 대한 자세한 내용은 . https://access.redhat.com/kb/docs/DOC-40821의 Red Hat 고객 포털에 있는 "Red Hat Enterprise Linux 클러스터, 고가용성, GFS 배포 성공 사례" 문서를 참조하십시오.

다음 부분에서는 Red Hat Enterprise Linux 6 초기 릴리즈 이후 Red Hat 고가용성 추가 기능에 추가된 기능 및 업데이트에 대해 요약된 내용을 제공합니다. 그 다음 Red Hat 고가용성 추가 기능 설정 및 관리에 대해 설명합니다.

1.1. 새로운 기능 및 변경된 기능

다음 부분에서는 Red Hat Enterprise Linux 6 최초 릴리즈 이후 추가된 Red Hat 고가용성 추가 기능의 새로운 기능 및 변경된 기능에 대해 설명합니다.

1.1.1. Red Hat Enterprise Linux 6.1에서 새로운 기능 및 변경된 기능

Red Hat Enterprise Linux 6.1에는 다음과 같은 문서와 기능 업데이트 및 변경 사항이 포함되어 있습니다.

- Red Hat Enterprise Linux 6.1 릴리즈와 그 이후 버전에서, Red Hat 고가용성 추가 기능은 SNMP 트랩에 대한 지원을 제공합니다. Red Hat 고가용성 추가 기능으로 SNMP 트랩을 설정하는 방법은 10장. *Red Hat 고가용성 추가 기능을 사용하여 SNMP 설정*에서 참조하십시오.
- Red Hat Enterprise Linux 6.1 릴리즈와 그 이후 버전에서 Red Hat 고가용성 추가 기능은 **ccs** 클러스터 설정 명령에 대한 지원을 제공합니다. **ccs** 명령에 대한 자세한 내용은 5장. ***ccs** 명령으로 Red Hat 고가용성 추가 기능 설정* 및 6장. ***ccs**로 Red Hat 고가용성 추가 기능 관리*에서 참조하십시오.
- Conga를 사용하여 Red Hat 고가용성 추가 기능 소프트웨어를 설정 및 관리에 대한 문서는 업데이트되어 업데이트된 Conga 화면 및 기능 지원이 반영되어 있습니다.
- Red Hat Enterprise Linux 6.1 릴리즈와 그 이후 버전에서 `ricci` 를 사용하는 경우 특정 노드에서 업데이트된 클러스터 설정을 처음 전달할 때 암호가 필요합니다. `ricci`에 대한 자세한 내용은 2.13절. “`ricci` 사용 시 고려 사항 ” 에서 참조하십시오.
- 서비스의 *Restart-Disable* 실패 정책을 지정할 수 있습니다. 이는 시스템이 실패할 경우 서비스를 다시 시작해야 하지만 서비스를 다시 시작해도 실패하면 클러스터에 있는 다른 노드로 이동하지 않고 서비스를 비활성화하게 됨을 나타냅니다. 이러한 기능은 3.10절. “클러스터에 클러스터 서비스 추가 ” 및 부록 B. *HA 리소스 매개 변수*에서 설명하고 있습니다.
- 리소스가 실패하면 그 리소스만 비활성화됨을 나타내는 독립적 하위 트리를 중요하지 않음 (non-critical)으로 설정할 있습니다. 이러한 기능에 대한 자세한 내용은 3.10절. “클러스터에 클러스터 서비스 추가 ” 및 C.4절. “장애 복구 및 독립적 하위 트리 ” 에서 참조하십시오.
- 다음 부분에는 새로운 장으로 9장. *클러스터에 있는 문제를 진단 및 수정*이 포함되어 있습니다.

또한 문서 전체에 걸쳐 일부 내용을 수정 및 명료화하였습니다.

1.1.2. Red Hat Enterprise Linux 6.2에서 새로운 기능 및 변경된 기능

Red Hat Enterprise Linux 6.2에는 다음과 같은 문서와 기능 업데이트 및 변경 사항이 포함되어 있습니다.

- 현재 Red Hat Enterprise Linux에서는 active/active 설정으로 클러스터 Samba를 실행하기 위한 지원을 제공합니다. 클러스터 Samba 설정에 대한 보다 자세한 내용은 11장. *클러스터 Samba 설정* 에서 참조하십시오.

- **luci**를 호스팅하는 시스템에서 인증할 수 있는 모든 사용자는 **luci**에 로그인할 수 있습니다. 하지만 Red Hat Enterprise Linux 6.2 이상에서는 관리자 (root 사용자 또는 관리 권한이 있는 사용자)가 사용자에게 권한을 설정하기 전 까지 **luci**를 실행하고 있는 시스템에서의 root 사용자만이 **luci** 구성요소에 액세스할 수 있습니다. 사용자에게 **luci** 권한을 설정하는 방법에 대한 자세한 내용은 3.3절. "luci로의 액세스 제어 "에서 참조하십시오.

- 클러스터에 있는 노드는 UDP 유니캐스트 전송 메커니즘을 사용하여 노드 간에 통신할 수 있습니다. UDP 유니캐스트를 설정하는 방법에 대한 자세한 내용은 2.12절. "UDP 유니캐스트 트래픽 "에서 참조하십시오.

- **/etc/sysconfig/luci** 파일을 사용하여 **luci**의 일부 동작을 설정할 수 있습니다. 예를 들어 **luci**가 작동하는 IP 주소만을 특별하게 설정할 수 있습니다. **luci**가 작동하는 IP 주소만을 설정하는 방법에 대한 자세한 내용은 표 2.2. "**luci**를 실행하는 컴퓨터에서 활성화된 IP 포트 " 에서 참조하십시오. **/etc/sysconfig/luci** 파일에 대한 일반적인 내용은 2.4절. "**/etc/sysconfig/luci**로 **luci** 설정"에서 참조하십시오.

- **ccs** 명령에는 **--lsfenceopts** 옵션이 포함되어 있어 사용 가능한 차단 장치 목록을 출력하며 **--lsfenceopts** *fence_type* 옵션은 사용 가능한 차단 유형을 출력합니다. 이러한 옵션에 대한 보다 자세한 내용은 5.6절. "차단 장치 및 차단 장치 옵션 목록 " 에서 참조하십시오.

- **ccs** 명령에는 **--lsserviceopts** 옵션이 포함되어 있으며 이 옵션은 클러스터에서 현재 사용 가능한 클러스터 서비스 목록을 출력합니다. **--lsserviceopts** *service_type* 옵션은 특정 서비스 유형에 지정할 수 있는 옵션 목록을 출력합니다. 이러한 옵션에 대한 자세한 내용은 5.11절. "사용 가능한 클러스터 서비스 목록 나열 "에서 참조하십시오.

- Red Hat Enterprise Linux 6.2 릴리즈에서는 VMware (SOAP 인터페이스) 차단 에이전트에 대한 지원을 제공합니다. 차단 장치 매개 변수에 대한 보다 자세한 내용은 부록 A. *차단 장치 매개 변수* 에서 참조하십시오.

- Red Hat Enterprise Linux 6.2 릴리즈에서는 RHEV 3.0 이상에 대해 RHEV-M REST API 차단 에이전트에 대한 지원을 제공합니다. 차단 장치 매개 변수에 대한 보다 자세한 내용은 부록 A. *차단 장치 매개 변수* 에서 참조하십시오.

- Red Hat Enterprise Linux 6.2 릴리즈 이후 **ccs** 명령을 사용하여 클러스터에서 가상 머신을 설정할 때 **--addvm** 옵션을 (**addservice** 옵션이 아니라) 사용할 수 있습니다. 이렇게 하면 클러스터 설정 파일에 있는 **rm** 설정 노드로 **vm** 리소스를 직접 정의하게 됩니다. **ccs** 명령을 사용하여 가상 머신 리소스를 설정하는 방법에 대한 자세한 내용은 5.12절. "가상 머신 리소스 "에서 참조하십시오.

- 이 문서에는 새로운 부록 부록 D. *클러스터 서비스 리소스 검사 및 페일 오버 시간 제한*이 포함되어 있습니다. 이 부록에서는 **rgmanager**가 클러스터 리소스 상태를 모니터링하는 방법과 상태 확인 간격을 수정하는 방법을 설명하고 있습니다. 또한 작업 시간 초과로 인해 서비스가 실패하는 것을 보여주는 **__enforce_timeouts** 서비스 매개 변수에 대해 설명합니다.

- 이 문서에는 새로운 2.3.3절. "클러스터 구성 요소를 허용하기 위해 iptables 방화벽 설정" 섹션이 포함되어 있습니다. 이 섹션에서는 다양한 클러스터 구성 요소에 대해 **iptables** 방화벽을 통해 멀티캐스트 트래픽을 허용하기 위해 사용할 수 있는 필터링에 대해 설명합니다.

또한 문서 전체에 걸쳐 일부 내용을 수정 및 명료화하였습니다.

1.1.3. Red Hat Enterprise Linux 6.3에서 새로운 기능 및 변경된 기능

Red Hat Enterprise Linux 6.3에는 다음과 같은 문서와 기능 업데이트 및 변경 사항이 포함되어 있습니다.

- Red Hat Enterprise Linux 6.3 릴리스에서는 **condor** 리소스 에이전트에 대한 지원을 제공합니다. HA 리소스 매개 변수에 대한 자세한 내용은 부록 B. *HA 리소스 매개 변수*에서 참조하십시오.
- 이 문서에는 새로운 부록 부록 F. *고가용성 LVM (HA-LVM)*이 포함되어 있습니다.
- 이 문서의 내용을 통해 클러스터를 다시 시작할 때 필요한 설정 변경 사항에 대해 명확하게 확인할 수 있습니다. 이러한 변경 사항에 대한 요약 내용은 9.1절. "설정 변경 사항은 적용되지 않음"에서 참조하십시오.
- 이 문서에는 **luci**에서 15 분 동안 작업을 수행하지 않으면 로그 아웃되는 유휴 시간 제한에 대해 설명하고 있습니다. **luci** 시작에 대한 내용은 3.2절. "**luci** 시작 "에서 참조하십시오.
- **fence_ipmilan** 차단 장치는 권한 수준 매개 변수를 지원합니다. 차단 장치 매개 변수에 대한 자세한 내용은 부록 A. *차단 장치 매개 변수*에서 참조하십시오.
- 이 문서에는 새로운 섹션으로 2.14절. "클러스터 환경에서 가상 머신 설정 " 이 포함되어 있습니다.
- 이 문서에는 새로운 섹션으로 4.6절. "luci 설정 백업 및 복구"가 포함되어 있습니다.
- 이 문서에는 새로운 섹션으로 9.4절. "클러스터 데몬 충돌"이 포함되어 있습니다.
- 이 문서에서는 5.14.4절. "로깅 ", 7.7절. "디버그 옵션 설정 ", 9.13절. "DLM (Distributed Lock Manager) 용 디버그 로깅은 활성화되어 있어야 함"에 디버깅 옵션을 설정하는 방법에 대해 설명하고 있습니다.
- Red Hat Enterprise Linux 6.3 이후 root 사용자 또는 **luci** 관리자 권한이 부여된 사용자는 **luci** 인터페이스를 사용하여 사용자를 시스템에 추가할 수 있습니다. 자세한 내용은 3.3절. "luci로의 액세스 제어 "에서 설명하고 있습니다.
- Red Hat Enterprise Linux 6.3 릴리즈에서 **ccs** 명령은 **-h** 옵션으로 지정한 노드에서 **/usr/share/cluster/cluster.rng**에 있는 클러스터 스키마에 따라 설정을 확인합니다. 이전에는 **ccs** 명령은 **ccs** 명령 자체와 패키지화된 로컬 시스템의 **/usr/share/ccs/cluster.rng**에 있는 클러스터 스키마를 항상 사용했었습니다. 설정 검증에 대한 자세한 내용은 5.1.6절. "설정 유효성 검사 "에서 참조하십시오.
- 부록 A. *차단 장치 매개 변수*에 있는 차단 장치 매개 변수를 설명하는 표 및 부록 B. *HA 리소스 매개 변수* 에 있는 HA 리소스 매개 변수를 설명하는 표에는 **cluster.conf** 파일에 표시된 대로 매개 변수의 이름도 포함되어 있습니다.

또한 문서 전체에 걸쳐 일부 내용을 수정 및 명료화하였습니다.

1.1.4. Red Hat Enterprise Linux 6.4에서 새로운 기능 및 변경된 기능

Red Hat Enterprise Linux 6.4에는 다음과 같은 문서와 기능 업데이트 및 변경 사항이 포함되어 있습니다.

- Red Hat Enterprise Linux 6.4 릴리즈에서는 Eaton Network Power Controller (SNMP 인터페이

스) 차단 에이전트, HP BladeSystem 차단 에이전트, IBM iPDU 차단 에이전트에 대한 지원을 제공합니다. 차단 장치 매개 변수에 대한 자세한 내용은 부록 A. *차단 장치 매개 변수*에서 참조하십시오.

- 부록 B. *HA 리소스 매개 변수*에서는 NFS 서버 리소스 에이전트에 대한 설명을 제공합니다.
- Red Hat Enterprise Linux 6.4에서는 root 사용자 또는 luci 관리 권한이 부여된 사용자가 luci 인터페이스를 사용하여 시스템에서 사용자를 제거할 수 있습니다. 이에 대해서는 3.3절. "luci로의 액세스 제어 "에 설명되어 있습니다.
- 부록 B. *HA 리소스 매개 변수*에서는 파일 시스템 및 GFS2 HA 리소스의 새로운 `nfsrestart` 매개 변수에 대해 설명합니다.
- 이 문서에는 새로운 5.1.5절. "이전 설정을 덮어쓰기하는 명령 " 섹션이 포함되어 있습니다.
- 2.3절. "IP 포트 사용 "에는 `igmp`의 `iptables` 방화벽을 필터링하는 내용이 포함되어 있습니다.
- IPMI LAN 차단 에이전트는 부록 A. *차단 장치 매개 변수*에 설명되어 있는것 처럼 IPMI 장치의 권한 수준을 설정하기 위한 매개 변수를 지원합니다.
- 클러스터에 있는 노드간 통신하기 위해 이더넷 본딩 모드 1, 본딩 모드 0 및 2에 대한 지원이 추가되었습니다. 지원되는 본딩 모드만을 사용할 것을 제안하는 이 문서의 문제 해결 조언은 이에 대해 기재하고 있습니다.
- VLAN 태그 네트워크 장치는 클러스터 하트 비트 통신에 대해 지원합니다. 이러한 기능이 지원되지 않음을 나타내는 문제 해결 조언은 이 문서에서 삭제되어 있습니다.
- Red Hat 고가용성 애드온은 중복 링 프로토콜의 설정을 지원합니다. 이러한 기능의 사용 및 `cluster.conf` 설정 파일 설정에 대한 일반적인 내용은 7.6절. "중복 링 프로토콜 설정 "에서 참조하십시오. luci로 중복 링 프로토콜을 설정하는 방법은 3.5.4절. "중복 링 프로토콜 설정 "에서 참조하십시오. ccs로 중복 링 프로토콜을 설정하는 방법은 5.14.5절. "중복 링 프로토콜 설정 "에서 참조하십시오.

또한 문서 전체에 걸쳐 일부 내용을 수정 및 명료화하였습니다.

1.2. 설정 기본

클러스터를 설정하려면, 노드를 특정 클러스터 하드웨어에 연결하여 노드를 클러스터 환경으로 설정합니다. Red Hat 고가용성 추가기능 설정 및 관리는 다음과 같은 기본 단계로 구성됩니다:

1. 하드웨어 설정. 1.3절. "하드웨어 설정 "에서 참조하십시오.
2. Red Hat 고가용성 추가 기능 소프트웨어 설치. 1.4절. "Red Hat 고가용성 추가 기능 소프트웨어 설치 "에서 참조하십시오.
3. Red Hat 고가용성 추가 기능 소프트웨어 설정. 1.5절. "Red Hat 고가용성 추가 기능 소프트웨어 설정 "에서 참조하십시오.

1.3. 하드웨어 설정

하드웨어 설정은 Red Hat 고가용성 추가 기능을 실행하기 위해 필요한 다른 하드웨어에 클러스터 노드를 연결하는 작업으로 이루어집니다. 하드웨어 수량과 종류는 클러스터의 목적과 사용 요구에 따라 달라집니다. 일반적으로 엔터프라이즈급 클러스터는 다음의 하드웨어 유형을 필요로 합니다. (그림 1.1. "Red Hat 고가용성 추가 기능 하드웨어 개요 " 참조) 하드웨어 및 다른 클러스터 설정에 관한 내용은 2장. *Red Hat 고가용성 추가 기능 설정 이전 작업* 에서 참조하거나 또는 Red Hat 담당자에게 확인하십시오.

- 클러스터 노드 – 최소 RAM 1GB로 Red Hat Enterprise Linux 6 소프트웨어를 실행할 수 있는 컴퓨터
- 공공 네트워크의 이더넷 스위치 또는 허브 – 이는 클라이언트가 클러스터에 액세스하는 데 필요합니다.
- 개인 네트워크의 이더넷 스위치 또는 허브 – 이는 네크워트 전원 스위치 또는 파이버 채널 스위치와 같은 클러스터 하드웨어 및 클러스터 노드간의 통신에 필요합니다.
- 네트워크 전원 스위치 – 엔터프라이즈급 클러스터에서 펜싱 (fencing)을 실행하기 위해 네트워크 전원 스위치를 권장합니다.
- 파이버 채널 스위치 – 파이버 채널 스위치는 파이버 채널 스토리지로의 액세스를 제공합니다. 기타 옵션은 iSCSI와 같은 스토리지 인터페이스의 유형에 따라 스토리지에 대해 사용할 수 있습니다. 파이버 채널 스위치는 펜싱 (fencing)을 실행하기 위해 설정할 수 있습니다.
- 스토리지 – 일부 유형의 스토리지가 클러스터에 필요합니다. 필요한 유형은 클러스터 용도에 따라 다릅니다.

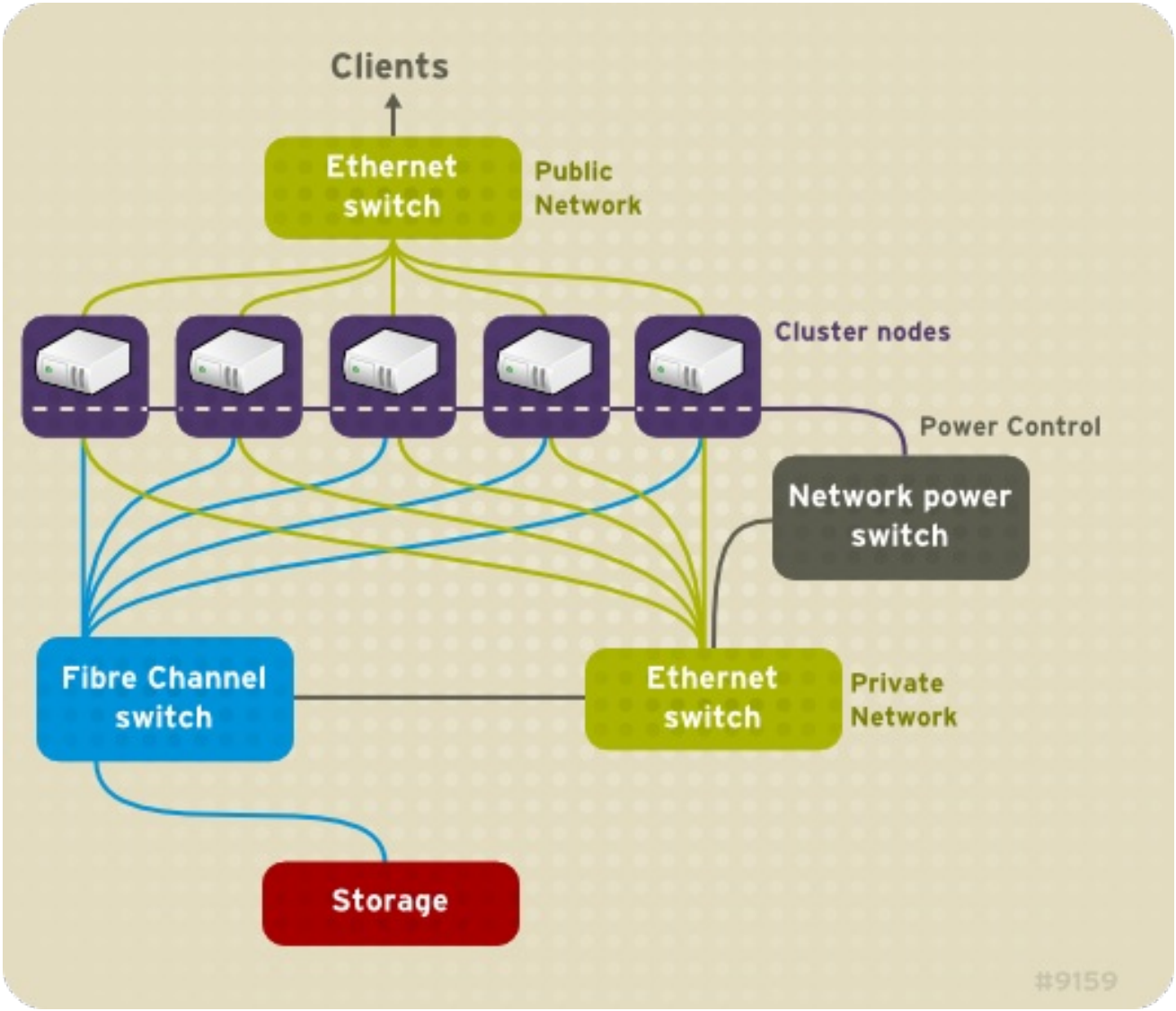


그림 1.1. Red Hat 고가용성 추가 기능 하드웨어 개요

1.4. RED HAT 고가용성 추가 기능 소프트웨어 설치

Red Hat 고가용성 애드온 소프트웨어를 설치하려면, 소프트웨어의 인타이틀먼트가 있어야 합니다. **luci**

설정 GUI를 사용하고 있을 경우 GUI로 클러스터 소프트웨어를 설치합니다. 클러스터를 설정하기 위해 다른 도구를 사용하고 있을 경우 Red Hat Enterprise Linux 소프트웨어를 설치하는 것처럼 안전하게 소프트웨어를 설치합니다.

다음과 같은 **yum install** 명령을 사용하여 Red Hat 고가용성 애드온 소프트웨어 패키지를 설치할 수 있습니다:

```
# yum install rgmanager lvm2-cluster gfs2-utils
```

rgmanager만 설치해도 HA 클러스터를 생성하기 위해 필요한 모든 의존 패키지를 고가용성 채널에서 불러오게 됩니다. **lvm2-cluster** 및 **gfs2-utils** 패키지는 ResilientStorage 채널의 일부분으로 귀하의 사이트에 필요하지 않을 수 도 있습니다.

1.4.1. Red Hat 고가용성 추가 기능 소프트웨어 업그레이드

클러스터를 프로덕션에서 분리하지 않고 Red Hat Enterprise Linux의 모든 주요 릴리즈에 있는 클러스터 소프트웨어를 업그레이드할 수 있습니다. 이를 위해 한번에 하나의 호스트에 있는 클러스터 소프트웨어를 비활성화하고 소프트웨어를 업그레이드한 후 그 호스트의 클러스터 소프트웨어를 다시 시작해야 합니다.

1. 단일 클러스터 노드에서 모든 클러스터 서비스를 종료합니다. 노드에서 클러스터 소프트웨어를 중지하는 방법에 대한 설명은 8.1.2절. “클러스터 소프트웨어 중지 ”에서 참조하십시오. **rgmanager**를 중지하기 전에 클러스터 관리 서비스 및 가상 시스템을 수동으로 호스트에서 이동하는 것이 좋습니다.

2. **yum update** 명령을 실행하여 설치된 패키지를 업데이트합니다.

3. 수동으로 클러스터 노드나 클러스터 서비스를 다시 시작합니다. 노드에서 클러스터 소프트웨어를 시작하는 방법은 8.1.1절. “클러스터 소프트웨어 시작 ”에서 참조하십시오.

1.5. RED HAT 고가용성 추가 기능 소프트웨어 설정

Red Hat 고가용성 추가 기능 소프트웨어 설정은 클러스터 구성 요소 간의 관계를 지정하기 위한 설정 도구의 사용으로 구성되어 있습니다. 다음의 클러스터 설정 도구는 Red Hat 고가용성 추가 기능과 함께 사용할 수 있습니다:

- **Conga** – Red Hat 고가용성 추가 기능을 설치, 설정, 관리하기 위한 종합적인 사용자 인터페이스입니다. **Conga**를 사용하여 고가용성 추가 기능을 설정 및 관리하는 방법에 대한 자세한 내용은 3장. *Conga*를 사용하여 *Red Hat* 고가용성 추가 기능 설정 및 4장. *Conga*를 사용하여 *Red Hat* 고가용성 추가 기능 관리에서 참조하십시오.

- **ccs** 명령 – 이 명령은 Red Hat 고가용성 추가 기능을 설정 및 관리합니다. **ccs** 명령으로 고가용성 추가 기능을 설정 및 관리하는 방법에 관한 내용은 5장. *ccs* 명령으로 *Red Hat* 고가용성 추가 기능 설정 및 6장. *ccs*로 *Red Hat* 고가용성 추가 기능 관리에서 참조하십시오.

- 명령행 도구 – 이는 Red Hat 고가용성 추가 기능의 설정 및 관리를 위한 명령행 도구 모음입니다. 명령행 도구를 사용하여 클러스터를 설정 및 관리하는 방법에 대한 자세한 내용은 7장. *명령행 도구로 Red Hat 고가용성 추가 기능 설정* 및 8장. *명령행 도구로 Red Hat 고가용성 추가 기능 관리*에서 참조하십시오. 우선하는 명령행 도구에 대한 요약은 부록 E. *명령행 도구 요약*에서 참조하십시오.

참고

system-config-cluster는 Red Hat Enterprise Linux 6에서 사용 가능하지 않습니다.

2장. RED HAT 고가용성 추가 기능 설정 이전 작업

다음 부분에서는 Red Hat 고가용성 추가 기능을 설치 및 구성하기 전 실행 및 고려해야할 작업에 대해 설명합니다. 이는 다음과 같은 부분으로 구성되어 있습니다.

중요

Red Hat 고가용성 추가 기능의 도입이 요구에 부합되며 지원되는지 확인합니다. Red Hat 담당자에게 문의하여 배포 전 설정을 확인하십시오. 또한 장애 모드를 테스트하기 위해 설정 번인 (burn-in) 기간을 두도록 합니다.

2.1. 일반적인 설정 고려 사항

Red Hat 고가용성 추가 기능을 사용자의 요구에 맞게 다양한 방식으로 설정할 수 있습니다. 계획, 설정, 운용 구현 시 다음과 같은 일반적 사항을 고려하십시오.

지원되는 클러스터 노드 수

고가용성 추가 기능에서 지원되는 클러스터 노드의 최대 개수는 16개입니다.

단일 사이트 클러스터

이번에는 단일 사이트 클러스터에만 완전하게 지원됩니다. 여러 물리적 위치에 퍼져있는 클러스터는 공식적으로 지원되지 않습니다. 여러 사이트 클러스터에 대한 설명과 자세한 내용은 Red Hat 영업 또는 지원 담당자에게 문의하시기 바랍니다.

GFS2

GFS2 파일 시스템은 독립형 시스템이나 클러스터 구성의 일부로 구현될 수 있지만, Red Hat은 단일 노

드 파일 시스템으로 GFS2 사용을 지원하지 않습니다. Red Hat은 단일 노드에 최적화되어 있는 고성능 단일 노드 파일 시스템의 대부분을 지원하므로 일반적으로 클러스터 파일 시스템 보다 낮은 오버헤드를 갖습니다. Red Hat은 파일 시스템을 마운트하기 위해 단일 노드만이 필요한 경우 GFS2 보다 이러한 파일 시스템의 사용을 권장합니다. Red Hat은 기존 고객에 대해 단일 노드 GFS2 파일 시스템을 계속 지원합니다.

GFS2 파일 시스템을 클러스터 파일 시스템으로 설정할 경우 클러스터의 모든 노드가 공유 파일 시스템에 액세스할 수 있는지 확인하십시오. 일부 노드가 파일 시스템에 액세스되어 있고 다른 노드는 액세스 권한이 없는 비대칭 클러스터 구성은 지원되지 않습니다. 이는 모든 노드가 실제로 GFS2 파일 시스템 자체를 마운트하도록 요청하지 않습니다.

단일 지점 장애가 없는 하드웨어 설정

클러스터는 애플리케이션 다운 타임에서의 단순 장애나 데이터 손실이 발생하지 않는다는 것을 확인하기 위해 듀얼 컨트롤러 RAID 어레이, 다중 연결 네트워크 채널, 클러스터 멤버와 스토리지 간의 다중 경로 및 이중 UPS (un-interruptible power supply) 시스템을 포함시킬 수 있습니다.

다른 방법으로 단일 지점 장애가 없는 클러스터 보다 낮은 가용성을 제공하는 낮은 비용의 클러스터를 설정할 수 있습니다. 예를 들어, 단일 컨트롤러 RAID 어레이와 단일 이더넷 채널만을 갖는 클러스터를 설치할 수 있습니다.

호스트 RAID 컨트롤러, 클러스터가 지원하지 않는 소프트웨어 RAID 및 멀티 초기 병렬 SCSI 설정 같은 일부 저가 대체 설정은 공유되는 클러스터 스토리지로 사용하기에 적합 또는 호환되지 않습니다.

데이터 무결성 보장

데이터 무결성을 보장하기 위해, 한번에 한 개의 노드만 클러스터 서비스를 실행하여 클러스터 서비스 데이터에 액세스할 수 있습니다. 클러스터 하드웨어 구성에서 전원 스위치의 사용은 장애 복구 프로세스 동안 한 개의 노드가 해당 노드의 HA 서비스를 다시 시작하기 전에 다른 노드로 전원을 껐다가 전력을 양도할 수 있도록 합니다. 따라서 두 개의 노드가 동시에 같은 데이터에 액세스하지 못하게 하고 손상을 방지합니다. 모든 장애 상황에서 데이터 무결성 보장을 위해, *차단 장치 (fence devices)*(원격 조작으로 클러스터 노드 집합의 전력 공급, 종료 및 다시 시작하는 하드웨어 및 소프트웨어 솔루션) 사용이 권장됩니다.

이더넷 채널 본딩

클러스터 쿼럼 (quorum)과 노드 상태는 이더넷을 통한 클러스터 노드 간 메세지의 통신에 의해 결정됩니다. 또한, 클러스터 노드는 여러 기타 다른 중요한 클러스터 기능 (예: 펜싱)에 대해 이더넷을 사용합니다. 멀티 이더넷 인터페이스는 이더넷 채널 본딩과 함께 하나로 동작하도록 설정되어, 클러스터 노드와 다른 클러스터 하드웨어 간의 전형적인 이더넷 스위치 연결에서 단일 지점 장애 현상의 위험을 줄일 수 있습니다.

Red Hat Enterprise Linux 6.4에서 본딩 모드 0, 1, 2가 지원됩니다.

IPv4 및 IPv6

고가용성 추가 기능은 IPv4 및 IPv6 인터넷 프로토콜을 지원합니다. 고가용성 추가 기능에서 IPv6 지원은 Red Hat Enterprise Linux 6에서 새로운 사항입니다.

2.2. 호환 가능 하드웨어

Red Hat 고가용성 추가 기능 소프트웨어를 설정하기 전에 사용하는 클러스터가 적절한 하드웨어 (예 : 지원되는 차단 장치, 저장 장치 및 파이버 채널 스위치)를 사용하고 있는지 확인합니다. 최신 하드웨어 호환성 정보는 http://www.redhat.com/cluster_suite/hardware/의 하드웨어 설정 지침을 참조하십시오.

2.3. IP 포트 사용

Red Hat 고가용성 추가 기능을 도입하기 전에 클러스터 노드에서와 **luci** (**Conga** 사용자 인터페이스 서버) 를 실행하는 컴퓨터에서 특정 IP 포트를 설정해야합니다. 다음 부분에서 설정해야 할 IP 포트를 확인합니다:

- 2.3.1절. "클러스터 노드에서 IP 포트 사용 "
- 2.3.2절. "**luci**의 IP 포트를 사용 "

다음 부분에서는 Red Hat 고가용성 애드온에 필요한 IP 포트를 활성화하기 위해 **iptables** 규칙을 제공합니다:

- 2.3.3절. "클러스터 구성 요소를 허용하기 위해 iptables 방화벽 설정"

2.3.1. 클러스터 노드에서 IP 포트 사용

클러스터의 노드간에 통신할 수 있도록 특정 Red Hat 고가용성 애드온 구성 요소에 할당된 IP 포트를 활성화해야 합니다. 표 2.1. "Red Hat 고가용성 추가 기능 노드에서 IP 포트 사용 " 에는 IP 포트 번호, 해당 프로토콜, 포트 번호가 할당된 구성 요소가 나열되어 있습니다. 각 클러스터 노드에서 표 2.1. "Red Hat 고가용성 추가 기능 노드에서 IP 포트 사용 "에 따라 IP 포트를 활성화합니다. **system-config-firewall**을 사용하여 IP 포트를 활성화할 수 있습니다.

표 2.1. Red Hat 고가용성 추가 기능 노드에서 IP 포트 사용

IP 포트 번호	프로토콜	구성 요소
5404, 5405	UDP	**corosync/cman** (클러스터 관리자)
11111	TCP	**ricci** (업데이트된 클러스터 정보를 전달)
21064	TCP	**dlm** (분산형 잠금 관리자)
16851	TCP	**modclusterd**

2.3.2. luci의 IP 포트를 사용

클라이언트 컴퓨터가 **luci** (**Conga** 사용자 인터페이스 서버)를 실행하는 컴퓨터와 통신할 수 있게 하려면 **luci**에 할당된 IP 포트를 사용해야 합니다. **luci**를 실행하는 각 컴퓨터에서 표 2.2. "**luci**를 실행하는 컴퓨터에서 활성화된 IP 포트 "에 따라 IP 포트를 활성화합니다.

참고

클러스터 노드가 **luci**를 실행하는 경우, 11111 포트는 이미 활성화되어 있습니다.

표 2.2. luci를 실행하는 컴퓨터에서 활성화된 IP 포트

IP 포트 번호	프로토콜	구성 요소
8084	TCP	**luci** (**Conga** 사용자 인터페이스 서버)

/etc/sysconfig/luci 파일을 사용하여 설정을 활성화하는 Red Hat Enterprise Linux 6.1 릴리즈 이후 **luci**가 작동하는 IP 주소만 특정하게 설정할 수 있습니다. 사용하는 서버 인프라에 여러 네트워크가 내장

되어 있고 내부 네트워크에서만 **luci**에 액세스하고자 할 경우 이 기능을 사용할 수 있습니다. 이를 실행하려면 파일에서 **host**를 지정하는 행을 주석 해제하여 편집합니다. 예를 들어, 파일에서 **host** 설정을 10.10.10.10으로 변경하려면 다음과 같이 **host** 행을 편집합니다:

```
host = 10.10.10.10
```

/etc/sysconfig/luci 파일에 대한 자세한 내용은 2.4절. “**/etc/sysconfig/luci**로 **luci** 설정”에서 참조하십시오.

2.3.3. 클러스터 구성 요소를 허용하기 위해 iptables 방화벽 설정

다음에서는 Red Hat Enterprise Linux 6 (고가용성 애드온 포함) 에서 필요로하는 IP 포트를 활성화하기 위한 iptable 규칙의 예가 나열되어 있습니다. 이러한 예에서는 서브넷으로 192.168.1.0/24를 사용하지만 이러한 규칙을 사용할 경우 적절한 서브넷으로 192.168.1.0/24를 대체해야 합니다.

cman (Cluster Manager)의 경우 다음과 같은 필터링을 사용합니다.

```
$ iptables -I INPUT -m state --state NEW -m multiport -p udp -s
192.168.1.0/24 -d 192.168.1.0/24 --dports 5404,5405 -j ACCEPT
$ iptables -I INPUT -m addrtype --dst-type MULTICAST -m state --state NEW
-m multiport -p udp -s 192.168.1.0/24 --dports 5404,5405 -j ACCEPT
```

dlm (Distributed Lock Manager)의 경우:

```
$ iptables -I INPUT -m state --state NEW -p tcp -s 192.168.1.0/24 -d
192.168.1.0/24 --dport 21064 -j ACCEPT
```

ricci (Conga 원격 에이전트의 일부)의 경우:

```
$ iptables -I INPUT -m state --state NEW -p tcp -s 192.168.1.0/24 -d
192.168.1.0/24 --dport 11111 -j ACCEPT
```

modclusterd (Conga 원격 에이전트의 일부)의 경우:

```
$ iptables -I INPUT -m state --state NEW -p tcp -s 192.168.1.0/24 -d
192.168.1.0/24 --dport 16851 -j ACCEPT
```

luci (Conga 사용자 인터페이스 서버)의 경우:

```
$ iptables -I INPUT -m state --state NEW -p tcp -s 192.168.1.0/24 -d
192.168.1.0/24 --dport 16851 -j ACCEPT
```

igmp (Internet Group Management Protocol)의 경우:

```
$ iptables -I INPUT -p igmp -j ACCEPT
```

이 명령을 실행한 후, 다음과 같은 명령을 실행하여 현재 설정을 저장하여 다시 시작해도 변경 사항이 유지되도록 합니다.

```
$ service iptables save ; service iptables restart
```

2.4. /ETC/SYSCONFIG/LUCI로 LUCI 설정

Red Hat Enterprise Linux 6.1 릴리즈 이후 **/etc/sysconfig/luci** 파일을 사용하여 **luci**의 동작 일부 몇 가지를 설정할 수 있습니다. 이 파일에서 변경할 수 있는 매개 변수에는 서버 설정 뿐 만 아니라 init 스크립트에 의해 사용되는 실행 환경의 보조 설정도 포함됩니다. 또한 이 파일을 편집하여 일부 애플리케이션 매개 변수를 수정할 수 있습니다. 파일 자체에 이 파일을 편집하여 변경할 수 있는 설정 매개 변수를 설명하는 절차가 들어 있습니다.

원하는 형식을 유지하기 위해 파일을 편집할 때 **/etc/sysconfig/luci** 파일의 비설정 행을 변경하지 않는 것이 좋습니다. 또한 필요한 파일 구문을 따르도록 합니다. 특히 **INITSCRIPT** 부분의 경우 등호 앞뒤에 공백을 넣을 수 없고 공백이 포함된 문자열은 따옴표를 사용해야 합니다.

다음 예제에서는 **/etc/sysconfig/luci** 파일을 편집하여 **luci**가 작동하는 포트를 변경하는 방법을 보여줍니다.

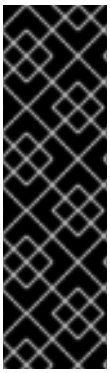
1. **/etc/sysconfig/luci** 파일에서 다음 행을 주석 해제 처리합니다:

```
#port = 4443
```

2. 4443을 원하는 포트 번호로 변경합니다. 이는 1024 (권한이 있는 포트가 아님) 또는 그 이상이어야 합니다. 예를 들어 **luci**가 8084에서 작동하는 포트를 설정하도록 다음과 같이 파일의 행을 편집할 수 있습니다.

```
port = 8084
```

3. 변경 사항을 적용하기 위해 **luci** 서비스를 다시 시작합니다.



중요

디폴트 값을 다시 정의하기 위해 **/etc/sysconfig/luci** 파일에 있는 설정 매개 변수를 수정할 때 디폴트 값 대신 새 값을 사용하는 것이 좋습니다. 예를 들어, **luci**가 작동하는 포트를 수정하는 경우 2.3.2절. “luci의 IP 포트를 사용 ”에서 설명하고 있듯이 **luci**의 IP 포트를 활성화할 때 수정된 값을 지정해야 합니다.

수정된 포트와 호스트 매개 변수는 3.2절. “luci 시작 ”에서 설명하고 있듯이 **luci** 서비스를 시작할 때 표시되는 URL에 자동으로 반영됩니다. **luci**에 액세스하려면 이 URL을 사용해야 합니다.

/etc/sysconfig/luci 파일로 설정할 수 있는 매개 변수에 대한 전체 내용은 파일 자체에 있는 문서에서 참조하십시오.

2.5. 통합 차단 (FENCE) 장치와 함께 사용하는 ACPI 설정

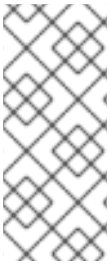
클러스터가 통합 차단 장치를 사용하는 경우, 신속하고 완벽한 차단을 위해 ACPI (Advanced Configuration and Power Interface)를 설정해야 합니다.

참고

Red Hat 고가용성 추가 기능이 지원하는 통합 차단 장치에 대한 가장 최신 정보는 http://www.redhat.com/cluster_suite/hardware/에서 참조하십시오.

클러스터 노드가 통합 차단 장치에 의해 차단되도록 설정되어 있을 경우, 해당 노드의 ACPI Soft-Off를 비

활성화합니다. ACPI Soft-Off를 비활성화하면 완전 종료 (예: **shutdown -h now**)를 시도하지 않고 통합 차단 장치가 즉시 완전하게 노드를 종료하게 합니다. 그렇지 않을 경우, ACPI Soft-Off가 활성화되어 있을 경우, 통합 차단 장치는 노드를 종료하는데 4 초 이상이 걸릴 수 있습니다. (아래 내용 참조) 또한 ACPI Soft-Off가 활성화되어 종료 도중 노드 패닉이나 정지 상태가 발생하면, 통합 차단 장치는 노드를 종료하지 못할 수 있습니다. 이러한 경우 차단 기능은 지연되거나 실패하게 됩니다. 결과적으로 노드가 통합 차단 장치로 차단되고 ACPI Soft-Off가 활성화되면, 클러스터의 복구가 느려지거나 복구를 위해 관리자의 개입이 필요하게 됩니다.



참고

노드를 차단하는데 필요한 시간은 사용되는 통합 차단 장치에 따라 달라집니다. 일부 통합 차단 장치는 전원 버튼을 누르고 있으면 이에 상당하는 동작을 합니다. 따라서 차단 장치는 4에서 5초내에 노드를 종료합니다. 다른 통합 차단 장치는 전원 버튼을 잠시 누르면 노드를 종료하기 위한 운영 체제에 따라 동작하게 됩니다. 따라서 차단 장치는 4에서 5 초보다 더 긴 시간 간격으로 노드를 종료합니다.

ACPI Soft-Off를 비활성화하려면 **chkconfig** 관리를 사용하고 차단되었을 때 노드가 즉시 꺼지는지를 확인합니다. ACPI Soft-Off를 비활성화하기 위한 우선적인 방법은 **chkconfig** 관리를 사용하는 것이지만 클러스터에 이러한 방식이 적합하지 않은 경우, 다음의 대체 방법 중 하나를 사용하여 ACPI Soft-Off를 비활성화할 수 있습니다:

- 지연없이 노드를 끄기 위해 BIOS 설정을 "instant-off" 또는 비슷한 설정 변경

  참고

  BIOS에서 ACPI Soft-Off를 비활성화하는 것은 일부 컴퓨터에서 불가능할 수 도 있습니다.

- **/boot/grub/grub.conf** 파일의 커널 부트 명령행에 **acpi=off** 추가

  중요

  이 방식은 완전히 ACPI를 비활성화합니다; ACPI가 완전히 비활성화될 경우 일부 컴퓨터는 제대로 시작되지 않습니다. 클러스터에서 다른 방식이 효과가 없을 *경우에만* 이 방식을 사용하십시오.

다음 부분에서는 ACPI Soft-Off를 비활성화는 우선적인 방법과 대체 방법에 대한 지침을 설명합니다:

- 2.5.1절. “**chkconfig** 관리를 사용하여 ACPI Soft-Off 비활성화 ” – 우선적인 방법
- 2.5.2절. “BIOS를 사용하여 ACPI Soft-Off 비활성화 ” – 첫번째 대체 방법
- 2.5.3절. “**grub.conf** 파일에서 ACPI를 완전하게 비활성화 ” – 두번째 대체 방법

2.5.1. chkconfig 관리를 사용하여 ACPI Soft-Off 비활성화

chkconfig 관리를 사용하여 **chkconfig** 관리에서 ACPI 데몬 (**acpid**)을 제거하거나 또는 **acpid**를 비활성화하여 ACPI Soft-Off 를 해제할 수 있습니다.

참고

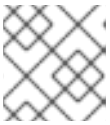
이는 ACPI Soft-Off 비활성화를 위한 우선적인 방법입니다.

다음과 같이 각각의 클러스터 노드에서 **chkconfig** 관리를 사용하여 ACPI Soft-Off를 비활성화합니다:

1. 다음 명령 중 하나를 실행합니다:
   - **chkconfig --del acpid** – 이 명령은 **chkconfig** 관리에서 **acpid**를 삭제합니다.

     – 또는 –

   - **chkconfig --level 2345 acpid off** – 이 명령은 **acpid**를 비활성화합니다.
2. 노드를 재부팅합니다.
3. 클러스터가 설정되어 실행될 때 차단되면 바로 노드가 비활성화되는지를 확인합니다.

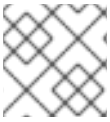


참고

fence_node 명령 또는 **Conga**로 노드를 차단할 수 있습니다.

2.5.2. BIOS를 사용하여 ACPI Soft-Off 비활성화

ACPI Soft-Off를 비활성화하는 우선적인 방법은 **chkconfig** 관리 (2.5.1절. "**chkconfig** 관리를 사용하여 ACPI Soft-Off 비활성화 ")를 사용하는 것입니다. 그러나 우선적인 방법이 사용자의 클러스터에 효과적이지 않을 경우, 다음 부분의 절차를 따릅니다.



참고

BIOS에서 ACPI Soft-Off를 비활성화하는 것은 일부 컴퓨터에서 불가능할 수 도 있습니다.

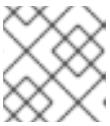
다음과 같이 각각의 클러스터 노드의 BIOS를 설정하여 ACPI Soft-Off를 비활성화할 수 있습니다:

1. 노드를 재부팅하고 **BIOS CMOS Setup Utility** 프로그램을 시작합니다.
2. **Power** 메뉴 (또는 동등의 전원 관리 메뉴)로 이동합니다.
3. **Power** 메뉴에서 **Soft-Off by PWR-BTTN** 기능 (또는 동등한 기능)을 **Instant-Off** (또는 지연없이 전원 버튼을 통해 노드를 끄는 것과 동일한 설정)로 설정합니다. 예 2.1. "**BIOS CMOS Setup Utility: Soft-Off by PWR-BTTN**는 **Instant-Off**로 설정 "에서는 **ACPI Function**이 **Enabled**로 설정되고, **Soft-Off by PWR-BTTN**이 **Instant-Off**로 설정되어있는 **Power** 메뉴를 보여줍니다.

참고

ACPI Function, **Soft-Off by PWR-BTTN** 및 **Instant-Off**과 같은 기능은 컴퓨터에 따라 다릅니다. 하지만 실행 목적은 전원 버튼을 통해 대기 시간없이 컴퓨터가 꺼지도록 BIOS를 설정하는 것입니다.

4. **BIOS CMOS Setup Utility** 프로그램을 종료하고 BIOS 설정을 저장합니다.
5. 클러스터가 설정되어 실행될 때 차단되면 바로 노드가 비활성화되는지를 확인합니다.



참고

fence_node 명령 또는 **Conga**로 노드를 차단할 수 있습니다.

예 2.1. BIOS CMOS Setup Utility: Soft-Off by PWR-BTTN는 Instant-Off로 설정

```
+---------------------------------------------|-------------------+
|    ACPI Function             [Enabled]      |    Item Help      |
|    ACPI Suspend Type         [S1(POS)]      |-------------------|
|  x Run VGABIOS if S3 Resume   Auto          |   Menu Level   *  |
|    Suspend Mode              [Disabled]     |                   |
|    HDD Power Down            [Disabled]     |                   |
|    Soft-Off by PWR-BTTN      [Instant-Off   |                   |
|    CPU THRM-Throttling       [50.0%]        |                   |
|    Wake-Up by PCI card       [Enabled]      |                   |
|    Power On by Ring          [Enabled]      |                   |
|    Wake Up On LAN            [Enabled]      |                   |
|  x USB KB Wake-Up From S3     Disabled      |                   |
|    Resume by Alarm           [Disabled]     |                   |
|  x  Date(of Month) Alarm       0            |                   |
|  x  Time(hh:mm:ss) Alarm       0 :  0 :     |                   |
|    POWER ON Function         [BUTTON ONLY   |                   |
|  x KB Power ON Password       Enter         |                   |
|  x Hot Key Power ON           Ctrl-F1       |                   |
|                                             |                   |
|                                             |                   |
+---------------------------------------------|-------------------+
```

이 예제에서는 **ACPI Function**이 **Enabled**로 설정되어 있고 **Soft-Off by PWR-BTTN**이 **Instant-Off**로 설정 되어 있음을 보여주고 있습니다.

2.5.3. `grub.conf` 파일에서 ACPI를 완전하게 비활성화

ACPI Soft-Off를 비활성화하기 위한 우선적인 방법은 **chkconfig** 관리 (2.5.1절. "**chkconfig** 관리를 사용하여 ACPI Soft-Off 비활성화 ")를 사용하는 것입니다. 사용중인 클러스터에 우선적인 방법이 효과적이지 않다면, BIOS 전원 관리 (2.5.2절. "BIOS를 사용하여 ACPI Soft-Off 비활성화 ")를 사용하여 ACPI Soft-Off를 비활성화할 수 있습니다 . 이 두가지 방법 모두 효과가 없으면 **grub.conf** 파일에 있는 커널 부트 명령행에 **acpi=off**을 추가하면 ACPI를 완전히 비활성화할 수 있습니다.

중요

이 방식은 완전히 ACPI를 비활성화합니다; ACPI가 완전히 비활성화될 경우 일부 컴퓨터는 제대로 시작되지 않습니다. 클러스터에서 다른 방식이 효과가 없을 *경우에만* 이 방식을 사용하십시오.

다음과 같이 각 클러스터 노드의 **grub.conf** 파일을 편집하여 ACPI를 완전하게 비활성화시킬 수 있습니다:

1. 텍스트 편집기로 **/boot/grub/grub.conf**를 엽니다.
2. **/boot/grub/grub.conf**에서 **acpi=off**를 커널 부트 명령행에 추가합니다. (예 2.2. "**acpi=off**로 추가된 커널 부트 명령행 " 에서 참조)
3. 노드를 재부팅합니다.
4. 클러스터가 설정되어 실행될 때 차단되면 바로 노드가 비활성화되는지를 확인합니다.

참고

fence_node 명령 또는 **Conga**로 노드를 차단할 수 있습니다.

예 2.2. acpi=off로 추가된 커널 부트 명령행

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this
file
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/mapper/vg_doc01-lv_root
#          initrd /initrd-[generic-]version.img
#boot=/dev/hda
default=0
timeout=5
serial --unit=0 --speed=115200
terminal --timeout=5 serial console
title Red Hat Enterprise Linux Server (2.6.32-193.el6.x86_64)
        root (hd0,0)
        kernel /vmlinuz-2.6.32-193.el6.x86_64 ro
root=/dev/mapper/vg_doc01-lv_root console=ttyS0,115200n8 acpi=off
        initrd /initramrs-2.6.32-131.0.15.el6.x86_64.img
```

이 예제에서 **acpi=off**는 커널 부트 명령행 **("kernel /vmlinuz-2.6.32-193.el6.x86_64.img"**로 시작하는 **)**에 추가되어 있습니다.

2.6. HA 서비스 설정 시 고려 사항

HA **(high-availability)** 서비스를 설정하여 필요에 따라 고가용성 클러스터를 생성할 수 있습니다. **Red Hat** 고가용성 추가 기능의 HA 서비스 관리의 핵심 구성 요소인 **rgmanager**는 상용 어플리케이션에 대해 콜드 페일 오버 **(cold failover)**를 구현합니다. **Red Hat** 고가용성 추가 기능에서 어플리케이션은 클러스터 클라이언트에 방해하지 않고 하나의 클러스터 노드에서 다른 노드로 장애 조치할 수 있도록 HA 서비스를 구축하기 위해 다른 클러스트 리소스와 함께 설정됩니다. 클러스터 노드에 문제가 발생한 경우 또는 클러스터 시스템 관리자가 서비스를 하나의 클러스터 노드에서 다른 노드로 이전 **(**예: 정기적 클러스터 노드 정비**)** 할 경우 HA 서비스 장애 조치가 실행될 수 있습니다.

HA 서비스를 생성하려면, 클러스터 설정 파일에서 이를 설정해야 합니다. HA 서비스는 클러스터 *리소스 (resources)*로 구성되어 있습니다. 클러스터 리소스에서는 클러스터 설정 파일에서 생성하고 관리할 요소를 구축합니다. – 예: IP 주소, 어플리케이션 초기화 스크립트, Red Hat GFS2 공유 파티션

HA 서비스는 데이터 무결성을 유지하기 위해 한 번에 하나의 클러스터 노드에서만 실행될 수 있습니다. 장애 조치 도메인에서 장애 조치 우선 순위를 지정할 수 있습니다. 장애 조치 우선 순위 지정은 장애 조치 도메인에 있는 각 노드에 우선 순위 수준을 할당하여 이루어 집니다. 우선 순위로 장애 조치 순서 즉, HA 서비스가 장애 조치해야 하는 노드의 순서를 결정합니다. 장애 조치 우선 순위를 지정하지 않으셨을 경우, HA 서비스는 장애 조치 도메인에 있는 아무 노드에 장애 조치를 실행할 수 있습니다. 또한, HA 서비스가 관련 장애 조치 도메인의 노드에서만 실행하도록 제한할 지의 여부도 지정할 수 있습니다. (제한되지 않는 장애 조치 도메인과 관련될 경우, HA 서비스는 장애 조치 도메인의 구성원이 사용할 수 없는 상황에서 아무 클러스터 노드에서 시작될 수 있습니다.)

그림 2.1. “웹 서버 클러스터 서비스 예시 ” 에서는 **"content-webserver"**라는 웹서버인 HA 서비스의 예를

보여주고 있습니다. 이는 클러스터 노드 B에서 실행되고 있으며 노드 A, B 및 D로 구성된 장애 조치 도메인에 존재합니다. 또한, 장애 조치 도메인은 노드 A 전에 노드 D로 장애 조치하고 이 장애 조치 도메인에 있는 노드에만 장애 조치를 제한하는 장애 조치 우선 순위로 설정되어 있습니다. HA 서비스는 다음과 같은 클러스터 리소스로 구성됩니다:

- IP 주소 리소스 – IP 주소 10.10.10.201.
- "httpd-content"라는 어플리케이션 리소스 – 웹 서버 어플리케이션 init 스크립트 **/etc/init.d/httpd** (**httpd**를 지정).
- 파일 시스템 리소스 – "gfs2-content-webserver"라는 Red Hat GFS2

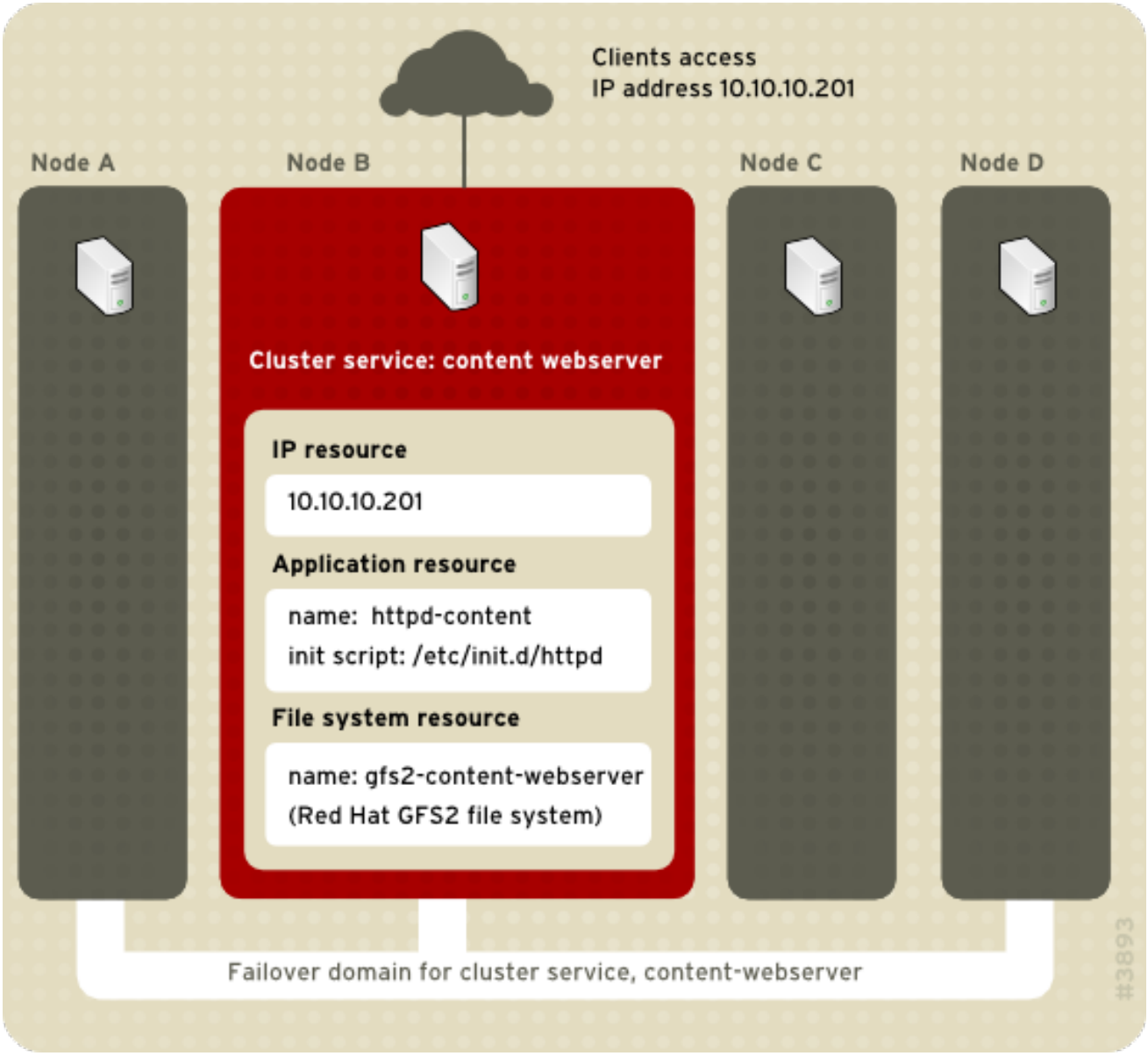
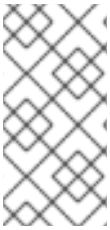


그림 2.1. 웹 서버 클러스터 서비스 예시

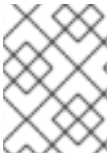
클라이언트는 IP 주소 10.10.10.201을 통해 HA 서비스에 액세스하여 웹 서버 어플리케이션, **httpd-content**와 통신할 수 있습니다. **httpd-content** 어플리케이션은 **gfs2-content-webserver** 파일 시스템을 사용합니다. 노드 B가 실패했을 경우, **content-webserver** HA 서비스는 노드 D로 장애 조치하게 됩니다. 노드 D를 사용할 수 없거나 실패한 경우, 서비스는 노드 A로 장애 조치하게 됩니다. 장애 조치는 클러스터 클라이언트에 대해 방해를 최소화하면서 발생합니다. 예를 들어, HTTP 서비스에서 특정 상태 정보 (세션 데이터 등)가 손실될 수 있습니다. HA 서비스는 장애 조치 이전에 가능했던 동일한 IP 주소를 통해 다른 클러스터 노드에서 액세스할 수 있습니다.



참고

HA 서비스 및 장애 복구 도메인에 대한 자세한 내용은 *고가용성 추가 기능 개요*에서 참조하십시오. 장애 복구 도메인 설정에 대한 자세한 내용은 3장. *Conga를 사용하여 Red Hat 고가용성 추가 기능 설정* (**Conga** 사용) 또는 7장. *명령행 도구로 Red Hat 고가용성 추가 기능 설정* (명령행 유틸리티 사용)에서 참조하십시오.

HA 서비스는 일관된 엔티티로 구성된 클러스터 리소스 그룹으로 클라이언트에 전문화된 서비스를 제공합니다. HA 서비스는 클러스터 구성 파일 **/etc/cluster/cluster.conf** (각 클러스터 노드)에서 리소스 트리로 표시됩니다. 클러스터 구성 파일에서 각 리소스 트리는 각 리소스와 리소스의 특성 그리고 리소스 트리에서 다른 리소스와의 관계 (부모, 자식, 형제 관계)를 지정하는 XML 표현입니다.



참고

HA 서비스는 계층 트리 형식으로 조직화된 리소스로 구성되어 있기 때문에 서비스는 *리소스 트리* 또는 *리소스 그룹*이라고 부릅니다. 두가지 모두 *HA 서비스*와 동의어입니다.

각 리소스 트리의 root에는 특수한 리소스 유형 – *서비스 리소스*가 있습니다. 다른 유형의 리소스는 서비스의 나머지 부분을 구성하고 그 특성을 결정합니다. HA 서비스 설정은 서비스 리소스 생성, 종속적 클러스터 리소스 생성 및 서비스의 계층적 제한에 따라 일관된 엔티티로 조직화하는 것으로 이루어져 있습니다.

HA 서비스를 설정할 때 두 가지 주요 사항을 고려해야 합니다:

- 서비스 생성에 필요한 리소스 유형
- 리소스 간의 부모, 자식, 형제 관계

리소스 유형 및 리소스 계급은 설정되는 서비스 유형에 따라 다릅니다.

클러스터 리소스 유형은 부록 B. *HA 리소스 매개 변수*에 열거되어 있습니다. 리소스 사이의 부모, 자식, 형제 관계는 부록 C. *HA 리소스 동작*에 설명되어 있습니다.

2.7. 설정 확인

클러스터 설정은 시작할 때와 설정이 다시 로드되었을 때 **/usr/share/cluster/cluster.rng**에 있는 클러스터 스키마에 따라 자동으로 유효성이 검사됩니다. 또한 **ccs_config_validate** 명령을 사용하면 언제든지 클러스터 설정의 유효성을 확인할 수 있습니다. **ccs** 명령을 사용할 때 설정 유효성 검사에 대한 내용은 5.1.6절. “설정 유효성 검사 ” 에서 참조하십시오.

주석 처리된 스키마는 **/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html** (예: **/usr/share/doc/cman-3.0.12/cluster_conf.html**)에서 확인할 수 있습니다.

다음과 같은 기본적인 오류에 대해 설정 사항을 확인합니다:

- XML 유효성 – 설정 파일이 유효한 XML 파일임을 확인합니다.
- 설정 옵션 – 옵션 (XML 요소 및 특성)이 유효한지 확인합니다.
- 옵션 값 – 옵션에 유효한 데이터 (제한적)가 포함되어 있는지 확인합니다.

다음 예에서는 유효성 검사를 보여주기 위해 유효한 설정 및 잘못된 설정을 나타내고 있습니다:

- 유효한 설정 – 예 2.3. “**cluster.conf** 설정 예: 유효한 파일 ”

- 잘못된 XML – 예 2.4. “cluster.conf 설정 예: 잘못된 XML ”
- 잘못된 옵션 – 예 2.5. “cluster.conf 설정 예: 잘못된 옵션 ”
- 잘못된 옵션 값 – 예 2.6. “cluster.conf 설정 예: 잘못된 옵션 값 ”

예 2.3. cluster.conf 설정 예: 유효한 파일

```
<cluster name="mycluster" config_version="1">
  <logging debug="off"/>
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
         </fence>
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
         </fence>
     </clusternode>
     <clusternode name="node-03.example.com" nodeid="3">
         <fence>
         </fence>
     </clusternode>
   </clusternodes>
   <fencedevices>
   </fencedevices>
   <rm>
   </rm>
</cluster>
```

예 2.4. cluster.conf 설정 예: 잘못된 XML

```
<cluster name="mycluster" config_version="1">
  <logging debug="off"/>
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
         </fence>
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
         </fence>
     </clusternode>
     <clusternode name="node-03.example.com" nodeid="3">
         <fence>
         </fence>
     </clusternode>
   </clusternodes>
   <fencedevices>
   </fencedevices>
```

```
   <rm>
   </rm>
<cluster>         <----------------INVALID
```

이 예제에서 설정의 마지막 행 (여기서는 "INVALID"로 되어 있습니다)에는 슬래시가 생략되어 있습니다. — **</cluster>** 대신 **<cluster>**로 되어 있습니다.

예 2.5. cluster.conf 설정 예: 잘못된 옵션

```
<cluster name="mycluster" config_version="1">
  <loging debug="off"/>         <----------------INVALID
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
         </fence>
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
         </fence>
     </clusternode>
     <clusternode name="node-03.example.com" nodeid="3">
         <fence>
         </fence>
     </clusternode>
   </clusternodes>
   <fencedevices>
   </fencedevices>
   <rm>
   </rm>
<cluster>
```

이 예제에서 설정의 2 번째 행 (여기서는 "INVALID"로 되어 있음)에는 잘못된 XML 요소가 들어 있습니다. — **logging** 대신 **loging**로 되어 있습니다.

예 2.6. cluster.conf 설정 예: 잘못된 옵션 값

```
<cluster name="mycluster" config_version="1">
  <loging debug="off"/>
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="-1">  <--------
INVALID
         <fence>
         </fence>
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
```

```
        </fence>
      </clusternode>
      <clusternode name="node-03.example.com" nodeid="3">
          <fence>
          </fence>
      </clusternode>
    </clusternodes>
    <fencedevices>
    </fencedevices>
    <rm>
    </rm>
 <cluster>
```

이 예제에서 설정의 4 번째 행 (여기서는 "INVALID"로 되어 있음)에는 XML 속성의 잘못된 값이 포함되어 있습니다. **node-01.example.com**에 대한 **clusternode** 행에 있는 **nodeid**가 그것입니다. 이 값은 양수 ("1") 대신 음수 ("-1")로 되어 있습니다. **nodeid** 속성의 경우 값은 양수 값이어야 합니다.

2.8. NETWORKMANAGER 사용시 고려 사항

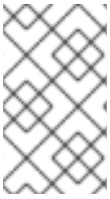
클러스터 노드에서 **NetworkManager** 사용이 지원되지 않습니다. 클러스터 노드에 **NetworkManager**를 설치했을 경우, 이를 제거하거나 비활성화해야 합니다.

참고

NetworkManager가 실행되고 있고 **chkconfig** 명령과 함께 실행하도록 설정되어 있을 경우 **cman** 서비스는 시작하지 않습니다.

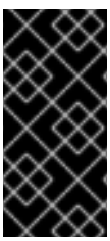
2.9. 쿼럼 디스크 (QUORUM DISK) 사용 시 고려 사항

쿼럼 디스크 (Quorum Disk)는 디스크 기반 쿼럼 (quorum, 정족수) 데몬인 **qdiskd**입니다. 이는 노드 상태를 결정하기 위한 휴리스틱스 (heuristics)를 제공합니다. 휴리스틱스를 사용하여 네트워크 파티션에 있는 노드 실행에 중요한 요소를 설정할 수 있게 합니다. 예를 들어, 3:1 분할로 된 4 노드 클러스터에서 일반적으로 3 개의 노드는 3대 1의 다수결에 의해 자동으로 "승리"합니다. 이러한 상황에서 하나의 노드가 차단됩니다. 하지만 **qdiskd**로 휴리스틱스를 설정하여 중요 리소스 (예: 중요 네트워크 경로)에 대한 액세스를 바탕으로 하나의 노드가 이길 수 있도록 할 수 있습니다. 클러스터가 노드 상태를 결정하기 위해 추가 메서드를 필요로 하는 경우 **qdiskd**를 설정해야 합니다.



참고

노드 상태에 특정 요구 사항이 없는 한 **qdiskd**를 설정할 필요가 없습니다. 특정 요구 사항의 예로는 "all-but-one" 설정입니다. all-but-one 설정에서 하나의 노드가 작동하는 경우에도 정족수를 유지하기 위해 충분한 쿼럼 투표를 제공하도록 **qdiskd**가 설정됩니다.



중요

결론적으로 휴리스틱스와 **qdiskd** 매개 변수는 사이트 환경과 필요한 특정 요건에 따라 다릅니다. 휴리스틱스와 다른 **qdiskd** 매개 변수 사용을 이해하려면 **qdisk(5) man** 페이지를 참조하십시오. 사용하는 사이트에 대한 **qdiskd** 사용 이해에 대한 도움이 필요하신 경우 Red Hat 지원 담당자에게 문의하여 주십시오.

qdiskd를 사용해야 할 경우, 다음과 같은 사항을 고려해야 합니다:

클러스터 노드 투표

쿼럼 디스크를 사용하는 경우, 각 클러스터 노드는 하나의 투표를 가져야 합니다.

CMAN 구성원 제한 시간 값

CMAN 구성원 제한 시간 값 (노드가 중지되었는지 및 구성원이 아닌 지를 CMAN이 확인하기 전 노드가 반응할 필요가 없는 시간)은 적어도 **qdiskd** 구성원 시간 제한 값의 두 배가 되어야 합니다. 그 이유는 쿼럼 데몬은 실패한 노드를 직접 감지해야 하므로 CMAN 보다 실행하는데 더 오랜 시간이 걸릴 수 있습니다. CMAN 구성원의 기본 제한 시간 값은 10초입니다. 다른 특정 사이트의 조건은 CMAN과 **qdiskd**의 구성원 제한 시간과의 관계에 영향을 미칠 수 있습니다. CMAN 구성원 제한 시간 값 조정에 대한 도움이 필요하신 경우 Red Hat 지원 담당자에게 문의해 주십시오.

펜싱 (Fencing)

qdiskd 사용시 신뢰할 수있는 펜싱을 확실하게 하려면, 파워 펜싱을 사용합니다. 다른 유형의 펜싱은 **qdiskd**로 설정되지 않은 클러스터에 대해 신뢰할 수 있을지도 모르지만 **qdiskd**로 설정된 클러스터는 신뢰할 수 없습니다.

최대 노드

qdiskd로 설정된 클러스터는 최대 16 개의 노드를 지원합니다. 이러한 제한을 두는 이유는 확장성 때문으로 노드 수가 증가하면 공유 쿼럼 디스크 장치에 있는 동기화 I/O 경쟁 수가 증가하게 됩니다.

쿼럼 디스크 장치

쿼럼 디스크 장치는 클러스터의 모든 노드에서 동시에 읽기/쓰기 액세스 권한을 갖는 공유 블록 장치여야 합니다. 블록 장치의 최소 크기는 10 메가 바이트입니다. **qdiskd**가 사용할 수 있는 공유 블록 장치의 예로는 멀티 포트 SCSI RAID 어레이, 파이버 채널 RAID SAN, 또는 RAID 제어 iSCSI 대상이 있습니다. 클러스터 쿼럼 디스크 유틸리티 (Cluster Quorum Disk Utility)인 **mkqdisk**로 쿼럼 디스크 장치를 생성할 수 있습니다. 유틸리티 사용에 대한 내용은 mkqdisk(8) man 페이지에서 참조하십시오.

참고

JBOD를 쿼럼 디스크로 사용하는 것은 권장하지 않습니다. JBOD는 신뢰할 수 있는 성능을 제공할 수 없으므로 노드가 이를 신속하게 작성하지 못하게 할 수 있습니다. 노드가 쿼럼 디스크 장치에 신속하제 작성할 수 없을 경우 노드는 클러스터에서 쫓겨나게 됩니다.

2.10. RED HAT 고가용성 추가 기능 및 SELINUX

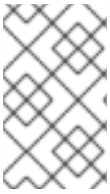
Red Hat Enterprise Linux 6 용 고가용성 추가 기능은 SELinux 정책 유형이 **targeted**로 설정된 상태에서 **enforcing** 상태의 SELinux를 지원합니다.

SELinux에 관한 보다 자세한 내용은 Red Hat Enterprise Linux 6의 *운용 가이드*를 참조하십시오.

2.11. 멀티캐스트 주소

클러스터에 있는 노드는 멀티캐스트 주소를 사용하여 서로 통신합니다. 따라서 각 네트워크 스위치와 Red Hat 고가용성 애드온에 있는 관련 네트워크 장치는 멀티캐스트 주소를 활성화하고 IGMP (Internet Group Management Protocol)를 지원하도록 설정해야 합니다. 각 네트워크 스위치와 Red Hat 고가용성 애드온에 있는 관련 네트워크 장치가 멀티캐스트 주소와 IGMP를 지원 가능한 지를 확인합니다. 지원하는 경우

멀티캐스트 주소와 IGMP를 활성화합니다. 멀티캐스트 및 IGMP가 없이 모든 노드가 클러스터에 참여할 수 없어 클러스터에 문제가 발생할 수 있습니다. 2.12절. “UDP 유니캐스트 트래픽 ”에서 설명하고 있듯이 이러한 환경에서는 UDP 유니캐스트를 사용합니다.



참고

네트워크 스위치와 관련 네트워크 장비 설정에 대한 절차는 각 제품에 따라 다릅니다. 네트워크 스위치와 관련 네트워크 장비 설정에 대한 해당 벤더 문서와 다른 정보를 참조하여 멀티 캐스트 주소와 IGMP를 활성화합니다.

2.12. UDP 유니캐스트 트래픽

Red Hat Enterprise Linux 6.2 릴리즈에서 클러스터에 있는 노드는 UDP 유니캐스트 전송 메커니즘을 사용하여 서로 통신할 수 있습니다. 하지만 클러스터 네트워크의 경우 IP 멀티캐스트를 사용할 것을 권장합니다. UDP 유니캐스트는 IP 멀티캐스트를 사용할 수 없는 경우 사용할 수 있는 대안입니다.

cluster.conf 설정 파일에서 **cman transport="udpu"** 매개 변수를 설정하여 UDP 유니캐스트를 사용하도록 Red Hat 고가용성 추가 기능을 설정할 수 있습니다. 또한 3.5.3절. “네트워크 설정 ”에서 설명하고 있듯이 **Conga** 사용자 인터페이스의 **네트워크 설정** 페이지에서 유니캐스트를 지정할 수 있습니다.

2.13. RICCI 사용 시 고려 사항

Red Hat Enterprise Linux 6의 경우 **ricci**는 **ccsd**로 교체됩니다. 따라서 **cman_tool version -r**, **ccs** 명령이나 **luci** 사용자 인터페이스 서버를 통해 업데이트된 클러스터 구성 정보를 전달할 수 있는 각 클러스터 노드에서 **ricci**를 실행해야 합니다. **service ricci start**를 사용하여 **ricci**를 시작하거나 **chkconfig**를 통해 부팅시 시작하도록 활성화합니다. **ricci**에 대한 IP 포트를 활성화하는 내용은 2.3.1절. “클러스터 노드에서 IP 포트 사용 ” 에서 참조하십시오.

Red Hat Enterprise Linux 6.1 및 그 이후 릴리즈에서 **ricci** 를 사용하는 경우, 특정 노드에서 업데이트된 클러스터 설정을 처음으로 전달할 때 암호가 필요합니다. 시스템에 **ricci**를 설치하고 root로 **ricci** 사용자에 대해 **passwd ricci** 명령을 사용하여 **ricci** 암호를 설정합니다.

2.14. 클러스터 환경에서 가상 머신 설정

가상 머신 리소스로 클러스터를 설정하려면 가상 머신을 시작 및 중지하기 위해 **rgmanager** 도구를 사용해야 합니다. **virsh**를 사용하여 컴퓨터를 시작하면 여러 곳에서 가상 머신을 실행할 수 있으나 가상 머신에서 데이터가 손상될 수 있습니다.

관리자가 클러스터 환경에서 클러스터 도구와 비 클러스터 도구 모두를 사용하여 가상 머신을 동시에 "2대 가동"하는 가능성을 줄이기 위해 가상 머신 설정 파일을 기본이 아닌 위치에 저장하도록 시스템을 설정할 수 있습니다. 기본 위치 이외이 장소에 가상 머신 설정 파일을 저장하면 설정 파일이 **virsh**이외의 것은 인식하지 않기 때문에 **virsh**를 사용하여 가상 머신을 실수로 시작하는 것이 어려워지게 됩니다.

가상 머신 설정 파일의 디폴트가 아닌 위치는 어느곳이든 될 수 있습니다. NFS 공유 또는 GFS2 공유 파일 시스템의 사용 이점은 관리자가 클러스터 멤버 전체에 걸쳐 설정 파일을 계속 동기화할 필요가 없다는 것입니다. 하지만 관리자가 어떤 방법으로 클러스터 전역에 걸쳐 컨텐츠를 지속적을 동기화하는 한 로컬 디렉토리를 사용하는 것도 허용됩니다.

클러스터 설정에서 가상 머신은 가상 머신 리소스의 **path** 속성을 사용하여 디폴트가 아닌 위치를 참조할 수 있습니다. **path** 속성은 디렉토리 또는 콜론 ':'으로 구분되는 디렉토리 모음이며 특정 파일로의 경로가 아닌 점에 유의하십시오.

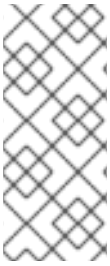
주의

libvirt-guests 서비스는 **rgmanager**가 실행되고 있는 모든 노드에서 비활성화 되어 있어야 합니다. 가상 머신이 자동으로 시작하거나 다시 시작하면 여러 위치에서 가상 머신이 실행될 수 있으며 이로 인해 가상 머신에 있는 데이터가 손상될 수 있습니다.

가상 머신 리소스 속성에 대한 자세한 내용은 표 B.24. "가상 머신 " 에서 참조하십시오.

3장. CONGA를 사용하여 RED HAT 고가용성 추가 기능 설정

다음 부분에서는 **Conga**를 사용하여 **Red Hat** 고가용성 추가 기능을 설정하는 방법에 대해 설명합니다. 실행 중인 클러스터를 관리하기 위해 **Conga**를 사용하는 방법에 대한 보다 자세한 내용은 *4장. Conga를 사용하여 Red Hat 고가용성 추가 기능 관리*에서 참조하십시오.



참고

Conga는 Red Hat 고가용성 추가 기능을 관리하기 위해 사용할 수 있는 그래픽 사용자 인터페이스입니다. 하지만 이러한 인터페이스를 효과적으로 사용하려면 기본 개념에 대해 명확하게 이해하고 있어야 합니다. 사용자 인터페이스에서 사용 가능한 기능을 검색하여 클러스터 설정을 배우는 것은 권장하지 않습니다. 구성 요소가 실패할 때 모든 서비스 실행이 유지되도록 시스템이 견고하지 않을 수 있기 때문입니다.

이는 다음과 같은 부분으로 구성되어 있습니다:

- 3.1절. “설정 작업 ”
- 3.2절. “luci 시작 ”
- 3.3절. “luci로의 액세스 제어 ”
- 3.4절. “클러스터 생성 ”
- 3.5절. “글로벌 클러스터 등록 정보 ”
- 3.6절. “차단 (Fence) 장치 설정 ”
- 3.7절. “클러스터 멤버에 대한 차단 장치 설정 ”
- 3.8절. “장애 조치 도메인 설정 ”
- 3.9절. “글로벌 클러스터 리소스 설정 ”
- 3.10절. “클러스터에 클러스터 서비스 추가 ”

3.1. 설정 작업

Conga를 사용한 **Red Hat** 고가용성 추가 기능 소프트웨어 설정은 다음과 같은 단계로 구성되어 있습니다:

1. **Conga** 설정 사용자 인터페이스 구성 및 실행 – **luci** 서버. 3.2절. “luci 시작 ”에서 참조하십시오.
2. 클러스터 생성. 3.4절. “클러스터 생성 ”에서 참조하십시오.
3. 글로벌 클러스터 등록 정보 설정. 3.5절. “글로벌 클러스터 등록 정보 ” 에서 참조하십시오.
4. 차단 장치 설정. 3.6절. “차단 (Fence) 장치 설정 ” 에서 참조하십시오.
5. 클러스터 멤버에 대한 차단 장치 설정. 3.7절. “클러스터 멤버에 대한 차단 장치 설정 ” 에서 참조하십시오.
6. 장애 조치 도메인 생성. 3.8절. “장애 조치 도메인 설정 ” 에서 참조하십시오.
7. 리소스 생성. 3.9절. “글로벌 클러스터 리소스 설정 ” 에서 참조하십시오.
8. 클러스터 서비스 생성. 3.10절. “클러스터에 클러스터 서비스 추가 ” 에서 참조하십시오.

3.2. LUCI 시작

참고

luci를 사용하여 클러스터를 설정하려면, 2.13절. "**ricci** 사용 시 고려 사항 "에서 설명하고 있듯이, **ricci**가 설치되어 클러스터 노드에서 실행하고 있어야 합니다. 이 부분에서 설명하고 있듯이, **ricci**를 사용하려면 3.4절. "클러스터 생성 "에서 설명하고 있듯이 클러스터를 생성할 때 각 클러스터 노드에 대해 **luci**가 요구하는 암호가 필요합니다.

luci를 시작하기 전 **luci**가 통신하게 될 노드 중 하나에 있는 **luci** 서버에서 클러스터 노드에 있는 IP포트가 포트 11111로 연결을 허용하는지를 확인합니다. 클러스터 노드에서 IP 포트를 활성화하는 방법에 대한 자세한 내용은 2.3.1절. "클러스터 노드에서 IP 포트 사용 "에서 참조하십시오.

Conga로 Red Hat 고가용성 추가 기능을 관리하려면 다음과 같이 **luci**를 설치 및 실행합니다:

1. **luci**를 호스트하기 위해 컴퓨터를 선택하고 해당 컴퓨터에 **luci** 소프트웨어를 설치합니다. 예:

```
# yum install luci
```

참고

일반적으로 서버 패키지 또는 데이터 센터에 있는 컴퓨터는 **luci**를 호스팅합니다. 하지만 클러스터 컴퓨터가 **luci**를 호스팅할 수 있습니다.

2. **service luci start**를 사용하여 **luci**를 시작합니다. 예:

```
# service luci start
Starting luci: generating https SSL certificates...  done
                                                           [  OK
]

Please, point your web browser to https://nano-01:8084 to access
luci
```

참고

Red Hat Enterprise Linux release 6.1 이후에서는 **/etc/sysconfig/luci** 파일을 사용하여 포트 및 호스트 매개 변수를 포함하여 **luci** 동작 일부를 설정할 수 있습니다. 자세한 내용은 2.4절. "**/etc/sysconfig/luci**로 luci 설정"에서 참조하십시오. 수정된 포트와 호스트 매개 변수는 **luci** 서비스가 시작할 때 표시되는 URL에 자동으로 반영됩니다.

3. 웹 브라우저에서 **luci** 서버의 URL을 URL 주소 표시줄에 배치한 후 **Go** (또는 이에 해당하는 것)을 클릭합니다. **luci** 서버의 URL 구문은 **https://*luci_server_hostname*:*luci_server_port***입니다. *luci_server_port*의 기본값은 **8084**입니다.

   처음으로 **luci**에 액세스할 때 (**luci** 서버의) 자체 서명된 SSL 인증서에 대한 웹 브라우저의 메세지가 나타납니다. 대화 상자를 승인하면 웹 브라우저는 **luci** 로그인 페이지를 표시합니다.

4. **luci**를 호스팅하는 시스템에서 인증할 수 있는 모든 사용자는 **luci**에 로그인할 수 있습니다. 하지만 Red Hat Enterprise Linux 6.2 이후에서는 관리자 (root 사용자 또는 관리 권한이 있는 사용자)가 사용자에 대한 권한을 설정하기 전 까지 **luci**를 실행하고 있는 시스템에서 root 사용자만이 **luci** 구성 요소에 액세스할 수 있습니다. 사용자에 대한 **luci** 권한 설정에 대한 자세한 내용은 3.3 절. “luci로의 액세스 제어 ”에서 참조하십시오.

   luci에 로그인하면 그림 3.1. “luci Homebase 페이지 ” 에서와 같이 **luci Homebase**가 나타납니다.

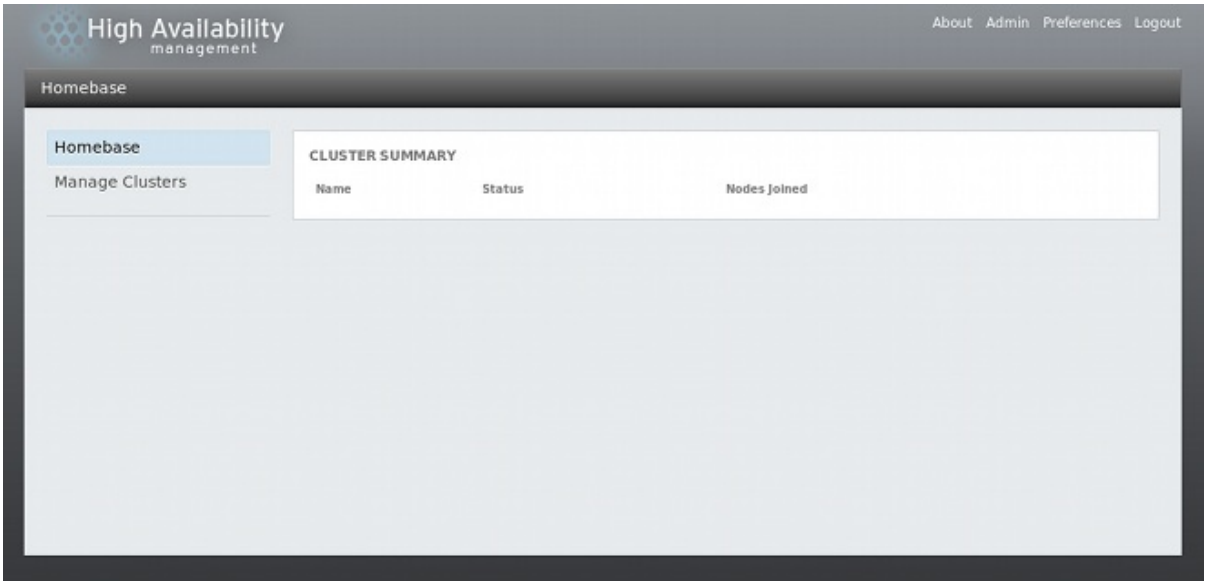


그림 3.1. luci Homebase 페이지

참고

luci에서는 15 분간 작업이 수행되지 않으면 로그아웃되는 유휴 시간 제한이 있습니다.

3.3. LUCI로의 액세스 제어

Red Hat Enterprise Linux 6의 초기 릴리즈에서 다음과 같은 기능이 **사용자 및 권한** 페이지에 추가되었습니다.

- Red Hat Enterprise Linux 6.2에서 root 사용자 또는 **luci**가 실행되고 있는 시스템에서 **luci** 관리 권한을 갖는 사용자는 시스템의 개별 사용자에게 권한을 설정하여 다양한 **luci** 구성 요소에 대한 액세스를 제어할 수 있습니다.
- Red Hat Enterprise Linux 6.3에서 root 사용자 또는 **luci** 관리 권한을 갖는 사용자는 **luci** 인터페이스를 사용하여 사용자를 시스템에 추가할 수 있습니다.
- Red Hat Enterprise Linux 6.4에서 root 사용자 또는 **luci** 관리 권한을 갖는 사용자는 **luci** 인터페이스를 사용하여 시스템에서 사용자를 삭제할 수 있습니다.

사용자 추가, 사용자 삭제, 사용자 권한을 설정하려면 **root** 또는 사전에 관리자 권한이 부여된 사용자로 **luci**에 로그인한 후 **luci** 화면의 오른쪽 상단 코너에 있는 **관리** 섹션을 클릭합니다. 그러면 **사용자 및 권한** 페이지가 나타나 기존 사용자를 표시합니다.

사용자를 삭제하려면 사용자를 선택하고 **선택 사항 삭제**를 클릭합니다.

사용자를 추가하려면 **사용자 추가**를 클릭하고 추가할 사용자 이름을 입력합니다.

사용자 권한을 설정하거나 변경하려면 **사용자 권한**의 드롭 다운 메뉴에서 사용자를 선택합니다. 여기서 다음과 같은 권한을 설정할 수 있습니다:

Luci 관리자

root 사용자와 동일한 권한을 사용자에게 부여합니다. 모든 클러스터에서 완전한 권한이며 root 이외의 다른 모든 사용자 권한을 설정 또는 삭제할 수 있습니다. 이러한 권한은 제한할 수 없습니다.

클러스터 생성 가능

3.4절. "클러스터 생성 "에서 설명하고 있듯이 사용자는 새 클러스터를 생성할 수 있습니다.

기존 클러스터 가져오기 가능

4.1절. "기존 클러스터를 luci 인터페이스에 추가 " 에서 설명하고 있듯이 사용자가 기존 클러스터를 **luci** 인터페이스에 추가할 수 있습니다.

luci로 가져오기하거나 생성된 각 클러스터의 경우 사용자에게 다음과 같은 권한을 설정할 수 있습니다:

클러스터 보기 가능

사용자가 특정 클러스터를 볼 수 있습니다.

클러스터 설정 변경 가능

클러스터 노드를 추가 및 제거하는 것을 제외하고 사용자가 특정 클러스터 설정을 수정할 수 있습니다.

서비스 그룹 활성화, 비활성화, 재배치, 마이그레이션 가능

4.5절. "고가용성 서비스 관리 "에서 설명하고 있듯이 사용자가 고가용성 서비스를 관리할 수 있습니다.

클러스터 노드 중지, 시작, 재부팅 가능

4.3절. "클러스터 노드 관리 "에서 설명하고 있듯이 사용자는 클러스터의 개별 노드를 관리할 수 있습니다.

노드 추가 및 삭제 가능

3.4절. "클러스터 생성 "에서 설명하고 있듯이 사용자는 클러스터에서 노드를 추가 및 삭제할 수 있습니다.

Luci에서 클러스터 삭제 가능

4.4절. "클러스터 시작, 중지, 다시 시작, 삭제 " 에서 설명하고 있듯이 사용자는 **luci** 인터페이스에서 클러스터를 삭제할 수 있습니다.

제출 (Submit)을 클릭하면 권한이 적용됩니다. 초기 값으로 되돌아가려면 **다시 설정 (Reset)**을 클릭합니다.

3.4. 클러스터 생성

luci를 사용하여 클러스터를 생성하기 위해 클러스터 이름 지정, 클러스터에 클러스터 노드 추가, 각 노드에 대해 **ricci** 암호 입력, 클러스터를 생성하기 위해 요청 제출이 필요합니다. 노드 정보와 암호가 정확할 경우, **Conga**는 자동으로 (현재 해당 소프트웨어 패키지가 설치되어 있지 않을 경우) 소프트웨어를 클러스터 노드에 설치하여 클러스터를 시작합니다. 다음과 같이 클러스터를 생성합니다:

1. **luci Homebase** 페이지의 왼쪽에 있는 메뉴에서 **클러스터 관리 (Manage Clusters)**를 클릭하면 그림 3.2. "luci 클러스터 관리 페이지 " 에서와 같이 **클러스터 (Clusters)** 화면이 나타납니다.

그림 3.2. luci 클러스터 관리 페이지

2. **생성 (Create)**을 클릭합니다. 그림 3.3. "luci 클러스터 생성 대화 상자 " 에서와 같이 **새 클러스터 생성 (Create New Cluster)** 대화 상자가 나타납니다.

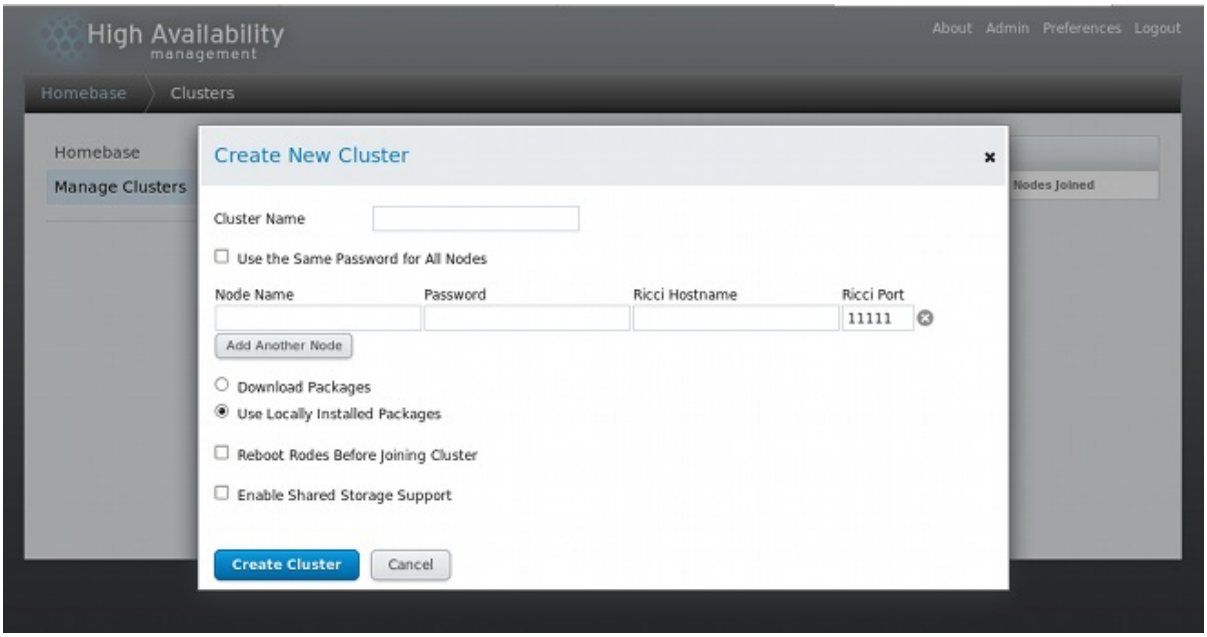


그림 3.3. luci 클러스터 생성 대화 상자

3. 필요에 따라 **새 클러스터 생성 (Create New Cluster)** 대화 상자에 다음과 같은 매개 변수를 입력합니다:

   - **클러스터 이름 (Cluster Name)** 텍스트 상자에 클러스터 이름을 입력합니다. 클러스터 이름은 15자를 초과할 수 없습니다.

   - 클러스터에 있는 각 노드가 동일한 **ricci** 암호를 갖는 경우, **모든 노드에 동일한 암호 사용 (Use the same password for all nodes)**을 선택하면 노드 추가 시 **암호 (password)**란에 자동 입력할 수 있습니다.

- **노드 이름 (Node Name)** 란에 클러스터에 있는 노드의 노드 이름을 입력하고 **암호 (Password)**란에 노드의 **ricci** 암호를 입력합니다.
- 사용중인 시스템이 클러스터 트래픽에만 사용되는 개인 전용 네트워크로 설정된 경우 **luci**를 설정하여 클러스터 노드 이름을 해결하는 주소와 다른 주소에서 **ricci**와 통신하고자 할 수 있습니다. 이를 위해 주소를 **Ricci Hostname**으로 입력합니다.
- **ricci** 에이전트에 대해 기본 포트인 11111 외에 다른 포트를 사용하고 있을 경우 해당 매개 변수를 변경할 수 있습니다.
- **다른 노드 추가 (Add Another Node)** 를 클릭하여 클러스터의 추가 노드마다 노드 이름과 **ricci** 암호를 입력합니다.
- 클러스터를 만들 때 이미 노드에 설치되어 있는 클러스터 소프트웨어 패키지를 업그레이드하지 않을 경우, **로컬로 설치된 패키지 사용 (Use locally installed packages)** 옵션을 선택한 상태로 둡니다. 모든 클러스터 소프트웨어 패키지를 업그레이그하고자 할 경우, **패키지 다운로드 (Download Packages)** 옵션을 선택합니다.

참고

로컬로 설치된 패키지 사용 (Use locally installed packages) 또는 **패키지 다운로드 (Download Packages)** 옵션 중 어느 것을 선택해도 기본 클러스터 구성 요소의 일부가 결여되어 있을 경우 (`cman`, `rgmanager`, `modcluster` 및 모든 종속성), 이는 설치됩니다. 설치할 수 없을 경우, 노드 생성은 실패하게 됩니다.

- 원하는 경우 **클러스터에 참여하기 전 노드 재부팅 (Reboot nodes before joining cluster)** 을 선택합니다.
- 클러스터 스토리지가 필요한 경우 **공유 스토리지 지원 활성화 (Enable shared storage support)**를 선택합니다. 이렇게 하면 클러스터 스토리지를 지원하는 패키지를 다운로드하여 클러스터된 LVM을 활성화합니다. 이는 장애 복구형 스토리지 추가 기능 또는 확장 가능한 파일 시스템 추가 기능을 액세스할 수 있는 경우에만 이 옵션을 선택하셔야 합니다.

4. **클러스터 생성 (`Create Cluster`)**을 클릭합니다. **클러스터 생성 (`Create Cluster`)**을 클릭하면 다음과 같은 동작이 실행됩니다:
   1. **패키지 다운로드 (Download Packages)** 를 선택한 경우, 클러스터 소프트웨어 패키지는 노드에 다운로드됩니다.
   2. 클러스터 소프트웨어는 노드에 설치됩니다 (또는 적절한 소프트웨어가 설치되었는지를 확인합니다).
   3. 클러스터 설정 파일이 업데이트되어 클러스터에 있는 각 노드에 전달됩니다.
   4. 추가된 노드는 클러스터에 참여합니다.

   클러스터가 생성되고 있음을 가리키는 메세지가 나타납니다. 클러스터가 준비되면 그림 3.4. “클러스터 노드 표시 ”에서 보여주듯이 새로 생성된 클러스터의 상태가 나타납니다. **ricci**가 노드에서 실행되고 있지 않으면 클러스터 생성은 실패하게 됩니다.

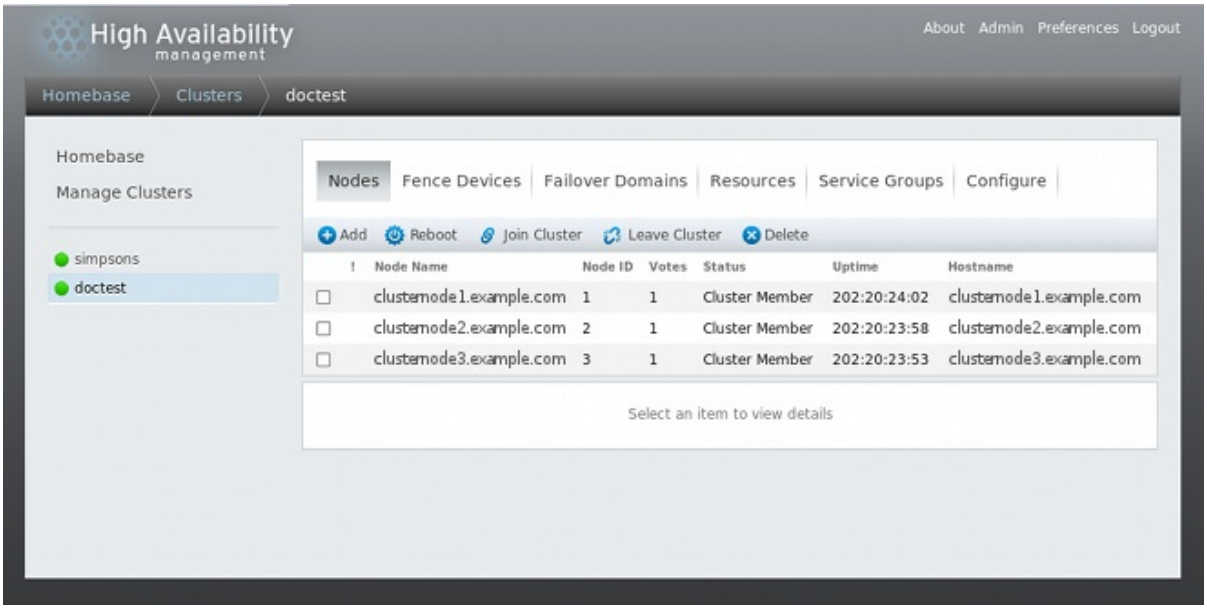
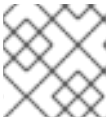


그림 3.4. 클러스터 노드 표시

5. 클러스터를 생성하기 위해 **클러스터 생성 (Create Cluster)**을 클릭한 후, 클러스터 노드 표시 페이지의 상단 메뉴에서 **추가 (Add)** 또는 **삭제 (Delete)** 기능을 클릭하여 클러스터에서 노드를 추가 또는 삭제할 수 있습니다. 전체 클러스터를 삭제하는 경우를 제외하고 삭제하기 전 노드를 중지해야 합니다. 현재 실행 중인 기존 클러스터에서 노드를 삭제하는 방법에 대한 내용은 4.3.4절. "클러스터에서 멤버 삭제 "에서 참조하십시오.



참고

클러스터에서 클러스터 노드를 제거하는 것은 되돌릴 수 없는 파괴식 작업입니다.

3.5. 글로벌 클러스터 등록 정보

설정할 클러스터를 선택하면 특정 클러스터 페이지가 표시됩니다. 이 페이지에는 클러스터 전역 속성을 설정할 수 있는 인터페이스가 있습니다. 클러스터 상단의 **설정**을 클릭하여 클러스터 전역 속성을 설정할 수 있습니다. 이는 **일반, 차단 데몬, 네트워크, 중복 링, QDisk, 로깅**과 같은 탭을 제공하는 탭 인터페이스를 생성합니다. 이러한 탭에서 매개 변수를 설정하려면 다음과 같은 섹션에 있는 단계를 따릅니다. 이 탭에서 매개 변수를 설정할 필요가 없는 경우 이러한 탭 색션을 건너뛰기합니다.

3.5.1. 일반 등록 정보 설정

일반 (General) 탭을 클릭하면 **일반 등록 정보 (General Properties)** 페이지가 나타나고 설정 버전을 변경하기 위한 인터페이스가 제공됩니다.

- **클러스터 이름 (Cluster Name)** 텍스트 상자는 클러스터 이름을 표시합니다; 이는 클러스터 이름 변경을 허용하지 않습니다. 클러스터의 이름을 변경하기 위한 유일한 방법은 새 이름으로 새 클러스터 설정을 생성하는 것입니다.

- 클러스터 생성 시 **설정 버전 (Configuration Version)** 값은 **1**로 설정되며 클러스터 설정을 변경할 때 마다 자동으로 값이 증가합니다. 하지만 다른 값을 설정해야 하는 경우, **설정 버전 (Configuration Version)** 텍스트 상자에서 이를 지정할 수 있습니다.

설정 버전 (Configuration Version) 값을 변경했을 경우 **적용 (Apply)**을 클릭하여 변경 내용을 적용시킵니다.

3.5.2. 차단 데몬 등록 정보 설정

차단 데몬 (Fence Daemon) 탭을 클릭하면 **실패 후 대기 시간 (Post Fail Delay)** 및 **참여 후 대기 시간 (Post Join Delay)** 설정을 위한 인터페이스를 제공하는 **차단 데몬 등록 정보 (Fence Daemon Properties)** 페이지가 나타납니다. 이러한 매개 변수에 대해 설정하는 값은 클러스터의 일반적인 차단 등록 정보입니다. 클러스터의 노드에 대해 특정 차단 장치를 설정하려면 3.6절. "차단 (Fence) 장치 설정 " 에서 보여주듯 이 클러스터 보기의 **차단 장치 (Fence Devices)** 메뉴에 있는 항목을 사용합니다.

- **실패 후 대기 시간 (Post Fail Delay)** 매개 변수는 노드 실패 후 노드 (차단 도메인의 멤버)를 차단하기 전까지 차단 데몬 (**fenced**)이 기다리는 시간 (초)입니다. **실패 후 대기 시간 (Post Fail Delay)** 기본값은 **0**입니다. 이 값은 클러스터와 네트워크 성능에 맞게 변경할 수 있습니다.

- **참여 후 대기시간** 매개 변수는 노드가 차단 데몬에 연결한 후 노드를 차단하기 전 까지 차단 데몬 (**fenced**)의 대기 시간 (초)입니다. **참여 후 대기시간** 기본값은 **6**입니다. **참여 후 대기시간**은 일반적으로 20 초에서 30 초 사이로 설정되어 있지만 클러스터와 네트워크 성능에 따라 다를 수 있습니다.

필요한 값을 입력하고 **적용 (Apply)**을 클릭하여 변경 사항을 적용합니다.

참고

참여 후 대기시간 (Post Join Delay) 및 **실패 후 대기시간 (Post Fail Delay)** 에 대한 자세한 내용은 fenced(8) man 페이지에서 참조하십시오.

3.5.3. 네트워크 설정

네트워크 (Network) 탭을 클릭하면 **네트워크 설정 (Network Configuration)** 페이지가 나타나서 네트워크 전송 유형 설정을 위한 인터페이스를 제공합니다.

이 탭을 통해 다음 옵션 중 하나를 선택할 수 있습니다:

- **UDP 멀티캐스트 및 클러스터가 멀티캐스트 주소를 선택하게 함**

  이는 기본 설정입니다. 이 옵션이 선택되어 있는 경우, Red Hat 고가용성 추가 기능 소프트웨어는 클러스터 ID를 기반으로 멀티캐스트 주소를 생성합니다. 이는 주소의 하위 16 비트를 생성하여 이를 IP 프로토콜이 IPV4 또는 IPV6 인지에 대한 여부에 따라 주소의 상위 부분에 추가합니다:

  - IPv4 용 – 형성되는 주소는 239.192입니다. 이에 Red Hat 고가용성 추가 기능 소프트웨어에 의해 생성된 하위 16 비트가 추가됩니다.
  - IPv6 용 – 형성되는 주소는 FF15::입니다. 이에 Red Hat 고가용성 추가 기능 소프트웨어에 의해 생성된 하위 16 비트가 추가됩니다.

  참고

  클러스터 ID는 각 클러스터에 대해 **cman**을 생성하는 고유한 식별자입니다. 클러스터 ID를 표시하려면 클러스터 노드에서 **cman_tool status** 명령을 실행합니다.

- **UDP 멀티캐스트 및 멀티 캐스트 주소를 수동으로 지정**

  특정 멀티 캐스트 주소를 사용해야 하는 경우, 이 옵션을 선택하고 **멀티 캐스트 주소 (Multicast Address)** 텍스트 상자에 멀티 캐스트 주소를 입력합니다.

  멀티캐스트 주소를 지정하지 않을 경우, **cman**을 사용하는 239.192.x.x 시리즈 (또는 IPv6 용 FF15::)를 사용해야 합니다. 그렇지 않으면 이 범위 이외의 멀티캐스트 주소의 사용은 예기치 못한

결과를 초래할 수 있습니다. 예를 들어, 224.0.0.x (이는 "네트워크 상의 모든 호스트") 사용은 올바르게 라우트되지 않거나 일부 하드웨어에 의해 전혀 라우트되지 않을 수 도 있습니다.

멀티캐스트 주소를 지정 또는 수정하면 이를 적용하기 위해 클러스터를 다시 시작해야 합니다. **Conga**를 사용하여 클러스터를 시작 및 중지하는 방법에 대한 내용은 4.4절. “클러스터 시작, 중지, 다시 시작, 삭제 ”에서 참조하십시오.

참고

멀티캐스트 주소를 지정할 경우, 클러스터 패킷이 통과하는 라우터의 설정을 검사하였는지 확인합니다. 일부 라우터는 주소를 인식하는데 시간이 오래 걸릴 수 있으므로, 클러스터 성능에 영향을 줄 수 있습니다.

- **UDP 유니캐스트 (UDPU)**

  Red Hat Enterprise Linux 6.2 릴리즈에서 클러스터에 있는 노드는 UDP 유니캐스트 전송 메커니즘을 사용하여 서로 통신할 수 있습니다. 하지만 클러스터 네트워크의 경우 IP 멀티캐스트를 사용할 것을 권장합니다. UDP 유니캐스트는 IP 멀티캐스트를 사용할 수 없는 경우 사용할 수 있는 대안입니다. UDP 유니캐스트를 사용하여 GFS2 배포는 권장하지 않습니다.

적용 (Apply)을 클릭합니다. 전송 유형을 변경할 때 변경 내용을 적용하기 위해 클러스터를 다시 시작해야 합니다.

3.5.4. 중복 링 프로토콜 설정

Red Hat Enterprise Linux 6.4에서 Red Hat 고가용성 애드온은 중복 링 프로토콜 설정을 지원합니다. 중복 링 프로토콜을 사용할 때 7.6절. “중복 링 프로토콜 설정 ” 에서 설명하고 있듯이 여러 가지 고려해야 할 사항이 있습니다.

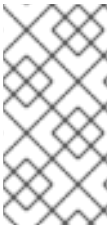
중복 링 탭을 클릭하면 **중복 링 프로토콜 설정** 페이지가 나타납니다. 이 페이지에서는 클러스터에 현재 설정된 모든 노드가 표시됩니다. 중복 링 프로토콜을 사용하도록 시스템을 설정하는 경우 두 번째 링의 각 노드에 **대체 이름**을 지정해야 합니다.

중복 링 프로토콜 설정 페이지에서는 옵션으로 두 번째 링의 **대체 링 멀티캐스트 주소, 대체 링 CMAN 포트, 대체 링 멀티캐스트 패킷 TTL** 을 지정할 수 있습니다.

두 번째 링의 멀티캐스트 주소를 지정하는 경우 대체 멀티캐스트 주소 또는 대체 포트는 첫 번째 링의 멀티캐스트 주소와 달라야 합니다. 대체 포트를 지정하는 경우 시스템 자체가 작업을 수행하기 위해 포트 및 포트 1을 사용하므로 첫 번째 링과 두 번째 링의 포트 번호는 최소 두 개의 다른 것이어야 합니다. 대체 멀티캐스트 주소를 지정하지 않은 경우, 시스템은 두 번째 링에 대해 자동으로 다른 멀티캐스트 주소를 사용하게 됩니다.

3.5.5. 쿼럼 디스크 (Quorum Disk) 설정

QDisk 탭을 클릭하면 **쿼럼 디스크 설정 (Quorum Disk Configuration)** 페이지가 표시되어 쿼럼 디스크 사용을 필요로 하는지에 대한 여부를 설정하는 쿼럼 디스크 매개 변수 설정을 위한 인터페이스가 제공됩니다.



참고

쿼럼 디스크 매개 변수 및 휴리스틱스는 사이트 환경과 필요한 특정 요건에 따라 달라집니다. 쿼럼 디스크 매개 변수 및 휴리스틱스 사용을 이해하려면 **qdisk(5) man** 페이지에서 참조하십시오. 쿼럼 디스크의 이해 및 사용에 대한 지원이 필요하실 경우 **Red Hat** 지원 담당자에게 문의하십시오.

쿼럼 디스크를 사용하지 않음 (Do Not Use a Quorum Disk) 매개 변수는 기본값으로 활성화되어 있습니다. 쿼럼 디스크를 사용해야 할 경우, **쿼럼 디스크 사용 (Use a Quorum Disk)** 을 클릭하고 쿼럼 디스크 매개 변수를 입력, **적용 (Apply)**을 클릭한 후 변경 사항이 반영되도록 클러스터를 다시 시작합니다.

표 3.1. "쿼럼 디스크 매개 변수 " 에서는 쿼럼 디스크 매개 변수에 대해 설명합니다.

표 3.1. 쿼럼 디스크 매개 변수

매개 변수	설명
물리적 장치를 지정: 장치 레이블 사용	**mkqdisk** 유틸리티에 의해 생성된 쿼럼 디스크 레이블을 지정합니다. 이 영역을 지정하면, 쿼럼 데몬은 **/proc/partitions** 파일을 읽어 발견된 모든 블록 장치에서 **qdisk** 서명을 확인하고 지정된 레이블에 대해 레이블을 비교합니다. 이는 쿼럼 장치 이름이 노드 간에 다르게 설정되어 있는 경우 유용합니다.
휴리스틱스 (Heuristics)	**프로그램으로의 경로** – 휴리스틱을 사용할 수 있는지를 확인하는데 사용하는 프로그램입니다. **/bin/sh -c**에서 실행할 수 있는 것이라면 무엇이든 상관없습니다. 반환 값 0은 성공을 나타내며 그 외의 값은 실패를 의미합니다. 이 필드는 필수 사항입니다. **간격 (Interval)** – 휴리스틱이 폴링되는 빈도 수 (초)입니다. 모든 휴리스틱 방법의 기본 간격은 2 초입니다. **점수 (Score)** – 휴리스틱 방법의 가중치입니다. 휴리스틱 방법의 점수를 결정할 때는 주의하셔야 합니다. 각 휴리스틱 방법의 기본 점수는 1입니다. **TKO** – 이러한 휴리스틱 방법을 사용 불가 확인 전 까지 필요한 연속 실패 수입니다.
최소 점수 합계	노드가 "실행 중"(alive)이라고 간주되는데 필요한 최소 점수. 생략되어 있거나 0으로 설정되는 경우, 기본 함수 **floor((*n*+1)/2)**가 사용됩니다. 여기서 *n*은 휴리스틱스 점수의 합계입니다. **최소 점수 합계 (Minimum Total Score)**값은 휴리스틱 점수의 합계를 초과해서는 안되며, 초과한 경우 쿼럼 디스크를 사용할 수 없습니다.

참고

QDisk 설정 탭의 **적용 (Apply)**을 클릭하여 변경 사항을 각 클러스터 노드에 있는 클러스터 설정 파일에 (**/etc/cluster/cluster.conf**) 전달합니다. 하지만, 쿼럼 디스크를 실행하기 위해서나 또는 쿼럼 디스크 매개 변수에 변경 사항을 적용하려면 클러스터를 다시 시작하여 (4.4절. "클러스터 시작, 중지, 다시 시작, 삭제 " 참조) 각 노드에서 **qdiskd** 데몬이 다시 시작하는 지를 확인해야 합니다.

3.5.6. 로깅 설정

로깅 (Logging) 탭을 클릭하면 **로깅 설정 (Logging Configuration)** 페이지가 나타납니다. 이는 로깅 설정에 대한 인터페이스를 제공합니다.

글로벌 로깅 설정에 대해 다음과 같은 설정을 구성할 수 있습니다:

- **디버깅 메세지 로그**를 클릭하면 로그 파일에 있는 디버깅 메세지를 활성화합니다.

- **Syslog에 있는 메세지 로그**를 클릭하면 **syslog**에 있는 메세지를 사용하게 됩니다. **Syslog 메세지 기능** 및 **Syslog 메세지 우선 순위**를 선택할 수 있습니다. **Syslog 메세지 우선 순위** 설정은 선택된 수준이나 그 이상에 있는 메세지가 **syslog**에 전송되는지를 표시합니다.

- **로그 파일에 있는 메세지 로그**를 클릭하면 로그 파일에 메세지가 활성화됩니다. 여기서 **로그 파일 경로** 이름을 지정할 수 있습니다. **로그 파일 메세지 우선 순위** 설정은 선택한 수준 또는 그 이상에 있는 메세지가 로그 파일에 기록되어 있음을 나타냅니다.

로깅 설정 (Logging Configuration) 페이지 하단에 있는 **특정 데몬 로깅 설정 덮어쓰기** 아래에 나열된 데몬 하나를 선택하여 특정 데몬에 대한 글로벌 로깅 설정을 덮어쓰기할 수 있습니다. 데몬을 선택한 후, 특정 데몬에 대한 디버깅 메시지를 기록했는지에 대한 여부를 확인할 수 있습니다. 또한 데몬에 대해 **syslog** 및 로그 파일 설정을 지정할 수 있습니다.

적용 (Apply)을 클릭하여 지정한 로깅 설정 변경 사항을 반영합니다.

3.6. 차단 (FENCE) 장치 설정

차단 (Fence) 장치 설정은 클러스터의 차단 장치 생성, 업데이트, 삭제로 이루어 집니다. 클러스터의 노드에 대해 차단 장치를 설정하기 전 클러스터에서 차단 장치를 설정해야 합니다.

차단 장치 생성은 차단 장치 유형 선택 및 차단 장치의 매개 변수 기입 (예: 이름, IP 주소, 로그인, 암호)으로 이루어 집니다. 차단 장치 업데이트는 기존 차단 장치 선택과 차단 장치의 매개 변수 변경으로 이루어 집니다. 차단 장치 삭제는 기존 차단 장치 선택 및 삭제로 이루어 집니다.

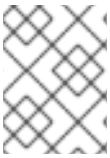
이 부분에서는 다음과 같은 작업에 대한 절차를 설명합니다:

- 차단 장치 생성 – 3.6.1절. “차단 장치 생성 ” 에서 참조하십시오. 차단 장치를 생성하고 이름을 지정한 후, 3.7절. “클러스터 멤버에 대한 차단 장치 설정 ” 에서 보여주듯이 클러스터에 있는 각 노드에 해당하는 차단 장치를 설정할 수 있습니다.

- 차단 장치 업데이트 – 3.6.2절. “차단 장치 수정 ” 에서 참조하십시오.

- 차단 장치 삭제 – 3.6.3절. “차단 장치 삭제 ” 에서 참조하십시오.

특정 클러스터 페이지에서 클러스터 보기 상단의 **차단 장치 (Fence Devices)** 를 클릭하여 클러스터의 차단 장치를 설정할 수 있습니다. 이렇게 하면 클러스터의 차단 장치가 표시되고 차단 장치 설정의 메뉴 항목 **추가 (Add)** 및 **삭제 (Delete)**가 표시됩니다. 이는 다음 부분에서 설명하는 각 단계의 출발점입니다.



참고

처음으로 클러스터를 설정하는 것일 경우, 차단 장치는 생성되지 않기 때문에 아무것도 나타나지 않습니다.

그림 3.5. “luci 차단 장치 설정 페이지 ” 에서는 차단 장치가 생성되기 전 차단 장치 설정 화면을 보여줍니다.

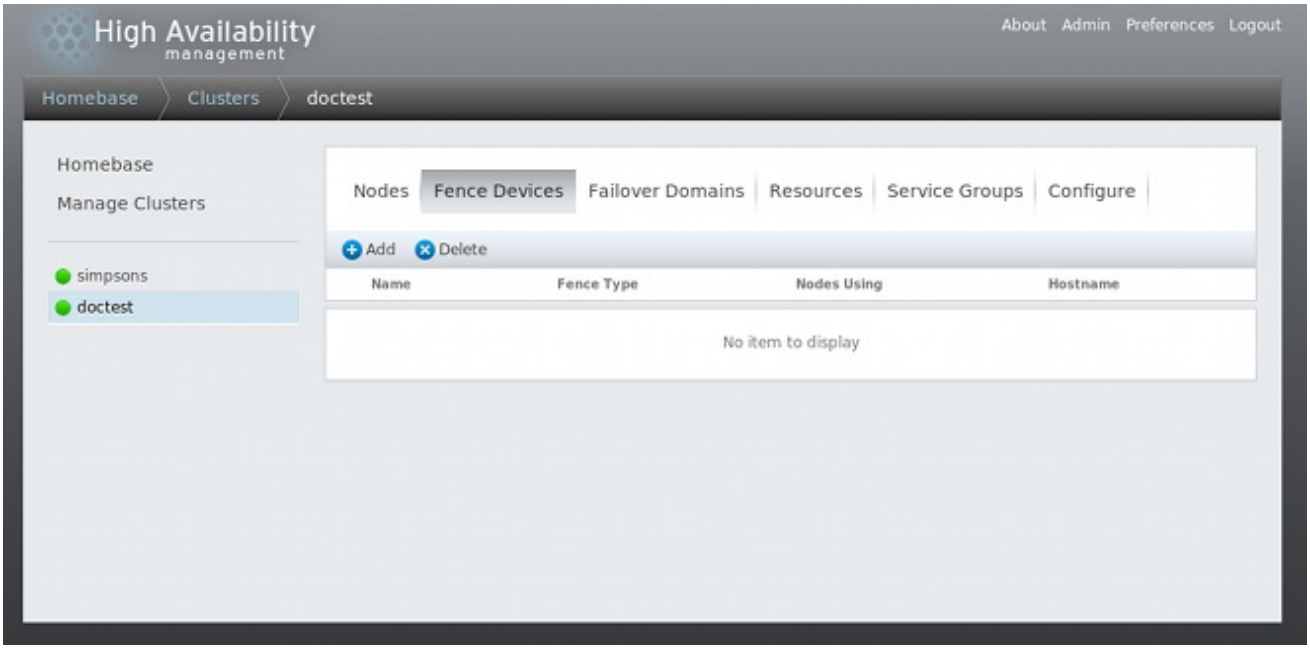


그림 3.5. luci 차단 장치 설정 페이지

3.6.1. 차단 장치 생성

차단 장치를 생성하려면 다음 절차를 따르십시오:

1. **차단 장치 (Fence Devices)** 설정 페이지에서 **추가 (Add)**를 클릭합니다. **추가 (Add)**를 클릭하면 **차단 장치 (인스턴스) 추가 (Add Fence Device [Instance])** 대화 상자가 나타납니다. 대화 상자에서 설정할 차단 장치 유형을 선택합니다.

2. 차단 장치 유형에 따라 **차단 장치 추가 (Add Fence Device [Instance])** 대화 상자의 정보를 지정합니다. 차단 장치 매개 변수에 대한 자세한 내용은 부록 A. *차단 장치 매개 변수*에서 참조하십시오. 일부 경우 3.7절. "클러스터 멤버에 대한 차단 장치 설정 " 에서 보여주듯이, 개별 노드에 대해 차단 장치를 설정할 때 차단 장치에 대해 추가 노드 특정 매개 변수를 지정해야 합니다.

3. **제출 (Submit)**을 클릭합니다.

차단 장치가 추가되면 이는 **차단 장치 (Fence Devices)** 설정 페이지에 나타납니다.

3.6.2. 차단 장치 수정

차단 장치를 수정하려면 다음 절차를 따르십시오:

1. **차단 장치 (Fence Devices)** 설정 페이지에서 수정할 차단 장치 이름을 클릭합니다. 그러면 장치에 설정된 값과 함께 차단 장치의 대화 상자가 나타납니다.

2. 차단 장치를 수정하려면, 표시된 매개 변수에 변경 사항을 입력합니다. 자세한 내용은 부록 A. *차단 장치 매개 변수* 에서 참조하십시오.

3. **적용 (Apply)**을 클릭하고 설정이 업데이트되는 것을 기다립니다.

3.6.3. 차단 장치 삭제

참고

사용 중인 차단 장치는 삭제할 수 없습니다. 현재 사용 중인 차단 장치를 삭제하려면 먼저 장치를 사용하는 노드의 차단 설정을 업데이트하여 해당 장치를 삭제합니다.

차단 장치를 삭제하려면 다음 절차를 따르십시오:

1. **차단 장치 (Fence Devices)** 설정 페이지에서 차단 장치 왼쪽에 있는 상자를 클릭하여 삭제하려는 장치를 선택합니다.
2. **삭제 (Delete)**를 클릭하여 업데이트될 설정을 기다립니다. 어떤 장치가 삭제될 지를 가리키는 메세지가 나타납니다.

설정이 업데이트되면 삭제된 차단 장치는 더이상 나타나지 않습니다.

3.7. 클러스터 멤버에 대한 차단 장치 설정

클러스터 생성 및 차단 장치 생성의 초기 단계를 완료한 후, 클러스터 노드의 차단 장치를 설정해야 합니다. 새로운 클러스터 생성 및 클러스터의 차단 장치 설정 후 노드의 차단 장치를 설정하려면 다음 부분에 있는 단계를 따르십시오. 클러스터에 있는 각 노드의 차단 장치를 설정해야 함에 유의하십시오.

다음 부분에서는 노드의 단일 차단 장치 설정, 백업 차단 장치가 있는 노드 설정 및 이중 전원 공급을 갖는 노드 설정에 대한 절차를 설명합니다:

- 3.7.1절. “노드에 대해 단일 차단 장치 설정 ”
- 3.7.2절. “백업 차단 장치 설정 ”
- 3.7.3절. “이중 전원 공급을 갖는 노드 설정 ”

3.7.1. 노드에 대해 단일 차단 장치 설정

다음 단계를 사용하여 단일 차단 장치가 있는 노드를 설정합니다.

1. 특정 클러스터 페이지에서 클러스터 보기 상단의 **노드 (Nodes)**를 클릭하여 클러스터에 있는 노드의 차단 장치를 설정할 수 있습니다. 이는 클러스터를 구성하는 노드를 표시합니다. 또한 이는 **luci 홈베이스 (Homebase)** 페이지의 왼쪽에 있는 메뉴에서 **클러스터 관리 (Manage Clusters)** 아래의 클러스터 이름을 클릭할 때 나타나는 기본 페이지입니다.
2. 노드 이름을 클릭합니다. 노드에 대한 링크를 클릭하면 해당 노드가 설정된 방법을 보여주는 링크된 페이지가 표시됩니다.

   특정 노드 페이지에서는 현재 노드에서 실행중인 서비스와 노드가 속해 있는 장애 조치 도메인을 표시합니다. 기존 장애 조치 도메인 이름을 클릭하여 이를 수정할 수 있습니다. 장애 조치 도메인 설정에 대한 자세한 내용은 3.8절. “장애 조치 도메인 설정 ” 에서 참조하십시오.
3. 특정 노드 페이지의 **차단 장치 (Fence Devices)** 아래에 있는 **차단 방식 추가 (Add Fence Method)**를 클릭합니다. **노드에 차단 방식 추가 (Add Fence Method to Node)** 대화 상자가 나타납니다.
4. 이 노드에 대해 설정하고 있는 차단 방식의 **방식 이름 (Method Name)**을 입력합니다. 이는 Red Hat 고가용성 추가 기능에 의해 사용되는 임의의 이름으로 장치의 DNS 이름과 동일하지 않습니다.

5. **제출 (Submit)**을 클릭합니다. 이는 **차단 장치 (Fence Devices)** 아래에 추가한 방식을 표시하는 특정 노드 화면을 표시합니다.

6. 차단 방식 아래에 나타나는 **차단 인스턴스 추가 (Add Fence Instance)**를 클릭하여 이 방식에 대한 차단 인스턴스를 설정합니다. 이는 3.6.1절. "차단 장치 생성 "에서 설명하고 있듯이 이전에 설정된 차단 장치를 선택할 수 있는 **차단 장치 추가 (Add Fence Device (Instance))** 드롭 다운 메뉴를 표시합니다.

7. 이러한 방식의 차단 장치를 선택합니다. 이 차단 장치가 특정 노드의 매개 변수 설정을 필요로 하는 경우, 설정해야 할 매개변수가 나타납니다. 차단 장치 매개 변수에 대한 자세한 내용은 부록 A. *차단 장치 매개 변수*에서 참조하십시오.

참고

비전원 (non-power) 차단 방식 (즉, SAN/스토리지 펜싱)의 경우, 특정 노드 매개 변수 표시에서 기본값으로 **Unfencing (펜싱 제거)**이 선택됩니다. 이는 노드가 다시 시작될 때 까지 스토리지로 차단된 노드의 액세스가 다시 활성화되지 않는지 확인합니다. 노드 펜싱 제거 (unfencing)에 대한 자세한 내용은 **fence_node**(8) man 페이지에서 참조하십시오.

8. **제출 (Submit)**을 클릭합니다. 클릭하면 차단 방식과 차단 인스턴스가 표시된 특정 노드 화면으로 돌아갑니다.

3.7.2. 백업 차단 장치 설정

한 노드에 여러 차단 방식을 정의할 수 있습니다. 첫 번째 방식을 사용하여 차단이 실패하면, 시스템은 두 번째 방식을 사용하여 노드를 차단하려 합니다. 그 후 설정한 추가 방식으로 계속 시도합니다.

다음 절차를 사용하여 노드에 백업 차단 장치를 설정합니다.

1. 3.7.1절. "노드에 대해 단일 차단 장치 설정 "에 제공된 절차를 사용하여 노드에 주요 차단 방식을 설정합니다.

2. 정의된 주요 방식 보기 아래에 있는 **차단 방식 추가 (Add Fence Method)**를 클릭합니다.

3. 이 노드에 설정된 백업 차단 방식의 이름을 입력하고 **제출 (Submit)**을 클릭합니다. 이는 주요 차단 방식 아래에 추가한 방식을 보여주는 특정 노드 화면을 표시합니다.

4. **차단 인스턴스 추가 (Add Fence Instance)**를 클릭하여 이 방식에 대한 차단 인스턴스를 설정합니다. 클릭하면 3.6.1절. "차단 장치 생성 "에서 설명하고 있듯이 이전에 설정한 차단 장치를 선택할 수 있는 드롭 다운 메뉴가 나타납니다.

5. 이러한 방식의 차단 장치를 선택합니다. 이 차단 장치가 특정 노드의 매개 변수 설정을 필요로 하는 경우, 설정해야 할 매개변수가 나타납니다. 차단 장치 매개 변수에 대한 자세한 내용은 부록 A. *차단 장치 매개 변수*에서 참조하십시오.

6. **제출 (Submit)**을 클릭합니다. 클릭하면 차단 방식과 차단 인스턴스가 표시된 특정 노드 화면으로 돌아갑니다.

필요에 따라 차단 방식 추가를 계속 진행할 수 있습니다. **위로 이동 (Move Up)** 및 **아래로 이동 (Move Down)**을 클릭하여 노드에 사용되는 차단 방식의 순서를 재구성할 수 있습니다.

3.7.3. 이중 전원 공급을 갖는 노드 설정

클러스터가 노드에 대해 이중 전원 공급으로 설정되어 있는 경우, 차단 장치가 설정되어 있는지 확인하여

노드가 차단되어야 할 때 노드를 완전히 종료하도록 합니다. 각 전원 공급을 개별 차단 방식으로 설정하려면, 각각의 전원 공급은 개별적으로 차단됩니다; 첫 번째 전원 공급이 차단되었을 때 두 번째 전원 공급으로 시스템을 계속 실행하게 하여 시스템이 완전히 차단되지 않게 합니다. 이중 전원 공급으로 시스템을 설정하려면, 차단 장치를 설정하여 두 전원 공급 모두를 종료하고 시스템을 완전히 종료해야 합니다. **Conga**를 사용하여 시스템을 설정할 때, 단일 차단 방식에서 두 인스턴스를 설정해야 합니다.

이중 전원 공급을 갖는 노드에 대한 차단 장치를 설정하려면 다음 부분에 있는 절차를 따릅니다.

1. 이중 전원 공급을 갖는 노드에 대한 차단 장치를 설정하기 전 클러스터의 차단 장치로 각 전원 스위치를 설정해야 합니다. 차단 장치 설정에 대한 자세한 내용은 3.6절. “차단 (Fence) 장치 설정 ”에서 참조하십시오.

2. 특정 클러스터 페이지에서 클러스터 보기의 상단의 **노드 (Nodes)**를 클릭하면 클러스터를 구성하는 노드가 나타납니다. 이는 **luci 홈베이스 (Homebase)** 페이지의 왼쪽에 있는 메뉴에서 **클러스터 관리 (Manage Clusters)** 아래의 클러스터 이름을 클릭할 때 나타나는 기본 페이지입니다.

3. 노드 이름을 클릭합니다. 노드에 대한 링크를 클릭하면 해당 노드가 설정된 방법을 보여주는 링크된 페이지가 표시됩니다.

4. 특정 노드 페이지에서 **차단 방식 추가 (Add Fence Method)**를 클릭합니다.

5. 이 노드에 설정하고 있는 차단 방식의 이름을 입력합니다.

6. **제출 (Submit)**을 클릭합니다. 이는 **차단 장치 (Fence Devices)** 아래에 추가한 방식을 표시하는 특정 노드 화면을 표시합니다.

7. **차단 인스턴스 추가 (Add Fence Instance)**를 클릭하여 이 방식에 대한 차단 인스턴스로 첫 번째 전원 공급을 설정합니다. 그러면 3.6.1절. “차단 장치 생성 ” 에서 설명하고 있듯이 이전에 설정한 전원 차단 장치 중 하나를 선택할 수 있는 드롭 다운 메뉴가 표시됩니다.

8. 이 방식에 대한 전원 차단 장치 중 하나를 선택하여 이 장치에 대한 적절한 매개 변수를 입력합니다.

9. **제출 (Submit)**을 클릭합니다. 클릭하면 차단 방식과 차단 인스턴스가 표시된 특정 노드 화면으로 돌아갑니다.

10. 첫 번째 전원 차단 장치를 설정한 것과 동일한 차단 방식 아래에서 **차단 인스턴스 추가 (Add Fence Instance)**를 클릭합니다. 이는 3.6.1절. “차단 장치 생성 ” 에서 설명하고 있듯이 이전에 설정한 두 번째 전원 차단 장치를 선택할 수 있는 드롭 다운 메뉴를 표시합니다.

11. 이 방식의 두 번째 전원 차단 장치를 선택하고 이 장치에 대한 적절한 매개 변수를 입력합니다.

12. **제출 (Submit)**을 클릭합니다. 이는 차단 방식과 차단 인스턴스를 표시하는 특정 노드 화면으로 돌아가 각 장치가 연속으로 시스템 전원을 끄고 연속으로 시스템 전원을 켜는 것을 보여줍니다. 이는 그림 3.6. “듀얼 전원 차단 장치 설정 ” 에서 보여주고 있습니다.

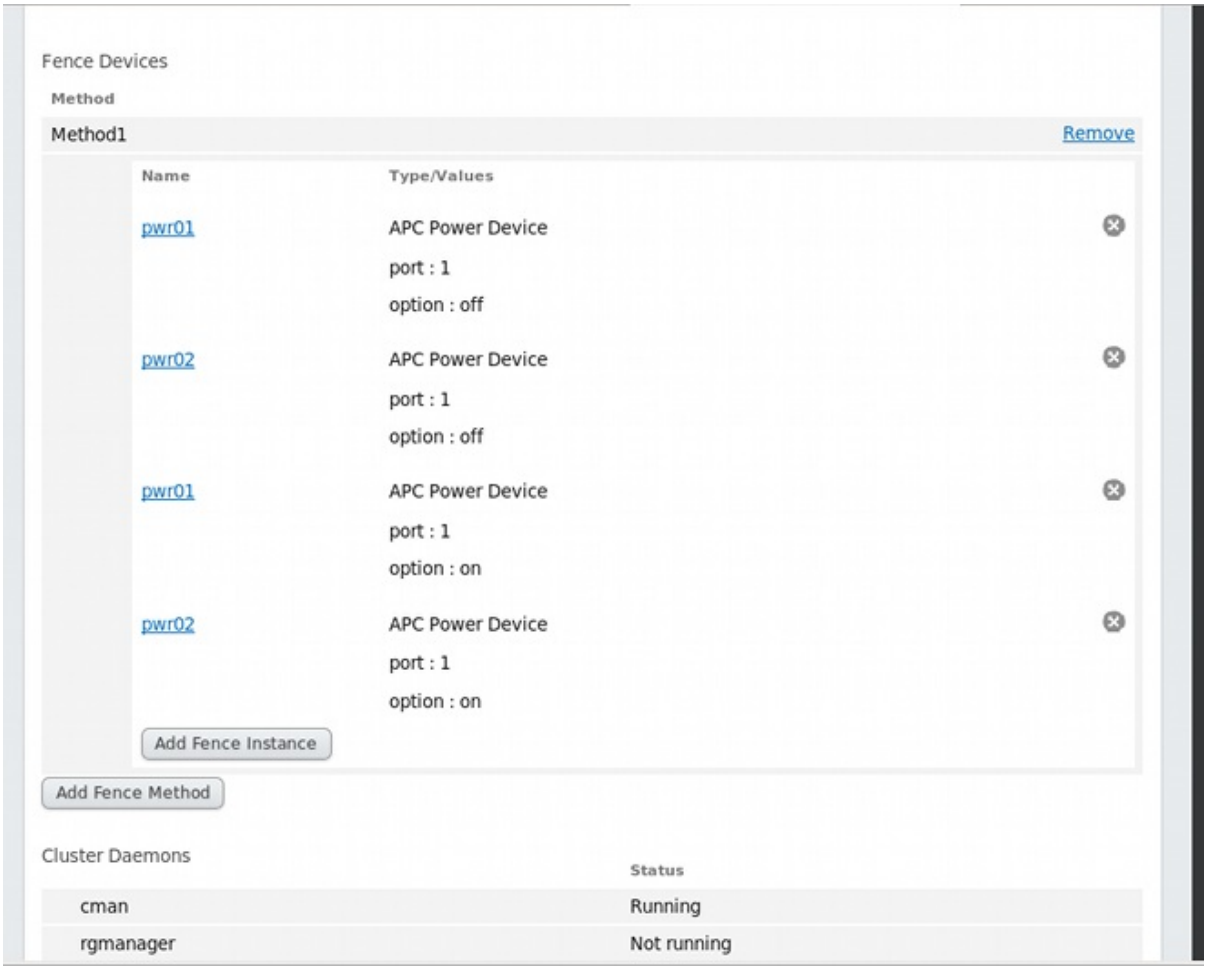
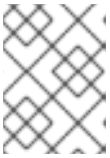


그림 3.6. 듀얼 전원 차단 장치 설정

3.8. 장애 조치 도메인 설정

장애 조치 도메인은 노드 장애 발생 시 클러스터 서비스를 실행할 수 있는 클러스터 노드의 이름이 지정된 하위 집합입니다. 장애 조치 도메인은 다음과 같은 특징을 갖을 수 있습니다:

- 제한 없음 (Unrestricted) – 우선 순위를 갖는 멤버의 하위 집합을 지정할 수 있게 합니다. 하지만 이러한 도메인에 할당된 클러스터 서비스는 사용 가능한 멤버에서 실행할 수 있습니다.
- 제한 (Restricted) – 특정 클러스터 서비스를 실행할 수 있는 멤버를 제한할 수 있게 합니다. 제한된 장애 조치 도메인에 있는 사용 가능한 멤버가 없을 경우, 클러스터 서비스는 (수동으로 또는 클러스터 소프트웨어로도) 시작할 수 없습니다.
- 우선 순위 없음 (Unordered) – 클러스터 서비스가 우선 순위가 없는 장애 조치 도메인에 할당될 때 클러스터 서비스를 실행할 멤버는 우선 순위가 없는 사용 가능한 장애 조치 도메인 멤버에서 선택됩니다.
- 순서 지정 (Ordered) – 장애 조치 도메인의 멤버 사이에서 우선 순위를 지정할 수 있게 합니다. 목록 상단에 있는 멤버가 최상의 우선 순위에 있고 목록의 두 번째 멤버가 그 다음의 우선 순위에 있게 됩니다.
- 장애 복구 (Failback) – 장애 조치 도메인의 서비스가 노드 장애 이전 원래 실행하고 있는 노드로 장애 복구할지에 대한 여부를 지정할 수 있게 합니다. 이러한 기능 설정은 노드가 반복적으로 실패하고 이것이 우선 순위를 갖는 장애 조치 도메인의 일부분인 경우에 유용합니다. 이러한 상황에서 노드가 장애 조치 도메인에 있는 우선 순위를 갖는 노드일 경우, 서비스를 장애 조치하여 우선 순위를 갖는 노드와 다른 노드 사이에서 반복적으로 장애 복구할 가능성이 있으므로 이는 성능에 심각한 영향을 미칠 수 있습니다.

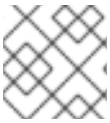


참고

우선 순위가 지정된 장애 조치가 설정된 경우에만 장애 복구 기능을 사용할 수 있습니다.

참고

장애 조치 도메인 설정 변경은 현재 실행 중인 서비스에 영향을 주지 않습니다.

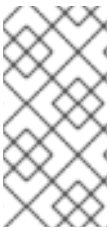


참고

장애 조치 도메인은 운용에 필요하지 *않습니다*.

기본값으로 장애 조치 도메인은 제한이 없고 우선 순위가 없습니다.

여러 멤버를 갖는 클러스터에서 제한된 장애 조치 도메인을 사용하면 클러스터 서비스 (예: **httpd**) 실행을 위한 클러스터 설치 작업을 최소화할 수 있습니다. 이때 여기서 클러스터 서비스를 실행하는 모든 멤버에서 동일한 설정을 구성해야 합니다. 클러스터 서비스를 실행하기 위해 전체 클러스터를 설정하는 대신 클러스터 서비스와 관련된 제한된 장애조치 도메인에 있는 멤버만을 설정할 수 있습니다.



참고

우선 순위를 갖는 멤버를 설정하려면, 하나의 클러스터 멤버로 이루어진 제한이 없는 장애 조치 도메인을 생성할 수 있습니다. 이렇게 할 경우 클러스터 서비스가 주로 클러스터 멤버 (우선 순위를 갖는 멤버)에서 실행되지만 클러스터 서비스가 다른 멤버라도 장애 조치할 수 있게 합니다.

다음 부분에서는 장애 조치 도메인의 추가, 수정, 삭제에 대해 설명합니다:

- 3.8.1절. “장애 조치 도메인 추가 ”
- 3.8.2절. “장애 조치 도메인 수정 ”
- 3.8.3절. “장애 조치 도메인 삭제 ”

3.8.1. 장애 조치 도메인 추가

장애 조치 도메인을 추가하려면 다음 절차를 따르십시오:

1. 특정 클러스터 페이지에서 클러스터 보기 상단의 **장애 조치 도메인 (Failover Domains)** 을 클릭하여 클러스터의 장애 조치 도메인을 설정할 수 있습니다. 이는 클러스터에 설정된 장애 조치 도메인을 표시합니다.
2. **추가 (Add)**를 클릭합니다. **추가 (Add)**를 클릭하면 그림 3.7. “luci 장애 조치 도메인 설정 대화 상자 ”에서 보여주듯이 **클러스터에 장애 조치 도메인 추가 (Add Failover Domain to Cluster)** 창이 나타납니다.

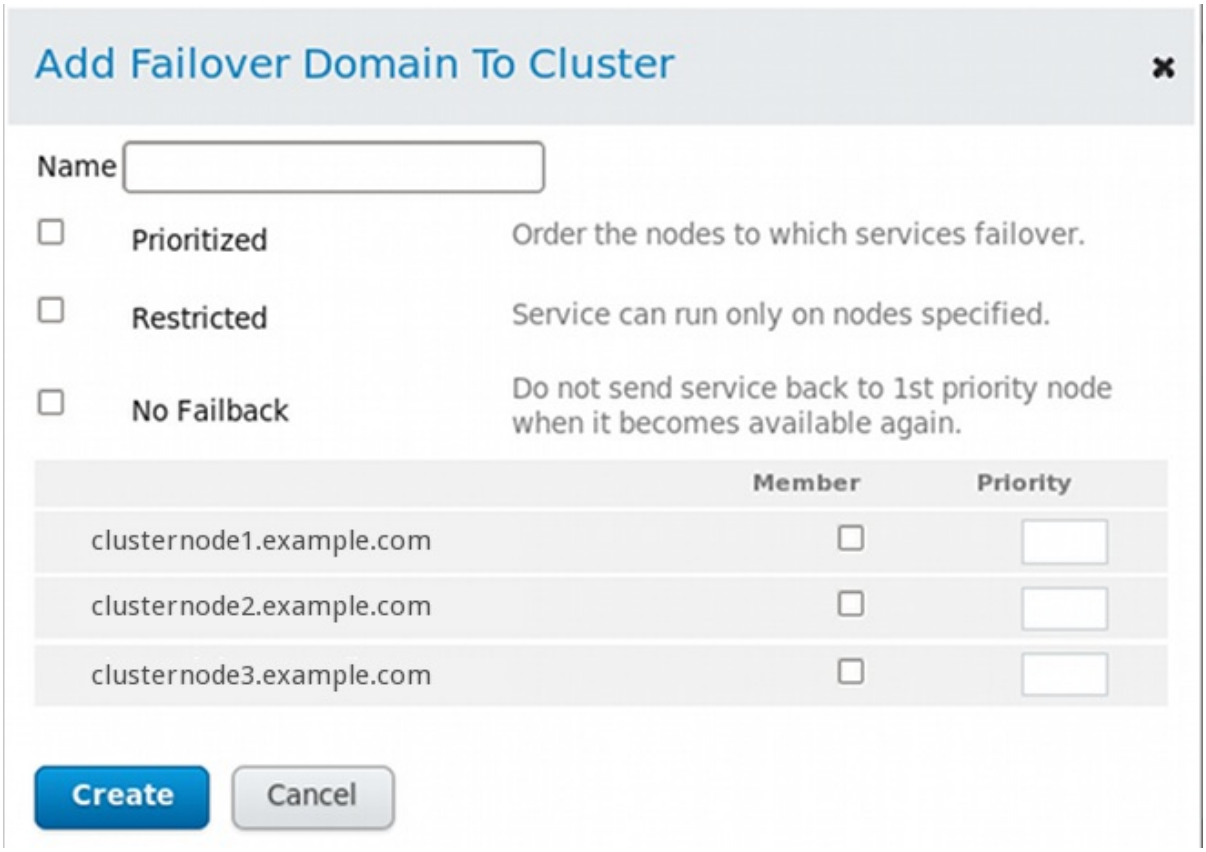


그림 3.7. luci 장애 조치 도메인 설정 대화 상자

3. **클러스터에 장애 조치 도메인 추가 (Add Failover Domain to Cluster)** 대화 상자에서 **이름 (Name)** 텍스트 상자에 있는 장애 조치 도메인 이름을 지정합니다.

참고

이름은 클러스터에 사용하는 다른 이름과 비교해 그 목적을 구별할 수 있도록 설명적인 이름이어야 합니다.

4. 장애 조치 도메인에 있는 멤버의 장애 조치 우선 순위 설정을 활성화하려면 **우선 순위 설정 (Prioritized)** 체크 박스를 클릭합니다. **우선 순위 (Prioritized)** 선택하여 장애 조치 도메인의 멤버로서 선택된 각 노드의 우선 순위 값 **Priority**를 설정할 수 있습니다.

5. 장애 조치 도메인에 있는 멤버에게 장애 조치를 제한하려면 **제한 설정 (Restricted)** 체크 박스를 클릭합니다. **제한 설정 (Restricted)** 체크 박스를 클릭하면 장애 조치 도메인에 할당된 서비스가 장애 조치 도메인에 있는 노드에서만 장애 조치합니다.

6. 노드가 장애 조치 도메인에 장애 복구하지 않도록 지정하려면, **장애 복구 없음 (No Failback)** 체크 박스를 클릭합니다. **장애 복구 없음 (No Failback)** 체크 박스를 클릭하여 서비스가 우선 순위에 있는 노드에서 장애 조치될 경우 서비스가 복구된 후 원래 노드로 장애 복구되지 않습니다.

7. 장애 조치 도메인의 멤버를 설정합니다. 장애 조치 도메인의 멤버가 될 각 노드의 **멤버 (Member)** 체크 박스를 클릭합니다. **우선 순위 설정 (Prioritized)**이 선택되어 있을 경우, 장애 조치 도메인의 각 멤버에 대한 **우선 순위 (Priority)** 텍스트 상자에서 우선 순위를 설정합니다.

8. **생성 (Create)**을 클릭합니다. 그러면 새로 생성된 장애 조치 도메인과 함께 **장애 조치 도메인 (Failover Domains)** 페이지가 나타납니다. 새로운 도메인이 생성되고 있다는 메세지가 나타납니다. 업데이트된 상태를 확인하려면 페이지를 새로 고침합니다.

3.8.2. 장애 조치 도메인 수정

장애 조치 도메인을 수정하려면 다음 절차를 따르십시오.

1. 특정 클러스터 페이지에서 클러스터 보기 상단의 **장애 조치 도메인 (Failover Domains)**을 클릭하여 클러스터의 장애 조치 도메인을 설정할 수 있습니다. 이렇게 하면 클러스터에 설정된 장애 조치 도메인을 표시할 수 있습니다.

2. 장애 조치 도메인 이름을 클릭합니다. 그러면 장애 조치 도메인에 대한 설정 페이지가 표시됩니다.

3. 장애 조치 도메인의 **우선 순위 설정 (Prioritized)**, **제한 설정 (Restricted)**, **장애 복구 없음 (No Failback)** 정보를 수정하려면, 등록 정보 옆에 있는 체크박스를 선택 또는 선택 해제한 후 **등록 정보 업데이트 (Update Properties)**를 클릭합니다.

4. 장애 조치 도메인 멤버쉽을 수정하려면, 클러스터 멤버 옆에 있는 체크 박스를 선택 또는 선택 해제합니다. 장애 조치 도메인의 우선 순위가 설정되어 있으며 클러스터 멤버의 우선 순위 설정도 수정할 수 있습니다. 그리고 **설정 업데이트 (Update Settings)**를 클릭합니다.

3.8.3. 장애 조치 도메인 삭제

장애 조치 도메인을 삭제하려면 다음 절차를 따르십시오.

1. 특정 클러스터 페이지에서 클러스터 보기 상단의 **장애 조치 도메인 (Failover Domains)**을 클릭하여 클러스터의 장애 조치 도메인을 설정할 수 있습니다. 이렇게 하면 클러스터에 설정된 장애 조치 도메인을 표시할 수 있습니다.

2. 삭제할 장애 조치 도메인의 체크 박스를 선택합니다.

3. **삭제 (Delete)**를 클릭합니다.

3.9. 글로벌 클러스터 리소스 설정

클러스터에서 실행하고 있는 서비스 중에서 사용할 수 있는 글로벌 리소스를 설정할 수 있으며 특정 서비스에서만 사용 가능한 리소스를 설정할 수 있습니다.

글로벌 클러스터 리소스를 추가하려면, 다음의 단계를 따르십시오. 3.10절. “클러스터에 클러스터 서비스 추가 ”에서 설명하고 있듯이 서비스를 설정할 때 특정 서비스에 로컬 리소스를 추가할 수 있습니다.

1. 특정 클러스터 페이지에서 클러스터 보기 상단의 **리소스 (Resources)**를 클릭하여 클러스터에 리소스를 추가할 수 있습니다. 그러면 클러스터에 설정된 리소스가 표시됩니다.

2. **추가 (Add)**를 클릭합니다. 그러면 **클러스터에 리소스 추가 (Add Resource to Cluster)** 드롭 다운 메뉴가 나타납니다.

3. **클러스터에 리소스 추가 (Add Resource to Cluster)** 아래의 드롭 다운 상자를 클릭하여 설정할 리소스 유형을 선택합니다.

4. 추가하려는 리소스의 리소스 매개 변수를 입력합니다. 부록 B. *HA 리소스 매개 변수*에서는 리소스 매개 변수에 대해 설명합니다.

5. **제출 (Submit)**을 클릭합니다. **제출 (Submit)**을 클릭하면 **리소스 (Resources)** 보기를 보여주는 리소스 페이지로 돌아갑니다. 이는 추가된 리소스 (및 기타 다른 리소스)를 표시합니다.

기존 리소스를 수정하려면 다음 절차를 실행하십시오.

1. **luci 리소스 (Resources)** 페이지에서 수정할 리소스 이름을 클릭합니다. 그러면 해당 리소스의 매개 변수가 나타납니다.
2. 리소스 매개 변수를 편집합니다.
3. **적용 (Apply)**을 클릭합니다.

기존 리소스를 삭제하려면 다음 절차를 실행합니다.

1. **luci 리소스 (Resources)** 페이지에서 삭제할 리소스의 체크 박스를 클릭합니다.
2. **삭제 (Delete)**를 클릭합니다.

3.10. 클러스터에 클러스터 서비스 추가

클러스터에 클러스터 서비스를 추가하려면, 다음 절차를 따르십시오.

1. 특정 클러스터 페이지에서 클러스터 보기 상단의 **서비스 그룹 (Service Groups)**을 클릭하여 클러스터에 서비스를 추가할 수 있습니다. 이렇게 하면 클러스터에 설정된 서비스를 표시할 수 있습니다. (4.5절. "고가용성 서비스 관리 "에서 설명하고 있듯이 **서비스 그룹 (Service Groups)** 페이지에서 서비스를 시작, 다시 시작, 비활성화할 수 있습니다.)
2. **추가 (Add)**를 클릭합니다. 이는 **클러스터에 서비스 그룹 추가 (Add Service Group to Cluster)** 대화 상자를 표시합니다.
3. **클러스터에 서비스 그룹 추가 (Add Service Group to Cluster)** 대화 상자에서 **서비스 이름 (Service name)** 텍스트 상자에 서비스 이름을 입력합니다.

참고

클러스터의 다른 서비스와 명백하게 구별할 수 있도록 설명적인 이름을 사용합니다.

4. 클러스터가 시작되어 실행될 때 서비스가 자동으로 시작하게 하려면 **자동으로 서비스 시작 (Automatically Start This Service)** 체크 박스를 선택합니다. 체크 박스가 선택되어 있지 *않을*경우 정지 상태에서 클러스터가 나타나는 때에 수동으로 서비스를 시작해야 합니다.
5. **배타적 실행 (Run Exclusive)** 체크 박스를 선택하여 다른 서비스를 실행하지 않는 노드에서만 서비스를 실행하는 정책을 설정합니다.
6. 클러스터에 장애 조치 도메인을 설정한 경우, **장애 조치 도메인 (Failover Domain)** 매개 변수의 드롭 다운 메뉴를 사용하여 서비스에 장애 조치 도메인을 선택할 수 있습니다. 장애 조치 도메인 설정에 대한 자세한 내용은 3.8절. "장애 조치 도메인 설정 "에서 참조하십시오.
7. **복구 정책 (Recovery Policy)** 드롭 다운 상자를 사용하여 서비스의 복구 정책을 선택합니다. 옵션에는 **재배치 (Relocate)**, **다시 시작 (Restart)**, **다시 시작-비활성화 (Restart-Disable)**, **비활성화 (Disable)**가 있습니다.

   다시 시작 (Restart) 옵션을 선택하면 시스템이 서비스를 재배치하기 전에 실패한 서비스를 다시 시작 시도해야 함을 가리킵니다. **재배치 (Relocate)** 옵션을 선택한 경우 시스템은 다른 노드에서 서비스를 다시 시작 시도해야 함을 가리킵니다. **비활성화 (Disable)** 옵션을 선택한 경우 시스템 구성 요소에 장애가 발생하면 리소스 그룹을 비활성화해야 함을 가리킵니다. **다시 시작-비활성화 (Restart-Disable)** 옵션을 선택하면 시스템이 실패한 경우 그 자리에서 시스템이 서비스를 다시 시작 시도해야 함을 가리킵니다. 하지만 서비스 다시 시작을 실패한 경우, 서비스는 클러스터에 있는 다른 호스트로 이동하지 않고 비활성화됩니다.

서비스의 복구 정책으로 **다시 시작 (Restart)** 또는 **다시 시작-비활성화 (Restart-Disable)**를 선택한 경우, 서비스를 이동 또는 비활성화하기 전 까지 다시 시작 실패의 최대 횟수를 지정할 수 있으며 다시 시작을 잊어버린 후 시간을 초 단위로 지정할 수 있습니다.

8. 서비스에 리소스를 추가하려면, **리소스 추가 (Add Resource)**를 클릭합니다. **리소스 추가 (Add Resource)**를 클릭하면 **서비스에 리소스 추가 (Add Resource To Service)** 드롭 다운 상자가 나타나 기존 글로벌 리소스를 추가하거나 *서비스에서만* 사용할 수 있는 새로운 리소스를 추가할 수 있습니다.

   - 기존 글로벌 리소스를 추가하려면, **서비스에 리소스 추가 (Add Resource To Service)** 드롭 다운 상자에서 기존 리소스 이름을 클릭합니다. 그러면 설정한 서비스에 대한 **서비스 그룹 (Service Groups)** 페이지에 리소스와 매개 변수가 표시됩니다. 글로벌 리소스 추가 및 수정에 대한 자세한 내용은 3.9절. “글로벌 클러스터 리소스 설정 ” 에서 참조하십시오.

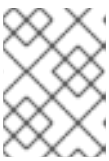
   - 서비스에서만 사용할 수 있는 새로운 리소스를 추가하려면, **서비스에 리소스 추가 (Add Resource To Service)** 드롭 다운 상자에서 설정할 리소스 유형을 선택하고 추가하려는 리소스에 대한 리소스 매개 변수를 입력합니다. 부록 B. *HA 리소스 매개 변수*에서는 리소스 매개 변수에 대해 설명합니다.

   - 리소스를 서비스에 추가할 때, 이것이 기존 글로벌 리소스인지 또는 서비스에서만 사용 가능한 리소스 인지를 지정할 수 있으며 리소스가 **독립 하위 트리 (Independent Subtree)** 또는 **중요하지 않은 리소스 (Non-Critical Resource)**인지를 지정할 수 있습니다.

     리소스가 독립 하위 트리라고 지정하여 리소스가 실패할 경우, 시스템이 정상적으로 복구 시도하기 전 (서비스 전체가 아닌) 해당 리소스만이 다시 시작됩니다. 서비스에 대한 복구 정책을 구현하기 전 노드의 리소스를 다시 시작 시도하는 최대 횟수를 지정할 수 있습니다. 또한 시스템이 서비스에 대한 복구 정책을 구현한 후 시간을 초단위로 지정할 수 있습니다.

     리소스가 중요하지 않은 리소스라고 지정하고 이러한 리소스가 실패할 경우 해당 리소스만이 다시 시작됩니다. 또한 리소스가 계속해서 실패하면 서비스 전체가 아닌 해당 리소스만 비활성화됩니다. 리소스를 비활성화하기 전 노드에 있는 리소스에 대해 다시 시작을 시도하는 최대 횟수를 지정할 수 있습니다. 또한 시스템이 리소스를 비활성화한 후 시간을 초 단위로 지정할 수 있습니다.

9. 정의된 리소스에 자식 리소스를 추가하려면 **자식 리소스 추가 (Add Child Resource)**를 클릭합니다. **자식 리소스 추가 (Add Child Resource)**를 클릭하면 **서비스에 리소스 추가 (Add Resource To Service)** 드롭 다운 상자가 나타납니다. 여기서 기존 글로벌 리소스를 추가하거나 서비스에서만 사용할 수 있는 새로운 리소스를 추가할 수 있습니다. 사용자 요건에 맞게 리소스에 자식 리소스를 계속 추가할 수 있습니다.



참고

Samba 서비스 리소스를 추가하는 경우, 다른 자식 리소스로서가 *아닌* 이를 서비스에 직접 추가합니다.

10. 서비스에 리소스를 추가 완료하고 리소스에 자식 리소스를 추가 완료한 후, **제출 (Submit)**을 클릭합니다. **제출 (Submit)**을 클릭하면 추가된 서비스 (및 다른 서비스)가 나타나는 **서비스 그룹 (Service Groups)** 페이지로 돌아갑니다.

참고

클러스터 서비스에서 사용하는 IP 서비스 리소스가 있는지 확인하려면 클러스터 노드에서 (폐지된 **ifconfig** 명령이 아니라) **/sbin/ip addr show** 명령을 사용할 수 있습니다. 다음은 클러스터 서비스가 실행되고 있는 노드에서 **/sbin/ip addr show** 명령을 실행하였을 경우의 출력 결과를 보여줍니다:

```
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1356 qdisc pfifo_fast
qlen 1000
    link/ether 00:05:5d:9a:d8:91 brd ff:ff:ff:ff:ff:ff
    inet 10.11.4.31/22 brd 10.11.7.255 scope global eth0
    inet6 fe80::205:5dff:fe9a:d891/64 scope link
    inet 10.11.4.240/22 scope global secondary eth0
       valid_lft forever preferred_lft forever
```

기존 서비스를 수정하려면 다음의 절차를 실행하십시오.

1. **서비스 그룹 (Service Groups)** 대화 상자에서 수정할 서비스 이름을 클릭합니다. 그러면 서비스에 설정되는 매개 변수와 리소스가 나타납니다.
2. 서비스 매개 변수를 편집합니다.
3. **제출 (Submit)**을 클릭합니다.

하나 이상의 기존 서비스를 삭제하려면 다음 절차를 실행합니다.

1. **luci 서비스 그룹 (Service Groups)** 페이지에서 삭제할 서비스의 체크 박스를 클릭합니다.
2. **삭제 (Delete)**를 클릭합니다.
3. Red Hat Enterprise Linux 6.3 이후 **luci**가 서비스를 삭제하기 전 삭제하고자 하는 서비스 그룹 또는 그룹을 확인하는 메세지가 나타나고 서비스를 구성하는 리소스를 중지합니다. 서비스를 삭제하지 않고 대화 상자를 닫으려면 **취소 (Cancel)**를 클릭합니다. 또는 **계속 진행 (Proceed)**을 클릭하여 선택한 서비스를 제거합니다.

4장. CONGA를 사용하여 RED HAT 고가용성 추가 기능 관리

다음 부분에서는 Red Hat 고가용성 추가 기능 관리를 위한 다양한 관리 작업을 설명하고 있으며 이는 다음과 같은 부분으로 구성되어 있습니다:

- 4.1절. “기존 클러스터를 luci 인터페이스에 추가 ”
- 4.2절. “luci 인터페이스에서 클러스터 삭제”
- 4.3절. “클러스터 노드 관리 ”
- 4.4절. “클러스터 시작, 중지, 다시 시작, 삭제 ”
- 4.5절. “고가용성 서비스 관리 ”
- 4.6절. “luci 설정 백업 및 복구”

4.1. 기존 클러스터를 LUCI 인터페이스에 추가

고가용성 추가 기능 클러스터를 이전에 생성한 경우 해당 클러스터를 **luci** 인터페이스에 쉽게 추가할 수 있으므로 **Conga**를 사용하여 클러스터를 관리할 수 있습니다.

기존 클러스터를 **luci** 인터페이스에 추가하려면, 다음 절차를 따르십시오:

1. **luci Homebase** 페이지의 왼쪽에 있는 메뉴에서 **클러스터 관리 (Manage Clusters)**를 클릭하면 **클러스터 (Clusters)** 화면이 나타납니다.
2. **추가 (Add)**를 클릭합니다. **기존 클러스터 추가 (Add an Existing Cluster)** 화면이 나타납니다.
3. 기존 클러스터에 있는 노드의 호스트 이름과 **ricci** 암호를 입력합니다. 클러스터에 있는 각 노드에는 클러스터에 대한 모든 설정 정보가 들어 있으므로 이는 **luci** 인터페이스에 클러스터를 추가하기 위해 충분한 정보를 제공합니다.
4. **연결 (Connect)**을 클릭합니다. **기존 클러스터 추가 (Add an Existing Cluster)** 화면에서 클러스터 이름과 클러스터에 있는 남아있는 노드가 나타납니다.
5. 클러스터에 있는 각 노드에 대해 별도의 **ricci** 암호를 입력하거나 하나의 암호를 입력하고 **모든 노드에 동일한 암호를 사용 (Use the same password for all nodes)**을 선택합니다.
6. **클러스터 추가 (Add Cluster)**를 클릭합니다. 이전에 설정된 클러스터가 **클러스터 관리 (Manage Clusters)** 화면에 나타납니다.

4.2. LUCI 인터페이스에서 클러스터 삭제

클러스터 서비스와 클러스터 멤버쉽에 영향을 주지 않고 **luci** 관리 GUI에서 클러스터를 삭제할 수 있습니다. 클러스터를 삭제하면 나중에 해당 클러스터를 다시 추가하거나 4.1절. “기존 클러스터를 luci 인터페이스에 추가 ”에서 설명하고 있듯이 다른 **luci** 인스턴스에 이를 추가할 수 있습니다.

클러스터 서비스 또는 클러스터 멤버쉽에 영향을 주지 않고 **luci** 관리 GUI에서 클러스터를 삭제하려면 다음 절차를 따르십시오:

1. **luci Homebase** 페이지의 왼쪽에 있는 메뉴에서 **클러스터 관리 (Manage Clusters)**를 클릭하면 **클러스터 (Clusters)** 화면이 나타납니다.
2. 삭제하고자 하는 클러스터를 선택합니다.

3. **삭제 (Remove)**를 클릭합니다.

전체 클러스터를 삭제하는 방법, 모든 클러스터 서비스를 중지하는 방법, 노드에서 클러스터 설정 정보를 제거하는 방법은 4.4절. "클러스터 시작, 중지, 다시 시작, 삭제 "에서 참조하십시오.

4.3. 클러스터 노드 관리

다음 부분에서는 **Conga**의 **luci** 서버 연결을 통해 다음과 같은 노드 관리 기능을 실행하는 방법에 대해 설명합니다:

- 4.3.1절. "클러스터 노드 재부팅 "
- 4.3.2절. "노드가 클러스터를 탈퇴 또는 참여하는 원인 "
- 4.3.3절. "실행중인 클러스터에 멤버 추가 "
- 4.3.4절. "클러스터에서 멤버 삭제 "

4.3.1. 클러스터 노드 재부팅

클러스터에 있는 노드를 재부팅하려면 다음 절차를 실행합니다:

1. 특정 클러스터 페이지에서 클러스터 보기의 상단의 **노드 (Nodes)**를 클릭하면 클러스터를 구성하는 노드가 나타납니다. 이는 **luci 홈베이스 (Homebase)** 페이지의 왼쪽에 있는 메뉴에서 **클러스터 관리 (Manage Clusters)** 아래의 클러스터 이름을 클릭할 때 나타나는 기본 페이지입니다.
2. 해당 노드의 체크 박스를 클릭하여 재부팅하기 위해 노드를 선택합니다.
3. 페이지 상단의 메뉴에서 **재부팅 (Reboot)** 기능을 선택합니다. 이렇게 하면 선택한 노드를 재부팅하여 그 노드가 재부팅되고 있음을 알리는 메세지가 페이지 상단에 나타납니다.
4. 노드의 업데이트된 상태를 확인하려면 페이지를 새로고침합니다.

재부팅 (Reboot)을 클릭하기전 재부팅하기 위한 모든 노드를 선택하여 한번에 하나 이상의 노드를 재부팅할 수 있습니다.

4.3.2. 노드가 클러스터를 탈퇴 또는 참여하는 원인

Conga의 **luci** 서버 구성 요소를 사용하여 노드의 모든 클러스터 서비스를 중지하면 해당 노드가 활성 클러스터를 탈퇴하게 할 수 있습니다. 또한 **Conga**의 **luci** 서버 구성 요소를 사용하여 클러스터를 탈퇴한 노드가 다시 클러스터에 참여하게 할 수 있습니다.

노드가 클러스터에서 탈퇴하는 원인이 되어도 노드에서 클러스터 설정 정보를 제거하지 않고 노드는 클러스터 노드 보기에 **클러스터 멤버가 아님 (`Not a cluster member`)** 상태로 남아 있게 됩니다. 클러스터 설정에서 노드를 완전히 삭제하는 방법은 4.3.4절. "클러스터에서 멤버 삭제 "에서 참조하십시오.

노드가 클러스터에서 탈퇴하게 하려면 다음 절차를 실행합니다. 이는 노드의 클러스터 소프트웨어를 종료합니다. 노드가 클러스터에서 탈퇴하게 하면 클러스터를 다시 시작할 때 노드는 클러스터에 자동으로 참여하지 못하게 됩니다.

1. 특정 클러스터 페이지에서 클러스터 보기의 상단의 **노드 (Nodes)**를 클릭하면 클러스터를 구성하는 노드가 나타납니다. 이는 **luci 홈베이스 (Homebase)** 페이지의 왼쪽에 있는 메뉴에서 **클러스터 관리 (Manage Clusters)** 아래의 클러스터 이름을 클릭할 때 나타나는 기본 페이지입니다.
2. 클러스터를 탈퇴하려는 노드의 체크박스를 클릭하여 해당 노드를 선택합니다.

3. 페이지 상단의 메뉴에서 **클러스터 탈퇴 (Leave Cluster)** 기능을 선택합니다. 이는 페이지 상단에 노드가 중지되었음을 가리키는 메세지가 나타나게 합니다.

4. 노드의 업데이트된 상태를 확인하려면 페이지를 새로고침합니다.

클러스터 탈퇴 (Leave Cluster) 를 클릭하기 전 클러스터를 탈퇴할 모든 노드를 선택하여 한 번에 하나 이상의 노드를 클러스터에서 탈퇴시킬 수 있습니다.

클러스터를 노드에 다시 참여하게 하려면, 클러스터를 다시 참가시키려는 노드의 체크박스를 클릭하여 노드를 선택하고 **클러스터 참여 (Join Cluster)** 를 선택합니다. 이렇게 하면 선택한 노드가 클러스터에 참여하게 하고 재부팅 시 클러스터에 참가하게 합니다.

4.3.3. 실행중인 클러스터에 멤버 추가

실행중인 클러스터에 멤버를 추가하려면 다음과 같은 절차를 따릅니다.

1. 특정 클러스터 페이지에서 클러스터 보기 상단의 **노드 (Nodes)**를 클릭하면 클러스터를 구성하는 노드가 나타납니다. 이는 **luci 홈베이스 (Homebase)** 페이지의 왼쪽에 있는 메뉴에서 **클러스터 관리 (Manage Clusters)** 아래의 클러스터 이름을 클릭할 때 나타나는 기본 페이지입니다.

2. **추가 (Add)**를 클릭합니다. **추가 (Add)**를 클릭하면 **클러스터에 노드 추가 (Add nodes to this cluster)** 창이 나타납니다.

3. **노드 호스트 이름 (Node Hostname)** 란에 노드 이름을 입력하고 **암호 (Password)** 란에 **ricci** 암호를 입력합니다. **ricci** 에이전트에 대해 기본 포트인 11111 외에 다른 포트를 사용하고 있을 경우 사용하고 있는 포트로 매개 변수를 변경할 수 있습니다.

4. 클러스터 스토리지가 필요한 경우 **공유 스토리지 지원 활성화 (EEnable Shared Storage Support)**를 선택합니다. 이렇게 하면 클러스터 스토리지를 지원하는 패키지를 다운로드하여 클러스터된 LVM을 활성화합니다. 이는 장애 복구형 스토리지 추가 기능 또는 확장 가능한 파일 시스템 추가 기능을 액세스할 수 있는 경우에만 이 옵션을 선택하셔야 합니다.

5. 노드를 더 추가하려면, **다른 노드 추가 (Add Another Node)** 를 클릭하여 클러스터의 추가 노드마다 노드 이름과 root 암호를 입력합니다.

6. **노드 추가 (Add Nodes)**를 클릭합니다. **노드 추가 (Add Nodes)**를 클릭하면 다음과 같은 작업이 시작됩니다:

    1. **패키지 다운로드 (Download Packages)** 를 선택한 경우, 클러스터 소프트웨어 패키지는 노드에 다운로드됩니다.

    2. 클러스터 소프트웨어는 노드에 설치됩니다 (또는 적절한 소프트웨어가 설치되었는지를 확인합니다).

    3. 클러스터 설정 파일이 업데이트되고 클러스터에 있는 각 노드에 전달됩니다 – 추가된 노드 포함

    4. 추가된 노드는 클러스터에 참여합니다.

    노드 (Nodes) 페이지에 노드가 클러스터에 추가되어 있음을 가리키는 메세지와 함께 나타납니다. 상태를 업데이트하기 위해 페이지를 새로 고침합니다.

7. 노드 추가 작업이 완료된 후, 3.6절. “차단 (Fence) 장치 설정 ” 에서 설명하듯이 해당 노드에 대한 펜싱을 설정하기 위해 새로 추가된 노드의 노드 이름을 클릭합니다.

4.3.4. 클러스터에서 멤버 삭제

현재 실행 중인 기존 클러스터에서 멤버를 제거하려면 다음의 절차를 실행합니다. 클러스터에 있는 모든 노드를 한번에 삭제하는 경우 외에는 삭제하기 전 노드를 중지해야 함에 유의하십시오.

1. 특정 클러스터 페이지에서 클러스터 보기 상단의 **노드 (Nodes)**를 클릭하면 클러스터를 구성하는 노드가 나타납니다. 이는 **luci 홈베이스 (Homebase)** 페이지의 왼쪽에 있는 메뉴에서 **클러스터 관리 (Manage Clusters)** 아래의 클러스터 이름을 클릭할 때 나타나는 기본 페이지입니다.

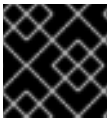
참고

노드가 삭제될 때 노드에서 실행중인 서비스를 복구하려면 다음 단계을 생략합니다.

2. 삭제될 노드에서 실행 중인 서비스를 비활성화하거나 또는 재배치합니다. 서비스 비활성화 및 재배치에 대한 내용은 4.5절. “고가용성 서비스 관리 ”에서 참조하십시오.

3. 삭제할 노드를 선택합니다.

4. **삭제 (Delete)**를 클릭합니다. **노드 (Nodes)** 페이지에서는 삭제된 노드가 나타납니다. 현재 상태를 확인하려면 페이지를 새로 고침합니다.



중요

클러스터에서 클러스터 노드를 제거하는 것은 되돌릴 수 없는 파괴식 작업입니다.

4.4. 클러스터 시작, 중지, 다시 시작, 삭제

클러스터에 있는 개별 노드에서 이러한 작업을 실행하여 클러스터를 시작, 중지, 재시작할 수 있습니다. 특정 클러스터 페이지에서 클러스터 보기 상단의 **노드 (Nodes)**를 클릭합니다. 그러면 클러스터를 구성하는 노드 집합이 나타납니다.

클러스터 서비스가 중지되거다 다시 시작되고 있는 노드에서 실행되고 있기 때문에 이를 다른 클러스터 멤버로 이동시켜야 할 경우, 클러스터 노드 또는 전체 클러스터의 시작 및 다시 시작 동작을 수행하여 클러스터 서비스를 단시간 중지할 수 있습니다.

클러스터를 중지하려면 다음 단계를 수행합니다. 이는 노드의 클러스터 소프트웨어를 종료하지만 노드에서의 클러스터 설정 정보를 삭제하지 않습니다. 그리고 노드는 **클러스터 멤버가 아님 (Not a cluster member)**의 상태로 클러스터 노드 보기에 계속 나타납니다.

1. 각 노드 옆에 있는 체크박스를 클릭하여 클러스터에 있는 모든 노드를 선택합니다.

2. 페이지 상단의 메뉴에서 **클러스터 탈퇴 (Leave Cluster)** 기능을 선택합니다. 이렇게 하면 페이지 상단에 각 노드가 중지되었음을 가리키는 메세지가 나타납니다.

3. 노드의 업데이트된 상태를 확인하기 위해 페이지를 새로고침합니다.

클러스터를 시작하려면, 다음의 절차를 실행합니다:

1. 각 노드 옆에 있는 체크박스를 클릭하여 클러스터에 있는 모든 노드를 선택합니다.

2. 페이지 상단의 메뉴에서 **클러스터 참여 (Join Cluster)** 기능을 선택합니다.

3. 노드의 업데이트된 상태를 확인하기 위해 페이지를 새로고침합니다.

실행중인 클러스터를 다시 시작하려면, 먼저 클러스터에 있는 모든 노드를 중지하고 앞에서 설명했듯이 클러스터에 있는 모든 노드를 시작합니다.

클러스터를 완전하게 제거하려면 다음 단계를 수행합니다. 이는 모든 클러스터 서비스를 중지하고 노드에서 클러스터 설정 정보를 삭제하며 클러스터 보기에서 노드를 삭제합니다. 나중에 삭제된 노드를 사용하여 기존 클러스터를 추가하려고 할 경우 **luci**는 노드가 클러스터 멤버가 아니라고 표시합니다.

중요

클러스터를 삭제하는 것은 실행 취소할 수 없는 파괴적인 작업입니다. 클러스터 삭제 후 이를 복구하려면 처음부터 클러스터를 다시 생성하여 다시 정의해야 합니다.

1. 각 노드 옆에 있는 체크박스를 클릭하여 클러스터에 있는 모든 노드를 선택합니다.
2. 페이지 상단 메뉴에서 **삭제 (Delete)** 기능을 선택합니다.

클러스터 멤버쉽을 변경하거나 클러스터 서비스를 중지하지 않고 **luci** 인터페이스에서 클러스터를 제고하고자 할 경우 4.2절. "luci 인터페이스에서 클러스터 삭제"에서 설명하고 있듯이 **Manage Clusters** 페이지에 있는 **Remove** 옵션을 사용할 수 있습니다.

4.5. 고가용성 서비스 관리

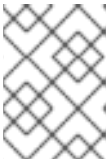
3.10절. "클러스터에 클러스터 서비스 추가 " 에서 설명하고 있듯이 서비스를 추가 및 수정에 더하여 **Conga**의 **luci** 서버 구성 요소를 통해 고가용성 서비스의 다음과 같은 관리 기능을 실행할 수 있습니다.

- 서비스 시작
- 서비스 재시작
- 서비스 비활성화
- 서비스 삭제
- 서비스 재배치

특정 클러스터 페이지에서 클러스터 보기 상단의 **서비스 그룹 (Service Groups)**을 클릭하여 클러스터의 서비스를 관리할 수 있습니다. 이렇게 하면 클러스터에 설정된 서비스를 표시할 수 있습니다.

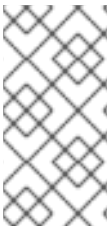
- **서비스 시작** – 현재 실행되지 않는 서비스 중 하나를 시작하려면 시작하려는 서비스에 대한 체크박스를 클릭하여 해당 서비스를 선택하고 **시작 (Start)**을 클릭합니다.
- **서비스 다시 시작** – 현재 실행 중인 서비스를 다시 시작하려면, 다시 시작하려는 서비스의 체크박스를 클릭하여 서비스를 선택하고 **다시 시작 (Restart)**을 클릭합니다.
- **서비스 비활성화 (Disabling a service)** – 현재 실행 중인 서비스를 비활성화하려면, 비활성화하려는 서비스의 체크박스를 클릭하여 서비스를 선택하고 **비활성화 (Disable)**를 클릭합니다.
- **서비스 삭제 (Deleting a service)** – 현재 실행되지 않는 서비스를 삭제하려면, 삭제하려는 서비스의 체크박스를 클릭하여 해당 서비스를 선택하고 **삭제 (Delete)**를 클릭합니다.
- **서비스 이동 (Relocating a service)** – 실행 중인 서비스를 이동하려면, 서비스 보기에서 서비스 이름을 클릭합니다. 이렇게 하면 현재 어떤 노드에서 서비스가 실행되고 있는 지를 나타내는 디스플레이와 함께 서비스의 서비스 설정 페이지가 나타납니다.

  노드에서 시작 (`Start on node...`) 드롭 다운 상자에서 서비스를 이동할 노드를 선택하고 **시작 (Start)** 아이콘을 클릭합니다. 화면 상단에 서비스가 시작되었다는 메세지가 나타납니다. 선택한 노드에서 서비스가 실행되고 있음이 나타나는 새로운 디스플레이를 보려면 화면을 새로고침해야 합니다.



참고

선택하여 실행 중인 서비스가 **vm** 서비스일 경우, 드롭 다운 상자에서 **relocate** 옵션 대신 **migrate** 옵션을 표시합니다.



참고

서비스 (Services) 페이지에서 서비스의 이름을 클릭하여 개별적으로 서비스를 시작, 재시작,비활성화, 삭제할 수 있습니다. 서비스 이름을 클릭하면 서비스 설정 페이지가 나타납니다. 서비스 설정 페이지의 오른쪽 위에는 **시작 (Start)**, **재시작 (Restart)**, **비활성화 (Disable)**, **삭제 (Delete)**와 동일한 아이콘이 있습니다.

4.6. LUCI 설정 백업 및 복구

Red Hat Enterprise Linux 6.2 릴리즈 이후, 다음 단계를 실행하여 **/var/lib/luci/data/luci.db** 파일에 저장되는 **luci** 데이터베이스를 백업할 수 있습니다. 이는 **cluster.conf** 파일에 저장되는 클러스터 설정 자체가 아닙니다. 대신 이에는 사용자와 클러스터 목록 및 **luci**가 관리하는 관련 속성 목록이 포함되어 있습니다. 기본값으로 이 단계에서 생성된 백업은 **luci.db** 파일과 동일한 디렉토리에 기록됩니다.

1. **service luci stop** 명령을 실행합니다.
2. **service luci backup-db** 명령을 실행합니다.

   옵션으로 **backup-db** 명령에 대한 매개 변수로 파일 이름을 지정할 수 있습니다. 이렇게 하면 파일에 **luci** 데이터베이스를 기록하게 됩니다. 예를 들어 **/root/luci.db.backup** 파일에 **luci** 데이터베이스를 쓰려면 **service luci backup-db /root/luci.db.backup** 명령을 실행합니다. 하지만 **/var/lib/luci/data/** 이외의 장소에 작성되는 백업 파일 (**service luci backup-db**를 사용할 때 지정한 파일 이름의 백업)은 **list-backups** 명령의출력 결과에 나타나지 않는다는 점에 주의하십시오.
3. **service luci start** 명령을 실행합니다.

다음 절차를 사용하여 **luci** 데이터베이스를 복구합니다.

1. **service luci stop** 명령을 실행합니다.
2. **service luci list-backups** 명령을 실행하여 복구할 파일 이름을 적어둡니다.
3. **service luci restore-db /var/lib/luci/data/*lucibackupfile***을 실행합니다. 여기서 *lucibackupfile*은 복구할 백업 파일입니다.

   예를 들어 다음 명령은 **luci-backup20110923062526.db** 백업 파일에 저장된 **luci** 설정 정보를 복구합니다:

   ```
   service luci restore-db /var/lib/luci/data/luci-
   backup20110923062526.db
   ```
4. **service luci start** 명령을 실행합니다.

luci 데이터베이스를 복원해야 하지만 완전한 재설치로 인해 백업을 생성한 컴퓨터에서 **host.pem** 파일을 잃어버린 경우 클러스터 노드를 다시 인증하기 위해 수동으로 클러스터를 **luci**에 다시 추가해야 합니다.

다음 단계를 실행하여 백업이 생성된 컴퓨터가 아닌 다른 컴퓨터에 **luci** 데이터베이스를 복구합니다. 데이터베이스 자체를 복구할 뿐 만 아니라 SSL 인증서를 복사하여 **luci**가 **ricci** 노드에 대해 인증되었는지 확인해야 합니다. 예제에서 백업은 **luci1** 시스템에 생성되어 백업은 **luci2** 시스템에 복구되어 있습니다.

1. 다음 일련의 명령을 실행하여 **luci1**에 **luci** 백업을 생성하고 SSL 인증서 파일과 **luci** 백업 모두를 **luci2**에 복사합니다.

```
[root@luci1 ~]# service luci stop
[root@luci1 ~]# service luci backup-db
[root@luci1 ~]# service luci list-backups
/var/lib/luci/data/luci-backup20120504134051.db
[root@luci1 ~]# scp /var/lib/luci/certs/host.pem
/var/lib/luci/data/luci-backup20120504134051.db root@luci2:
```

2. **luci2** 시스템에서 **luci**가 설치되어 실행되고 있지 않은지 확인합니다. 설치되지 않은 경우 패키지를 설치합니다.

3. 다음 일련의 명령을 실행하여 인증이 이루어 지고 있는지 확인하고 **luci2**에 **luci1**에서의 **luci** 데이터베이스를 복구합니다.

```
[root@luci2 ~]# cp host.pem /var/lib/luci/certs/
[root@luci2 ~]# chown luci: /var/lib/luci/certs/host.pem
[root@luci2 ~]# /etc/init.d/luci restore-db ~/luci-
backup20120504134051.db
[root@luci2 ~]# shred -u ~/host.pem ~/luci-backup20120504134051.db
[root@luci2 ~]# service luci start
```

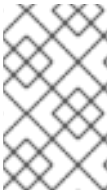
5장. CCS 명령으로 RED HAT 고가용성 추가 기능 설정

Red Hat Enterprise Linux 6.1 릴리즈와 그 이후 버전에서 Red Hat 고가용성 추가 기능은 **ccs**클러스터 설정 명령을 지원합니다. **ccs** 명령을 사용하면 관리자가 **cluster.conf** 클러스터 설정 파일을 생성, 수정, 확인할 수 있습니다. **ccs** 명령을 사용하여 클러스터 설정 파일을 로컬 파일 시스템이나 원격 노드에서 설정할 수 있습니다. **ccs** 명령을 사용하여 관리자는 설정된 클러스터에 있는 하나 이상의 노드에서 클러스터 서비스를 시작 및 중지할 수 있습니다.

다음 부분에서는 **ccs** 명령을 사용하여 Red Hat 고가용성 추가 기능 클러스터 설정 파일을 설정하는 방법에 대해 설명합니다. 실행 중인 클러스터를 관리하기 위해 **ccs** 명령을 사용하는 방법은 *6장. ccs로 Red Hat 고가용성 추가 기능 관리* 에서 참조하십시오.

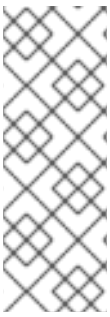
이는 다음과 같은 부분으로 구성되어 있습니다:

- 5.1절. “옵션 개요 ”
- 5.2절. “설정 작업 ”
- 5.3절. “**ricci** 시작 ”
- 5.4절. “클러스터 생성 ”
- 5.5절. “차단 장치 설정 ”
- 5.7절. “클러스터 멤버에 대해 차단 장치 설정 ”
- 5.8절. “장애 조치 도메인 설정 ”
- 5.9절. “글로벌 클러스터 리소스 설정 ”
- 5.10절. “클러스터에 클러스터 서비스 추가 ”
- 5.13절. “쿼럼 (Quorum) 디스크 설정 ”
- 5.14절. “기타 다른 클러스터 설정 ”
- 5.14절. “기타 다른 클러스터 설정 ”
- 5.15절. “클러스터 노드에 설정 파일 전달 ”



참고

고가용성 추가 기능의 사용이 자신의 요구에 부합하고 지원될 수 있는지 확인하십시오. 사용하기 전 설정을 확인하기 위해 Red Hat 담당자에게 문의하시기 바랍니다. 또한 설정 번인 (burn-in) 기간을 두어 장애 모드를 테스트하십시오.



참고

다음 부분에서는 일반적으로 사용되는 **cluster.conf** 요소와 속성을 참조합니다. **cluster.conf** 요소와 속성의 전체적 목록과 설명은 **/usr/share/cluster/cluster.rng**에 있는 클러스터 스키마와 **/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html** (예: **/usr/share/doc/cman-3.0.12/cluster_conf.html**)의 주석 스키마를 참조하십시오.

5.1. 옵션 개요

다음 부분에서는 클러스터를 설정하기 위한**ccs** 명령 사용의 일반적인 작업에 대해 설명합니다:

- 5.1.1절. “로컬 시스템에서 클러스터 설정 파일 생성 ”
- 5.1.2절. “현재 클러스터 설정 보기 ”
- 5.1.3절. “ccs 명령으로 ricci 암호 지정 ”
- 5.1.4절. “클러스터 설정 구성 요소 수정 ”

5.1.1. 로컬 시스템에서 클러스터 설정 파일 생성

ccs 명령을 사용하여 클러스터 노드에 클러스터 설정 파일을 생성하거나 로컬 파일 시스템에 클러스터 설정 파일을 생성한 후 이를 클러스터에 있는 호스트로 전송할 수 있습니다. 이렇게 하면 로컬 컴퓨터의 파일에서 작업할 수 있으므로 버전 제어 하에서 이를 유지 관리할 수 있습니다. 그렇지 않으며 필요에 따라 파일에 태그를 붙일 수 있습니다. **ccs** 명령을 사용 시 **root** 권한이 필요하지 않습니다.

ccs 명령을 사용하여 클러스터 노드에서 클러스터 설정 파일을 생성하거나 편집할 때, **-h** 옵션을 사용하여 호스트 이름을 지정할 수 있습니다. 이는 호스트에서 **cluster.conf** 파일을 생성하고 편집합니다:

```
ccs -h host [options]
```

로컬 시스템에서 클러스터 설정 파일을 생성하고 편집하려면, 클러스터 작업을 실행할 때 **ccs** 명령의 **-f** 옵션을 사용하여 설정 파일의 이름을 지정합니다. 이 파일의 이름을 원하는 대로 지정할 수 있습니다.

```
ccs -f file [options]
```

로컬 파일을 생성한 후 **ccs** 명령의 **--setconf** 옵션을 사용하여 파일을 클러스터 노드로 전송할 수 있습니다. 클러스터에 있는 호스트 컴퓨터에서 전송된 파일은 **cluster.conf**로 이름이 지정되어 **/etc/cluster** 디렉토리에 배치됩니다.

```
ccs -h host -f file --setconf
```

ccs 명령의 **--setconf** 옵션 사용에 대한 자세한 내용은 5.15절. “클러스터 노드에 설정 파일 전달 ” 에서 참조하십시오.

5.1.2. 현재 클러스터 설정 보기

클러스터 설정 파일을 생성할 때 현재 파일을 출력하려면 다음과 같은 명령을 사용하여 호스트로 클러스터에 노드를 지정합니다:

```
ccs -h host --getconf
```

로컬 시스템에 클러스터 설정 파일을 생성하고 있을 경우, 5.1.1절. “로컬 시스템에서 클러스터 설정 파일 생성 ”에서 설명하고 있듯이 **-h** 옵션 대신 **-f** 옵션을 지정할 수 있습니다.

5.1.3. ccs 명령으로 ricci 암호 지정

ccs 명령을 실행하면 **cluster.conf** 파일의 복사본을 클러스터의 노드로 배포하지만 이를 위해서 2.13절. “**ricci** 사용 시 고려 사항 ” 에서 설명하고 있듯이 클러스터 노드에서 **ricci**를 설치하여 실행하고 있어야 합니다. **ricci** 사용을 위해 특정 컴퓨터에서 **ricci**와 처음으로 통신할 때 암호가 필요합니다.

사용하고 있는 컴퓨터에서 특정 시스템의 **ricci** 인스턴스에 대한 암호를 입력하지 않으면 **ccs** 명령이 필요한 경우 암호를 묻습니다. 다른 방법으로 **-p** 옵션을 사용하여 명령행에서 **ricci** 암호를 지정할 수 있습니다.

```
ccs -h host -p password --sync --activate
```

ccs 명령의 **--sync** 옵션을 사용하여 클러스터의 모든 노드에 **cluster.conf** 파일을 전달하고 명령에 대해 **ricci** 암호를 지정할 때, **ccs** 명령은 클러스터에 있는 각 노드에 대한 암호를 사용합니다. 개별 노드에서 **ricci**에 대해 다른 암호를 설정해야 하는 경우, **-p** 옵션과 함께 **--setconf**를 사용하여 한 번에 하나의 노드에 설정 파일을 배포할 수 있습니다.

5.1.4. 클러스터 설정 구성 요소 수정

클러스터 설정 파일에서 클러스터 구성 요소 및 속성을 설정하려면 **ccs** 명령을 사용합니다. 파일에 클러스터 구성 요소를 추가한 후, 해당 구성 요소의 속성을 수정하기 위해 정의된 구성 요소를 제거하고 속성을 수정하여 다시 구성 요소를 추가해야 합니다. 각 구성 요소에 대해 이러한 작업을 실행하는 방법에 대한 자세한 내용은 이 장의 개별 섹션에서 설명하고 있습니다.

cman 클러스터 구성 요소의 속성은 클러스터 구성 요소를 변경하기 위한 절차에 예외를 추가합니다. 이러한 속성을 변경하려면 **ccs** 명령의 **--setcman** 옵션을 실행하여 새 속성을 지정합니다. 5.1.5절. “이전 설정을 덮어쓰기하는 명령 ”에서 설명하고 있듯이 이러한 옵션을 지정하는 것은 기본값으로 명시적으로 지정되지 않은 모든 값을 재설정할 수 있으므로 이에 유의합니다.

5.1.5. 이전 설정을 덮어쓰기하는 명령

속성을 설정할 때 의미론을 덮어쓰기하는 **ccs** 명령의 여러 옵션이 있습니다. 즉 아무 설정을 지정하지 않고 이러한 옵션 중 하나로 **ccs** 명령을 실행할 수 있지만 모든 설정을 기본값으로 재설정하는 것입니다. 이러한 옵션에는 다음과 같은 것이 있습니다:

- **--settotem**
- **--setdlm**
- **--setrm**
- **--setcman**
- **--setmulticast**
- **--setaltmulticast**
- **--setfencedaemon**
- **--setlogging**
- **--setquorumd**

예를 들어 모든 차단 데몬 속성을 재설정하려면 다음과 같은 명령을 실행할 수 있습니다.

```
# ccs -h hostname --setfencedaemon
```

하지만 이러한 명령 중 하나를 사용하여 속성을 다시 설정하면 명령의 다른 속성은 기본값으로 재설정되는 점에 유의하십시오. 예를 들어 다음과 같은 명령을 사용하여 **post_fail_delay** 속성을 **5**로 설정할 수 있습니다:

```
# ccs -h hostname --setfencedaemon post_fail_delay=5
```

명령 실행 후 다음과 같은 명령을 실행하여 **post_join_delay** 속성을 10으로 재설정할 경우 **post_fail_delay** 속성은 기본값으로 복원됩니다:

```
# ccs -h hostname --setfencedaemon post_join_delay=10
```

post_fail_delay 및 **post_join_delay** 속성 모두를 다시 설정하려면 다음 예제와 같이 동일한 명령에서 모두 지정합니다:

```
# ccs -h hostname --setfencedaemon post_fail_delay=5 post_join_delay=10
```

차단 장치를 설정하는 방법에 대한 자세한 내용은 5.5절. “차단 장치 설정 ” 에서 참조하십시오.

5.1.6. 설정 유효성 검사

ccs 명령을 사용하여 클러스터 설정 파일을 생성 및 편집할 때 설정은 클러스터 스키마에 따라 자동으로 확인됩니다. Red Hat Enterprise Linux 6.3 릴리즈 이후 **ccs** 명령은 **-h** 옵션으로 지정된 노드의 **/usr/share/cluster/cluster.rng**에 있는 클러스터 스키마에 따라 설정을 확인합니다. 이전에 **ccs** 명령은 **ccs** 명령으로 패키지된 로컬 시스템에 있는 **/usr/share/ccs/cluster.rng**에 있는 클러스터 스키마를 항상 사용했습니다. **-f** 옵션을 사용하여 로컬 시스템을 지정할 때 **ccs** 명령은 시스템에서 **ccs** 명령으로 패키지된 **/usr/share/ccs/cluster.rng**에 있는 클러스터 스키마를 계속 사용합니다.

5.2. 설정 작업

ccs로 Red Hat 고가용성 추가 기능 소프트웨어를 설정하는 것은 다음과 같은 절차로 구성되어 있습니다:

1. 클러스터에 있는 모든 노드에서 **ricci**가 실행되고 있는지 확인. 5.3절. “**ricci** 시작 ”에서 참조하십시오.
2. 클러스터 생성. 5.4절. “클러스터 생성 ”에서 참조하십시오.
3. 차단 장치 설정. 5.5절. “차단 장치 설정 ” 에서 참조하십시오.
4. 클러스터 멤버에 대한 펜싱 설정. 5.7절. “클러스터 멤버에 대해 차단 장치 설정 ” 에서 참조하십시오.
5. 장애 조치 도메인 생성. 5.8절. “장애 조치 도메인 설정 ” 에서 참조하십시오.
6. 리소스 생성. 5.9절. “글로벌 클러스터 리소스 설정 ” 에서 참조하십시오.
7. 클러스터 서비스 생성. 5.10절. “클러스터에 클러스터 서비스 추가 ” 에서 참조하십시오.
8. 필요한 경우 쿼럼 디스크 생성. 5.13절. “쿼럼 (Quorum) 디스크 설정 ” 에서 참조하십시오.
9. 글로벌 클러스터 등록 정보 설정. 5.14절. “기타 다른 클러스터 설정 ” 에서 참조하십시오.
10. 클러스터 설정 파일을 모든 클러스터 노드에 전달. 5.15절. “클러스터 노드에 설정 파일 전달 ” 에서 참조하십시오.

5.3. RICCI 시작

클러스터의 노드에 클러스터 설정 파일을 생성하여 배포하려면 각 노드에서 **ricci** 서비스가 실행되고 있어야 합니다. **ricci**를 시작하기 전 다음과 같이 시스템이 설정되어 있는지 확인하십시오:

1. 클러스터 노드의 IP 포트는 **ricci**에 대해 활성화되어 있어야 합니다. 클러스터 노드에서 IP 포트를 활성화하는 방법은 2.3.1절. "클러스터 노드에서 IP 포트 사용 " 에서 참조하십시오.

2. **ricci** 서비스는 클러스터에 있는 모든 노드에 설치되어 있으며 2.13절. "**ricci** 사용 시 고려 사항 "에서 설명하고 있듯이 **ricci** 암호가 할당되어 있습니다.

각 노드에 **ricci**를 설치 및 설정한 후, 각 노드에서 **ricci** 서비스를 시작합니다:

```
# service ricci start
Starting ricci:                                            [  OK  ]
```

5.4. 클러스터 생성

다음 부분에서는 **ccs** 명령을 사용하여 차단 장치, 페일오버 도메인, HA 서비스가 없는 스켈튼 클러스터 설정을 생성, 수정, 삭제하는 방법에 대해 설명합니다. 그 다음으로 이러한 설정 부분을 구성하는 방법에 대해 설명합니다.

스켈튼 클러스터 설정 파일을 생성하려면, 다음의 절차에서와 같이 먼저 클러스터를 만든 후 이름을 지정하고 클러스터에 노드를 추가합니다:

1. 클러스터에 있는 노드 중 하나에서 클러스터 설정 파일을 생성하려면, **ccs** 명령을 실행합니다. 여기서 **-h** 매개 변수를 사용하면 파일을 생성하기 위한 노드를 지정할 수 있으며 **createcluster** 옵션으로는 클러스터 이름을 지정할 수 있습니다:

   ```
   ccs -h host --createcluster clustername
   ```

   예를 들어, 다음과 같은 명령은 **node-01.example.com**에 **mycluster**라는 이름의 설정 파일을 생성합니다:

   ```
   ccs -h node-01.example.com --createcluster mycluster
   ```

   클러스터 이름은 15자를 초과할 수 없습니다.

   cluster.conf 파일이 지정한 호스트에 이미 존재하는 경우, 이 명령을 실행하여 기존 파일을 교체합니다.

   로컬 시스템에서 클러스터 설정 파일을 생성하려면 **-h** 옵션 대신 **-f** 옵션을 지정할 수 있습니다. 로컬시스템에서 파일을 생성하는 방법에 대한 자세한 내용은 5.1.1절. "로컬 시스템에서 클러스터 설정 파일 생성 "에서 참조하십시오.

2. 클러스터가 들어있는 노드를 설정하려면, 클러스터에 있는 각 노드에 대해 다음과 같은 명령을 실행합니다:

   ```
   ccs -h host --addnode node
   ```

   예를 들어, 다음의 세 개의 명령은 노드 **node-01.example.com**, **node-02.example.com**, **node-03.example.com**을 **node-01.example.com**에 있는 설정 파일에 추가합니다:

```
ccs -h node-01.example.com --addnode node-01.example.com
ccs -h node-01.example.com --addnode node-02.example.com
ccs -h node-01.example.com --addnode node-03.example.com
```

클러스터에 설정된 노드 목록을 보려면 다음 명령을 실행합니다:

```
ccs -h host --lsnodes
```

예 5.1. "세 개의 노드를 추가한 후 **cluster.conf** 파일 "에서는 **node-01.example.com**, **node-02.example.com**, **node-03.example.com** 노드가 들어 있는 클러스터 **mycluster**를 생성한 후 **cluster.conf** 설정 파일을 보여주고 있습니다.

예 5.1. 세 개의 노드를 추가한 후 cluster.conf 파일

```
<cluster name="mycluster" config_version="2">
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
         </fence>
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
         </fence>
     </clusternode>
     <clusternode name="node-03.example.com" nodeid="3">
         <fence>
         </fence>
     </clusternode>
   </clusternodes>
   <fencedevices>
   </fencedevices>
   <rm>
   </rm>
</cluster>
```

클러스터에 노드를 추가할 때, 쿼럼 (정족수)이 있는지 확인하기 위해 노드가 제공하는 표 수를 지정할 수 있습니다. 클러스터 노드에 대해 표 수를 설정하려면 다음 명령을 사용합니다:

```
ccs -h host --addnode host --votes votes
```

노드를 추가할 때, **ccs**는 노드 식별자로 사용되는 고유한 정수를 노드에 할당합니다. 노드를 생성할 때 수동으로 노드 식별자를 지정하려면 다음 명령을 사용합니다:

```
ccs -h host --addnode host --nodeid nodeid
```

클러스터에서 노드를 제거하려면 다음 명령을 실행합니다:

```
ccs -h host --rmnode node
```

클러스터의 모든 구성 요소 설정을 완료하면, 5.15절. "클러스터 노드에 설정 파일 전달 " 에서 설명하고 있

듯이 클러스터 설정 파일을 모든 노드에 동기화해야 합니다.

5.5. 차단 장치 설정

차단 장치 설정에는 클러스터의 차단 장치 생성, 업데이트, 삭제로 구성되어 있습니다. 클러스터에 있는 노드의 차단 장치를 설정하기 전 클러스터에 있는 차단 장치를 생성 및 이름을 지정해야 합니다. 클러스터에 있는 개별적 노드의 차단 장치 설정에 대한 자세한 내용은 5.7절. “클러스터 멤버에 대해 차단 장치 설정”에서 참조하십시오.

차단 장치를 설정하기 전 차단 데몬 등록 정보의 일부분을 기본값에서 변경하고자 할 수 있습니다. 차단 데몬에 대해 설정한 값은 클러스터에 대한 일반적인 값입니다. 수정하고자 하는 클러스터에 대한 일반적인 차단 장치 등록 정보는 다음과 같이 요약됩니다:

- **post_fail_delay** 속성은 노드 실패 후 노드 (차단 도메인의 멤버)를 차단하기 전 까지 차단 데몬 (**fenced**)이 기다리는 시간 (초)입니다. **post_fail_delay** 기본값은 **0**입니다. 이 값은 클러스터와 네트워크 성능에 맞게 변경할 수 있습니다.
- **post-join_delay** 속성은 노드가 차단 도메인을 결합하고 노드를 차단하기 전까지 차단 데몬 (**fenced**)의 대기 시간 (초)입니다. **post_join_delay** 기본값은 **6**입니다. **post_join_delay**의 일반 설정은 20 초에서 30 초 사이이지만 클러스터와 네트워크 성능에 따라 달라질 수 있습니다.

ccs 명령의 **--setfencedaemon** 옵션과 함께 **post_fail_delay** 및 **post_join_delay** 속성 값을 재설정합니다. 하지만 **ccs --setfencedaemon** 명령을 실행하면 명시적으로 설정된 기존의 차단 데몬 속성을 모두 덮어쓰기하고 기본 값으로 복구되는 점에 유의합니다.

예를 들어 **post_fail_delay** 속성 값을 설정하려면 다음 명령을 실행합니다. 이 명령은 이 명령으로 설정한 기존 차단 데몬 속성 값을 덮어쓰기하고 이를 기본값으로 복구합니다.

```
ccs -h host --setfencedaemon post_fail_delay=value
```

post_join_delay 속성 값을 설정하려면 다음 명령을 실행합니다. 이 명령은 이 명령으로 설정한 기존 차단 데몬 속성 값을 덮어쓰기하고 이를 기본값으로 복구합니다.

```
ccs -h host --setfencedaemon post_join_delay=value
```

post_join_delay 속성 및 **post_fail_delay** 속성 값을 설정하려면 다음 명령을 실행합니다:

```
ccs -h host --setfencedaemon post_fail_delay=value post_join_delay=value
```

참고

post_join_delay 및 **post_fail_delay** 속성 및 수정할 수 있는 추가 차단 데몬 등록 정보에 대한 자세한 내용은 fenced(8) man 페이지와 **/usr/share/cluster/cluster.rng**에 있는 클러스터 스키마 및 **/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html**의 주석 스키마를 참조하십시오.

클러스터의 차단 장치를 설정하려면 다음 명령을 실행합니다:

```
ccs -h host --addfencedev devicename [fencedeviceoptions]
```

예를 들어, IP 주소 **apc_ip_example**, 로그인 **login_example**, 암호 **password_example**를 갖는 **myfence**라는 이름으로 클러스터 노드 **node1**에 있는 설정 파일에 APC 차단 장치를 설정하려면 다음 명령을 실행합니다:

```
ccs -h node1 --addfencedev myfence agent=fence_apc ipaddr=apc_ip_example
login=login_example passwd=password_example
```

다음 예제에서는 이러한 APC 차단 장치를 추가한 후 **cluster.conf** 설정 파일의 **fencedevices** 부분을 보여주고 있습니다:

```
<fencedevices>
      <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="myfence" passwd="password_example"/>
</fencedevices>
```

클러스터에 차단 장치를 설정할 때 클러스터에 사용 할 수 있는 장치 또는 장치에 사용할 수 있는 옵션 목록을 확인하는 것은 유용할 수 있습니다. 또한 클러스터에 현재 설정된 차단 장치 목록을 확인하는 것도 유용할 수 있습니다. **ccs** 명령을 사용하여 사용 가능한 차단 장치 및 옵션 목록을 출력하거나 클러스터에 현재 설정된 차단 장치 목록을 출력하는 방법은 5.6절. “차단 장치 및 차단 장치 옵션 목록 ” 에서 참조하십시오.

클러스터 설정에서 차단 장치를 제거하려면 다음 명령을 실행합니다:

```
ccs -h host --rmfencedev fence_device_name
```

예를 들어, **node1** 클러스터 노드에 있는 클러스터 설정 파일에서 **myfence**라고 이름을 지정한 차단 장치를 제거하려면 다음 명령을 실행합니다:

```
ccs -h node1 --rmfencedev myfence
```

이미 설정한 차단 장치의 속성을 수정해야 할 경우, 먼저 차단 장치를 제거하고 변경된 속성이 있는 차단 장치를 다시 추가합니다.

클러스터의 모든 구성요소 설정 완료시 5.15절. “클러스터 노드에 설정 파일 전달 ” 에서 설명하고 있듯이 클러스터 설정 파일을 모든 노드에 동기화해야 합니다.

5.6. 차단 장치 및 차단 장치 옵션 목록

ccs 명령을 사용하여 사용 가능한 차단 장치 목록 및 사용 가능한 차단 장치 별 옵션 목록을 출력할 수 있습니다. 또한 **ccs** 명령을 사용하여 현재 클러스터에 설정된 차단 장치 목록을 출력할 수 있습니다.

클러스터에 현재 사용 가능한 차단 장치 목록을 출력하려면 다음 명령을 실행합니다:

```
ccs -h host --lsfenceopts
```

예를 들어 다음 명령은 클러스터 노드 **node1**에서 사용할 수 있는 차단 장치를 나열합니다. 예시 출력 결과는 다음과 같습니다.

```
[root@ask-03 ~]# ccs -h node1 --lsfenceopts
fence_rps10 - RPS10 Serial Switch
fence_vixel - No description available
```

```
fence_egenera - No description available
fence_xcat - No description available
fence_na - Node Assassin
fence_apc - Fence agent for APC over telnet/ssh
fence_apc_snmp - Fence agent for APC over SNMP
fence_bladecenter - Fence agent for IBM BladeCenter
fence_bladecenter_snmp - Fence agent for IBM BladeCenter over SNMP
fence_cisco_mds - Fence agent for Cisco MDS
fence_cisco_ucs - Fence agent for Cisco UCS
fence_drac5 - Fence agent for Dell DRAC CMC/5
fence_eps - Fence agent for ePowerSwitch
fence_ibmblade - Fence agent for IBM BladeCenter over SNMP
fence_ifmib - Fence agent for IF MIB
fence_ilo - Fence agent for HP iLO
fence_ilo_mp - Fence agent for HP iLO MP
fence_intelmodular - Fence agent for Intel Modular
fence_ipmilan - Fence agent for IPMI over LAN
fence_kdump - Fence agent for use with kdump
fence_rhevm - Fence agent for RHEV-M REST API
fence_rsa - Fence agent for IBM RSA
fence_sanbox2 - Fence agent for QLogic SANBox2 FC switches
fence_scsi - fence agent for SCSI-3 persistent reservations
fence_virsh - Fence agent for virsh
fence_virt - Fence agent for virtual machines
fence_vmware - Fence agent for VMware
fence_vmware_soap - Fence agent for VMware over SOAP API
fence_wti - Fence agent for WTI
fence_xvm - Fence agent for virtual machines
```

특정 차단 유형을 지정할 수 있는 옵션 목록을 출력하려면 다음 명령을 실행합니다:

```
ccs -h host --lsfenceopts fence_type
```

예를 들어 다음 명령은 **fence_wti** 차단 에이전트의 차단 옵션을 나열합니다.

```
[root@ask-03 ~]# ccs -h node1 --lsfenceopts fence_wti
fence_wti - Fence agent for WTI
  Required Options:
  Optional Options:
    option: No description available
    action: Fencing Action
    ipaddr: IP Address or Hostname
    login: Login Name
    passwd: Login password or passphrase
    passwd_script: Script to retrieve password
    cmd_prompt: Force command prompt
    secure: SSH connection
    identity_file: Identity file for ssh
    port: Physical plug number or name of virtual machine
    inet4_only: Forces agent to use IPv4 addresses only
    inet6_only: Forces agent to use IPv6 addresses only
    ipport: TCP port to use for connection with device
    verbose: Verbose mode
    debug: Write debug information to given file
    version: Display version information and exit
```

```
    help: Display help and exit
    separator: Separator for CSV created by operation list
    power_timeout: Test X seconds for status change after ON/OFF
    shell_timeout: Wait X seconds for cmd prompt after issuing command
    login_timeout: Wait X seconds for cmd prompt after login
    power_wait: Wait X seconds after issuing ON/OFF
    delay: Wait X seconds before fencing is started
    retry_on: Count of attempts to retry power on
```

현재 클러스터에 설정된 차단 장치 목록을 인쇄하려면 다음 명령을 실행합니다:

```
ccs -h host --lsfencedev
```

5.7. 클러스터 멤버에 대해 차단 장치 설정

클러스터 생성 및 차단 장치 생성의 초기 단계를 완료하면 클러스터 노드에 대해 차단 장치를 설정해야 합니다. 새로운 클러스터를 생성하고 클러스터에 대한 차단 장치를 설정한 후에 노드에 대한 차단 장치를 설정하려면 다음 단계를 따르십시오. 클러스터에 있는 각 노드에 대해 차단 장치를 설정해야 함에 유의하십시오.

다음 부분에서는 다음과 같은 절차를 문서화하고 있습니다:

- 5.7.1절. “노드에 대해 단일 전원 기반 차단 장치 설정 ”
- 5.7.2절. “노드에 대해 단일 스토리지 기반 차단 장치 설정 ”
- 5.7.3절. “백업 차단 장치 설정 ”
- 5.7.4절. “이중 전원으로 노드 설정 ”
- 5.7.5절. “차단 방식 및 차단 인스턴스 제거 ”

5.7.1. 노드에 대해 단일 전원 기반 차단 장치 설정

다음 절차를 사용하여 단일 전원 기반 차단 장치로 노드를 설정합니다. 이는 **fence_apc** 차단 에이전트를 사용하는 **apc**라는 차단 장치를 사용합니다.

1. 노드의 차단 방식을 추가하고 차단 방식의 이름을 기입합니다.

   ```
   ccs -h host --addmethod method node
   ```

   예를 들어, 클러스터 노드 **node-01.example.com**에 있는 설정 파일에서 노드 **node-01.example.com**에 대한 **APC**라는 이름의 차단 방식을 설정하려면, 다음 명령을 실행합니다:

   ```
   ccs -h node01.example.com --addmethod APC node01.example.com
   ```

2. 차단 방식에 대한 차단 인스턴스를 추가합니다. 노드에 사용할 차단 장치, 이러한 인스턴스를 적용할 노드, 방식 이름, 노드 특유의 옵션을 지정해야 합니다:

   ```
   ccs -h host --addfenceinst fencedevicename node method [options]
   ```

예를 들어, 클러스터 노드 **node-01.example.com**에 있는 설정 파일에 차단 인스턴스를 설정하기 위해 **APC**라는 방식을 사용하여 클러스터 노드 **node-01.example.com**를 차단하기 위해 **apc**라는 차단 장치에서 APC 스위치 전원 포트 1을 사용하는 경우 다음 명령을 실행합니다:

```
ccs -h node01.example.com --addfenceinst apc node01.example.com APC
port=1
```

클러스터에 있는 각 노드에 대해 차단 방식을 추가해야 할 필요가 있습니다. 다음 명령을 사용하여 **APC**라는 방식 이름으로 각 노드에 대해 차단 방식을 설정합니다. 차단 방식에 대한 장치는 **apc**를 장치 이름으로 지정합니다. 이는 5.5절. "차단 장치 설정 "에서 설명하고 있듯이 **--addfencedev** 옵션을 사용하여 이전에 설정된 장치입니다. 각 노드는 고유한 APC 스위치 전원 포트 번호로 설정됩니다. **node-01.example.com**의 포트 번호는 **1**이고, **node-02.example.com**의 포트 번호는 **2**이며, **node-03.example.com**의 포트 번호는 **3**입니다.

```
ccs -h node01.example.com --addmethod APC node01.example.com
ccs -h node01.example.com --addmethod APC node02.example.com
ccs -h node01.example.com --addmethod APC node03.example.com
ccs -h node01.example.com --addfenceinst apc node01.example.com APC port=1
ccs -h node01.example.com --addfenceinst apc node02.example.com APC port=2
ccs -h node01.example.com --addfenceinst apc node03.example.com APC port=3
```

예 5.2. "전원 기반 차단 방식 추가 후 **cluster.conf**"에서는 클러스터에 있는 각 노드에 차단 방식과 인스턴스를 추가한 후 **cluster.conf** 설정 파일을 보여주고 있습니다.

예 5.2. 전원 기반 차단 방식 추가 후 cluster.conf

```
<cluster name="mycluster" config_version="3">
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
            <method name="APC">
              <device name="apc" port="1"/>
             </method>
         </fence>
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
            <method name="APC">
              <device name="apc" port="2"/>
            </method>
         </fence>
     </clusternode>
     <clusternode name="node-03.example.com" nodeid="3">
         <fence>
            <method name="APC">
              <device name="apc" port="3"/>
            </method>
         </fence>
     </clusternode>
   </clusternodes>
   <fencedevices>
         <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="apc" passwd="password_example"/>
```

```
    </fencedevices>
    <rm>
    </rm>
</cluster>
```

클러스터의 모든 구성요소 설정 완료시 5.15절. “클러스터 노드에 설정 파일 전달 ” 에서 설명하고 있듯이 클러스터 설정 파일을 모든 노드에 동기화해야 합니다.

5.7.2. 노드에 대해 단일 스토리지 기반 차단 장치 설정

노드를 차단하기 위해 비전원 차단 방법 (즉, SAN/storage 펜싱)을 사용할 때 차단 장치에 대해 *차단 장치 제거 (unfencing)*를 설정해야 합니다. 이는 노드가 다시 시작할 때까지 차단 노드가 다시 활성화되지 않는지를 확인합니다. 노드의 차단 장치 제거 설정 시 **on** 또는 **enable**의 명시적 추가 작업과 함께 설정한 해당 차단 장치를 미러하는 장치를 지정합니다.

노드의 차단 장치 제거 (unfencing)에 대한 자세한 내용은 **fence_node(8)** man 페이지를 참조하십시오.

다음 절차를 사용하여 **sanswitch1**라는 차단 장치를 사용하는 단일 스토리지 기반 차단 장치로 노드를 설정합니다. 이러한 차단 장치는 **fence_sanbox2** 차단 에이전트를 사용합니다.

1. 노드의 차단 방식을 추가하고 차단 방식의 이름을 기입합니다.

   ```
   ccs -h host --addmethod method node
   ```

   예를 들어, 클러스터 노드 **node-01.example.com**에 있는 설정 파일에서 노드 **node-01.example.com**에 대한 **SAN**이라는 차단 방식을 설정하려면 다음 명령을 실행합니다:

   ```
   ccs -h node01.example.com --addmethod SAN  node01.example.com
   ```

2. 차단 방식에 대한 차단 인스턴스를 추가합니다. 노드에 사용할 차단 장치, 이러한 인스턴스를 적용할 노드, 방식 이름, 노드 특유의 옵션을 지정해야 합니다:

   ```
   ccs -h host --addfenceinst fencedevicename node method [options]
   ```

   예를 들어, **SAN**이라는 방식을 사용하여 클러스터 노드 **node-01.example.com**를 차단하기 위해 **sanswitch1**라는 차단 장치에 있는 SAN 스위치 전원 포트 11을 사용하는 클러스터 노드 **node-01.example.com**의 설정 파일에 있는 차단 인스턴스를 설정하려면 다음 명령을 실행합니다:

   ```
   ccs -h node01.example.com --addfenceinst sanswitch1
   node01.example.com SAN port=11
   ```

3. 이 노드에 있는 스토리지 기반 차단 장치에 대해 차단 장치 제거 (unfencing)를 설정하려면 다음 명령을 실행합니다:

   ```
   ccs -h host --addunfence fencedevicename node action=on|off
   ```

클러스터에 있는 각 노드에 대해 차단 방식을 추가할 필요가 있습니다. 다음 명령은 **SAN**이라는 방식을 사용하는 각 노드에 대해 차단 방식을 설정합니다. 차단 방식의 장치는 장치 이름으로 **sanswitch**를 지정하지만, 5.5절. “차단 장치 설정 ” 에서 설명하고 있듯이 이는 **--addfencedev** 옵션으로 이전에 설정된 장치입

니다. 각 노드는 고유한 SAN 물리적 포트 번호로 설정됩니다. **node-01.example.com**의 포트 번호는 **11**이고, **node-02.example.com**의 포트 번호는 **12**이며, **node-03.example.com**의 포트 번호는 **13**입니다.

```
ccs -h node01.example.com --addmethod SAN node01.example.com
ccs -h node01.example.com --addmethod SAN node02.example.com
ccs -h node01.example.com --addmethod SAN node03.example.com
ccs -h node01.example.com --addfenceinst sanswitch1 node01.example.com SAN
port=11
ccs -h node01.example.com --addfenceinst sanswitch1 node02.example.com SAN
port=12
ccs -h node01.example.com --addfenceinst sanswitch1 node03.example.com SAN
port=13
ccs -h node01.example.com --addunfence sanswitch1 node01.example.com
port=11 action=on
ccs -h node01.example.com --addunfence sanswitch1 node02.example.com
port=12 action=on
ccs -h node01.example.com --addunfence sanswitch1 node03.example.com
port=13 action=on
```

예 5.3. "스토리지 기반 차단 방식을 추가한 후 **cluster.conf**"에서는 클러스터에 있는 각 노드에 차단 방식, 차단 인스턴스, 차단 장치 제거를 추가한 후 **cluster.conf** 설정 파일을 보여주고 있습니다.

예 5.3. 스토리지 기반 차단 방식을 추가한 후 cluster.conf

```
<cluster name="mycluster" config_version="3">
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
            <method name="SAN">
       <device name="sanswitch1" port="11"/>
             </method>
         </fence>
         <unfence>
             <device name="sanswitch1" port="11" action="on"/>
         </unfence>
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
            <method name="SAN">
       <device name="sanswitch1" port="12"/>
             </method>
         </fence>
         <unfence>
             <device name="sanswitch1" port="12" action="on"/>
         </unfence>
     </clusternode>
     <clusternode name="node-03.example.com" nodeid="3">
         <fence>
            <method name="SAN">
       <device name="sanswitch1" port="13"/>
             </method>
         </fence>
         <unfence>
```

```
                <device name="sanswitch1" port="13" action="on"/>
            </unfence>
        </clusternode>
    </clusternodes>
    <fencedevices>
        <fencedevice agent="fence_sanbox2" ipaddr="san_ip_example"
login="login_example" name="sanswitch1" passwd="password_example"/>
    </fencedevices>
    <rm>
    </rm>
</cluster>
```

클러스터의 모든 구성요소 설정 완료시 5.15절. “클러스터 노드에 설정 파일 전달 ” 에서 설명하고 있듯이 클러스터 설정 파일을 모든 노드에 동기화해야 합니다.

5.7.3. 백업 차단 장치 설정

노드에 대해 여러 차단 방식을 정의할 수 있습니다. 첫 번째 방식을 사용하여 차단 실패할 경우, 시스템은 두 번째 방식을 사용하여 노드를 차단 시도합니다. 그 후 설정한 추가 방식을 실행합니다. 노드에 백업 차단 방식을 설정하려면 노드에 대해 두 가지 방식을 설정하여 각 노드에 차단 인스턴스를 설정합니다.

참고

시스템이 설정한 차단 방식을 사용하는 순서는 클러스터 설정 파일에 있는 순서를 따릅니다. **ccs** 명령으로 설정한 첫 번째 방식이 주요 차단 방법이며 설정한 두 번째 방식은 백업 차단 방법입니다. 순서를 변경하려면, 설정 파일에서 주요 차단 방식을 제거한 후 그 방식을 다시 추가합니다.

다음 명령을 실행하여 노드에 대해 현재 설정된 차단 방식 및 인스턴스 목록을 언제든지 확인할 수 있습니다. 노드를 지정하지 않은 경우, 이 명령은 모든 노드에 대해 현재 설정된 차단 방식 및 인스턴스를 나열합니다.

```
ccs -h host --lsfenceinst [node]
```

다음 단계를 따라하시면 **fence_apc** 차단 에이전트를 사용하는 **apc**라는 차단 장치를 사용하는 주요 차단 방식과 **fence_sanbox2** 차단 에이전트를 사용하는 **sanswitch1**라는 차단 장치를 사용하는 백업 차단 장치가 있는 노드를 설정할 수 있습니다. **sanswitch1** 장치는 스토리지 기반 차단 에이전트이기 때문에 장치에 대한 차단 장치 제거 (unfencing)도 설정해야 합니다.

1. 노드에 대해 주요 차단 방식을 추가하고 차단 방식에 대한 이름을 지정합니다.

   ```
   ccs -h host --addmethod method node
   ```

   예를 들어, 클러스터 노드 **node-01.example.com**에 있는 설정 파일에 노드 **node-01.example.com**에 대한 주요 방식으로 **APC**라는 차단 방식을 설정하려면 다음 명령을 실행합니다:

   ```
   ccs -h node01.example.com --addmethod APC node01.example.com
   ```

2. 주요 방식에 대한 차단 인스턴스를 추가합니다. 노드에 사용할 차단 장치, 인스턴스의 적용 대상 노드, 방식 이름, 노드 특유의 옵션을 지정해야 합니다:

```
ccs -h host --addfenceinst fencedevicename node method [options]
```

예를 들어, 클러스터 노드 **node-01.example.com**에 있는 설정 파일에 차단 인스턴스를 설정하기 위해 **APC**라는 방식을 사용하여 클러스터 노드 **node-01.example.com**를 차단하기 위해 **apc**라는 차단 장치에서 APC 스위치 전원 포트 1을 사용하는 경우 다음 명령을 실행합니다:

```
ccs -h node01.example.com --addfenceinst apc node01.example.com APC
port=1
```

3. 노드에 대한 백업 차단 방식을 추가하고 차단 방식에 대한 이름을 지정합니다.

```
ccs -h host --addmethod method node
```

예를 들어, 클러스터 노드 **node-01.example.com**에 있는 설정 파일에서 노드 **node-01.example.com**에 **SAN**라는 백업 차단 방식을 설정하려면 다음 명령을 실행합니다:

```
ccs -h node01.example.com --addmethod SAN  node01.example.com
```

4. 백업 방식에 차단 인스턴스를 추가합니다. 노드에 사용할 차단 장치, 인스턴스의 적용 대상 노드, 방식 이름, 노드 특유의 옵션을 지정해야 합니다:

```
ccs -h host --addfenceinst fencedevicename node method [options]
```

예를 들어, **SAN**이라는 방식을 사용하여 클러스터 노드 **node-01.example.com**를 차단하기 위해 **sanswitch1**라는 차단 장치에 있는 SAN 스위치 전원 포트 11을 사용하는 클러스터 노드 **node-01.example.com**의 설정 파일에 있는 차단 인스턴스를 설정하려면 다음 명령을 실행합니다:

```
ccs -h node01.example.com --addfenceinst sanswitch1
node01.example.com SAN port=11
```

5. **sanswitch1** 장치가 스토리지 기반 장치이기 때문에 장치에 대한 차단 장치 제거를 설정해야 합니다.

```
ccs -h node01.example.com --addunfence sanswitch1 node01.example.com
port=11 action=on
```

필요에 따라 차단 방식을 계속 추가할 수 있습니다.

이 절차는 클러스터에 있는 하나의 노드에 차단 장치 및 백업 차단 장치를 설정합니다. 클러스터에 있는 다른 노드에도 차단 장치를 설정해야 합니다.

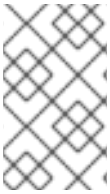
예 5.4. “백업 차단 방식을 추가한 후 cluster.conf ”에서는 클러스터에 있는 각 노드에 전원 기반 주요 차단 방식과 스토리지 기반 백업 차단 방식을 추가한 후 **cluster.conf** 설정 파일을 보여주고 있습니다.

예 5.4. 백업 차단 방식을 추가한 후 cluster.conf

```
<cluster name="mycluster" config_version="3">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
        <fence>
```

```
            <method name="APC">
              <device name="apc" port="1"/>
             </method>
            <method name="SAN">
          <device name="sanswitch1" port="11"/>
             </method>
        </fence>
        <unfence>
            <device name="sanswitch1" port="11" action="on"/>
        </unfence
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
        <fence>
           <method name="APC">
             <device name="apc" port="2"/>
           </method>
           <method name="SAN">
          <device name="sanswitch1" port="12"/>
             </method>
        </fence>
        <unfence>
            <device name="sanswitch1" port="12" action="on"/>
        </unfence
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
        <fence>
           <method name="APC">
             <device name="apc" port="3"/>
           </method>
           <method name="SAN">
          <device name="sanswitch1" port="13"/>
             </method>
        </fence>
        <unfence>
            <device name="sanswitch1" port="13" action="on"/>
        </unfence
    </clusternode>
  </clusternodes>
  <fencedevices>
       <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="apc" passwd="password_example"/>
       <fencedevice agent="fence_sanbox2" ipaddr="san_ip_example"
login="login_example" name="sanswitch1" passwd="password_example"/>
  </fencedevices>
  <rm>
  </rm>
</cluster>
```

클러스터의 모든 구성요소 설정 완료시 5.15절. “클러스터 노드에 설정 파일 전달 ” 에서 설명하고 있듯이 클러스터 설정 파일을 모든 노드에 동기화해야 합니다.



참고

설정한 차단 방식을 시스템에서 사용하는 순서는 클러스터 설정 파일의 순서를 따릅니다. 설정한 첫 번째 방식은 주요 차단 방식이고 설정한 두 번째 방식은 백업 차단 방식입니다. 순서를 변경하려면, 설정 파일에서 주요 차단 방식을 제거한 후 그 방식을 다시 추가합니다.

5.7.4. 이중 전원으로 노드 설정

클러스터가 노드에 대해 이중 전원 공급으로 설정되어 있을 경우, 노드 차단이 필요할 때 노드가 완전히 종료되도록 차단 장치를 설정해야 합니다. 각 전원 공급을 별도의 차단 방식으로 설정하면, 각 전원 공급은 별도록 차단됩니다. 따라서 첫 번째 전원 공급이 차단될 때 두 번째 전원 공급은 시스템을 계속 가동하게 하여 시스템이 전혀 차단되지 않을 수 있습니다. 시스템을 이중 전원 공급으로 설정하려면, 두 전원 공급을 종료하여 시스템이 완전히 중지하도록 차단 장치를 설정해야 합니다. 이를 위해 단일 차단 방식 안에서 두 개의 인스턴스를 설정해야 합니다. 각각의 인스턴스의 경우 **on**의 **action** 속성으로 각 장치를 설정하기 전 **off**의 **action** 속성으로 두 차단 장치를 설정해야 합니다.

이중 전원 공급으로 노드의 차단 장치를 설정하려면 다음 부분에 있는 절차를 따르십시오.

1. 이중 전원이 있는 노드에 대한 차단 장치를 설정하기 전 각 전원 스위치를 클러스터의 차단 장치로 설정해야 합니다. 차단 장치 설정에 대한 자세한 내용은 5.5절. “차단 장치 설정 ” 에서 참조하십시오.

   현재 클러스터에 설정된 차단 장치 목록을 인쇄하려면 다음 명령을 실행합니다:

   ```
   ccs -h host --lsfencedev
   ```

2. 노드의 차단 방식을 추가하고 차단 방식의 이름을 기입합니다.

   ```
   ccs -h host --addmethod method node
   ```

   예를 들어, 클러스터 노드 **node-01.example.com**에 있는 설정 파일에서 노드 **node-01.example.com**에 대해 **APC-dual**이라는 차단 방식을 설정하려면 다음 명령을 실행합니다:

   ```
   ccs -h node01.example.com --addmethod APC-dual node01.example.com
   ```

3. 차단 방식에 첫 번째 전원 공급에 대한 차단 인스턴스를 추가합니다. 노드에 사용할 차단 장치, 인스턴스를 적용할 노드, 방식 이름, 노드 특유의 옵션을 지정해야 합니다. 이 시점에서 **action** 속성을 **off**로 설정합니다.

   ```
   ccs -h host --addfenceinst fencedevicename node method [options]
   action=off
   ```

   예를 들어, **APC-dual**라는 방식을 사용하여 클러스터 노드 **node-01.example.com**를 차단하기 위해 **apc1**이라는 차단 장치에 있는 APC 스위치 전원 포트 1을 사용하는 클러스터 노드 **node-01.example.com**에 있는 설정 파일의 차단 인스턴스를 설정하고 **action** 속성을 **off**로 설정하려면 다음 명령을 실행합니다:

   ```
   ccs -h node01.example.com --addfenceinst apc1 node01.example.com
   APC-dual port=1 action=off
   ```

4. 차단 방식에 두 번째 전원 공급의 차단 인스턴스를 추가합니다. 노드에 사용할 차단 장치, 인스턴스의 적용 대상 노드, 방식 이름, 노드 특유의 옵션을 지정해야 합니다. 이 시점에서 인스턴스의 **action** 속성을 **off**로 설정해야 합니다:

```
ccs -h host --addfenceinst fencedevicename node method [options]
action=off
```

예를 들어, **APC-dual**이라는 첫 번째 인스턴스에 지정한 것과 동일한 방식을 사용하여 클러스터 노드 **node-01.example.com**를 차단하기 위해 **apc2**라는 차단 장치에 있는 APC 스위치 전원 포트 1을 사용하는 클러스터 노드 **node-01.example.com**에 있는 설정 파일의 두 번째 차단 인스턴스를 설정하고 **action** 속성을 **off**로 설정하려면 다음 명령을 실행합니다:

```
ccs -h node01.example.com --addfenceinst apc2 node01.example.com
APC-dual port=1 action=off
```

5. 이 시점에서 첫 번째 전원 공급에 대한 다른 차단 인스턴스를 차단 방식에 추가하여 **action** 속성을 **on**으로 설정합니다. 노드에 사용할 차단 장치, 인스턴스의 적용 대상 노드, 방식 이름, 노드 특유의 옵션을 지정하고 **action** 속성을 **on**으로 지정합니다:

```
ccs -h host --addfenceinst fencedevicename node method [options]
action=on
```

예를 들어, **APC-dual**라는 방식을 사용하는 클러스터 노드 **node-01.example.com**를 차단하기 위해 **apc1**라는 차단 장치에 있는 APC 스위치 전원 포트 1을 사용하는 클러스터 노드 **node-01.example.com**에 있는 설정 파일의 차단 인스턴스를 설정하고 **action** 속성을 **on**으로 설정하려면, 다음 명령을 실행합니다:

```
ccs -h node01.example.com --addfenceinst apc1 node01.example.com
APC-dual port=1 action=on
```

6. 두 번째 전원 공급의 다른 차단 인스턴스를 차단 방식에 추가하여, 인스턴스에 대한 **action** 속성을 **on**으로 지정합니다. 노드에 사용할 차단 장치, 인스턴스의 적용 대상 노드, 방식 이름, 노드 특유의 옵션을 지정해야 하며 **action** 속성을 **on**으로 지정해야 합니다.

```
ccs -h host --addfenceinst fencedevicename node method [options]
action=on
```

예를 들어, **APC-dual**이라는 첫 번째 인스턴스에 지정한 것과 동일한 방식을 사용하여 클러스터 노드 **node-01.example.com**를 차단하기 위해 **apc2**라는 차단 장치에 있는 APC 스위치 전원 포트 1을 사용하는 클러스터 노드 **node-01.example.com**에 있는 설정 파일의 두 번째 차단 인스턴스를 설정하고 **action** 속성을 **on**으로 설정하려면 다음 명령을 실행합니다:

```
ccs -h node01.example.com --addfenceinst apc2 node01.example.com
APC-dual port=1 action=on
```

예 5.5. “이중 전원 차단 장치를 추가한 후 **cluster.conf**”에서는 클러스터에 있는 각 노드에 대해 두 개의 전원 공급에 대한 차단 장치를 추가 한 후 **cluster.conf** 설정 파일을 보여주고 있습니다.

예 5.5. 이중 전원 차단 장치를 추가한 후 cluster.conf

```
<cluster name="mycluster" config_version="3">
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
```

```
            <method name="APC-dual">
              <device name="apc1" port="1"action="off"/>
              <device name="apc2" port="1"action="off"/>
              <device name="apc1" port="1"action="on"/>
              <device name="apc2" port="1"action="on"/>
             </method>
        </fence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
        <fence>
           <method name="APC-dual">
             <device name="apc1" port="2"action="off"/>
             <device name="apc2" port="2"action="off"/>
             <device name="apc1" port="2"action="on"/>
             <device name="apc2" port="2"action="on"/>
           </method>
        </fence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
        <fence>
           <method name="APC-dual">
             <device name="apc1" port="3"action="off"/>
             <device name="apc2" port="3"action="off"/>
             <device name="apc1" port="3"action="on"/>
             <device name="apc2" port="3"action="on"/>
           </method>
        </fence>
    </clusternode>
  </clusternodes>
  <fencedevices>
      <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="apc1" passwd="password_example"/>
      <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="apc2" passwd="password_example"/>
  </fencedevices>
  <rm>
  </rm>
</cluster>
```

클러스터의 모든 구성요소 설정 완료시 5.15절. “클러스터 노드에 설정 파일 전달 ” 에서 설명하고 있듯이 클러스터 설정 파일을 모든 노드에 동기화해야 합니다.

5.7.5. 차단 방식 및 차단 인스턴스 제거

클러스터 설정에서 차단 방식을 제거하려면 다음 명령을 실행합니다:

```
ccs -h host --rmmethod method node
```

예를 들어, 클러스터 노드 **node01.example.com**에 있는 클러스터 설정 파일에서 **node01.example.com**에 대해 설정한 **APC**라는 차단 방식을 제거하려면 다음 명령을 실행합니다:

```
ccs -h node01.example.com  --rmmethod APC node01.example.com
```

-

차단 방식에서 차단 장치의 모든 차단 인스턴스를 제거하려면 다음 명령을 실행합니다:

```
ccs -h host --rmfenceinst fencedevicename node method
```

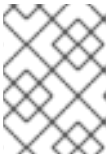
예를 들어, 클러스터 노드 **node01.example.com**에 있는 클러스터 설정 파일에서 **node01.example.com**에 대해 설정된 **APC-dual**라는 방식으로 부터 **apc1**라는 차단 장치의 모든 인스턴스를 제거하려면 다음 명령을 실행합니다:

```
ccs -h node01.example.com --rmfenceinst apc1 node01.example.com APC-dual
```

5.8. 장애 조치 도메인 설정

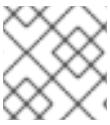
장애 조치 도메인은 노드 장애 발생 시 클러스터 서비스를 실행할 수 있는 클러스터 노드의 이름이 지정된 하부 집합입니다. 장애 조치 도메인은 다음과 같은 특징을 가지고 있습니다:

- 제한 없음 (Unrestricted) – 우선 순위를 갖는 멤버의 하위 집합을 지정할 수 있지만 이러한 도메인에 할당된 클러스터 서비스는 사용 가능한 멤버에서만 실행할 수 있습니다.
- 제한됨 (Restricted) – 특정 클러스터 서비스를 실행할 수 있는 멤버를 제한 할 수 있습니다. 제한된 장애 조치 도메인에서 사용할 수 있는 멤버가 없을 경우, 클러스터 서비스는 (수동으로 또는 클러스터 소프트웨어로도) 시작될 수 없습니다.
- 순서 없음 (Unordered) – 클러스터 서비스가 순서 없이 장애 조치 도메인에 할당되면, 클러스터 서비스가 실행되는 멤버는 우선 순위 없이 사용 가능한 장애 조치 도메인 멤버에서 선택됩니다.
- 순서 있음 (Ordered) – 장애 조치 도메인의 구성원 사이에서 우선 순위를 지정할 수 있습니다. 목록 상위에 위치하는 멤버가 최고 우선 순위를 갖는 것이고 그 다음으로 목록에 있는 두 번째 멤버가 우선 순위를 갖게 됩니다.
- 장애 복구 (Failback) – 장애 조치 도메인에 있는 서비스가 노드 장애 발생 이전에 원래 실행되고 있던 노드에 장애를 복구할 지에 대한 여부를 지정할 수 있습니다. 이러한 기능 설정은 노드가 반복적으로 장애가 발생하고 우선 순위가 있는 장애 조치 도메인의 일부일 경우에 유용합니다. 이러한 상황에서 노드가 장애 조치 도메인에 있는 우선 순위를 갖는 노드일 경우, 서비스가 우선 순위를 갖는 노드와 다른 노드 사이에서 장애 조치 및 장애 복구를 반복하여 성능에 심각한 영향을 미칠 수 있습니다.



참고

장애 복구 특징은 우선 순위가 있는 장애 조치가 설정되어 있을 때만 사용 가능합니다.



참고

현재 실행 중인 서비스에서 장애 조치 도메인 설정 변경은 영향을 미치지 않습니다.

참고

장애 조치 도메인은 운용에 필요하지 *않습니다*.

기본값으로 장애 조치 도메인은 제한 없음 (unrestricted) 및 순서 없음 (unordered)입니다.

여러 멤버가 있는 클러스터에서 제한이 있는 장애 조치 도메인을 사용할 경우 클러스터 서비스 (예:

httpd)를 실행하기 위해 클러스터 구성 작업을 최소화할 수 있습니다. 이는 클러스터 서비스를 실행하는 모든 멤버에 대해 동일한 설정을 구성해야 합니다. 클러스터 서비스를 실행하기 위해 전체 클러스터를 구성하는 대신 클러스터 서비스와 관련된 제한이 있는 장애 조치 도메인에 있는 구성원에만 설정할 수 있습니다.

참고

우선 순위를 갖는 멤버를 설정하려면, 하나의 클러스터 멤버로만 구성된 제한 없는 장애 조치 도메인을 생성합니다. 이렇게 하면 클러스터 서비스가 주로 클러스터 멤버 (우선 순위를 갖는 멤버)에서 실행되게 하지만 클러스터 서비스가 다른 멤버 중 하나에서 장애 조치를 수행할 수 있게 합니다.

장애 조치 도메인을 설정하려면, 다음 절차를 수행합니다:

1. 장애 조치 도메인을 추가하려면 다음 명령을 실행합니다:

```
ccs -h host --addfailoverdomain name [restricted] [ordered]
[nofailback]
```

참고

이름을 클러스터에서 사용되는 다른 이름과 그 목적에 있어서 구분할 수 있도록 충분히 설명적으로 합니다.

예를 들어, 다음 명령은 제한 없고 순서가 있으며 장애 복구 가능한 **example_pri**라는 장애 조치 도메인을 **node-01.example.com**에 설정합니다:

```
ccs -h node-01.example.com --addfailoverdomain example_pri ordered
```

2. 노드를 장애 조치 도메인에 추가하려면, 다음 명령을 실행합니다:

```
ccs -h host --addfailoverdomainnode failoverdomain node priority
```

예를 들어, **node-01.example.com**의 설정 파일에서 장애 조치 도메인 **example_pri**을 설정하여 우선 순위 1 인 **node-01.example.com**, 우선 순위 2인 **node-02.example.com**, 우선 순위 3인 **node-03.example.com**이 포함되도록 하려면 다음 명령을 실행합니다:

```
ccs -h node-01.example.com --addfailoverdomainnode example_pri node-
01.example.com 1
ccs -h node-01.example.com --addfailoverdomainnode example_pri node-
02.example.com 2
ccs -h node-01.example.com --addfailoverdomainnode example_pri node-
03.example.com 3
```

다음 명령을 사용하여 클러스터에 설정된 모든 장애 조치 도메인과 장애 조치 도메인 노드를 나열할 수 있습니다:

```
ccs -h host --lsfailoverdomain
```

장애 조치 도메인을 삭제하려면, 다음 명령을 실행합니다:

```
ccs -h host --rmfailoverdomain name
```

장애 조치 도메인에서 노드를 제거하려면 다음 명령을 실행합니다:

```
ccs -h host --rmfailoverdomainnode failoverdomain node
```

클러스터의 모든 구성요소 설정 완료시 5.15절. "클러스터 노드에 설정 파일 전달 " 에서 설명하고 있듯이 클러스터 설정 파일을 모든 노드에 동기화해야 합니다.

5.9. 글로벌 클러스터 리소스 설정

두 가지 유형의 리소스를 설정할 수 있습니다:

- 글로벌 (Global) – 클러스터에 있는 모든 서비스에서 사용 가능한 리소스입니다.
- 특정 서비스 (Service-specific) – 하나의 서비스에만 사용 가능한 리소스입니다.

현재 클러스터에 설정된 리소스와 서비스 목록을 보려면 다음 명령을 실행합니다:

```
ccs -h host --lsservices
```

글로벌 클러스터 리소스를 추가하려면 다음 명령을 실행합니다. 5.10절. "클러스터에 클러스터 서비스 추가 "에서 설명하고 있듯이 서비스를 설정할 때 특정 서비스에 대한 로컬 리소스를 추가할 수 있습니다.

```
ccs -h host --addresource resourcetype [resource options]
```

예를 들어 다음 명령은 글로벌 파일 시스템 리소스를 **node01.example.com**에 있는 클러스터 설정 파일에 추가합니다. 리소스 이름은 **web_fs**이고, 파일 시스템 장치는 **/dev/sdd2**이며, 파일 시스템 마운트 지점은 **/var/www**, 파일 시스템 유형은 **ext3**입니다.

```
ccs -h node01.example.com --addresource fs name=web_fs device=/dev/sdd2
mountpoint=/var/www fstype=ext3
```

사용 가능한 리소스 유형 및 리소스 옵션에 대한 자세한 내용은 부록 B. *HA 리소스 매개 변수*에서 참조하십시오.

글로벌 리소스를 삭제하려면 다음 명령을 실행합니다:

```
ccs -h host --rmresource resourcetype [resource options]
```

기존 글로벌 리소스의 매개 변수를 수정해야 할 경우, 리소스를 삭제하고 이를 다시 설정할 수 있습니다.

클러스터의 모든 구성요소 설정 완료시 5.15절. "클러스터 노드에 설정 파일 전달 " 에서 설명하고 있듯이 클러스터 설정 파일을 모든 노드에 동기화해야 합니다.

5.10. 클러스터에 클러스터 서비스 추가

클러스터에 클러스터 서비스를 설정하려면 다음 단계를 수행합니다:

1. 다음 명령을 사용하여 클러스터에 서비스를 추가합니다:

   ```
   ccs -h host --addservice servicename [service options]
   ```

참고

클러스터에 있는 다른 서비스와 명확하게 구별할 수 있도록 서비스에 설명적인 이름을 사용합니다.

클러스터 설정에 서비스를 추가할 때 다음 속성을 설정합니다:

- **autostart** – 클러스터가 시작될 때 서비스를 자동으로 시작할지에 대한 여부를 지정합니다. 활성화하려면 "1"을 비활성화하려면 "0"을 사용합니다. 디폴트 값은 활성화입니다.
- **domain** – 장애 조치 도메인을 지정합니다. (필요한 경우)
- **exclusive** – 다른 서비스가 실행되고 있지 않는 노드에서만 서비스를 실행하는 정책을 지정합니다.
- **recovery** – 서비스 복구 정책을 지정합니다. 옵션에는 서비스 재배치, 다시 시작, 비활성화, 다시 시작-비활성화가 있습니다. 다시 시작 복구 정책은 서비스를 다른 노드로 재배치하기 전 시스템이 장애가 발생한 서비스를 다시 시작 시도해야 함을 나타냅니다. 재배치 정책은 시스템이 다른 노드에서 서비스를 다시 시작 시도해야 함을 나타냅니다. 비활성화 정책은 구성 요소에 장애가 발생할 경우 시스템이 리소스 그룹을 비활성화해야 함을 나타냅니다. 다시 시작-비활성화 정책은 서비스에 장애가 발생할 경우 시스템은 서비스를 다시 시작 시도해야 하지만 서비스 재시작에 실패하면 서비스가 클러스터에 있는 다른 호스트로 이동하지 않고 비활성화됨을 나타냅니다.

  서비스의 복구 정책으로 **다시 시작 (Restart)** 또는 **다시 시작-비활성화 (Restart-Disable)** 를 선택한 경우, 서비스를 이동 또는 비활성화하기 전 까지 다시 시작 실패의 최대 횟수를 지정할 수 있으며 다시 시작을 잊어버린 후 시간을 초 단위로 지정할 수 있습니다.

예를 들어, 장애 조치 도메인 **example_pri**을 사용하여 **example_apache**라는 클러스터 노드 **node-01.example.com**에 있는 설정 파일에 서비스를 추가하려면 다음 명령을 실행합니다:

```
ccs -h node-01.example.com --addservice example_apache
domain=example_pri recovery=relocate
```

클러스터의 서비스를 설정할 때 클러스터에 사용 가능한 서비스 및 각 서비스에서 사용 가능한 옵션 목록을 검색할 때 유용할 수 있습니다. **ccs** 명령을 사용하여 사용 가능한 서비스 및 옵션 목록을 출력하는 방법은 5.11절. “사용 가능한 클러스터 서비스 목록 나열 ” 에서 참조하십시오.

2. 다음 명령을 사용하여 서비스에 리소스를 추가합니다:

```
ccs -h host --addsubservice servicename subservice [service options]
```

사용하려는 리소스 유형에 따라 서비스를 글로벌 (global) 또는 특정 서비스 (service-specific) 리소스로 배치할 수 있습니다. 글로벌 리소스를 추가하려면, **ccs**의 **--addsubservice** 옵션을 사용하여 리소스를 추가합니다. 예를 들어, **node-01.example.com**에 있는 클러스터 설정 파일에 **web_fs**라는 글로벌 파일 시스템 리소스를 **example_apache**라는 서비스에 추가하려면, 다음 명령을 실행합니다:

```
ccs -h node01.example.com --addsubservice example_apache fs
ref=web_fs
```

특정 서비스 리소스를 서비스에 추가하려면 모든 서비스 옵션을 지정해야 합니다. 예를 들어, 이전에 **web_fs**를 글로벌 서비스로 정의하지 않은 경우, 다음 명령을 사용하여 특정 서비스 리소스로 이를 추가할 수 있습니다:

```
ccs -h node01.example.com --addsubservice example_apache fs
name=web_fs device=/dev/sdd2 mountpoint=/var/www fstype=ext3
```

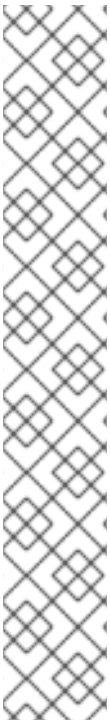
3. 서비스에 자식 리소스를 추가하려면 **ccs** 명령의 **--addsubservice** 옵션을 사용하여 서비스 옵션을 지정합니다.

   의존성 트리 구조에서 서비스를 추가할 필요가 있는 경우, 콜론 (":")을 사용하여 요소를 분리하고 괄호를 사용하여 동일한 유형의 하위 서비스를 식별할 수 있습니다. 다음 예제에서는 세 번째 **nfsclient** 서비스를 **nfsclient** 서비스의 하위 서비스로 추가하고 있습니다. 이는 서비스 자체로 **nfsclient** 서비스의 하위 서비스이며 이는 **service_a**라는 서비스의 하위 서비스입니다:

```
ccs -h node01.example.com --addsubservice service_a
nfsclient[1]:nfsclient[2]:nfsclient
```

참고

Samba 서비스 리소스를 추가하는 경우, 다른 리소스의 자식으로서가 *아닌* 이를 서비스에 직접 추가합니다.



참고

클러스터 서비스에서 사용하는 IP 서비스 리소스가 있는지 확인하려면 클러스터 노드에서 (폐지된 **ifconfig** 명령이 아니라) **/sbin/ip addr show** 명령을 사용할 수 있습니다. 다음은 클러스터 서비스가 실행되고 있는 노드에서 **/sbin/ip addr show** 명령을 실행하였을 경우의 출력 결과를 보여줍니다:

```
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1356 qdisc pfifo_fast
qlen 1000
    link/ether 00:05:5d:9a:d8:91 brd ff:ff:ff:ff:ff:ff
    inet 10.11.4.31/22 brd 10.11.7.255 scope global eth0
    inet6 fe80::205:5dff:fe9a:d891/64 scope link
    inet 10.11.4.240/22 scope global secondary eth0
       valid_lft forever preferred_lft forever
```

서비스 및 모든 하위 서비스를 제거하려면 다음 명령을 실행합니다:

```
ccs -h host --rmservice servicename
```

하위 서비스를 제거하려면 다음 명령을 실행합니다:

```
ccs -h host --rmsubservice servicename subservice [service options]
```

클러스터의 모든 구성요소 설정 완료시 5.15절. "클러스터 노드에 설정 파일 전달 " 에서 설명하고 있듯이 클러스터 설정 파일을 모든 노드에 동기화해야 합니다.

5.11. 사용 가능한 클러스터 서비스 목록 나열

ccs 명령을 사용하여 클러스터에서 사용 가능한 서비스 목록을 출력할 수 있습니다. 또한 **ccs** 명령을 사용하여 특정 서비스 유형에 지정할 수 있는 옵션 목록도 출력할 수 있습니다.

클러스터에서 현재 사용 가능한 클러스터 서비스 목록을 출력하려면 다음 명령을 실행합니다:

```
ccs -h host --lsserviceopts
```

예를 들어, 다음과 같은 명령은 클러스터 노드 **node1**에서 사용 가능한 클러스터 서비스를 나열합니다. 예시 출력 결과는 다음과 같습니다.

```
[root@ask-03 ~]# ccs -h node1 --lsserviceopts
service - Defines a service (resource group).
ASEHAagent - Sybase ASE Failover Instance
SAPDatabase - SAP database resource agent
SAPInstance - SAP instance resource agent
apache - Defines an Apache web server
clusterfs - Defines a cluster file system mount.
fs - Defines a file system mount.
ip - This is an IP address.
lvm - LVM Failover script
mysql - Defines a MySQL database server
named - Defines an instance of named server
netfs - Defines an NFS/CIFS file system mount.
nfsclient - Defines an NFS client.
nfsexport - This defines an NFS export.
nfsserver - This defines an NFS server resource.
openldap - Defines an Open LDAP server
oracledb - Oracle 10g Failover Instance
orainstance - Oracle 10g Failover Instance
oralistener - Oracle 10g Listener Instance
postgres-8 - Defines a PostgreSQL server
samba - Dynamic smbd/nmbd resource agent
script - LSB-compliant init script as a clustered resource.
tomcat-6 - Defines a Tomcat server
vm - Defines a Virtual Machine
action - Overrides resource action timings for a resource instance.
```

특정 서비스 유형을 지정할 수 있는 옵션 목록을 출력하려면 다음 명령을 실행합니다:

```
ccs -h host --lsserviceopts service_type
```

예를 들어 다음 명령은 **vm** 서비스의 서비스 옵션을 나열합니다.

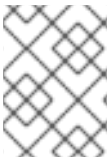
```
[root@ask-03 ~]# ccs -f node1 --lsserviceopts vm
vm - Defines a Virtual Machine
  Required Options:
    name: Name
  Optional Options:
    domain: Cluster failover Domain
    autostart: Automatic start after quorum formation
    exclusive: Exclusive resource group
    recovery: Failure recovery policy
    migration_mapping: memberhost:targethost,memberhost:targethost ..
    use_virsh: If set to 1, vm.sh will use the virsh command to manage
virtual machines instead of xm. This is required when using non-Xen
```

```
virtual machines (e.g. qemu / KVM).
    xmlfile: Full path to libvirt XML file describing the domain.
    migrate: Migration type (live or pause, default = live).
    path: Path to virtual machine configuration files.
    snapshot: Path to the snapshot directory where the virtual machine
image will be stored.
    depend: Top-level service this depends on, in service:name format.
    depend_mode: Service dependency mode (soft or hard).
    max_restarts: Maximum restarts for this service.
    restart_expire_time: Restart expiration time; amount of time before a
restart is forgotten.
    status_program: Additional status check program
    hypervisor: Hypervisor
    hypervisor_uri: Hypervisor URI (normally automatic).
    migration_uri: Migration URI (normally automatic).
    __independent_subtree: Treat this and all children as an independent
subtree.
    __enforce_timeouts: Consider a timeout for operations as fatal.
    __max_failures: Maximum number of failures before returning a failure
to a status check.
    __failure_expire_time: Amount of time before a failure is forgotten.
    __max_restarts: Maximum number restarts for an independent subtree
before giving up.
    __restart_expire_time: Amount of time before a failure is forgotten
for an independent subtree.
```

5.12. 가상 머신 리소스

가상 머신 리소스는 다른 클러스터 리소스와 다르게 설정됩니다. 특히 이는 서버 정의로 그룹화되지 않습니다. Red Hat Enterprise Linux 6.2 릴리즈 이후에서 **ccs** 명령으로 클러스터에 있는 가성 머신을 설정할 때 **--addvm** (**addservice** 옵션 대신) 옵션을 사용할 수 있습니다. 이는 **vm** 리소스가 클러스터 설정 파일에 있는 **rm** 클러스터 노드 아래에 직접 정의되게 합니다.

가상 머신 리소스에는 최소한 **name** 및 **path** 속성이 필요합니다. **name** 속성은 **libvirt** 도메인 이름과 일치해야 하고 **path** 속성은 공유 가상 머신 정의가 저장된 디렉토리를 지정해야 합니다.



참고

클러스터 설정 파일에 있는 **path** 속성은 개별 파일로의 경로가 아닌 경로 지정 또는 디렉토리 이름입니다.

가상 머신 정의가 **/mnt/vm_defs**라는 공유 디렉토리에 저장되어 있는 경우 다음과 같은 명령은 **guest1**이라는 가상 머신을 정의하게 됩니다:

```
# ccs -h node1.example.com --addvm guest1 path=/mnt/vm_defs
```

이 명령을 실행하면 다음과 같은 행이 **cluster.conf** 파일에 있는 **rm** 설정 노드에 추가됩니다:

```
<vm name="guest1" path="/mnt/vm_defs"/>
```

5.13. 쿼럼 (QUORUM) 디스크 설정

참고

Quorum-disk 매개 변수와 heuristics는 사이트 환경과 필요한 특정 요구 사항에 따라 달라집니다. quorum-disk 매개 변수와 heuristics의 사용을 이해하려면 qdisk(5) man 페이지를 참조하십시오. 쿼럼 디스크의 이해와 사용에 관한 지원이 필요하신 경우 Red Hat 지원 담당자에게 문의해 주십시오.

다음 명령을 사용하여 쿼럼 디스크 사용을 위해 시스템을 설정합니다:

```
ccs -h host --setquorumd [quorumd options]
```

5.1.5절. "이전 설정을 덮어쓰기하는 명령 " 에서 설명하고 있듯이 이 명령은 **--setquorumd** 옵션으로 설정할 수 있는 기타 모든 속성을 기본값으로 재설정하는 점에 유의하십시오.

표 5.1. "쿼럼 (Quorum) 디스크 옵션 " 에서는 설정해야 하는 쿼럼 디스크 옵션의 의미를 요약하고 있습니다. 쿼럼 디스크 매개 변수의 전체 목록은 **/usr/share/cluster/cluster.rng**에 있는 클러스터 스키마와 **/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html**에 있는 주석 스키마를 참조하십시오.

표 5.1. 쿼럼 (Quorum) 디스크 옵션

매개 변수	설명
interval	읽기/쓰기 사이클의 빈도, 초 단위
votes	점수가 충분히 높을 때 쿼럼 데몬이 **cman**에 공고하는 투표 수
tko	노드가 종료 선언을 무시해야 하는 사이클 수
min_score	노드가 "실행 중"(alive)이라고 간주되는데 필요한 최소 점수. 생략되어 있거나 **0**으로 설정되는 경우, 기본 함수 **floor((*n*+1)/2)**가 사용됩니다. 여기서 *n*은 휴리스틱 점수의 합계입니다. **최소 점수 (Minimum Score)** 값은 휴리스틱 점수의 합계를 초과해서는 안되며, 초과한 경우 쿼럼 디스크를 사용할 수 없습니다.
device	쿼럼 데몬이 사용하는 스토리지 장치입니다. 장치는 모든 노드에서 동일해야 합니다.
label	**mkqdisk** 유틸리티에 의해 생성되는 쿼럼 디스크 레이블을 지정합니다. 이 필드에 항목이 있는 경우 레이블은 **Device** 필드를 덮어쓰게 됩니다. 이 필드가 사용되면 쿼럼 데몬은 **/proc/partitions**를 읽어 발견된 모든 블록 장치에 있는 qdisk 서명을 확인하고, 레이블을 지정된 레이블과 비교합니다. 이는 쿼럼 장치 이름이 노드 사이에 다르게 되어 있는 설정에서 유용합니다.

다음 명령을 사용하여 쿼럼 디스크의 휴리스틱스를 설정합니다:

```
ccs -h host --addheuristic [heuristic options]
```

표 5.2. "쿼럼 디스크 휴리스틱스 " 에서는 설정해야 하는 쿼럼 디스크 휴리스틱스의 의미에 대해 요약하고 있습니다.

표 5.2. 쿼럼 디스크 휴리스틱스

매개 변수	설명
program	이 휴리스틱을 사용할 수 있는지 확인하기 위해 사용되는 프로그램으로의 경로입니다. **/bin/sh -c**로 실행할 수 있는 것이면 무엇이든 상관없습니다. 반환 값이 0은 성공을 나타내며 그 외의 값은 실패를 의미합니다. 이 매개 변수는 쿼럼 디스크를 사용하기 위해 필요합니다.
interval	휴리스틱이 투표하는 빈도 (초)입니다. 모든 휴리스틱의 기본 간격은 2초입니다.
score	휴리스틱의 중요성. 휴리스틱 점수를 지정할 때는 주의해야 합니다. 각 휴리스틱의 기본 점수는 1입니다.
tko	휴리스틱이 사용 불가능을 선언하기 전 까지 필요한 연속 실패 수입니다.

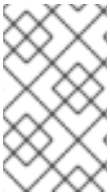
시스템에 설정된 쿼럼 디스크 옵션 및 휴리스틱스 목록을 확인하려면 다음 명령을 실행합니다:

```
ccs -h host --lsquorum
```

휴리스틱 옵션에 의해 지정된 휴리스틱을 제거하려면 다음 명령을 실행합니다:

```
ccs -h host rmheuristic [heuristic options]
```

클러스터의 모든 구성요소 설정 완료시 5.15절. “클러스터 노드에 설정 파일 전달 ” 에서 설명하고 있듯이 클러스터 설정 파일을 모든 노드에 동기화해야 합니다.



참고

동기화 및 활성화로 업데이트된 클러스터 설정 파일을 전달 및 활성화합니다. 하지만 쿼럼 디스크가 작동하게 하려면 클러스터를 다시 시작하여 (6.2절. “클러스터 시작 및 중지 ” 참조) 각 노드에서 **qdiskd** 데몬을 다시 시작해야 합니다.

5.14. 기타 다른 클러스터 설정

다음 부분에서는 다음과 같은 설정을 위해 **ccs** 명령을 사용하는 방법에 대해 설명합니다:

- 5.14.1절. “클러스터 설정 버전 ”
- 5.14.2절. “멀티캐스트 설정 ”
- 5.14.3절. “2 노드 클러스터 설정 ”
- 5.14.4절. “로깅 ”
- 5.14.5절. “중복 링 프로토콜 설정 ”

ccs 명령을 사용하여 **totem** 옵션, **dlm** 옵션, **rm** 옵션, **cman** 옵션을 포함한 고급 클러스터 설정 매개 변수를 구성할 수 있습니다. 이러한 매개 변수를 설정하는 방법에 대한 자세한 내용은 **ccs**(8) man 페이지 및 **/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html**에 있는 주석 클러스터 설정 파일 스키마를 참조하십시오.

클러스터에 설정된 다른 클러스터 속성 목록을 확인하려면 다음 명령을 실행합니다:

```
ccs -h host --lsmisc
```

5.14.1. 클러스터 설정 버전

클러스터 설정 파일에는 클러스터 설정 버전 값이 포함됩니다. 설정 버전 값은 클러스터 설정 파일을 생성할 때 기본값으로 **1**로 설정되고 클러스터 설정 파일을 수정할 때 마다 자동으로 증가합니다. 하지만 다른 값으로 설정해야 하는 경우 다음 명령을 사용하여 값을 지정할 수 있습니다:

```
ccs -h host --setversion n
```

다음 명령을 사용하여 기존 클러스터 설정 파일에 있는 현재 설정 버전 값을 얻을 수 있습니다:

```
ccs -h host --getversion
```

설정 버전 값을 클러스터에 있는 모든 노드의 클러스터 설정 파일에서 하나 증가시키려면 다음 명령을 실행합니다:

```
ccs -h host --incversion
```

5.14.2. 멀티캐스트 설정

클러스터 설정 파일에 멀티캐스트 주소를 지정하지 않을 경우, Red Hat 고가용성 추가 기능 소프트웨어는 클러스터 ID를 기반으로하여 생성합니다. 이는 주소의 하위 16 비트를 생성하여 이를 IP 프로토콜이 IPv4 또는 IPv6 인지에 대한 여부에 따라 주소의 상위 부분에 추가합니다:

- IPv4 용 – 형성되는 주소는 239.192입니다. 이에 Red Hat 고가용성 추가 기능 소프트웨어에 의해 생성된 하위 16 비트가 추가됩니다.
- IPv6 용 – 형성되는 주소는 FF15::입니다. 이에 Red Hat 고가용성 추가 기능 소프트웨어에 의해 생성된 하위 16 비트가 추가됩니다.

참고

클러스터 ID는 각 클러스터에 대해 **cman**이 생성하는 고유한 식별자입니다. 클러스터 ID를 확인하려면, 클러스터 노드에서 **cman_tool status** 명령을 실행합니다.

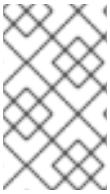
클러스터 설정파일에 있는 멀티캐스트 주소를 수동으로 지정하려면 다음 명령을 실행합니다:

```
ccs -h host --setmulticast multicastaddress
```

5.1.5절. “이전 설정을 덮어쓰기하는 명령 ” 에서 설명하고 있듯이 이 명령은 **--setmulticast** 옵션으로 설정할 수 있는 기타 모든 속성을 기본값으로 재설정하는 점에 유의하십시오.

멀티캐스트 주소를 지정하려면, **cman**을 사용하는 239.192.x.x 시리즈 (또는 IPv6의 경우 FF15::)를 사용해야 합니다. 그렇지 않으면 범위 이외의 멀티캐스트 주소의 사용으로 예측할 수 없는 결과를 일으킬 수 있습니다. 예를 들어, 224.0.0.x (이는 "네트워크 상의 모든 호스트")의 사용으로 올바르게 라우트되지 않거나 또는 일부 하드웨어에서는 전혀 라우트되지 않을 수 있습니다.

멀티 캐스트 주소를 지정하거나 수정하면 이를 적용하기 위해 클러스터를 다시 시작해야 합니다. **ccs** 명령을 사용하여 클러스터를 시작 및 중지하는 방법은 6.2절. “클러스터 시작 및 중지 ” 에서 참조하십시오.



참고

멀티캐스트 주소를 지정하려면 클러스터 패킷이 통과하는 라우터의 설정을 확인해야 합니다. 일부 라우터는 주소를 인식하는데 시간이 오래 걸릴 수 있으므로 클러스터 성능에 심각한 영향을 미칩니다.

설정 파일에서 멀티캐스트 주소를 제거하려면 **ccs**의 **--setmulticast** 옵션을 사용하지만 멀티캐스트 주소는 지정하지 않습니다:

```
ccs -h host --setmulticast
```

5.14.3. 2 노드 클러스터 설정

2 노드 클러스터를 설정하는 경우, 다음 명령을 실행하여 단일 노드가 쿼럼 (정족수)을 유지하게 합니다 (예: 하나의 노드가 실패했을 경우):

```
ccs -h host --setcman two_node=1 expected_votes=1
```

5.1.5절. "이전 설정을 덮어쓰기하는 명령 " 에서 설명하고 있듯이 이 명령은 **--setcman** 옵션으로 설정할 수 있는 기타 모든 속성을 기본값으로 재설정하는 점에 유의하십시오.

ccs --setcman 명령을 사용하여 **two_node** 옵션을 추가, 제거, 수정할 경우 이러한 변경 사항을 적용하기 위해 클러스터를 다시 시작해야 합니다. **ccs** 명령을 사용하여 클러스터를 시작 또는 중지하는 방법은 6.2절. "클러스터 시작 및 중지 " 에서 참조하십시오.

5.14.4. 로깅

클러스터에 있는 모든 데몬의 디버깅을 활성화하거나 특정 클러스터 프로세스에 대한 로깅을 활성화할 수 있습니다.

모든 데몬의 로깅을 활성화하려면 다음 명령을 실행합니다. 디폴트로 로깅은 **/var/log/cluster/*daemon*.log** 파일로 이동합니다.

```
ccs -h host --setlogging [logging options]
```

예를 들어 다음 명령은 모든 데몬의 디버깅을 활성화합니다.

```
# ccs -h node1.example.com --setlogging debug=on
```

5.1.5절. "이전 설정을 덮어쓰기하는 명령 " 에서 설명하고 있듯이 이 명령은 **--setlogging** 옵션으로 설정할 수 있는 기타 모든 속성을 기본값으로 재설정하는 점에 유의하십시오.

개별 클러스터 프로세스에 대한 디버깅을 활성화하려면 다음 명령을 실행합니다. 데몬 마다 로깅 설정은 글로벌 설정을 덮어쓰기합니다.

```
ccs -h host --addlogging [logging daemon options]
```

예를 들어 다음 명령은 **corosync** 및 **fenced** 데몬의 디버깅을 활성화합니다.

```
# ccs -h node1.example.com --addlogging name=corosync debug=on
# ccs -h node1.example.com --addlogging name=fenced debug=on
```

개별 데몬의 로그 설정을 제거하려면 다음 명령을 사용합니다.

```
ccs -h host --rmlogging name=clusterprocess
```

예를 들어 다음 명령은 **fenced** 데몬의 데몬 고유의 로그 설정을 제거합니다.

```
ccs -h host --rmlogging name=fenced
```

글로벌 및 데몬별 로깅 모두에 대해 설정할 수 있는 로깅 및 추가 로깅 옵션을 활성화할 수 있는 로깅 데몬 목록은 **cluster.conf**(5) man 페이지에서 참조하십시오.

클러스터의 모든 구성요소 설정 완료시 5.15절. "클러스터 노드에 설정 파일 전달 " 에서 설명하고 있듯이 클러스터 설정 파일을 모든 노드에 동기화해야 합니다.

5.14.5. 중복 링 프로토콜 설정

Red Hat Enterprise Linux 6.4에서 Red Hat 고가용성 애드온은 중복 링 프로토콜 설정을 지원합니다. 중복 링 프로토콜을 사용할 때 7.6절. "중복 링 프로토콜 설정 " 에서 설명하고 있듯이 여러 가지 고려해야 할 사항이 있습니다.

중복 링 프로토콜에 사용할 두 번째 네트워크 인터페이스를 지정하려면 **ccs** 명령의 **--addalt** 옵션을 사용하여 노드의 대체 이름을 추가합니다:

```
ccs -h host --addalt node_name alt_name
```

예를 들어 다음 명령은 클러스터 노드 **clusternet-node1-eth1**의 다른 이름 **clusternet-node1-eth2**을 설정합니다:

```
# ccs -h clusternet-node1-eth1 --addalt clusternet-node1-eth1 clusternet-
node1-eth2
```

옵션으로 두 번째 링에 대해 멀티캐스트 주소, 포트, TTL을 수동으로 지정할 수 있습니다. 두 번째 링의 멀티캐스트 주소를 지정하는 경우 대체 멀티캐스트 주소 또는 대체 포트는 첫 번째 링의 멀티캐스트 주소와 달라야 합니다. 대체 포트를 지정하는 경우 시스템 자체가 작업을 수행하기 위해 포트 및 포트 1을 사용하므로 첫 번째 링과 두 번째 링의 포트 번호는 최소 두 개의 다른 것이어야 합니다. 대체 멀티캐스트 주소를 지정하지 않은 경우, 시스템은 두 번째 링에 대해 자동으로 다른 멀티캐스트 주소를 사용하게 됩니다.

두 번째 링에 다른 멀티캐스트 주소, 포트, TTL을 지정하려면 **ccs** 명령의 **--setaltmulticast** 옵션을 사용합니다:

```
ccs -h host --setaltmulticast [alt_multicast_address]
[alt_multicast_options].
```

예를 들어 다음 명령은 **clusternet-node1-eth1** 노드에 있는 **cluster.conf** 파일에 정의된 클러스터에 대한 다른 멀티캐스트 주소 239.192.99.88, 포트 888, TTL3을 설정합니다:

```
ccs -h clusternet-node1-eth1 --setaltmulticast 239.192.99.88 port=888
ttl=3
```

대체 멀티케스트 주소를 삭제하려면 **ccs** 명령의 **--setaltmulticast** 옵션을 지정하지만 멀티캐스트 주소를 지정하지 않습니다. 5.1.5절. "이전 설정을 덮어쓰기하는 명령 " 에서 설명하고 있듯이 이 명령을 실행하면 **--setaltmulticast** 옵션으로 설정할 수 있는 모든 다른 속성을 기본값으로 재설정한다는 점에 유의하십시오.

클러스터의 모든 구성 요소 설정을 완료하면, 5.15절. “클러스터 노드에 설정 파일 전달 ” 에서 설명하고 있듯이 클러스터 설정 파일을 모든 노드에 동기화해야 합니다.

5.15. 클러스터 노드에 설정 파일 전달

클러스터에 있는 노드 중 하나에서 클러스터 설정 파일을 생성 또는 편집한 후 동일한 파일을 모든 클러스터 노드에 전달하고 설정을 활성화해야 합니다.

다음 명령을 사용하여 클러스터 설정 파일을 전달 및 활성화합니다:

```
ccs -h host --sync --activate
```

호스트 클러스터 설정 파일에 지정된 모든 노드가 동일한 클러스터 설정 파일을 가지고 있는지 확인하려면 다음 명령을 실행합니다:

```
ccs -h host --checkconf
```

로컬 노드에 설정 파일을 생성하거나 편집하는 경우 다음 명령을 사용하여 클러스터에 있는 노드 중 하나로 파일을 전송합니다:

```
ccs -f file -h host --setconf
```

로컬 파일에 지정된 모든 노드가 동일한 클러스터 설정 파일을 갖는지를 확인하려면 다음 명령을 실행합니다:

```
ccs -f file --checkconf
```

6장. CCS로 RED HAT 고가용성 추가 기능 관리

다음 부분에서는 **ccs** 명령을 사용하여 Red Hat 고가용성 추가 기능을 운용하기 위한 다양한 관리 작업에 대해 설명합니다. **ccs** 명령은 Red Hat Enterprise Linux 6.1 릴리즈와 이후 버전에서 지원됩니다. 이는 다음과 같은 부분으로 구성되어 있습니다:

- 6.1절. “클러스터 노드 관리 ”
- 6.2절. “클러스터 시작 및 중지 ”
- 6.3절. “클러스터에 있는 문제를 진단 및 수정 ”

6.1. 클러스터 노드 관리

이 부분에서는 **ccs** 명령을 사용하여 다음과 같은 노드 관리 기능을 수행하는 방법에 대해 설명합니다:

- 6.1.1절. “노드가 클러스터를 탈퇴 또는 참여하는 원인 ”
- 6.1.2절. “실행중인 클러스터에 멤버 추가 ”

6.1.1. 노드가 클러스터를 탈퇴 또는 참여하는 원인

ccs 명령을 사용하여 노드에 있는 클러스터 서비스를 중지하여 노드가 클러스터를 탈퇴하게 할 수 있습니다. 노드가 클러스터를 탈퇴하는 것으로 인해 노드에서 클러스터 설정 정보가 제거되지 않습니다. 노드가 클러스터를 탈퇴하면 다시 시작할 때 노드가 클러스터에 자동으로 참여하지 못하게 합니다.

노드를 클러스터에서 탈퇴시키려면 다음 명령을 실행합니다. 이는 **-h** 옵션으로 지정된 노드에 있는 클러스터 서비스를 중지시킵니다:

```
ccs -h host --stop
```

노드에서 클러스터 서비스를 중지하면 해당 노드에서 실행 중인 모든 서비스는 장애 조치됩니다.

클러스터 설정에서 노드를 완전히 삭제하려면 5.4절. “클러스터 생성 ”에서 설명하고 있듯이 **ccs** 명령의 **--rmnode** 옵션을 사용합니다.

노드가 클러스터에 다시 참여하게 하려면 다음 명령을 실행합니다. 이는 **-h** 옵션으로 지정된 노드에서 클러스터 서비스를 시작합니다:

```
ccs -h host --start
```

6.1.2. 실행중인 클러스터에 멤버 추가

실행 중인 클러스터에 멤버를 추가하려면, 5.4절. “클러스터 생성 ”에서 설명하고 있듯이 클러스터에 노드를 추가합니다. 클러스터 설정 파일을 업데이트한 후, 클러스터에 있는 모든 노드에 해당 파일을 전달하고 5.15절. “클러스터 노드에 설정 파일 전달 ” 에서 설명하고 있듯이 새로운 클러스터 설정 파일을 활성화합니다.

6.2. 클러스터 시작 및 중지

ccs 명령을 사용하여 클러스터를 중지하려면 다음 명령을 사용하여 클러스터의 모든 노드에서 클러스터 서비스를 중지합니다:

```
ccs -h host --stopall
```

ccs 명령을 사용하여 실행되고 있지 않는 클러스터를 시작하려면 다음 명령을 사용하여 클러스터의 모든 노드에서 클러스터 서비스를 시작합니다:

```
ccs -h host --startall
```

6.3. 클러스터에 있는 문제를 진단 및 수정

클러스터에 있는 문제 진단 및 수정에 대한 자세한 내용은 9장. *클러스터에 있는 문제를 진단 및 수정*에서 참조하십시오. **ccs** 명령을 사용하여 몇 가지 간단한 검사를 수행할 수 있습니다.

호스트 클러스터 설정 파일에 지정된 모든 노드가 동일한 클러스터 설정 파일을 갖는지를 확인하려면, 다음 명령을 실행합니다:

```
ccs -h host --checkconf
```

로컬 노드에 있는 설정 파일을 생성 또는 편집하는 경우, 다음 명령을 사용하여 로컬 파일에 지정된 모든 노드가 동일한 클러스터 설정 파일을 갖는지 확인할 수 있습니다:

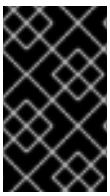
```
ccs -f file --checkconf
```

7장. 명령행 도구로 RED HAT 고가용성 추가 기능 설정

다음 부분에서는 클러스터 설정 파일 (**/etc/cluster/cluster.conf**)을 직접 편집하고 명령행 도구를 사용하여 Red Hat 고가용성 추가 기능을 설정하는 방법에 대해 설명합니다. 이 부분에서는 여기에서 제공하는 예제 파일을 시작으로 한번에 한 섹션 씩 설정 파일을 구축하는 방법에 대해 소개합니다. 여기에서 제공하는 예제 파일을 시작으로 하지 않고 **cluster.conf** man 페이지에서 스켈튼 설정 파일을 복사할 수도 있습니다. 하지만 이 방법은 이 부분에 있는 다음 절차에 있는 내용과 병행하여 진행할 필요는 없습니다. 다른 방법으로 클러스터 설정 파일을 생성하고 구성할 수 있으며, 이 장에서는 한 번에 한 섹션의 설정 파일을 구축하는 절차에 관해 설명합니다. 또한 이는 사용자의 클러스터링 요구에 부합하기 위한 설정 파일 개발의 출발점에 불과하다는 것을 염두해 두시기 바랍니다.

이는 다음과 같은 부분으로 구성되어 있습니다:

- 7.1절. “설정 작업 ”
- 7.2절. “기본적인 클러스터 설정 파일 생성 ”
- 7.3절. “차단 장치 설정 ”
- 7.4절. “장애 조치 도메인 설정 ”
- 7.5절. “HA 서비스 설정 ”
- 7.7절. “디버그 옵션 설정 ”
- 7.6절. “중복 링 프로토콜 설정 ”
- 7.8절. “설정 확인 ”



중요

고가용성 추가 기능의 사용이 자신의 요구에 부합하고 지원될 수 있는지 확인하십시오. 사용하기 전 설정을 확인하기 위해 Red Hat 담당자에게 문의하시기 바랍니다. 또한 설정 번인 (burn-in) 기간을 두어 장애 모드를 테스트하십시오.

중요

다음 부분에서는 일반적으로 사용되는 **cluster.conf** 요소와 속성을 참조합니다. **cluster.conf** 요소와 속성의 전체적 목록과 설명은 **/usr/share/cluster/cluster.rng**에 있는 클러스터 스키마와 **/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html** (예: **/usr/share/doc/cman-3.0.12/cluster_conf.html**)의 주석 스키마를 참조하십시오.

중요

다음 부분에서의 일부 단계는 클러스터 설정을 클러스터 전역에 전달하기 위해 **cman_tool version -r** 명령을 사용해야 합니다. 이 명령을 사용하려면 **ricci**가 실행되고 있어야 합니다. **ricci** 사용 시 특정 컴퓨터에서 처음으로 **ricci**와 통신할 때 암호가 필요합니다. **ricci** 서비스에 대한 자세한 내용은 2.13절. “**ricci** 사용 시 고려 사항 ” 에서 참조하십시오.

참고

이 부분에 있는 절차에는 부록 E. *명령행 도구 요약*에 나열된 명령행 도구의 일부 특정 명령을 포함하고 있습니다. 모든 명령 및 변수에 대한 자세한 내용은 각 명령행 도구의 man 페이지를 참조하십시오.

7.1. 설정 작업

명령행 도구로 Red Hat 고가용성 추가 기능 소프트웨어를 설정하는 방법은 다음과 같은 절차로 구성되어 있습니다:

1. 클러스터 생성. 7.2절. "기본적인 클러스터 설정 파일 생성 "에서 참조하십시오.
2. 차단 장치 설정. 7.3절. "차단 장치 설정 "에서 참조하십시오.
3. 장애 조치 도메인 설정. 7.4절. "장애 조치 도메인 설정 "에서 참조하십시오.
4. HA 서비스 설정. 7.5절. "HA 서비스 설정 "에서 참조하십시오.
5. 설정 확인. 7.8절. "설정 확인 "에서 참조하십시오.

7.2. 기본적인 클러스터 설정 파일 생성

클러스터 하드웨어가 제공되고, Red Hat Enterprise Linux 및 고가용성 추가 기능 소프트웨어가 설치되어 있는 경우, 클러스터 설정 파일 (**/etc/cluster/cluster.conf**)을 생성하여 고가용성 추가 기능의 실행을 시작할 수 있습니다. 다음에서는 단순히 출발점으로 차단 장치, 장애 조치 도메인, HA 서비스 없이 스켈튼 클러스터 설정 파일을 생성하는 방법에 대해 설명합니다. 그 후 설정 파일에서 이러한 부분을 설정하는 방법에 대해 설명합니다.

중요

여기에서는 단순히 클러스터 설정 파일을 생성하기 위한 중간 단계를 설명합니다. 이 결과로 생긴 파일에는 차단 장치가 없고 지원되는 설정은 고려되지 않습니다.

다음 절차에서는 스켈튼 클러스터 설정 파일을 생성 및 구성하는 방법에 대해 설명합니다. 궁극적으로 사용자의 클러스터에 대한 설정 파일은 노드 수, 차단 장치 유형, HA 서비스 유형 및 수량, 기타 특정 사이트의 요구에 따라 달라집니다.

1. 클러스터에 있는 노드 중 하나에서 예 7.1. "**cluster.conf** 예: 기본 설정 "에 있는 템플릿 예시를 사용하여 **/etc/cluster/cluster.conf**를 생성합니다.
2. **(옵션)** 2 노드 클러스터를 설정하는 경우, 설정 파일에 다음 행을 추가하여 단일 노드가 쿼럼을 유지하게 합니다. (예: 하나의 노드가 실패한 경우):

   ```
   <cman two_node="1" expected_votes="1"/>
   ```

   cluster.conf 파일에서 **two_node** 옵션을 추가 또는 삭제시 설정을 업데이트하면 변경 사항을 적용하기 위해 클러스터를 다시 시작해야 합니다. 클러스터 설정을 업데이트하는 방법에 대한 내용은 8.4절. "설정 업데이트 "에서 참조하십시오. **two_node** 옵션을 지정하는 예는 예 7.2. "**cluster.conf** 예: 기본적인 두 개의 노드 설정 "에서 참조하십시오.
3. **cluster** 속성을 사용하여 클러스터 이름 및 설정 버전 번호를 지정합니다: **name** 및 **config_version** (예 7.1. "**cluster.conf** 예: 기본 설정 " 또는 예 7.2. "**cluster.conf** 예: 기본적인 두 개의 노드 설정 " 참조).

4. **clusternodes** 부분에서 **clusternode** 속성을 사용하여 각 노드의 노드 이름 과 노드 ID를 지정합니다: **name** 및 **nodeid**.

5. **/etc/cluster/cluster.conf**를 저장합니다.

6. **ccs_config_validate** 명령을 사용하여 클러스터 스키마 (**cluster.rng**)에 대해 파일 유효성을 검증합니다. 예:

```
[root@example-01 ~]# ccs_config_validate
Configuration validates
```

7. 설정 파일을 각 클러스터 노드의 **/etc/cluster/**에 전달합니다. 예를 들어, **scp** 명령을 사용하여 파일을 다른 클러스터 노드에 전달할 수 있습니다.

참고

클러스터가 처음으로 생성되었을 때 이 방법으로 클러스터 설정 파일을 전달해야 합니다. 클러스터가 설치되어 실행되면, 클러스터 설정 파일은 **cman_tool version -r**을 사용하여 전달될 수 있습니다. 업데이트된 설정 파일을 전달하기 위해 **scp** 명령을 사용할 수 있지만, **scp** 명령을 사용하는 동안에는 모든 노드에 있는 클러스터 소프트웨어를 중지해야 합니다. 또한, **scp**를 통해 업데이트된 설정 파일을 전달하려면 **ccs_config_validate**를 실행해야 합니다.

참고

예시 설정 파일에는 다른 요소와 속성이 있지만 (예: **fence** 및 **fencedevices**) 이를 지금 배치할 필요는 없습니다. 이 장의 후반 부분에서 다른 요소와 속성을 지정하는 방법에 대해 설명합니다.

8. 클러스터를 시작합니다. 각각의 클러스터 노드는 다음과 같은 명령을 실행합니다:

service cman start

예:

```
[root@example-01 ~]# service cman start
Starting cluster:
   Checking Network Manager...                             [  OK
]
   Global setup...                                         [  OK
]
   Loading kernel modules...                               [  OK
]
   Mounting configfs...                                    [  OK
]
   Starting cman...                                        [  OK
]
   Waiting for quorum...                                   [  OK
]
   Starting fenced...                                      [  OK
]
   Starting dlm_controld...                                [  OK
]
```

```
   Starting gfs_controld...                                   [  OK
]
   Unfencing self...                                          [  OK
]
   Joining fence domain...                                    [  OK
]
```

9. 클러스터 노드에서 **cman_tool nodes**를 실행하여 노드가 클러스터에서 멤버로 작동하는지 확인합니다 (상태 컬럼 "Sts"에서 "M"로 표시). 예:

```
[root@example-01 ~]# cman_tool nodes
Node  Sts   Inc   Joined               Name
   1   M    548   2010-09-28 10:52:21  node-01.example.com
   2   M    548   2010-09-28 10:52:21  node-02.example.com
   3   M    544   2010-09-28 10:52:21  node-03.example.com
```

10. 클러스터가 실행되고 있을 경우 7.3절. “차단 장치 설정 ”를 실행합니다.

7.2.1. 기본적인 설정 예시

예 7.1. “**cluster.conf** 예: 기본 설정 ” 및 예 7.2. “**cluster.conf** 예: 기본적인 두 개의 노드 설정 ” (2 노드 클러스터)는 각각 출발점으로 매우 기본적인 클러스터 설정 파일의 샘플을 제공합니다. 이 장의 후반 부분에서는 차단 장치와 HA 서비스를 설정하는 방법을 설명합니다.

예 7.1. cluster.conf 예: 기본 설정

```
<cluster name="mycluster" config_version="2">
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
         </fence>
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
         </fence>
     </clusternode>
     <clusternode name="node-03.example.com" nodeid="3">
         <fence>
         </fence>
     </clusternode>
   </clusternodes>
   <fencedevices>
   </fencedevices>
   <rm>
   </rm>
</cluster>
```

예 7.2. cluster.conf 예: 기본적인 두 개의 노드 설정

```
<cluster name="mycluster" config_version="2">
   <cman two_node="1" expected_votes="1"/>
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
         </fence>
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
         </fence>
     </clusternode>
   </clusternodes>
   <fencedevices>
   </fencedevices>
   <rm>
   </rm>
</cluster>
```

7.2.2. 2 노드 클러스터에서 totem의 합의 (consensus) 값

2 노드 클러스터를 생성하여 나중에 클러스터에 추가 노드를 더하지 않고자 할 경우, **cluster.conf** 파일에 있는 **totem** 태그의 **합의** 값을 생략하여 **합의** 값이 자동으로 계산되게 합니다. **합의** 값이 자동으로 산출될 때 다음과 같은 규칙이 사용됩니다:

- 두 개 또는 그 이하의 노드가 있을 경우, **합의** 값은 최대 2000 msec에서 최저 200 msec을 갖는 (token * 0.2)가 됩니다.
- 세 개 또는 그 이상의 노드가 있는 경우, **합의** 값은 (token + 2000 msec)이 됩니다.

cman 유틸리티가 이렇게 합의 시간 제한을 설정하게 하면 나중에 2 노드에서 3 노드 (또는 그 이상)로 변경할 경우 클러스터를 다시 시작해야 합니다. 이는 token 시간 제한을 기준으로 합의 시간 제한 보다 큰 값으로 변경해야 하기 때문입니다.

2 노드 클러스터를 설정하고 있고 차후에 2개 이상의 노드로 업그레이드하고자 할 경우, 합의 시간 제한을 무시하고 2 노드에서 3 노드 (또는 그 이상)로 변경할 때 클러스터를 다시 시작하도록 요구하지 않게 할 수 있습니다. 이를 위해 **cluster.conf**에서 다음과 같이 실행합니다:

```
<totem token="X" consensus="X + 2000" />
```

설정 파서는 X + 2000을 자동으로 산출하는 것이 아님에 유의하십시오. 등식보다 정수 값을 사용해야 합니다.

2 노드 클러스터에 최적화된 합의 시간 제한을 사용하는 장점은 전체적 장애 복구 시간이 2 노드 경우에 대해 단축된다는 것입니다. 이는 합의 (consensus)가 token 시간 제한 함수가 아니기 때문입니다.

cman에서 2 노드 자동 탐색의 경우, 중요한 것은 물리적 노드이며 **cluster.conf** 파일에 있는 **two_node=1** 지시문의 존재가 아님에 유의하십시오.

7.3. 차단 장치 설정

차단 장치 설정은 (a) 클러스터에 하나 또는 그 이상의 차단 장치를 지정하고 (b) 각 노드에 하나 또는 그 이상의 차단 방식을 지정 (지정된 차단 장치 사용)하는 것으로 구성됩니다.

사용 설정에 필요한 차단 장치 및 차단 방식 유형을 기반으로 다음과 같이 **cluster.conf**를 설정합니다:

1. **fencedevices** 부분에서는 **fencedevice** 요소와 차단 장치 종속 속성을 사용하여 각 차단 장치를 지정합니다. 예 7.3. "**cluster.conf**에 추가된 APC 차단 장치 " 에서는 추가된 APC 차단 장치와 함께 설정 파일의 예를 보여주고 있습니다.

2. **clusternodes** 부분에서는 각 **clusternode** 섹션의 **fence** 요소에서 노드의 차단 방식을 지정합니다. **method** 속성, **name**을 사용하여 차단 방식 이름을 지정합니다. **device** 요소와 속성, **name**, 특정 차단 장치 매개 변수를 사용하여 차단 방식의 차단 장치를 지정합니다. 예 7.4. "**cluster.conf**에 추가된 차단 방식 " 에서는 클러스터에 있는 각 노드에 대해 하나의 차단 장치가 있는 차단 방식의 예를 보여주고 있습니다.

3. 비전원 (non-power) 차단 방식 (즉, SAN/스토리지 차단)의 경우, **clusternodes** 부분에 **unfence** 부분을 추가합니다. 이는 차단된 노드가 다시 시작할 때 까지 다시 활성화되지 않는지 확인합니다. 노드를 차단 해제하는 방법에 대한 자세한 내용은 **fence_node**(8) man 페이지에서 참조하십시오.

   unfence 부분에는 **fence** 부분에 있는 것과는 달리 **method** 부분이 들어 있지 않습니다. 하지만 이는 **device** 참조를 직접 포함하고 있어, "on" 또는 "enable"의 명시적 동작 (**action**)이 추가된 **fence**의 해당 장치 부분을 미러합니다. **fence**와 **unfence**에 의해 동일한 **fencedevice**는 참조되며 **device** 행과 동일한 노드 별 인수는 반복됩니다.

   action 속성을 "on" 또는 "enable"로 지정하면 시작할 때 노드를 활성화합니다. 예 7.4. "**cluster.conf**에 추가된 차단 방식 " 및 예 7.5. "**cluster.conf**: 노드 당 여러 차단 방식 " 에는 **unfence** 요소 및 속성 예제가 포함되어 있습니다.

   unfence 관한 보다 자세한 내용은 **fence_node** man 페이지를 참조하십시오.

4. 값을 증가시켜 **config_version** 속성을 업데이트합니다 (예: **config_version="2"**에서 **config_version="3">**로 변경)

5. **/etc/cluster/cluster.conf**를 저장합니다.

6. **(옵션) ccs_config_validate** 명령을 실행하여 클러스터 스키마 (**cluster.rng**)에 대해 업데이트된 파일의 유효성을 검사합니다. 예:

   ```
   [root@example-01 ~]# ccs_config_validate
   Configuration validates
   ```

7. **cman_tool version -r** 명령을 실행하여 나머지 클러스터 노드에 설정을 전달합니다. 이는 추가 유효성 검사도 실행하게 됩니다. 업데이트된 클러스터 설정 정보를 전달하려면 각 클러스터 노드에서 **ricci**가 실행되고 있어야 합니다.

8. 업데이트된 설정 파일이 전달되었는지 확인합니다.

9. 7.4절. "장애 조치 도메인 설정 " 으로 이동합니다.

필요한 경우, 노드 당 여러 차단 방식과 차단 방식 당 여러 차단 장치가 있는 복잡한 설정을 구성할 수 있습니다. 노드 마다 여러 차단 방식을 지정할 때, 첫 번째 방식을 사용하여 차단 실패할 경우, 차단 데몬인 **fenced**는 다음과 같은 방법을 시도하다가 성공할 때 까지 메소드 군을 차례로 반복하여 계속 실행합니다.

일부 경우, 노드 차단에는 두 개의 I/O 경로 또는 2 개의 파워 포트를 비활성해야 합니다. 이는 차단 방식에서 두 개 이상의 장치를 지정하여 수행할 수 있습니다. **fenced**는 각 차단 장치행에 대해 차단 에이전트를 한번 실행하지만, 차단이 성공적이라고 간주되려면 모두가 실행 성공해야 합니다.

보다 복잡한 설정은 "차단 장치 설정 예 "에서 보여주고 있습니다.

특정 차단 장치 설정에 관한 내용은 차단 장치 에이전트 man 페이지 (예: **fence_apc**의 man 페이지)에서 보실 수 있습니다. 또한, 부록 A. *차단 장치 매개 변수*에서는 차단 장치 매개 변수 내용, **/usr/sbin/**에서는 차단 에이전트 내용, **/usr/share/cluster/cluster.rng**에서는 클러스터 스키마 내용, **/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html** (예: **/usr/share/doc/cman-3.0.12/cluster_conf.html**)에서는 주석 스키마 내용을 확인할 수 있습니다.

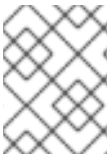
7.3.1. 차단 장치 설정 예

다음 예제에서는 노드 마다 하나의 차단 방식과 차단 방식 마다 하나의 차단 장치가 있는 간단설 설정을 보여주고 있습니다:

- 예 7.3. "**cluster.conf**에 추가된 APC 차단 장치 "
- 예 7.4. "**cluster.conf**에 추가된 차단 방식 "

다음 예시에서는 보다 복잡한 설정을 보여줍니다:

- 예 7.5. "**cluster.conf**: 노드 당 여러 차단 방식 "
- 예 7.6. "**cluster.conf**: 차단 장치, 멀티패스 다중 포트 "
- 예 7.7. "**cluster.conf**: 듀얼 전원 공급을 갖는 노드 차단 장치 "



참고

이 부분에 있는 예제가 전부는 아닙니다. 즉, 필요에 따라 차단 장치를 설정해야 하는 방법이 다를 수 있습니다.

예 7.3. cluster.conf에 추가된 APC 차단 장치

```
<cluster name="mycluster" config_version="3">
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
         </fence>
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
         </fence>
     </clusternode>
     <clusternode name="node-03.example.com" nodeid="3">
         <fence>
         </fence>
     </clusternode>
   </clusternodes>
   <fencedevices>
         <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
```

```
login="login_example" name="apc" passwd="password_example"/>
   </fencedevices>
   <rm>
   </rm>
</cluster>
```

이 예제에서 차단 장치 (**fencedevice**)는 **fencedevices** 요소에 추가되어, 차단 에이전트 (**agent**) 를 **fence_apc**로, IP 주소 (**ipaddr**)를 **apc_ip_example**로, 로그인 (**login**)을 **login_example**로, 차단 장치 이름 (**name**)을 **apc**로, 암호 (**passwd**)를 **password_example**로 지정하고 있습니다.

예 7.4. cluster.conf에 추가된 차단 방식

```
<cluster name="mycluster" config_version="3">
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
            <method name="APC">
              <device name="apc" port="1"/>
             </method>
         </fence>
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
            <method name="APC">
              <device name="apc" port="2"/>
            </method>
         </fence>
     </clusternode>
     <clusternode name="node-03.example.com" nodeid="3">
         <fence>
            <method name="APC">
              <device name="apc" port="3"/>
            </method>
         </fence>
     </clusternode>
   </clusternodes>
   <fencedevices>
         <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="apc" passwd="password_example"/>
   </fencedevices>
   <rm>
   </rm>
</cluster>
```

이 예제에서 차단 방식 (**method**)은 각 노드에 추가되어 있습니다. 각 노드의 차단 방식 이름 (**name**)은 **APC**입니다. 각 노드에 있는 차단 방식의 장치 (**device**)는 이름(**name**)을 **apc**로 지정하고, 각 노드에 대해 고유한 APC 스위치 전원 포트 번호 (**port**)를 지정하고 있습니다. 예를 들어, node-01.example.com의 포트 번호는 **1** (**port="1"**)입니다. 각 노드의 장치 이름 (**device name="apc"**) 은 **fencedevices** 요소의 행에서 **apc**의 이름 (**name**)에 의해 차단 장치를 가리킵니다: **fencedevice agent="fence_apc" ipaddr="apc_ip_example" login="login_example" name="apc" passwd="password_example"**.

예 7.5. cluster.conf: 노드 당 여러 차단 방식

```
<cluster name="mycluster" config_version="3">
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
            <method name="APC">
              <device name="apc" port="1"/>
             </method>
            <method name="SAN">
       <device name="sanswitch1" port="11"/>
             </method>
         </fence>
         <unfence>
             <device name="sanswitch1" port="11" action="on"/>
         </unfence
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
            <method name="APC">
              <device name="apc" port="2"/>
            </method>
            <method name="SAN">
       <device name="sanswitch1" port="12"/>
             </method>
         </fence>
         <unfence>
             <device name="sanswitch1" port="12" action="on"/>
         </unfence
     </clusternode>
     <clusternode name="node-03.example.com" nodeid="3">
         <fence>
            <method name="APC">
              <device name="apc" port="3"/>
            </method>
            <method name="SAN">
       <device name="sanswitch1" port="13"/>
             </method>
         </fence>
         <unfence>
             <device name="sanswitch1" port="13" action="on"/>
         </unfence
     </clusternode>
   </clusternodes>
   <fencedevices>
        <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="apc" passwd="password_example"/>
        <fencedevice agent="fence_sanbox2" ipaddr="san_ip_example"
login="login_example" name="sanswitch1" passwd="password_example"/>
   </fencedevices>
   <rm>
   </rm>
</cluster>
```

예 7.6. `cluster.conf`: 차단 장치, 멀티패스 다중 포트

```
<cluster name="mycluster" config_version="3">
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
            <method name="SAN-multi">
       <device name="sanswitch1" port="11"/>
       <device name="sanswitch2" port="11"/>
     </method>
         </fence>
         <unfence>
             <device name="sanswitch1" port="11" action="on"/>
             <device name="sanswitch2" port="11" action="on"/>
         </unfence
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
            <method name="SAN-multi">
       <device name="sanswitch1" port="12"/>
       <device name="sanswitch2" port="12"/>
            </method>
         </fence>
         <unfence>
             <device name="sanswitch1" port="12" action="on"/>
             <device name="sanswitch2" port="12" action="on"/>
         </unfence
     </clusternode>
     <clusternode name="node-03.example.com" nodeid="3">
         <fence>
            <method name="SAN-multi">
       <device name="sanswitch1" port="13"/>
       <device name="sanswitch2" port="13"/>
            </method>
         </fence>
         <unfence>
             <device name="sanswitch1" port="13" action="on"/>
             <device name="sanswitch2" port="13" action="on"/>
         </unfence
     </clusternode>
   </clusternodes>
   <fencedevices>
        <fencedevice agent="fence_sanbox2" ipaddr="san_ip_example"
login="login_example" name="sanswitch1" passwd="password_example"/>
        <fencedevice agent="fence_sanbox2" ipaddr="san_ip_example"
login="login_example" name="sanswitch2" passwd="password_example"/>
   </fencedevices>
   <rm>
   </rm>
```

```
</cluster>
```

예 7.7. `cluster.conf`: 듀얼 전원 공급을 갖는 노드 차단 장치

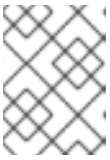
```
<cluster name="mycluster" config_version="3">
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
            <method name="APC-dual">
              <device name="apc1" port="1"action="off"/>
              <device name="apc2" port="1"action="off"/>
              <device name="apc1" port="1"action="on"/>
              <device name="apc2" port="1"action="on"/>
             </method>
         </fence>
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
            <method name="APC-dual">
              <device name="apc1" port="2"action="off"/>
              <device name="apc2" port="2"action="off"/>
              <device name="apc1" port="2"action="on"/>
              <device name="apc2" port="2"action="on"/>
            </method>
         </fence>
     </clusternode>
     <clusternode name="node-03.example.com" nodeid="3">
         <fence>
            <method name="APC-dual">
              <device name="apc1" port="3"action="off"/>
              <device name="apc2" port="3"action="off"/>
              <device name="apc1" port="3"action="on"/>
              <device name="apc2" port="3"action="on"/>
            </method>
         </fence>
     </clusternode>
   </clusternodes>
   <fencedevices>
       <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="apc1" passwd="password_example"/>
       <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="apc2" passwd="password_example"/>
   </fencedevices>
   <rm>
   </rm>
</cluster>
```

> 듀얼 전원 공급과 함께 노드를 차단하기 위해 전원 스위치를 사용할 때, 두 포트 모두에 전원을 복구하기 전 전원 포트를 끄도록 에이전트에 통지해야 합니다. 에이전트의 기본값 온/오프 동작은 전원이 노드에서 완전히 비활성화되지 않게 할 수 있습니다.

7.4. 장애 조치 도메인 설정

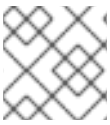
장애 조치 도메인은 노드 장애 발생 시 클러스터 서비스를 실행할 수 있는 클러스터 노드의 이름이 지정된 하위 집합입니다. 장애 조치 도메인은 다음과 같은 특징을 갖을 수 있습니다:

- 제한 없음 **(Unrestricted)** – 우선 순위를 갖는 멤버의 하위 집합을 지정할 수 있게 합니다. 하지만 이러한 도메인에 할당된 클러스터 서비스는 사용 가능한 멤버에서 실행할 수 있습니다.
- 제한 **(Restricted)** – 특정 클러스터 서비스를 실행할 수 있는 멤버를 제한할 수 있게 합니다. 제한된 장애 조치 도메인에 있는 사용 가능한 멤버가 없을 경우, 클러스터 서비스는 (수동으로 또는 클러스터 소프트웨어로도) 시작할 수 없습니다.
- 우선 순위 없음 **(Unordered)** – 클러스터 서비스가 우선 순위가 없는 장애 조치 도메인에 할당될 때 클러스터 서비스를 실행할 멤버는 우선 순위가 없는 사용 가능한 장애 조치 도메인 멤버에서 선택됩니다.
- 우선 순위 지정 **(Ordered)** – 장애 조치 도메인의 멤버 간에 우선 순위를 지정할 수 있게 합니다. 우선 순위가 지정된 장애 조치 도메인은 최하위 우선 순위 번호를 갖는 노드를 먼저 선택합니다. 즉, 장애 조치 도메인에 있는 우선 순위 번호 **"1"**을 갖는 노드는 최상의 우선 순위를 지정하게 되므로 이는 장애 조치 도메인에서 가장 우선 순위를 갖는 노드가 됩니다. 이 노드 다음에 다음의 우선 순위를 갖는 노드는 다음으로 최상위 우선 순위 번호를 갖는 노드가 됩니다.
- 장애 복구 **(Failback)** – 장애 조치 도메인의 서비스가 노드 장애 이전 원래 실행하고 있는 노드로 장애 복구할지에 대한 여부를 지정할 수 있게 합니다. 이러한 기능 설정은 노드가 반복적으로 실패하고 이것이 우선 순위를 갖는 장애 조치 도메인의 일부분인 경우에 유용합니다. 이러한 상황에서 노드가 장애 조치 도메인에 있는 우선 순위를 갖는 노드일 경우, 서비스를 장애 조치하여 우선 순위를 갖는 노드와 다른 노드 사이에서 반복적으로 장애 복구할 가능성이 있으므로 이는 성능에 심각한 영향을 미칠 수 있습니다.



참고

우선 순위가 지정된 장애 조치가 설정된 경우에만 장애 복구 기능을 사용할 수 있습니다.



참고

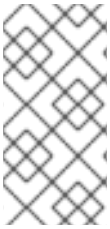
장애 조치 도메인 설정 변경은 현재 실행 중인 서비스에 영향을 주지 않습니다.

참고

장애 조치 도메인은 운용에 필요하지 *않습니다*.

기본값으로 장애 조치 도메인은 제한이 없고 우선 순위가 없습니다.

여러 멤버를 갖는 클러스터에서 제한된 장애 조치 도메인을 사용하면 클러스터 서비스 (예: **`httpd`**) 실행을 위한 클러스터 설치 작업을 최소화할 수 있습니다. 이때 여기서 클러스터 서비스를 실행하는 모든 멤버에서 동일한 설정을 구성해야 합니다. 클러스터 서비스를 실행하기 위해 전체 클러스터를 설정하는 대신 클러스터 서비스와 관련된 제한된 장애조치 도메인에 있는 멤버만을 설정할 수 있습니다.



참고

우선 순위를 갖는 멤버를 설정하려면, 하나의 클러스터 멤버로 이루어진 제한이 없는 장애 조치 도메인을 생성할 수 있습니다. 이렇게 할 경우 클러스터 서비스가 주로 클러스터 멤버(우선 순위를 갖는 멤버)에서 실행되지만 클러스터 서비스가 다른 멤버라도 장애 조치할 수 있게 합니다.

장애 조치 도메인을 설정하려면, 다음 절차를 사용합니다:

1. 클러스터의 노드 중 하나에서 **/etc/cluster/cluster.conf**를 엽니다.
2. 사용될 각각의 장애 조치 도메인에 대해 **rm** 요소 내에 다음과 같은 부분을 추가합니다.

```
        <failoverdomains>
            <failoverdomain name="" nofailback="" ordered=""
restricted="">
                <failoverdomainnode name="" priority=""/>
                <failoverdomainnode name="" priority=""/>
                <failoverdomainnode name="" priority=""/>
            </failoverdomain>
        </failoverdomains>
```

참고

failoverdomainnode 속성의 수는 장애 조치 도메인에 있는 노드 수에 따라 달라집니다. 위의 스켈튼 **failoverdomain** 부분에서는 세 개의 **failoverdomainnode** 요소(노드 이름이 지정되지 않음)를 보여주고 있으며 장애 조치 도메인에 세 개의 노드가 있음을 나타내고 있습니다.

3. **failoverdomain** 부분에서는 요소와 속성의 값을 제공합니다. 요소와 속성에 대한 설명은 주석 클러스터 스키마의 *failoverdomain* 부분을 참조하십시오. 주석 클러스터 스키마는 클러스터 노드에 있는 **/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html** (예: **/usr/share/doc/cman-3.0.12/cluster_conf.html**)에서 사용 가능합니다. **failoverdomains** 부분의 예제의 경우 예 7.8. “**cluster.conf**에 추가된 장애 조치 도메인 ” 에서 참조하십시오.
4. 값을 증가시켜 **config_version** 속성을 업데이트합니다 (예: **config_version="2"**에서 **config_version="3">**로 변경)
5. **/etc/cluster/cluster.conf**를 저장합니다.
6. **(옵션) ccs_config_validate** 명령을 실행하여 클러스터 스키마 (**cluster.rng**)로 파일의 유효성을 검증합니다. 예:

```
[root@example-01 ~]# ccs_config_validate
Configuration validates
```

7. **cman_tool version -r** 명령을 실행하여 설정을 나머지 클러스터 노드에 전달합니다.
8. 7.5절. “HA 서비스 설정 ”으로 이동합니다.

예 7.8. “cluster.conf에 추가된 장애 조치 도메인 ” 에서는 우선 순위로된 무제한 장애 조치 도메인과 함께 설정 예를 보여줍니다.

예 7.8. cluster.conf에 추가된 장애 조치 도메인

```
<cluster name="mycluster" config_version="3">
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
            <method name="APC">
              <device name="apc" port="1"/>
             </method>
         </fence>
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
            <method name="APC">
              <device name="apc" port="2"/>
            </method>
         </fence>
     </clusternode>
     <clusternode name="node-03.example.com" nodeid="3">
         <fence>
            <method name="APC">
              <device name="apc" port="3"/>
            </method>
         </fence>
     </clusternode>
   </clusternodes>
   <fencedevices>
         <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="apc" passwd="password_example"/>
   </fencedevices>
   <rm>
       <failoverdomains>
           <failoverdomain name="example_pri" nofailback="0"
ordered="1" restricted="0">
               <failoverdomainnode name="node-01.example.com"
priority="1"/>
               <failoverdomainnode name="node-02.example.com"
priority="2"/>
               <failoverdomainnode name="node-03.example.com"
priority="3"/>
           </failoverdomain>
       </failoverdomains>
   </rm>
</cluster>
```

failoverdomains 부분에는 클러스터에 있는 각 장애 조치 도메인의 **failoverdomain** 부분이 들어 있습니다. 이 예제에는 하나의 장애 조치 도메인이 있습니다. **failoverdomain** 행에서 이름 (**name**)은 **example_pri**로 지정되어 있습니다. 또한, 장애 복구 없음 (**failback="0"**)이 지정되어 있으며, 장애 조치는 (**ordered="1"**) 우선 순위를 갖고 장애 조치 도메인은 제한 없음 (**restricted="0"**)으로 지정되어 있습니다.

7.5. HA 서비스 설정

HA (High Availability) 서비스 설정은 리소스 설정과 이를 서비스에 할당하는 것으로 구성되어 있습니다.

다음 부분에서는 **/etc/cluster/cluster.conf**를 편집하여 리소스 및 서비스를 추가하는 방법을 설명합니다.

- 7.5.1절. "클러스터 리소스 추가 "
- 7.5.2절. "클러스터에 클러스터 서비스 추가 "

중요

고가용성 리소스와 서비스로 광범위하게 설정할 수 있습니다. 리소스 매개 변수 및 리소스 동작에 대한 보다 나은 이해를 위해 부록 B. *HA 리소스 매개 변수*및 부록 C. *HA 리소스 동작*을 참조하십시오. 최적의 성능과 사용 설정이 지원 가능한지를 확인하기 위해 Red Hat 지원 담당자에게 문의하십시오.

7.5.1. 클러스터 리소스 추가

두 가지 유형의 리소스를 설정할 수 있습니다:

- 글로벌 (Global) – 클러스터의 서비스에서 사용 가능한 리소스. 이는 설정 파일 (**rm** 요서 내에 있는)의 **resources** 부분에 설정되어 있습니다.
- 특정 서비스 (Service-specific) – 하나의 서비스에만 사용 가능한 리소스. 이는 설정 파일 (**rm** 요소에 있는) 의 각 **service** 부분에 설정되어 있습니다.

다음 부분에서는 글로벌 리소스를 추가하는 방법에 대해 설명합니다. 특정 서비스 리소스 설정에 대한 절차는 7.5.2절. "클러스터에 클러스터 서비스 추가 "에서 참조하십시오.

글로벌 클러스터 리소스를 추가하려면 다음 절차를 따르십시오.

1. 클러스터의 노드 중 하나에서 **/etc/cluster/cluster.conf**를 엽니다.
2. **rm** 요소에 **resources** 부분을 추가합니다. 예:

```
<rm>
    <resources>

    </resources>
</rm>
```

3. 생성하고자 하는 서비스에 따라 리소스로 이를 작성합니다. 예를 들어, Apache 서비스에서 사용할 수 있는 리소스가 있을 경우, 이는 파일 시스템 (**fs**) 리소스, IP (**ip**) 리소스, Apache (**apache**) 리소스로 구성됩니다.

```
<rm>
    <resources>
```

```
            <fs name="web_fs" device="/dev/sdd2"
mountpoint="/var/www" fstype="ext3"/>
            <ip address="127.143.131.100" monitor_link="yes"
sleeptime="10"/>
            <apache config_file="conf/httpd.conf"
name="example_server" server_root="/etc/httpd" shutdown_wait="0"/>
        </resources>
    </rm>
```

예 7.9. "리소스가 추가된 **cluster.conf** 파일 "에서는 **resources** 부분이 추가된 **cluster.conf** 파일의 예를 보여주고 있습니다.

4. 값이 증가함에 따라 (예: **config_version="2"**에서 **config_version="3"**로 변경) **config_version** 속성을 업데이트합니다.
5. **/etc/cluster/cluster.conf**를 저장합니다.
6. **(옵션) ccs_config_validate** 명령을 실행하여 클러스터 스키마 (**cluster.rng**)로 파일의 유효성을 검증합니다. 예:

```
[root@example-01 ~]# ccs_config_validate
Configuration validates
```

7. **cman_tool version -r** 명령을 실행하여 설정을 나머지 클러스터 노드에 전달합니다.
8. 업데이트된 설정 파일이 전달되었는지 확인합니다.
9. 7.5.2절. "클러스터에 클러스터 서비스 추가 "로 이동합니다.

예 7.9. 리소스가 추가된 cluster.conf 파일

```
<cluster name="mycluster" config_version="3">
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
            <method name="APC">
              <device name="apc" port="1"/>
             </method>
         </fence>
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
            <method name="APC">
              <device name="apc" port="2"/>
            </method>
         </fence>
     </clusternode>
     <clusternode name="node-03.example.com" nodeid="3">
         <fence>
            <method name="APC">
              <device name="apc" port="3"/>
            </method>
         </fence>
```

```
      </clusternode>
   </clusternodes>
   <fencedevices>
         <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="apc" passwd="password_example"/>
   </fencedevices>
   <rm>
       <failoverdomains>
           <failoverdomain name="example_pri" nofailback="0"
ordered="1" restricted="0">
               <failoverdomainnode name="node-01.example.com"
priority="1"/>
               <failoverdomainnode name="node-02.example.com"
priority="2"/>
               <failoverdomainnode name="node-03.example.com"
priority="3"/>
           </failoverdomain>
       </failoverdomains>
       <resources>
           <fs name="web_fs" device="/dev/sdd2" mountpoint="/var/www"
fstype="ext3"/>
           <ip address="127.143.131.100" monitor_link="yes"
sleeptime="10"/>
           <apache config_file="conf/httpd.conf" name="example_server"
server_root="/etc/httpd" shutdown_wait="0"/>
        </resources>

   </rm>
</cluster>
```

7.5.2. 클러스터에 클러스터 서비스 추가

클러스터에 클러스터 서비스를 추가하려면, 다음 절차를 따르십시오.

1. 클러스터의 노드 중 하나에서 **/etc/cluster/cluster.conf**를 엽니다.
2. 각 서비스에 대해 **rm** 요소 내에 **service** 부분을 추가합니다. 예:

```
    <rm>
        <service autostart="1" domain="" exclusive="0" name=""
recovery="restart">

        </service>
    </rm>
```

3. **service** 요소에서 다음의 매개 변수 (속성)를 설정합니다:
   - **autostart** – 클러스터가 시작될 때 서비스를 자동으로 시작할지에 대한 여부를 지정합니다. 활성화하려면 '1'을 비활성화하려면 '0'을 사용합니다. 디폴트 값은 활성화입니다.
   - **domain** – 장애 조치 도메인을 지정합니다. (필요한 경우)

- **exclusive** – 다른 서비스가 실행되고 있지 않는 노드에서만 서비스를 실행하는 정책을 지정합니다.
- **recovery** – 서비스의 복구 정책을 지정합니다. 이 옵션은 서비스를 재배치, 다시 시작, 비활성화, 다시 시작-비활성화합니다.

4. 사용하고자 하는 리소스의 유형에 따라, 글로벌 또는 특정 서비스 리소스를 갖는 서비스를 배치합니다.

   예를 들어 글로벌 리소스를 사용하는 Apache 서비스는 다음과 같습니다:

```
    <rm>
        <resources>
                <fs name="web_fs" device="/dev/sdd2"
mountpoint="/var/www" fstype="ext3"/>
                <ip address="127.143.131.100" monitor_link="yes"
sleeptime="10"/>
                <apache config_file="conf/httpd.conf"
name="example_server" server_root="/etc/httpd" shutdown_wait="0"/>
        </resources>
        <service autostart="1" domain="example_pri" exclusive="0"
name="example_apache" recovery="relocate">
                <fs ref="web_fs"/>
                <ip ref="127.143.131.100"/>
                <apache ref="example_server"/>
        </service>
    </rm>
```

   예를 들어, 특정 서비스 리소스를 사용하는 Apache 서비스는 다음과 같습니다:

```
    <rm>
        <service autostart="0" domain="example_pri" exclusive="0"
name="example_apache2" recovery="relocate">
                <fs name="web_fs2" device="/dev/sdd3"
mountpoint="/var/www2" fstype="ext3"/>
                <ip address="127.143.131.101" monitor_link="yes"
sleeptime="10"/>
                <apache config_file="conf/httpd.conf"
name="example_server2" server_root="/etc/httpd" shutdown_wait="0"/>
        </service>
    </rm>
```

   예 7.10. “서비스 추가와 함께 **cluster.conf**: 하나는 글로벌 리소스를 사용하고 하나는 특정 서비스 리소스를 사용 ”에서는 두 개의 서비스를 갖는 **cluster.conf** 파일의 예입니다:

   - **example_apache** – 이 서비스는 글로벌 리소스 **web_fs**, **127.143.131.100**, **example_server**를 사용합니다.
   - **example_apache2** – 이 서비스는 특정 서비스 리소스 **web_fs2**, **127.143.131.101**, **example_server2**를 사용합니다.

5. 값을 증가시켜 **config_version** 속성을 업데이트합니다 (예: **config_version="2"**에서 **config_version="3">**로 변경)

6. **/etc/cluster/cluster.conf**를 저장합니다.

7. **(옵션) ccs_config_validate** 명령을 실행하여 클러스터 스키마 (**cluster.rng**)에 대해 업데이트된 파일의 유효성을 검사합니다. 예:

```
[root@example-01 ~]# ccs_config_validate
Configuration validates
```

8. **cman_tool version -r** 명령을 실행하여 설정을 나머지 클러스터 노드에 전달합니다.

9. 업데이트된 설정 파일이 전달되었는지 확인합니다.

10. 7.8절. “설정 확인 ”으로 이동합니다.

예 7.10. 서비스 추가와 함께 cluster.conf: 하나는 글로벌 리소스를 사용하고 하나는 특정 서비스 리소스를 사용

```
<cluster name="mycluster" config_version="3">
   <clusternodes>
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
            <method name="APC">
              <device name="apc" port="1"/>
             </method>
         </fence>
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
            <method name="APC">
              <device name="apc" port="2"/>
            </method>
         </fence>
     </clusternode>
     <clusternode name="node-03.example.com" nodeid="3">
         <fence>
            <method name="APC">
              <device name="apc" port="3"/>
            </method>
         </fence>
     </clusternode>
   </clusternodes>
   <fencedevices>
         <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="apc" passwd="password_example"/>
   </fencedevices>
   <rm>
       <failoverdomains>
          <failoverdomain name="example_pri" nofailback="0"
ordered="1" restricted="0">
              <failoverdomainnode name="node-01.example.com"
priority="1"/>
```

```
                <failoverdomainnode name="node-02.example.com"
priority="2"/>
                <failoverdomainnode name="node-03.example.com"
priority="3"/>
            </failoverdomain>
        </failoverdomains>
        <resources>
            <fs name="web_fs" device="/dev/sdd2" mountpoint="/var/www"
fstype="ext3"/>
            <ip address="127.143.131.100" monitor_link="yes"
sleeptime="10"/>
            <apache config_file="conf/httpd.conf" name="example_server"
server_root="/etc/httpd" shutdown_wait="0"/>
        </resources>
        <service autostart="1" domain="example_pri" exclusive="0"
name="example_apache" recovery="relocate">
            <fs ref="web_fs"/>
            <ip ref="127.143.131.100"/>
            <apache ref="example_server"/>
        </service>
        <service autostart="0" domain="example_pri" exclusive="0"
name="example_apache2" recovery="relocate">
            <fs name="web_fs2" device="/dev/sdd3" mountpoint="/var/www2"
fstype="ext3"/>
            <ip address="127.143.131.101" monitor_link="yes"
sleeptime="10"/>
            <apache config_file="conf/httpd.conf" name="example_server2"
server_root="/etc/httpd" shutdown_wait="0"/>
        </service>
    </rm>
</cluster>
```

7.6. 중복 링 프로토콜 설정

Red Hat Enterprise Linux 6.4에서 **Red Hat** 고가용성 애드온은 중복 링 프로토콜 설정을 지원합니다.

중복 링 프로토콜을 사용하도록 시스템을 설정할 때 다음 사항을 고려해야 합니다:

- 두 개 이상의 링을 지정하지 않습니다.
- 각 링은 동일한 프로토콜을 사용해야 합니다; **IPv4**와 **IPv6**를 혼합하지 마십시오.
- 필요한 경우 두 번째 링에 대해 멀티캐스트 주소를 수동으로 지정할 수 있습니다. 두 번째 링의 멀티캐스트 주소를 지정하는 경우 대체 멀티캐스트 주소 또는 대체 포트는 첫 번째 링의 멀티캐스트 주소와 달라야 합니다. 대체 멀티캐스트 주소를 지정하지 않을 경우 시스템은 두 번째 링에 대해 다른 멀티캐스트 주소를 자동으로 사용하게 됩니다.

  대체 포트를 지정할 경우 시스템 자체가 작업을 수행하기 위해 포트 및 포트 1을 사용하므로 첫 번째 링과 두번째 링의 포트 번호는 최소 두개의 다른 것이어야 합니다.
- 동일한 서브넷에서 두 개의 다른 인터페이스를 사용하지 마십시오.
- 일반적으로 **NIC** 또는 스위치 중 하나에 문제가 발생했을 경우 두 개의 다른 **NIC** 및 두 개의 다른 스위치에 중복 링 프로토콜을 설정하는 것이 좋습니다.

- **ifdown** 명령이나 **service network stop** 명령을 사용하여 네트워크 오류 시뮬레이션을 수행하지 마십시오. 이는 클러스터 전체를 파괴하여 복구를 위해 클러스터에 있는 모든 노드를 다시 시작해야 합니다.
- 케이블이 빠지면 **ifdown**이 실행되므로 **NetworkManager**를 사용하지 마십시오.
- NIC의 하나의 노드에 문제가 발생하면 전체 링에 문제가 있다고 표시됩니다.
- 문제가 발생한 링을 복구하기 위해 수동 개입이 필요하지 않습니다. 복구하려면 잘못된 NIC 또는 스위치와 같은 문제의 원인이 되는 부분만 수정하면 됩니다.

중복 링 프로토콜에 사용할 두 번째 네트워크 인터페이스를 지정하려면 **cluster.conf** 설정 파일의 **clusternode** 섹션에 **altname** 구성 요소를 추가합니다. **altname**을 지정할 때 **name** 속성을 지정하여 노드의 두 번째 호스트 이름 또는 IP 주소를 지정합니다.

다음 예에서는 클러스터 노드 **clusternet-node1-eth1**의 대체 이름으로 **clusternet-node1-eth2**를 지정합니다.

```
<cluster name="mycluster" config_version="3" >
  <logging debug="on"/>
  <clusternodes>
    <clusternode name="clusternet-node1-eth1" votes="1" nodeid="1">
      <fence>
        <method name="single">
          <device name="xvm" domain="clusternet-node1"/>
        </method>
      </fence>
      <altname name="clusternet-node1-eth2"/>
    </clusternode>
```

clusternode 블록 내의 **altname** 섹션은 위치에 종속되지 않습니다. **fence** 섹션 앞이나 뒤에 넣을 수 있습니다. 클러스터 노드에 하나 이상의 **altname** 구성 요소를 지정하지 마십시오. 시스템이 시작 실패하게 됩니다.

옵션으로 **cluster.conf** 설정 파일의 **cman** 섹션에 **altmulticast** 구성 요소를 포함하여 두 번째 링의 멀티캐스트 주소, 포트, TTL을 수동으로 지정할 수 있습니다. **altmulticast** 구성 요소는 **addr**, **port**, **ttl** 매개 변수를 허용합니다.

다음 예에서는 두 번째 링의 멀티캐스트 주소, 포트, TTL을 설정하는 클러스터 설정 파일의 **cman** 섹션을 보여주고 있습니다.

```
<cman>
   <multicast addr="239.192.99.73" port="666" ttl="2"/>
   <altmulticast addr="239.192.99.88" port="888" ttl="3"/>
</cman>
```

7.7. 디버그 옵션 설정

클러스터에 있는 모든 데몬의 디버깅을 활성화하거나 특정 클러스터 프로세스에 대한 로깅을 활성화할 수 있습니다.

모든 데몬의 디버깅을 활성화하려면 다음을 **/etc/cluster/cluster.conf**에 추가합니다. 디폴트로 로깅은 **/var/log/cluster/*daemon*.log** 파일로 이동합니다.

```
<cluster config_version="7" name="rh6cluster">
  <logging debug="on"/>
   ...
</cluster>
```

개별 클러스터 프로세스의 디버깅을 활성화하려면 다음 행을 **/etc/cluster/cluster.conf** 파일에 추가합니다. 데몬별 로깅 설정은 글로벌 설정을 덮어쓰기합니다.

```
<cluster config_version="7" name="rh6cluster">
   ...
  <logging>
       <!-- turning on per-subsystem debug logging -->
       <logging_daemon name="corosync" debug="on" />
       <logging_daemon name="fenced" debug="on" />
       <logging_daemon name="qdiskd" debug="on" />
       <logging_daemon name="rgmanager" debug="on" />
       <logging_daemon name="dlm_controld" debug="on" />
       <logging_daemon name="gfs_controld" debug="on" />
  </logging>
   ...
</cluster>
```

글로벌 및 데몬 별 로깅 모두를 설정할 수 있는 추가 로깅 옵션 뿐 만 아니라 글로벌 로깅을 활성화할 수 있는 로깅 데몬 목록은 **cluster.conf**(5) man 페이지에서 참조하십시오.

7.8. 설정 확인

클러스터 설정 파일을 만든 후 다음의 절차를 실행하여 이것이 올바로 작동하고 있는지 확인합니다:

1. 각 노드에서 클러스터 소프트웨어를 다시 시작합니다. 이 작업은 시작 시에만 확인하는 추가 설정 사항이 실행 중인 설정에 포함되어 있는지를 확인합니다. **service cman restart**를 실행하여 클러스터 소프트웨어를 다시 시작할 수 있습니다. 예:

```
[root@example-01 ~]# service cman restart
Stopping cluster:
   Leaving fence domain...                                 [  OK
]
   Stopping gfs_controld...                                [  OK
]
   Stopping dlm_controld...                                [  OK
]
   Stopping fenced...                                      [  OK
]
   Stopping cman...                                        [  OK
]
   Waiting for corosync to shutdown:                       [  OK  ]
   Unloading kernel modules...                             [  OK
```

```
]
   Unmounting configfs...                                  [  OK
]
Starting cluster:
   Checking Network Manager...                             [  OK
]
   Global setup...                                         [  OK
]
   Loading kernel modules...                               [  OK
]
   Mounting configfs...                                    [  OK
]
   Starting cman...                                        [  OK
]
   Waiting for quorum...                                   [  OK
]
   Starting fenced...                                      [  OK
]
   Starting dlm_controld...                                [  OK
]
   Starting gfs_controld...                                [  OK
]
   Unfencing self...                                       [  OK
]
   Joining fence domain...                                 [  OK
]
```

2. CLVM이 클러스터 볼륨을 만드는데 사용한 경우, **service clvmd start**를 실행합니다. 예:

```
[root@example-01 ~]# service clvmd start
Activating VGs:                                            [  OK
]
```

3. Red Hat GFS2를 사용하고 있을 경우, **service gfs2 start**를 실행합니다. 예:

```
[root@example-01 ~]# service gfs2 start
Mounting GFS2 filesystem (/mnt/gfsA):                      [  OK  ]
Mounting GFS2 filesystem (/mnt/gfsB):                      [  OK  ]
```

4. 고가용성 (HA) 서비스를 사용하고 있는 경우 **service rgmanager start**를 실행합니다. 예:

```
[root@example-01 ~]# service rgmanager start
Starting Cluster Service Manager:                          [  OK  ]
```

5. 클러스터 노드에서 **cman_tool nodes**를 실행하여 노드가 클러스터에서 멤버로 작동하는지 확인합니다 (상태 컬럼 "Sts"에서 "M"로 표시). 예:

```
[root@example-01 ~]# cman_tool nodes
Node  Sts   Inc   Joined               Name
   1   M    548   2010-09-28 10:52:21  node-01.example.com
   2   M    548   2010-09-28 10:52:21  node-02.example.com
   3   M    544   2010-09-28 10:52:21  node-03.example.com
```

6. 노드에서 **clustat** 유틸리티를 사용하여, HA 서비스가 예상대로 실행되는지 확인합니다. 또한 **clustat**는 클러스터 노드의 상태를 표시합니다. 예:

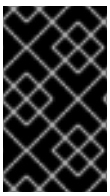
```
[root@example-01 ~]#clustat
Cluster Status for mycluster @ Wed Nov 17 05:40:00 2010
Member Status: Quorate

 Member Name                             ID   Status
 ------ ----                             ---- ------
 node-03.example.com                         3 Online, rgmanager
 node-02.example.com                         2 Online, rgmanager
 node-01.example.com                         1 Online, Local,
rgmanager

 Service Name                   Owner (Last)
State
 ------- ----                   ----- ------                    --
---
 service:example_apache         node-01.example.com
started
 service:example_apache2        (none)
disabled
```

7. 클러스터가 예상대로 작동하는 경우, 설정 파일 만들기가 완료됩니다. 8장. *명령행 도구로 Red Hat 고가용성 추가 기능 관리* 에서 설명하고 있듯이 명령행 도구를 사용하여 클러스터를 관리할 수 있습니다.

8장. 명령행 도구로 RED HAT 고가용성 추가 기능 관리

다음 부분에서는 Red Hat 고가용성 추가 기능 관리를 위한 다양한 관리 작업에 대해 설명하고 있으며 이는 다음과 같은 부분으로 구성되어 있습니다:

- 8.1절. “클러스터 소프트웨어 시작 및 중지 ”
- 8.2절. “노드 삭제 또는 추가 ”
- 8.3절. “고가용성 서비스 관리 ”
- 8.4절. “설정 업데이트 ”



중요

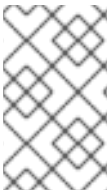
Red Hat 고가용성 추가 기능의 도입이 요구에 부합되며 지원되는지 확인합니다. Red Hat 담당자에게 문의하여 배포 전 설정을 확인하십시오. 또한 장애 모드를 테스트하기 위해 설정 번인 (burn-in) 기간을 두도록 합니다.

중요

다음 부분에서는 일반적으로 사용되는 **cluster.conf** 요소와 속성을 참조합니다. **cluster.conf** 요소와 속성의 전체적 목록과 설명은 **/usr/share/cluster/cluster.rng**에 있는 클러스터 스키마와 **/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html** (예: **/usr/share/doc/cman-3.0.12/cluster_conf.html**)의 주석 스키마를 참조하십시오.

중요

다음 부분에서의 일부 단계는 클러스터 설정을 클러스터 전역에 전달하기 위해 **cman_tool version -r** 명령을 사용해야 합니다. 이 명령을 사용하려면 **ricci**가 실행되고 있어야 합니다.



참고

이 부분에 있는 절차에는 부록 E. *명령행 도구 요약*에 나열된 명령행 도구의 일부 특정 명령을 포함하고 있습니다. 모든 명령 및 변수에 대한 자세한 내용은 각 명령행 도구의 man 페이지를 참조하십시오.

8.1. 클러스터 소프트웨어 시작 및 중지

8.1.1절. “클러스터 소프트웨어 시작 ” 및 8.1.2절. “클러스터 소프트웨어 중지 ”에 따라 노드에 있는 클러스터 소트웨어를 시작 또는 중지할 수 있습니다. 노드의 클러스터 소프트웨어를 시작하면 클러스터에 가입하게되고 노드의 클러스터 소프트웨어를 중지하면 클러스터에서 탈퇴하게 됩니다.

8.1.1. 클러스터 소프트웨어 시작

노드에서 클러스터 소프트웨어를 시작하려면 다음의 명령을 순서대로 입력합니다:

1. **service cman start**
2. **service clvmd start**, 클러스터 볼륨을 만들기 위해 CLVM를 사용하는 경우

3. **service gfs2 start**, Red Hat GFS2를 사용하는 경우

4. **service rgmanager start**, 고가용성 (HA) 서비스 (**rgmanager**)를 사용하는 경우

예:

```
[root@example-01 ~]# service cman start
Starting cluster:
   Checking Network Manager...                                [  OK  ]
   Global setup...                                            [  OK  ]
   Loading kernel modules...                                  [  OK  ]
   Mounting configfs...                                       [  OK  ]
   Starting cman...                                           [  OK  ]
   Waiting for quorum...                                      [  OK  ]
   Starting fenced...                                         [  OK  ]
   Starting dlm_controld...                                   [  OK  ]
   Starting gfs_controld...                                   [  OK  ]
   Unfencing self...                                          [  OK  ]
   Joining fence domain...                                    [  OK  ]
[root@example-01 ~]# service clvmd start
Starting clvmd:                                               [  OK  ]
Activating VG(s):   2 logical volume(s) in volume group "vg_example" now
active
                                                              [  OK  ]
[root@example-01 ~]# service gfs2 start
Mounting GFS2 filesystem (/mnt/gfsA):                         [  OK  ]
Mounting GFS2 filesystem (/mnt/gfsB):                         [  OK  ]
[root@example-01 ~]# service rgmanager start
Starting Cluster Service Manager:                             [  OK  ]
[root@example-01 ~]#
```

8.1.2. 클러스터 소프트웨어 중지

노드에서 클러스터 소프트웨어를 중지하려면 다음의 명령을 순서대로 입력합니다:

1. **service rgmanager stop**, 고가용성 (HA) 서비스 (**rgmanager**)를 사용하는 경우

2. **service gfs2 stop**, Red Hat GFS2를 사용하는 경우

3. **umount -at gfs2**, **rgmanager**와 함께 Red Hat GFS2를 사용하고 있는 경우, **rgmanager** 시작 시 마운트된 (하지만 종료 시 마운트 해제되지 않은) GFS2 파일이 모두 마운트 해제되었는지 확인합니다.

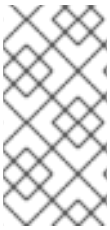
4. **service clvmd stop**, 클러스터 볼륨을 만드는데 CLVM을 사용하는 경우

5. **service cman stop**

예:

```
[root@example-01 ~]# service rgmanager stop
Stopping Cluster Service Manager:                             [  OK  ]
[root@example-01 ~]# service gfs2 stop
Unmounting GFS2 filesystem (/mnt/gfsA):                       [  OK  ]
Unmounting GFS2 filesystem (/mnt/gfsB):                       [  OK  ]
[root@example-01 ~]# umount -at gfs2
```

```
[root@example-01 ~]# service clvmd stop
Signaling clvmd to exit                                    [  OK  ]
clvmd terminated                                           [  OK  ]
[root@example-01 ~]# service cman stop
Stopping cluster:
   Leaving fence domain...                                 [  OK  ]
   Stopping gfs_controld...                                [  OK  ]
   Stopping dlm_controld...                                [  OK  ]
   Stopping fenced...                                      [  OK  ]
   Stopping cman...                                        [  OK  ]
   Waiting for corosync to shutdown:                       [  OK  ]
   Unloading kernel modules...                             [  OK  ]
   Unmounting configfs...                                  [  OK  ]
[root@example-01 ~]#
```



참고

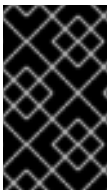
노드에서 클러스터 소프트웨어를 중지하면 HA 서비스가 다른 노드에서 장애 조치되는 요인이 됩니다. 다른 방법으로 클러스터 소프트웨어를 중지하기 전 HA 서비스를 다른 노드로 재배치 또는 이전할 수 있습니다. HA 서비스 관리에 대한 보다 자세한 내용은 8.3절. "고가용성 서비스 관리 "에서 참조하십시오.

8.2. 노드 삭제 또는 추가

다음 부분에서는 클러스터에서 노드를 삭제하고 클러스터에 노드를 추가하는 방법에 대해 설명합니다. 8.2.1절. "클러스터에서 노드를 삭제 "에 따라 클러스터에서 노드를 삭제할 수 있으며 8.2.2절. "클러스터에 노드 추가 "에 따라 클러스터에 노드를 추가할 수 있습니다.

8.2.1. 클러스터에서 노드를 삭제

클러스터에서 노드를 삭제하는 것은 삭제할 노드에서 클러스터 소프트웨어를 종료하고 변경 사항을 반영하기 위해 클러스터 설정을 업데이트하는 것으로 이루어집니다.



중요

클러스터에서 노드를 삭제하는 것이 2개 이상의 노드에서 2개의 노드로 변경되는 원인이 될 경우 클러스터 설정 파일을 업데이트한 후 각 노드에서 클러스터 소프트웨어를 다시 시작해야 합니다.

클러스터에서 노드를 제거하려면 다음 절차를 실행합니다:

1. 노드 중 하나에서 **clusvcadm** 유틸리티를 사용하여 클러스터에서 삭제할 노드에서 실행되고 있는 각각의 HA 서비스를 재배치, 이전, 중지합니다. **clusvcadm** 사용에 대한 자세한 내용은 8.3절. "고가용성 서비스 관리 "에서 참조하십시오.

2. 클러스터에서 삭제된 노드에서 8.1.2절. "클러스터 소프트웨어 중지 "에 따라 클러스터 소프트웨어를 중지합니다. 예:

```
[root@example-01 ~]# service rgmanager stop
Stopping Cluster Service Manager:                          [  OK  ]
[root@example-01 ~]# service gfs2 stop
Unmounting GFS2 filesystem (/mnt/gfsA):                    [  OK  ]
Unmounting GFS2 filesystem (/mnt/gfsB):                    [  OK  ]
```

```
[root@example-01 ~]# service clvmd stop
Signaling clvmd to exit                                    [  OK
]
clvmd terminated                                           [  OK
]
[root@example-01 ~]# service cman stop
Stopping cluster:
   Leaving fence domain...                                 [  OK
]
   Stopping gfs_controld...                                [  OK
]
   Stopping dlm_controld...                                [  OK
]
   Stopping fenced...                                      [  OK
]
   Stopping cman...                                        [  OK
]
   Waiting for corosync to shutdown:                       [  OK  ]
   Unloading kernel modules...                             [  OK
]
   Unmounting configfs...                                  [  OK
]
[root@example-01 ~]#
```

3. 클러스터에 있는 노드 중 하나에서 **/etc/cluster/cluster.conf**를 편집하여 삭제될 노드의 **clusternode** 부분을 제거합니다. 예를 들어 예 8.1. “3 노드 클러스터 설정 ” 에서 node-03.example이 삭제되어야 한다고 할 경우 해당 노드의 **clusternode** 부분을 제거합니다. 노드를 제거하는 것이 2-노드 클러스터가 되는 원인이 될 경우, 설정 파일에 다음 행을 추가하여 단일 노드가 쿼터를 유지하게 합니다 (예를 들어 1 개의 노드가 실패할 경우):

   <cman two_node="1" expected_votes="1"/>

   3 노드와 2 노드 구성 간의 비교는 8.2.3절. “3 노드 및 2-노드 설정의 예 ” 에서 참조하십시오.

4. 값을 증가시켜 **config_version** 속성을 업데이트합니다 (예: **config_version="2"**에서 **config_version="3">**로 변경)

5. **/etc/cluster/cluster.conf**를 저장합니다.

6. **(옵션) ccs_config_validate** 명령을 실행하여 클러스터 스키마 **(cluster.rng)**에 대해 업데이트된 파일의 유효성을 검사합니다. 예:

   ```
   [root@example-01 ~]# ccs_config_validate
   Configuration validates
   ```

7. **cman_tool version -r** 명령을 실행하여 설정을 나머지 클러스터 노드에 전달합니다.

8. 업데이트된 설정 파일이 전달되었는지 확인합니다.

9. 클러스터 노드 수가 2개 이상의 노드에서 2개의 노드로 변경된 경우 다음과 같이 클러스터 소프트웨어를 다시 시작해야 합니다:

   1. 각 노드에서 8.1.2절. “클러스터 소프트웨어 중지 ” 에 따라 클러스터 소프트웨어를 중지합니다. 예:

```
[root@example-01 ~]# service rgmanager stop
Stopping Cluster Service Manager:                          [  OK
]
[root@example-01 ~]# service gfs2 stop
Unmounting GFS2 filesystem (/mnt/gfsA):                    [  OK
]
Unmounting GFS2 filesystem (/mnt/gfsB):                    [  OK
]
[root@example-01 ~]# service clvmd stop
Signaling clvmd to exit                                    [
OK  ]
clvmd terminated                                           [
OK  ]
[root@example-01 ~]# service cman stop
Stopping cluster:
   Leaving fence domain...                                 [
OK  ]
   Stopping gfs_controld...                                [
OK  ]
   Stopping dlm_controld...                                [
OK  ]
   Stopping fenced...                                      [
OK  ]
   Stopping cman...                                        [
OK  ]
   Waiting for corosync to shutdown:                       [  OK
]
   Unloading kernel modules...                             [
OK  ]
   Unmounting configfs...                                  [
OK  ]
[root@example-01 ~]#
```

2. 각 노드에서 8.1.1절. “클러스터 소프트웨어 시작 ”에 따라 클러스터 소프트웨어를 시작합니다. 예:

```
[root@example-01 ~]# service cman start
Starting cluster:
   Checking Network Manager...                             [
OK  ]
   Global setup...                                         [
OK  ]
   Loading kernel modules...                               [
OK  ]
   Mounting configfs...                                    [
OK  ]
   Starting cman...                                        [
OK  ]
   Waiting for quorum...                                   [
OK  ]
   Starting fenced...                                      [
OK  ]
   Starting dlm_controld...                                [
OK  ]
   Starting gfs_controld...                                [
```

```
OK  ]
   Unfencing self...                                        [
OK  ]
   Joining fence domain...                                  [
OK  ]
[root@example-01 ~]# service clvmd start
Starting clvmd:                                             [
OK  ]
Activating VG(s):   2 logical volume(s) in volume group
"vg_example" now active
                                                            [
OK  ]
[root@example-01 ~]# service gfs2 start
Mounting GFS2 filesystem (/mnt/gfsA):                       [  OK
]
Mounting GFS2 filesystem (/mnt/gfsB):                       [  OK
]
[root@example-01 ~]# service rgmanager start
Starting Cluster Service Manager:                           [  OK
]
[root@example-01 ~]#
```

3. 클러스터 노드에서 **cman_tools nodes**를 실행하여 노드가 클러스터에서 멤버로 작동하는지 확인합니다 (상태 컬럼 **"Sts"**에서 **"M"**로 표시). 예:

```
[root@example-01 ~]# cman_tool nodes
Node  Sts   Inc   Joined               Name
   1   M    548   2010-09-28 10:52:21  node-01.example.com
   2   M    548   2010-09-28 10:52:21  node-02.example.com
```

4. 노드에서 **clustat** 유틸리티를 사용하여, HA 서비스가 예상대로 실행되는지 확인합니다. 또한 **clustat**는 클러스터 노드의 상태를 표시합니다. 예:

```
[root@example-01 ~]#clustat
Cluster Status for mycluster @ Wed Nov 17 05:40:00 2010
Member Status: Quorate

 Member Name                             ID   Status
 ------ ----                             ---- ------
 node-02.example.com                         2 Online, rgmanager
 node-01.example.com                         1 Online, Local,
rgmanager

 Service Name                   Owner (Last)
State
 ------- ----                   ----- ------
-----
 service:example_apache         node-01.example.com
started
 service:example_apache2        (none)
disabled
```

8.2.2. 클러스터에 노드 추가

클러스터에 노드를 추가하는 것은 클러스터 설정의 업데이트, 추가될 노드에 업데이트된 설정을 전달, 노드에서 클러스터 소프트웨어를 시작하는 것으로 구성됩니다. 클러스터에 노드를 추가하려면, 다음의 단계를 실행합니다:

1. 클러스터에 있는 노드 중 하나에서 **/etc/cluster/cluster.conf**를 편집하여 추가될 노드에 **clusternode** 부분을 추가합니다. 예를 들어 예 8.2. “2 노드 클러스터 설정 ” 에서 node-03.example.com이 추가된다고 가정할 경우, 해당 노드에 **clusternode** 부분을 추가합니다. 노드를 추가하여 클러스터가 2 노드 클러스터에서 3 개 이상의 노드가 있는 클러스터로 변경될 경우 **/etc/cluster/cluster.conf**에서 다음의 **cman** 속성을 제거합니다:

   - **cman two_node="1"**
   - **expected_votes="1"**

   3 노드와 2 노드 구성 간의 비교는 8.2.3절. “3 노드 및 2-노드 설정의 예 ” 에서 참조하십시오.

2. 값을 증가시켜 **config_version** 속성을 업데이트합니다 (예: **config_version="2"**에서 **config_version="3">**로 변경)

3. **/etc/cluster/cluster.conf**를 저장합니다.

4. **(옵션) ccs_config_validate** 명령을 실행하여 클러스터 스키마 (**cluster.rng**)에 대해 업데이트된 파일의 유효성을 검사합니다. 예:

   ```
   [root@example-01 ~]# ccs_config_validate
   Configuration validates
   ```

5. **cman_tool version -r** 명령을 실행하여 설정을 나머지 클러스터 노드에 전달합니다.

6. 업데이트된 설정 파일이 전달되었는지 확인합니다.

7. 업데이트된 설정 파일을 클러스터에 추가될 각 노드의 **/etc/cluster/**에 전달합니다. 예를 들어, **scp** 명령을 사용하여 클러스터에 추가될 각 노드에 업데이트된 설정 파일을 전달합니다.

8. 클러스터의 노드 수가 2개의 노드에서 2개 이상의 노드로 변경된 경우 다음과 같이 기존 클러스터 노드에서 클러스터 소프트웨어를 다시 시작해야 합니다.

   1. 각 노드에서 8.1.2절. “클러스터 소프트웨어 중지 ” 에 따라 클러스터 소프트웨어를 중지합니다. 예:

      ```
      [root@example-01 ~]# service rgmanager stop
      Stopping Cluster Service Manager:                          [  OK
      ]
      [root@example-01 ~]# service gfs2 stop
      Unmounting GFS2 filesystem (/mnt/gfsA):                    [  OK
      ]
      Unmounting GFS2 filesystem (/mnt/gfsB):                    [  OK
      ]
      [root@example-01 ~]# service clvmd stop
      Signaling clvmd to exit                                    [
      OK  ]
      clvmd terminated                                           [
      OK  ]
      [root@example-01 ~]# service cman stop
      Stopping cluster:
      ```

```
   Leaving fence domain...                                 [
OK  ]
   Stopping gfs_controld...                                [
OK  ]
   Stopping dlm_controld...                                [
OK  ]
   Stopping fenced...                                      [
OK  ]
   Stopping cman...                                        [
OK  ]
   Waiting for corosync to shutdown:                       [  OK
]
   Unloading kernel modules...                             [
OK  ]
   Unmounting configfs...                                  [
OK  ]
[root@example-01 ~]#
```

2. 각 노드에서 8.1.1절. “클러스터 소프트웨어 시작 ”에 따라 클러스터 소프트웨어를 시작합니다. 예:

```
[root@example-01 ~]# service cman start
Starting cluster:
   Checking Network Manager...                             [
OK  ]
   Global setup...                                         [
OK  ]
   Loading kernel modules...                               [
OK  ]
   Mounting configfs...                                    [
OK  ]
   Starting cman...                                        [
OK  ]
   Waiting for quorum...                                   [
OK  ]
   Starting fenced...                                      [
OK  ]
   Starting dlm_controld...                                [
OK  ]
   Starting gfs_controld...                                [
OK  ]
   Unfencing self...                                       [
OK  ]
   Joining fence domain...                                 [
OK  ]
[root@example-01 ~]# service clvmd start
Starting clvmd:                                            [
OK  ]
Activating VG(s):   2 logical volume(s) in volume group
"vg_example" now active
                                                           [
OK  ]
[root@example-01 ~]# service gfs2 start
Mounting GFS2 filesystem (/mnt/gfsA):                      [  OK
]
Mounting GFS2 filesystem (/mnt/gfsB):                      [  OK
```

```
]
[root@example-01 ~]# service rgmanager start
Starting Cluster Service Manager:                          [  OK
]
[root@example-01 ~]#
```

9. 클러스터에 추가될 각 노드에서 8.1.1절. “클러스터 소프트웨어 시작 ” 에 따라 클러스터 소프트웨어를 시작합니다. 예:

```
[root@example-01 ~]# service cman start
Starting cluster:
   Checking Network Manager...                             [  OK
]
   Global setup...                                         [  OK
]
   Loading kernel modules...                               [  OK
]
   Mounting configfs...                                    [  OK
]
   Starting cman...                                        [  OK
]
   Waiting for quorum...                                   [  OK
]
   Starting fenced...                                      [  OK
]
   Starting dlm_controld...                                [  OK
]
   Starting gfs_controld...                                [  OK
]
   Unfencing self...                                       [  OK
]
   Joining fence domain...                                 [  OK
]
[root@example-01 ~]# service clvmd start
Starting clvmd:                                            [  OK
]
Activating VG(s):   2 logical volume(s) in volume group "vg_example"
now active
                                                           [  OK
]
[root@example-01 ~]# service gfs2 start
Mounting GFS2 filesystem (/mnt/gfsA):                      [  OK  ]
Mounting GFS2 filesystem (/mnt/gfsB):                      [  OK  ]

[root@example-01 ~]# service rgmanager start
Starting Cluster Service Manager:                          [  OK  ]
[root@example-01 ~]#
```

10. 노드 중 하나에서 **clustat** 유틸리티를 사용하여 각 추가 노드가 클러스터의 일부로 실행되고 있는지를 확인합니다. 예:

```
[root@example-01 ~]#clustat
Cluster Status for mycluster @ Wed Nov 17 05:40:00 2010
Member Status: Quorate
```

```
 Member Name                              ID   Status
 ------ ----                              ---- ------
 node-03.example.com                          3 Online, rgmanager
 node-02.example.com                          2 Online, rgmanager
 node-01.example.com                          1 Online, Local,
rgmanager

 Service Name                    Owner (Last)
State
 ------- ----                    ----- ------                    --
---
 service:example_apache          node-01.example.com
started
 service:example_apache2         (none)
disabled
```

clustat 사용에 대한 보다 자세한 내용은 8.3절. "고가용성 서비스 관리 "에서 참조하십시오.

또한 **cman_tool status**를 사용하여 노드 투표, 노드 수, 쿼터 수를 확인할 수 있습니다. 예:

```
[root@example-01 ~]#cman_tool status
Version: 6.2.0
Config Version: 19
Cluster Name: mycluster
Cluster Id: 3794
Cluster Member: Yes
Cluster Generation: 548
Membership state: Cluster-Member
Nodes: 3
Expected votes: 3
Total votes: 3
Node votes: 1
Quorum: 2
Active subsystems: 9
Flags:
Ports Bound: 0 11 177
Node name: node-01.example.com
Node ID: 3
Multicast addresses: 239.192.14.224
Node addresses: 10.15.90.58
```

11. 노드에서 **clusvcadm** 유틸리티를 사용하여 실행중인 서비스를 새로 참여한 노드에 이전하거나 재배치합니다. 또한 비활성화된 서비스를 활성화할 수 있습니다. **clusvcadm** 사용에 대한 자세한 내용은 8.3절. "고가용성 서비스 관리 "에서 참조하십시오.

8.2.3. 3 노드 및 2-노드 설정의 예

3 노드와 2 노드 설정간의 비교는 다음의 예제를 참조하십시오.

예 8.1. 3 노드 클러스터 설정

```
<cluster name="mycluster" config_version="3">
   <cman/>
   <clusternodes>
```

```
     <clusternode name="node-01.example.com" nodeid="1">
         <fence>
            <method name="APC">
              <device name="apc" port="1"/>
             </method>
         </fence>
     </clusternode>
     <clusternode name="node-02.example.com" nodeid="2">
         <fence>
            <method name="APC">
              <device name="apc" port="2"/>
            </method>
         </fence>
     </clusternode>
     <clusternode name="node-03.example.com" nodeid="3">
         <fence>
            <method name="APC">
              <device name="apc" port="3"/>
            </method>
         </fence>
     </clusternode>
   </clusternodes>
   <fencedevices>
         <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="apc" passwd="password_example"/>
   </fencedevices>
   <rm>
       <failoverdomains>
           <failoverdomain name="example_pri" nofailback="0"
ordered="1" restricted="0">
              <failoverdomainnode name="node-01.example.com"
priority="1"/>
              <failoverdomainnode name="node-02.example.com"
priority="2"/>
              <failoverdomainnode name="node-03.example.com"
priority="3"/>
           </failoverdomain>
       </failoverdomains>
       <resources>
           <fs name="web_fs" device="/dev/sdd2" mountpoint="/var/www"
fstype="ext3"/>
           <ip address="127.143.131.100" monitor_link="yes"
sleeptime="10"/>
           <apache config_file="conf/httpd.conf" name="example_server"
server_root="/etc/httpd" shutdown_wait="0"/>
       </resources>
       <service autostart="0" domain="example_pri" exclusive="0"
name="example_apache" recovery="relocate">
           <fs ref="web_fs"/>
           <ip ref="127.143.131.100"/>
           <apache ref="example_server"/>
       </service>
       <service autostart="0" domain="example_pri" exclusive="0"
name="example_apache2" recovery="relocate">
           <fs name="web_fs2" device="/dev/sdd3" mountpoint="/var/www"
fstype="ext3"/>
```

```
            <ip address="127.143.131.101" monitor_link="yes"
sleeptime="10"/>
            <apache config_file="conf/httpd.conf" name="example_server2"
server_root="/etc/httpd" shutdown_wait="0"/>
        </service>
    </rm>
</cluster>
```

예 8.2. 2 노드 클러스터 설정

```
<cluster name="mycluster" config_version="3">
    <cman two_node="1" expected_votes="1"/>
    <clusternodes>
      <clusternode name="node-01.example.com" nodeid="1">
          <fence>
             <method name="APC">
               <device name="apc" port="1"/>
              </method>
          </fence>
      </clusternode>
      <clusternode name="node-02.example.com" nodeid="2">
          <fence>
             <method name="APC">
               <device name="apc" port="2"/>
             </method>
          </fence>
    </clusternodes>
    <fencedevices>
          <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="apc" passwd="password_example"/>
    </fencedevices>
    <rm>
        <failoverdomains>
            <failoverdomain name="example_pri" nofailback="0"
ordered="1" restricted="0">
                <failoverdomainnode name="node-01.example.com"
priority="1"/>
                <failoverdomainnode name="node-02.example.com"
priority="2"/>
            </failoverdomain>
        </failoverdomains>
        <resources>
            <fs name="web_fs" device="/dev/sdd2" mountpoint="/var/www"
fstype="ext3"/>
            <ip address="127.143.131.100" monitor_link="yes"
sleeptime="10"/>
            <apache config_file="conf/httpd.conf" name="example_server"
server_root="/etc/httpd" shutdown_wait="0"/>
        </resources>
        <service autostart="0" domain="example_pri" exclusive="0"
name="example_apache" recovery="relocate">
            <fs ref="web_fs"/>
```

```
            <ip ref="127.143.131.100"/>
            <apache ref="example_server"/>
        </service>
        <service autostart="0" domain="example_pri" exclusive="0"
name="example_apache2" recovery="relocate">
            <fs name="web_fs2" device="/dev/sdd3" mountpoint="/var/www"
fstype="ext3"/>
            <ip address="127.143.131.101" monitor_link="yes"
sleeptime="10"/>
            <apache config_file="conf/httpd.conf" name="example_server2"
server_root="/etc/httpd" shutdown_wait="0"/>
        </service>
    </rm>
</cluster>
```

8.3. 고가용성 서비스 관리

클러스터 상태 유틸리티 (Cluster Status Utility) , `clustat`, 클러스터 사용자 서비스 관리 유틸리티 (Cluster User Service Administration Utility), `clusvcadm`을 사용하여 고가용성 서비스를 관리할 수 있습니다. **`clustat`**는 클러스터의 상태를 표시하고 **`clusvcadm`**은 고가용성 서비스를 관리하는 방법을 제공합니다.

다음 부분에서는 **`clustat`** 및 **`clusvcadm`**을 사용하여 HA 서비스를 관리하기 위한 기본적인 내용을 설명합니다. 이는 다음과 같은 하위 섹션으로 구성되어 있습니다:

- 8.3.1절. “**`clustat`**를 사용하여 HA 서비스 상태 표시 ”
- 8.3.2절. “**`clusvcadm`**을 사용하여 HA 서비스 관리 ”

8.3.1. `clustat`를 사용하여 HA 서비스 상태 표시

`clustat`클러스터 전반의 상태를 표시합니다. 이는 멤버쉽 정보, 쿼터 보기, 모든 고가용성 서비스의 상태를 표시하고 **`clustat`** 명령이 실행되는 노드 (로컬)를 나타냅니다. 표 8.1. “서비스 상태 ”에서는 서비스가 될 수 있는 상태 및 **`clustat`**를 실행할 때 표시되는 상태를 설명합니다. 예 8.3. “**`clustat`** 표시 ”에서는 **`clustat`** 표시의 예를 보여줍니다. **`clustat`** 명령 실행에 대한 자세한 내용은 **`clustat`** man 페이지를 참조하십시오.

표 8.1. 서비스 상태

서비스 상태	설명
Started	서비스 리소스가 설정되어 서비스를 소유하는 클러스터 시스템에서 사용할 수 있습니다.
Recovering	서비스는 다른 노드에서 시작하기 위해 기다리고 있습니다.
Disabled	서비스가 비활성화되어 할당된 소유자가 없습니다. 비활성화된 서비스는 클러스터에 의해 자동으로 다시 시작되지 않습니다.

서비스 상태	설명
Stopped	중지된 상태입니다. 서비스는 다음의 서비스 또는 노드 변경 후에 시작을 위해 평가받게 됩니다. 이는 일시적인 상태입니다. 이 상태에서 서비스를 활성화하거나 비활성화할 수 있습니다.
Failed	서비스는 종료 상태로 추정됩니다. 서비스는 리소스의 *중지* 작업이 실패했을 때 이 상태를 유지합니다. 서비스가 이 상태가 된 후, **disable** 요청을 실행하기 전 할당된 리소스 (예: 마운트된 파일 시스템)가 없는지를 확인해야 합니다. 서비스가 이 상태가 되었을 때 실행할 수 있는 유일한 작업은 **disable**입니다.
Uninitialized	이 상태는 **clustat -f**를 시작 및 실행 중 특정 경우에 나타날 수 있습니다.

예 8.3. clustat 표시

```
[root@example-01 ~]#clustat
Cluster Status for mycluster @ Wed Nov 17 05:40:15 2010
Member Status: Quorate

 Member Name                             ID   Status
 ------ ----                             ---- ------
 node-03.example.com                         3 Online, rgmanager
 node-02.example.com                         2 Online, rgmanager
 node-01.example.com                         1 Online, Local,
rgmanager

 Service Name                   Owner (Last)                   State
 ------- ----                   ----- ------                   -----
 service:example_apache         node-01.example.com            started
 service:example_apache2        (none)
disabled
```

8.3.2. clusvcadm을 사용하여 HA 서비스 관리

clusvcadm 명령을 사용하여 HA 서비스를 관리할 수 있습니다. 이를 사용하여 다음의 작업을 실행할 수 있습니다:

- 서비스 활성화 및 시작
- 서비스 비활성화
- 서비스 중지
- 서비스 고정
- 서비스 고정 취소
- 서비스 마이그레이션 (가상 머신 서비스 전용)
- 서비스 재배치

- 서비스 재시작

표 8.2. "서비스 작업 "에서는 보다 자세하에 이 작업에 대해 설명하고 있습니다. 이러한 작업을 실행하는 방법에 대한 총괄적인 설명은 **clusvcadm** 유틸리티 man 페이지를 참조하십시오.

표 8.2. 서비스 작업

서비스 작업	설명	명령 구문
활성화 (Enable)	옵션으로 우선 대상 및 장애조치 도메인 규칙에 따라 서비스를 시작합니다. 이 두 옵션이 없는 상태에서 **clusvcadm**이 실행되는 로컬 호스트에서 서비스를 시작합니다. 원래의 *시작*이 실패하면, 서비스는 *재배치* 작업이 요구되는 것처럼 작동합니다 (표에서 **재배치 (Relocate)** 참조). 작업이 성공적으로 진행되는 서비스는 시작 (started) 상태가 됩니다.	**clusvcadm -e <service_name>** 또는 **clusvcadm -e <service_name> -m <member>** (-m 옵션을 사용하여 서비스를 시작하기 위해 우선 대상 멤버를 지정합니다.)
비활성화 (Disable)	서비스를 중지하고 비활성화 상태로 합니다. 이는 서비스가 *실패 (failed)* 상태에 있을 때 허용되는 유일한 동작입니다.	**clusvcadm -d <service_name>**
재배치 (Relocate)	서비스를 다른 노드로 이동합니다. 옵션으로 서비스를 받고자 하는 노드를 지정할 수 있지만, 해당 호스트에서 실행하기 위해 서비스의 실행 불가능 (예를 들어, 서비스 시작 실패 또는 호스트의 오프라인)이 이전을 못하게 하지 않으며 다른 노드가 선택됩니다. **rgmanager**는 클러스터에 있는 모든 권한있는 노드에서 서비스를 시작 시도합니다. 클러스터에 있는 권한 있는 대상 노드가 서비스를 시작할 수 없는 경우, 이전은 실패하고 서비스는 원래 소유자에서 다시 시작 시도합니다. 원래 소유자가 서비스를 다시 시작할 수 없는 경우 서비스는 *중지 (stopped)* 상태가 됩니다.	**clusvcadm -r <service_name>** 또는 **clusvcadm -r <service_name> -m <member>** (-m 옵션을 사용하여 서비스를 시작하기 위해 우선 대상 멤버를 지정합니다.)
중지 (Stop)	서비스를 중지하고 *중지 (stopped)* 상태가 되게 합니다.	**clusvcadm -s <service_name>**
고정 (Freeze)	현재 실행 중인 노드에서 서비스를 고정합니다. 이는 서비스의 상태 확인 및 노드가 실패하거나 rgmanager가 멈춘 상황에서 페일오버가 차단됩니다. 이 기능은 서비스를 보류하고 기본 리소스의 유지 보수를 가능하게 하는데 사용할 수 있습니다. *고정 (freeze)* 및 *고정 취소 (unfreeze)* 작업 사용에 대한 중요한 내용은"고정 (Freeze) 및 고정 취소 (Unfreeze) 작업의 사용을 고려 "에서 참조하십시오.	**clusvcadm -Z <service_name>**

서비스 작업	설명	명령 구문
고정 취소 (Unfreeze)	고정 취소 (Unfreeze) 서비스를 *고정 (freeze)* 상태에서 해제합니다. 이는 상태 점검을 다시 활성화합니다. *고정 (freeze)* 및 *고정 취소 (unfreeze)* 작업 사용에 대한 중요한 내용은"고정 (Freeze) 및 고정 취소 (Unfreeze) 작업의 사용을 고려 "에서 참조하십시오.	`clusvcadm -U <service_name>`
이전 (Migrate)	가상 머신을 다른 노드로 이동합니다. 대상 노드를 지정해야 합니다. 마이그레이션 실패는 실패의 내용에 따라 가상 머신이 *실패 (failed)* 상태가 되거나 또는 원래 소유자에서 시작 (started) 상태가 될 수 있습니다.	`clusvcadm -M <service_name> -m <member>`  중요 *이전 (migrate)* 작업의 경우, **`-m <member>`** 옵션을 사용하는 대상 노드를 *지정해야 합니다.*
다시 시작 (Restart)	현재 실행중인 노드에서 서비스를 다시 시작합니다.	`clusvcadm -R <service_name>`

8.3.2.1. 고정 (Freeze) 및 고정 취소 (Unfreeze) 작업의 사용을 고려

고정 작업을 사용하면 **rgmanager** 서비스의 부분적인 관리가 가능합니다. 예를 들어, 하나의 **rgmanager** 서비스에 데이터베이스와 웹 서버가 있을 경우, **rgmanager** 서비스를 고정하고 데이터베이스를 중지한 후, 관리를 수행하고 데이터베이스를 다시 시작 그리고 서비스를 고정 취소할 수 있습니다.

서비스가 고정되면 이는 다음과 같이 작동합니다:

- *상태* 검사가 비활성화됩니다.
- *시작* 동작이 비활성화됩니다.
- *중지* 동작이 비활성화됩니다.
- 장애 조치가 발생하지 않습니다 (서비스 소유자의 전원을 껐을 경우에도)

중요

이러한 지침을 준수하지 않으면 리소스가 여러 호스트에 할당될 수 있습니다:

- rgmanager를 다시 시작하기 전에 호스트를 다시 시작할 계획이 없는한 서비스가 고정되었을 때 rgmanager의 모든 인스턴스를 중지하지 *않습니다.*
- 보고된 서비스의 소유자가 클러스터에 다시 참가하여 rgmanager를 다시 시작할 때 까지 서비스를 고정 해제하지 *않습니다.*

8.4. 설정 업데이트

클러스터 설정 파일 업데이트는 클러스터 설정 파일 (**/etc/cluster/cluster.conf**)을 편집하고 이를 클러스터에 있는 각 노드에 전달함으로써 성립됩니다. 다음 절차 중 하나를 사용하여 설정을 업데이트할 수 있습니다:

- 8.4.1절. "**cman_tool version -r** 명령을 사용하여 설정 업데이트 "
- 8.4.2절. "**scp**를 사용하여 설정 업데이트 "

8.4.1. **cman_tool version -r** 명령을 사용하여 설정 업데이트

cman_tool version -r 명령을 사용하여 설정을 업데이트하려면, 다음 절차를 실행합니다:

1. 클러스터에 있는 노드 중 하나에서 **/etc/cluster/cluster.conf** 파일을 편집합니다.
2. 값을 증가시켜 **config_version** 속성을 업데이트합니다 (예: **config_version="2"**에서 **config_version="3">**로 변경)
3. **/etc/cluster/cluster.conf**를 저장합니다.
4. **cman_tool version -r** 명령을 실행하여 설정을 나머지 클러스터 노드에 전달합니다. 업데이트된 클러스터 설정 정보를 전달하려면 각 클러스터 노드에서 **ricci**가 실행되고 있어야 합니다.
5. 업데이트된 설정 파일이 전달되었는지 확인합니다.
6. 다음의 설정을 변경하면 이 단계 (클러스터 소프트웨어를 다시 시작)를 생략할 수 있습니다:
   - 클러스터 설정에서 노드 삭제 – *예외:*노드 수가 2개 이상에서 2개의 노드로 변경하는 경우. 클러스터에서 노드를 제거하는 방법 및 2개 이상의 노드에서 2 개의 노드로 변경하는 방법에 대한 자세한 내용은 8.2절. "노드 삭제 또는 추가 "에서 참조하십시오.
   - 클러스터 설정에 노드 추가 – *예외:* 노드 수가 2 개의 노드에서 2 개 이상의 노드로 변경하는 경우. 클러스터에 노드를 추가하는 방법 및 2 개의 노드에서 2 개 이상의 노드로 변경하는 방법에 대한 자세한 내용은 8.2.2절. "클러스터에 노드 추가 "에서 참조하십시오.
   - 데몬이 정보를 기록하는 방법에서의 변경 사항
   - HA 서비스/VM 관리 (추가, 편집 또는 삭제)
   - 리소스 관리 (추가, 편집 또는 삭제)
   - 장애 조치 도메인 관리 (추가, 편집, 또는 삭제)

   그렇지 않을 경우 다음과 같이 클러스터 소프트웨어를 다시 시작해야 합니다:

   1. 각 노드에서 8.1.2절. "클러스터 소프트웨어 중지 "에 따라 클러스터 소프트웨어를 중지합니다. 예:

```
[root@example-01 ~]# service rgmanager stop
Stopping Cluster Service Manager:                          [  OK
]
[root@example-01 ~]# service gfs2 stop
Unmounting GFS2 filesystem (/mnt/gfsA):                    [  OK
]
Unmounting GFS2 filesystem (/mnt/gfsB):                    [  OK
]
[root@example-01 ~]# service clvmd stop
```

```
Signaling clvmd to exit                                    [
OK  ]
clvmd terminated                                           [
OK  ]
[root@example-01 ~]# service cman stop
Stopping cluster:
   Leaving fence domain...                                 [
OK  ]
   Stopping gfs_controld...                                [
OK  ]
   Stopping dlm_controld...                                [
OK  ]
   Stopping fenced...                                      [
OK  ]
   Stopping cman...                                        [
OK  ]
   Waiting for corosync to shutdown:                       [  OK
]
   Unloading kernel modules...                             [
OK  ]
   Unmounting configfs...                                  [
OK  ]
[root@example-01 ~]#
```

2. 각 노드에서 8.1.1절. “클러스터 소프트웨어 시작 ”에 따라 클러스터 소프트웨어를 시작합니다. 예:

```
[root@example-01 ~]# service cman start
Starting cluster:
   Checking Network Manager...                             [
OK  ]
   Global setup...                                         [
OK  ]
   Loading kernel modules...                               [
OK  ]
   Mounting configfs...                                    [
OK  ]
   Starting cman...                                        [
OK  ]
   Waiting for quorum...                                   [
OK  ]
   Starting fenced...                                      [
OK  ]
   Starting dlm_controld...                                [
OK  ]
   Starting gfs_controld...                                [
OK  ]
   Unfencing self...                                       [
OK  ]
   Joining fence domain...                                 [
OK  ]
[root@example-01 ~]# service clvmd start
Starting clvmd:                                            [
OK  ]
Activating VG(s):   2 logical volume(s) in volume group
"vg_example" now active
```

```
                                                           [
OK  ]
[root@example-01 ~]# service gfs2 start
Mounting GFS2 filesystem (/mnt/gfsA):                      [  OK
]
Mounting GFS2 filesystem (/mnt/gfsB):                      [  OK
]
[root@example-01 ~]# service rgmanager start
Starting Cluster Service Manager:                          [  OK
]
[root@example-01 ~]#
```

클러스터 소프트웨어 시작 및 중지는 시작 시에만 확인하는 설정 변경이 실행 중인 설정에 포함되도록 합니다.

7. 클러스터 노드에서 **cman_tools nodes**를 실행하여 노드가 클러스터에서 멤버로 작동하는지 확인합니다 (상태 컬럼 "Sts"에서 "M"로 표시). 예:

```
[root@example-01 ~]# cman_tool nodes
Node  Sts   Inc   Joined               Name
   1   M    548   2010-09-28 10:52:21  node-01.example.com
   2   M    548   2010-09-28 10:52:21  node-02.example.com
   3   M    544   2010-09-28 10:52:21  node-03.example.com
```

8. 노드에서 **clustat** 유틸리티를 사용하여, HA 서비스가 예상대로 실행되는지 확인합니다. 또한 **clustat**는 클러스터 노드의 상태를 표시합니다. 예:

```
[root@example-01 ~]#clustat
Cluster Status for mycluster @ Wed Nov 17 05:40:00 2010
Member Status: Quorate

 Member Name                             ID   Status
 ------ ----                             ---- ------
 node-03.example.com                         3 Online, rgmanager
 node-02.example.com                         2 Online, rgmanager
 node-01.example.com                         1 Online, Local,
rgmanager

 Service Name                   Owner (Last)
State
 ------- ----                   ----- ------                    --
---
 service:example_apache         node-01.example.com
started
 service:example_apache2        (none)
disabled
```

9. 클러스터가 예상대로 작동하고 있을 경우, 설정 업데이트가 완료됩니다.

8.4.2. scp를 사용하여 설정 업데이트

scp 명령을 사용하여 설정을 업데이트하려면 다음 절차를 실행합니다:

1. 각 노드에서 8.1.2절. "클러스터 소프트웨어 중지 "에 따라 클러스터 소프트웨어를 중지합니다. 예:

```
[root@example-01 ~]# service rgmanager stop
Stopping Cluster Service Manager:                          [  OK  ]
[root@example-01 ~]# service gfs2 stop
Unmounting GFS2 filesystem (/mnt/gfsA):                    [  OK  ]
Unmounting GFS2 filesystem (/mnt/gfsB):                    [  OK  ]
[root@example-01 ~]# service clvmd stop
Signaling clvmd to exit                                    [  OK
]
clvmd terminated                                           [  OK
]
[root@example-01 ~]# service cman stop
Stopping cluster:
   Leaving fence domain...                                 [  OK
]
   Stopping gfs_controld...                                [  OK
]
   Stopping dlm_controld...                                [  OK
]
   Stopping fenced...                                      [  OK
]
   Stopping cman...                                        [  OK
]
   Waiting for corosync to shutdown:                       [  OK  ]
   Unloading kernel modules...                             [  OK
]
   Unmounting configfs...                                  [  OK
]
[root@example-01 ~]#
```

2. 클러스터에 있는 노드 중 하나에서 **/etc/cluster/cluster.conf** 파일을 편집합니다.

3. 값을 증가시켜 **config_version** 속성을 업데이트합니다 (예: **config_version="2"**에서 **config_version="3">**로 변경)

4. **/etc/cluster/cluster.conf**를 저장합니다.

5. **ccs_config_validate** 명령을 실행하여 클러스터 스키마 (**cluster.rng**)에 대한 업데이트된 파일의 유효성을 검사합니다.

```
[root@example-01 ~]# ccs_config_validate
Configuration validates
```

6. 업데이트된 파일이 유효한 경우, **scp** 명령을 사용하여 각 클러스터 노드에 있는 **/etc/cluster/**에 전달합니다.

7. 업데이트된 설정 파일이 전달되었는지 확인합니다.

8. 각 노드에서 8.1.1절. “클러스터 소프트웨어 시작 ”에 따라 클러스터 소프트웨어를 시작합니다. 예:

```
[root@example-01 ~]# service cman start
Starting cluster:
   Checking Network Manager...                             [  OK
]
   Global setup...                                         [  OK
]
```

```
   Loading kernel modules...                                  [  OK
]
   Mounting configfs...                                       [  OK
]
   Starting cman...                                           [  OK
]
   Waiting for quorum...                                      [  OK
]
   Starting fenced...                                         [  OK
]
   Starting dlm_controld...                                   [  OK
]
   Starting gfs_controld...                                   [  OK
]
   Unfencing self...                                          [  OK
]
   Joining fence domain...                                    [  OK
]
[root@example-01 ~]# service clvmd start
Starting clvmd:                                               [  OK
]
Activating VG(s):   2 logical volume(s) in volume group "vg_example"
now active
                                                              [  OK
]
[root@example-01 ~]# service gfs2 start
Mounting GFS2 filesystem (/mnt/gfsA):                         [  OK  ]
Mounting GFS2 filesystem (/mnt/gfsB):                         [  OK  ]
[root@example-01 ~]# service rgmanager start
Starting Cluster Service Manager:                             [  OK  ]
[root@example-01 ~]#
```

9. 클러스터 노드에서 **cman_tools nodes**를 실행하여 노드가 클러스터에서 멤버로 작동하는지 확인합니다 (상태 컬럼 "Sts"에서 "M"로 표시). 예:

```
[root@example-01 ~]# cman_tool nodes
Node  Sts   Inc   Joined               Name
   1   M    548   2010-09-28 10:52:21  node-01.example.com
   2   M    548   2010-09-28 10:52:21  node-02.example.com
   3   M    544   2010-09-28 10:52:21  node-03.example.com
```

10. 노드에서 **clustat** 유틸리티를 사용하여, HA 서비스가 예상대로 실행되는지 확인합니다. 또한 **clustat**는 클러스터 노드의 상태를 표시합니다. 예:

```
[root@example-01 ~]#clustat
Cluster Status for mycluster @ Wed Nov 17 05:40:00 2010
Member Status: Quorate

 Member Name                             ID   Status
 ------ ----                             ---- ------
 node-03.example.com                         3 Online, rgmanager
 node-02.example.com                         2 Online, rgmanager
 node-01.example.com                         1 Online, Local,
rgmanager
```

```
 Service Name                    Owner (Last)
State
 ------- ----                    ----- ------                    --
---
 service:example_apache          node-01.example.com
started
 service:example_apache2         (none)
disabled
```

11. 클러스터가 예상대로 작동하고 있을 경우, 설정 업데이트가 완료됩니다.

9장. 클러스터에 있는 문제를 진단 및 수정

원래 클러스터 문제는 해결하기 어려울 수 있습니다. 이는 단일 시스템 상의 문제 진단과는 다르게 시스템 클러스터의 고도의 복잡성 때문입니다. 하지만 시스템 관리자가 클러스터를 배포 또는 관리할 때 발생할 수 있는 일반적인 문제가 있습니다. 이러한 문제를 해결하는 방법을 이해하면 클러스터 배포 및 관리를 보다 용이하게 할 수 있습니다.

다음 부분에서는 일반적인 클러스터 문제와 해결 방법에 대해 설명합니다. 추가 지원은 **knowledge base**를 참조하시거나 **Red Hat** 지원 담당자에게 문의하시기 바랍니다. 문제가 **GFS2** 파일 시스템과 관련되어 있는 경우, *Global File System 2*문서에서 일반적인 **GFS2** 문제 해결 방법을 참조하십시오.

9.1. 설정 변경 사항은 적용되지 않음

클러스터 설정을 변경한 경우 이러한 변경 사항을 클러스터에 있는 모든 노드에 전달해야 합니다.

- **Conga**를 사용하여 클러스터를 설정할 때 변경 사항을 적용하면 **Conga**는 자동으로 변경 내용을 전달합니다.
- **ccs** 명령을 사용하여 클러스터 설정 변경 사항 전달에 대한 자세한 내용은 5.15절. “클러스터 노드에 설정 파일 전달 ”에서 참조하십시오.
- 명령행 도구를 사용하여 클러스터 설정 변경사항 전달에 대한 자세한 내용은 8.4절. “설정 업데이트 ”에서 참조하십시오.

클러스터에 다음과 같은 설정을 변경하는 경우 변경 사항을 적용하기 위해 변경 사항을 전달한 후 클러스터를 다시 시작할 필요가 없습니다.

- 클러스터 설정에서 노드 삭제 – *예외:*노드 수가 **2**개 이상에서 **2**개의 노드로 변경하는 경우.
- 클러스터 설정에 노드 추가 – *예외:* 노드 수를 **2** 개의 노드에서 **2** 개 이상의 노드로 변경하는 경우.
- 로깅 설정을 변경합니다.
- HA 서비스 또는 VM 구성 요소를 추가, 편집 또는 삭제합니다.
- 클러스터 리소스를 추가, 편집 또는 삭제합니다.
- 페일오버 도메인을 추가, 편집 또는 삭제합니다.

클러스터에 다른 설정을 변경하는 경우 이러한 변경 사항을 반영하기 위해 클러스터를 다시 시작해야 합니다. 다음과 같은 클러스터 설정 변경 사항의 경우 이를 반영하기 위해 클러스터를 다시 시작해야 합니다:

- 클러스터 설정 파일에서 **two_node** 옵션을 추가 또는 삭제
- 클러스터 이름을 변경
- **corosync** 또는 **openais** 타이머를 변경
- 쿼럼 디스크의 휴리스틱스를 추가/ 변경/ 삭제, 쿼럼 디스크 타이머 변경, 쿼럼 디스크 장치 변경. 이러한 변경 사항을 적용하려면 **qdiskd** 데몬의 글로벌 재시작이 필요합니다.
- **rgmanager**의 **central_processing** 모드를 변경합니다. 이러한 변경 사항을 적용하려면 **rgmanager**의 글로벌 재시작이 필요합니다.
- 멀티캐스트 주소를 변경

- 전송 모드를 UDP 멀티캐스트에서 UDP 유니캐스트로 전환 또는 UDP 유니캐스트에서 UDP 멀티캐스트로 전환

Conga, **ccs** 명령, 명령행 도구를 사용하여 클러스터를 다시 시작할 수 있습니다.

- **Conga**를 사용하여 클러스터를 다시 시작하는 내용은 4.4절. “클러스터 시작, 중지, 다시 시작, 삭제 ”에서 참조하십시오.
- **ccs** 명령을 사용하여 클러스터를 다시 시작하는 내용은 6.2절. “클러스터 시작 및 중지 ” 에서 참조하십시오.
- 명령행 도구를 사용하여 클러스터를 다시 시작하는 내용은 8.1절. “클러스터 소프트웨어 시작 및 중지 ”에서 참조하십시오.

9.2. 클러스터를 구성할 수 없음

새 클러스터 구성에 문제가 있을 경우, 다음 사항을 확인하십시오:

- 이름이 올바르게 설정되어 있는지 확인합니다. **cluster.conf** 파일에 있는 클러스터 노드 이름은 클러스터가 통신을 위해 사용하는 네트워크에서 클러스터의 주소를 해결하는데 사용되는 이름에 해당해야 합니다. 예들 들어, 클러스터 노드 이름이 **nodea** 및 **nodeb**인 경우, 두 노드가 **/etc/cluster/cluster.conf** 파일 및 **/etc/hosts** 파일에 이 이름과 일치하는 항목이 있는지 확인합니다.
- 클러스터가 노드 간의 통신을 위해 멀티캐스트를 사용하는 경우 멀티캐스트 트래픽이 차단되어 있지 않는지, 지연되지 않는지를 확인하거나, 통신을 위해 클러스터가 사용하고 있는 네트워크에서 방해되지 않는지를 확인합니다. 일부 Cisco 스위치에는 멀티캐스트 트래픽을 지연시킬 수 있는 기능이 있다는 점에 유의하십시오.
- **telnet** 또는 **SSH**를 사용하여 원격 노드에 도달할 수 있는지 확인합니다.
- **ethtool eth1 | grep link** 명령을 실행하여 이더넷 링크가 활성화되어 있는지 확인합니다.
- 각 노드에서 **tcpdump** 명령을 사용하여 네트워크 트래픽을 확인합니다.
- 노드 간에 방화벽 규칙이 통신을 차단하지 않는지 확인합니다.
- 인터 노드 통신에 클러스터가 사용하는 인터페이스가 0, 1, 2 이외의 본딩 모드를 사용하지 않도록 합니다. (본딩 모드 0 및 2는 Red Hat Enterprise Linux 6.4에서 지원됩니다.)

9.3. 차단 또는 재부팅 후 노드가 클러스터에 다시 참여할 수 없음

차단 또는 재부팅 후 노드가 클러스터에 다시 참여하지 않으면 다음 사항을 확인합니다:

- Cisco Catalyst 스위치를 통해 트래픽을 통과시키는 클러스터에 이러한 문제가 발생할 수 있습니다.
- 모든 클러스터 노드에 동일한 버전의 **cluster.conf** 파일이 있는지 확인합니다. **cluster.conf** 파일이 노드 중 하나에서라도 다른 경우 노드는 차단 후 클러스터에 참여할 수 없을 수 있습니다.

  Red Hat Enterprise Linux 6.1에서 다음 명령을 사용하여 호스트의 클러스터 설정 파일에 지정된 모든 노드가 동일한 클러스터 설정 파일을 갖는지를 확인할 수 있습니다:

  ```
  ccs -h host --checkconf
  ```

ccs 명령에 대한 보다 자세한 내용은 *5장. ccs 명령으로 Red Hat 고가용성 추가 기능 설정*및 *6장. ccs로 Red Hat 고가용성 추가 기능 관리*에서 참조하십시오.

- 클러스터에 참여 시도하고 있는 노드에서 클러스터 서비스에 대해 **chkconfig on**을 설정했는지 확인합니다.
- 방화벽 규칙이 클러스터에 있는 다른 노드외의 통신에서 노드를 차단하지 않는지 확인합니다.

9.4. 클러스터 데몬 충돌

RGManager에는 주요 **rgmanager** 프로세스가 예기치 않게 실패하면 호스트를 재부팅하는 워치독 프로세스가 있습니다. 이렇게 하면 클러스터 노드가 차단되어 **rgmanager**가 다른 호스트에서 서비스를 복구합니다. 워치독 데몬이 주요 **rgmanager** 프로세스가 충돌한 것을 감지하면 클러스터 노드를 재부팅하여 활성 클러스터 노드는 클러스터 노드가 이탈한 것을 감지하고 이를 클러스터에서 제거합니다.

프로세스 ID(PID)의 낮은 번호는 워치독 프로세스로 자식 프로세스 (PID 번호가 높은 프로세스)가 충돌한 경우 조치를 취합니다. **gcore**를 사용하여 높은 PID 번호를 갖는 프로세스 코어를 캡처하여 충돌된 데몬의 문제 해결을 지원합니다.

코어를 캡쳐 및 보기에 필요한 패키지를 설치하고 **rgmanager** 및 **rgmanager-debuginfo**가 동일한 버전인지 확인합니다. 그렇지 않으면 캡쳐한 애플리케이션 코어를 사용하지 못 할 수 있습니다.

```
$ yum -y --enablerepo=rhel-debuginfo install gdb rgmanager-debuginfo
```

9.4.1. 런타임에서 rgmanager 코어 캡처

시작할 때 실행하는 두 개의 **rgmanager** 프로세스가 있습니다. 큰 PID를 갖는 **rgmanager** 프로세스의 코어를 캡처해야 합니다.

다음은 **ps** 명령의 출력 결과의 예로 **rgmanager**의 두 개의 프로세스를 보여주고 있습니다.

```
$ ps aux | grep rgmanager | grep -v grep

root     22482  0.0  0.5  23544  5136 ?        S<Ls Dec01   0:00 rgmanager
root     22483  0.0  0.2  78372  2060 ?        S<l  Dec01   0:47 rgmanager
```

다음 예제에서 **pidof** 프로그램을 사용하여 코어를 생성하는데 적절한 PID인 큰 PID 번호를 자동으로 지정하게 합니다. 전체 명령은 큰 PID 번호를 갖는 프로세스 **22483**의 애플리케이션 코어를 캡처합니다.

```
$ gcore -o /tmp/rgmanager-$(date '+%F_%s').core $(pidof -s rgmanager)
```

9.4.2. 데몬 충돌 시 코어를 캡처

디폴트로 **/etc/init.d/functions** 스크립트는 **/etc/init.d/rgmanager**에 의해 호출된 데몬에서 코어 파일을 차단합니다. 애플리케이션 코어를 생성하기 위한 데몬의 경우 해당 옵션을 활성화해야 합니다. 이 단계는 애플리케이션 코어를 캡처해야 하는 모든 클러스터 노드에서 실행해야 합니다.

rgmanager 데몬 충돌 시 코어 파일을 생성하려면 **/etc/sysconfig/cluster** 파일을 편집합니다. ***DAEMONCOREFILELIMIT*** 매개 변수는 프로세스가 충돌한 경우 데몬이 코어 파일을 만들 수 있도록 합니다. **-w** 옵션은 워치독 프로세스가 실행되지 않도록 합니다. 워치독 데몬은 **rgmanager**가 충돌하는 경우,

경우에 따라 워치독 데몬이 실행되고 있고 코어 파일이 생성되지 않을 경우 클러스터 노드를 다시 시작해야 합니다 따라서 코어 파일을 캡처하기 위해 이를 비활성화해야 합니다.

```
DAEMONCOREFILELIMIT="unlimited"
RGMGR_OPTS="-w"
```

rgmanager를 다시 시작하여 새 설정 옵션을 활성화합니다:

```
service rgmanager restart
```

참고

클러스터 노드에서 클러스터 서비스가 실행되고 있을 경우 이는 실행 중인 서비스를 잘못된 상태로 내버려둘 가능성이 있습니다.

코어 파일은 **rgmanager** 프로세스의 충돌에 의해 생성되었을 때 작성됩니다.

```
ls /core*
```

출력 결과는 다음과 유사하게 나타납니다:

```
/core.11926
```

애플리케이션 코어를 캡처하기 위해 **rgmanager**를 다시 시작하기 전 **/** 디렉토리 아래에 있는 모든 오래된 코어 파일을 이동하거나 삭제합니다. **rgmanager** 충돌 경험이 있는 클러스터 노드는 워치독 프로세스가 실행되고 있지 않은지 확인하기 위해 코어가 캡처된 후 재부팅 또는 차단해야 합니다.

9.4.3. gdb 백트레이스 세션 기록

코어 파일을 캡처한 후, GNU 디버거인 **gdb**를 사용하여 내용을 확인할 수 있습니다. 영향을 받은 시스템에서 코어 파일에 있는 **gdb**의 스크립트 세션을 기록하려면 다음 명령을 실행합니다:

```
$ script /tmp/gdb-rgmanager.txt
$ gdb /usr/sbin/rgmanager /tmp/rgmanager-.core.
```

gdb 세션이 시작되고 **script**는 이를 적절한 텍스트 파일에 기록합니다. **gdb**에서 다음 명령을 실행합니다:

```
(gdb) thread apply all bt full
(gdb) quit
```

ctrl-D를 눌러 스크립트 세션을 중지하고 이를 텍스트 파일로 저장합니다.

9.5. 클러스터 서비스 중지

클러스터 서비스가 노드를 차단하려 할 때, 클러스터 서비스는 차단 작업이 성공적으로 완료될 때 까지 중지됩니다. 따라서 클러스터 제어 스토리지 또는 서비스가 중단되어 클러스터 노드가 클러스터 멤버쉽 보기를 잘못 표시하거나 노드를 차단 시도할 때 클러스터가 중단되어 복구를 위해 노드를 다시 시작해야 할 경우 다음 사항을 확인하십시오:

- 클러스터는 노드를 차단 시도하여 차단 작업이 실패했을 수 있습니다.

- 모든 노드에 있는 **/var/log/messages** 파일을 확인하여 실패한 차단 메세지가 있는지 확인합니다. 있을 경우, 클러스터에 있는 노드를 다시 시작하고 차단 장치를 올바르게 설정합니다.
- 9.8절. "2 노드 클러스터에 있는 각 노드는 두 번째 노드 정지를 보고 " 에서 설명하고 있듯이 네트워크 파티션이 발생하지 않았는지를 확인하고 노드 간의 통신이 아직 가능하며 네트워크가 활성화되어 있는지를 확인합니다.
- 노드가 클러스터를 탈퇴하면 나머지 노드는 쿼럼 (정족수)에 부족할 수 도 있습니다. 클러스터는 작동하기 위해 쿼럼에 도달해야 합니다. 클러스터가 쿼럼을 충족하지 않은 상태로 노드가 삭제되는 경우 서비스와 스토리지는 중단됩니다. 예상 투표 수를 조정하거나 클러스터에 필요한 노드 양을 복원해야 합니다.

참고

fence_node 명령이나 **Conga**를 사용하여 노드를 수동으로 차단할 수 있습니다. 보다 자세한 내용은 **fence_node** man 페이지 및 4.3.2절. "노드가 클러스터를 탈퇴 또는 참여하는 원인 "에서 참조하십시오.

9.6. 클러스터 서비스가 시작되지 않음

클러스터 제어 서비스가 시작되지 않을 경우 다음 사항을 확인합니다.

- **cluster.conf** 파일에 있는 서비스 설정에 구문 오류가 있을 수 있습니다. **rg_test** 명령을 사용하여 설정 구문을 확인할 수 있습니다. 설정 또는 구문 오류가 있을 경우, **rg_test**는 문제점에 대해 알려줍니다.

```
$ rg_test test /etc/cluster/cluster.conf start service servicename
```

rg_test 명령에 대한 자세한 내용은 C.5절. "서비스 및 리소스 순서 디버그 및 테스트 " 에서 참조하십시오.

설정이 유효하면 리소스 그룹 관리자의 로깅을 증가하여 메세지 로그를 읽고 서비스 실패 원인에 대해 확인합니다. **cluster.conf** 파일에 있는 **loglevel="7"** 매개 변수를 **rm** 태그에 추가하여 로그 수준을 증가시킬 수 있습니다. 그러면 클러스터 서비스 시작, 중지, 마이그레이션과 관련한 로그 메세지의 증가된 세부 정보를 얻을 수 있습니다.

9.7. 클러스터 제어 서비스의 마이그레이션 실패

클러스터 제어 서비스가 다른 노드로 마이그레이션 실패하였으나 서비스가 일부 특정 노드에서 시작할 경우, 다음 사항을 확인합니다.

- 서비스를 실행하는데 필요한 리소스가 서비스를 실행하기 위해 필요한 클러스터의 모든 노드에 있는지 확인합니다. 예를 들어, 클러스터된 서비스가 특정 위치에 있는 스크립트 파일이거나 특정 마운트 지점으로 마운트된 파일 시스템일 경우, 이러한 리소스는 클러스터의 모든 노드에 있는 예상되는 위치에서 사용할 수 있는지 확인해야 합니다.
- 장애 조치 도메인, 서비스 종속성, 서비스 배타성이 예상대로 서비스를 노드로 마이그레이션할 수 없는 방식으로 설정되어 있지 않는지 확인합니다.
- 문제가 되는 서비스가 가상 머신 리소스일 경우, 올바른 설정 작업 모두가 완료되었는지 확인하기 위해 문서를 체크합니다.

- 9.6절. “클러스터 서비스가 시작되지 않음 ” 에서 설명하고 있듯이 리소스 그룹 관리자의 로깅을 증가시켜 마이그레이션을 위한 서비스 시작 실패의 원인이 무엇인지를 확인하기 위해 메세지 로그를 읽습니다.

9.8. 2 노드 클러스터에 있는 각 노드는 두 번째 노드 정지를 보고

클러스터가 2 노드 클러스터이며 각 노드가 활성화되어 있지만 다른 노드는 비활성화되어 있다고 보고하는 경우, 이는 클러스터 노드가 클러스터 하트비트 네트워크에서 멀티캐스트를 통해 서로 통신할 수 없음을 나타내는 것입니다. 이는 **"split brain"** 또는 **"network partition"**이라고 합니다. 이 문제를 확인하려면 9.2절. “클러스터를 구성할 수 없음 ” 에 요약된 사항을 확인합니다.

9.9. LUN 경로 장애에서 노드가 차단됨

클러스터에 있는 노드 또는 노드 집합이 LUN 경로 실패시 마다 차단되는 경우, 멀티패스된 스토리지에서 쿼럼 디스크를 사용해야 할 수 도 있습니다. 쿼럼 디스크를 사용하고 있고 쿼럼 디스크가 멀티패스된 스토리지에 있는 경우 경로 장애를 허용하기 위해 모든 시간을 제대로 설정했는지 확인합니다.

9.10. 쿼럼 디스크가 클러스터 멤버로 표시되지 않음

쿼럼 디스크를 사용하도록 시스템이 구성되어 있으나 쿼럼 디스크가 클러스터 멤버로 표시되지 않을 경우 다음 사항을 확인하십시오.

- **`qdisk`** 서비스에 대해 **`chkconfig on`**이 설정되어 있는지 확인합니다.
- **`qdisk`** 서비스를 시작했는지 확인합니다.
- 쿼럼 디스크가 클러스터로 등록되려면 몇 분 정도 걸릴 수 있음에 유의하십시오. 이는 일반적이고 정상적인 동작입니다.

9.11. 비정상적인 장애 조치 동작

클러스터 서버에서 흔히 발생하는 문제는 비정상적인 장애 조치 동작입니다. 다른 서비스가 시작했을 때 서비스를 중지하거나 또는 장애 조치시 서비스는 시작을 거부할 수 있습니다. 이는 장애 조치 도메인, 서비스 종속성, 서비스 배타성으로 구성된 장애 조치에 있어서 시스템의 복잡성으로 인한 것일 수 있습니다. 보다 간단한 서비스 또는 장애 조치 도메인 설정을 사용 시도 후에도 문제가 계속 발생하는지 확인합니다. 모든 상황에서 이러한 기능이 장애 조치에 어떻게 영향을 미치는 지를 명확하게 파악하지 않는한 서비스 배타성 및 종속성과 같은 기능은 피해야 합니다.

9.12. 노드 차단이 무작위로 발생

노드가 무작위로 차단되는 경우 다음 사항을 확인하십시오.

- 차단 장치의 근본적인 문제는 *항상* 노드가 토큰을 분실하는 것입니다. 이는 다른 클러스터와의 통신을 상실하여 하트비트 전송이 중지됨을 의미합니다.
- 지정된 토큰 간격 내에서 하트 비트를 시스템에 반환하지 않는 경우 모두 차단될 수 있습니다. 기본값으로 토큰 간격은 10초입니다. 이는 원하는 값 (밀리토 단위)을 **`cluster.conf`** 파일에 있는 totem 태그의 토큰 매개변수에 추가하여 지정할 수 있습니다. (예: 30 초의 경우 **`totem token="30000"`**을 설정)
- 네트워크가 예상하는데로 정상적으로 작동하는지 확인합니다.
- 인터 노드 통신에 클러스터가 사용하는 인터페이스가 0, 1, 2 이외의 본딩 모드를 사용하지 않도록 합니다. (본딩 모드 0 및 2는 Red Hat Enterprise Linux 6.4에서 지원됩니다.)

- 시스템이 **"freezing"** 또는 커널 패닉을 일으키고 있는지를 확인하기 위해 측정합니다. **kdump** 유틸리티를 설정하고 차단 장치 중 하나에서 코어를 얻는지 확인합니다.
- 차단에 잘못된 원인을 발생시킬 수 있는 상황이 발생하지 않도록 확인합니다.예를 들어, 스토리지 장애로 인해 쿼럼 디스크가 노드를 배출하거나 Oracle RAC과 같은 타사 제품이 외부 조건으로 인해 노드를 다시 시작하는 등입니다. 메세지 로그는 이러한 문제를 판단하는데 있어서 매우 유용합니다. 차단 장치 또는 노드를 다시 시작할 때 마다, 이것이 발생한 시점에서 클러스터에 있는 모든 노드의 메세지 로그를 검사하는 것은 표준 작업이 되어야 합니다.
- 예상대로 시스템이 하트비트에 반응하지 않을 수 있는 하드웨어 오류에 대해 시스템을 철저히 검사합니다.

9.13. DLM (DISTRIBUTED LOCK MANAGER) 용 디버그 로깅은 활성화되어 있어야 함

필요한 경우 활성화할 수 있는 두 가지 DLM (Distributed Lock Manager) 용 디버그 옵션이 있습니다. DLM 커널 디버깅과 POSIX 잠금 디버깅 옵션입니다.

DLM 디버깅을 활성화하려면, **/etc/cluster/cluster.conf** 파일을 편집하여 설정 옵션을 **dlm** 태그에 추가합니다. **log_debug** 옵션은 DLM 커널 디버깅 메세지를 활성화하며 **plock_debug** 옵션은 POSIX 잠금 디버깅 메세지를 활성화합니다.

다음의 **/etc/cluster/cluster.conf** 파일의 예제 부분에서는 두 가지 DLM 디버깅 옵션 모두를 활성화하는 **dlm** 태그를 보여줍니다:

```
<cluster config_version="42" name="cluster1">
  ...
  <dlm log_debug="1" plock_debug="1"/>
  ...
</cluster>
```

/etc/cluster/cluster.conf 파일을 편집한 후, **cman_tool version -r** 명령을 실행하여 나머지 클러스터 노드에 설정을 전달합니다.

10장. RED HAT 고가용성 추가 기능을 사용하여 SNMP 설정

Red Hat Enterprise Linux 6.1 릴리즈 및 이후 버전에서 Red Hat 고가용성 추가 기능은 SNMP 트랩에 대한 지원을 제공합니다. 다음 부분에서는 SNMP를 위한 시스템 설정 방법에 대해 설명하고 다음으로 Red Hat 고가용성 추가 기능이 특정 클러스터 이벤트에 대해 생성하는 트랩에 대해 요약 설명합니다.

10.1. SNMP 및 RED HAT 고가용성 추가 기능

Red Hat 고가용성 추가 기능 SNMP 서브 에이전트는 **foghorn**으로 SNMP 트랩을 생성합니다. **foghorn** 서브 에이전트는 AgentX 프로토콜을 통해 **snmpd** 데몬과 통신합니다. **foghorn** 서브에이전트는 SNMP 트랩만을 생성하며 **get** 또는 **set**과 같은 다른 SNMP 옵션을 지원하지 않습니다.

현재 **foghorn** 서브 에이전트에 대해 **config** 옵션이 없습니다. 이는 특정 소켓을 사용하기 위해 설정될 수 없으며 기본 AgentX 소켓만이 현재 지원되고 있습니다.

10.2. RED HAT 고가용성 추가 기능을 사용하여 SNMP 설정

Red Hat 고가용성 추가 기능을 사용하여 SNMP를 설정하려면, 클러스터에 있는 각 노드에서 다음 단계를 수행하여 필요한 서비스가 활성화되어 실행되고 있는지를 확인합니다.

1. Red Hat 고가용성 추가 기능으로 SNMP 트랩을 사용하려면 **snmpd** 서비스가 필요하며 이는 마스터 에이전트처럼 작동합니다. **foghorn** 서비스가 하위 에이전트이며 AgentX 프로토콜을 사용하기 때문에 **/etc/snmp/snmpd.conf** 파일에 다음과 같은 행을 추가하여 AgentX 지원을 활성화해야 합니다:

   ```
   master agentx
   ```

2. SNMP 트랩 통지가 전송되어야 할 호스트를 지정하려면 다음 행을 **/etc/snmp/snmpd.conf** 파일에 추가합니다:

   ```
   trap2sink host
   ```

   통지 처리에 대한 자세한 내용은 **snmpd.conf** man 페이지를 참조하십시오.

3. 다음 명령을 실행하여 **snmpd** 데몬이 활성화되어 실행되고 있는지 확인합니다:

   ```
   # chkconfig snmpd on
   # service snmpd start
   ```

4. **messagebus** 데몬이 아직 활성화되지 않아 실행되고 있지 않으면 다음과 같은 명령을 실행합니다:

   ```
   # chkconfig messagebus on
   # service messagebus start
   ```

5. 다음 명령을 실행하여 **foghorn** 데몬이 활성화되어 실행되고 있는지 확인합니다:

   ```
   # chkconfig foghorn on
   # service foghorn start
   ```

6. **COROSYNC-MIB**가 SNMP 트랩을 생성하고 **corosync-notifyd** 데몬이 활성화되어 실행되고 있는지를 확인하기 위해 다음 명령을 실행하여 시스템을 설정합니다:

```
# echo "OPTIONS=\"-d\" " > /etc/sysconfig/corosync-notifyd
# chkconfig corosync-notifyd on
# service corosync-notifyd start
```

SNMP를 위해 클러스터에 있는 각 노드를 설정하고 필요한 서비스가 실행되고 있는지 확인한 후, D-bus 신호가 **foghorn** 서비스에 의해 전송되어 SNMPv2 트랩으로 해석됩니다. 그 다음으로 이러한 트랩은 SNMPv2 트랩을 수신하기 위해 **trapsink** 항목으로 정의된 호스트로 전달됩니다.

10.3. SNMP 트랩 전송

SNMP 트랩을 클러스터의 일부가 아닌 컴퓨터에 전송할 수 있습니다. 외부 컴퓨터에서 **snmptrapd** 데몬을 사용하여 통지에 응답하는 방법을 사용자 정의할 수 있습니다.

다음 단계를 수행하여 클러스터에 있는 SNMP 트랩을 클러스터 노드중 하나가 아닌 시스템에 전송할 수 있습니다:

1. 클러스터에 있는 각 노드에 대해 10.2절. “Red Hat 고가용성 추가 기능을 사용하여 SNMP 설정”에서 설명하고 있는 단계를 따릅니다. **/etc/snmp/snmpd.conf** 파일에 있는 **trap2sink *host*** 항목을 설정하고 **snmptrapd** 데몬을 실행하게 될 외부 호스트를 지정합니다.

2. 트랩을 수신하는 외부 호스트에서 **/etc/snmp/snmptrapd.conf** 설정 파일을 편집하여 커뮤니티 문자열을 지정합니다. 예를 들어, 다음 항목을 사용하여 **snmptrapd** 데몬이 **public** 커뮤니티 문자열을 사용하여 통지를 처리할 수 있게 합니다.

```
authCommunity log,execute,net public
```

3. 트랩을 수신하는 외부 호스트에서 다음 명령을 실행하여 **snmptrapd** 데몬이 활성화되어 실행되고 있는지를 확인합니다:

```
# chkconfig snmptrapd on
# service snmptrapd start
```

SNMP 통지 처리에 관한 자세한 내용은 **snmptrapd.conf** man 페이지를 참조하십시오.

10.4. RED HAT 고가용성 추가 기능에 의해 만들어진 SNMP 트랩

foghorn 데몬은 다음과 같은 트랩을 생성합니다:

- **fenceNotifyFenceNode**

  이 트랩은 차단된 노드가 다른 노드를 차단 시도할 때 마다 발생합니다. 이 트랩은 하나의 노드 (차단 작업을 시도한 노드)에서만 생성되는 점에 유의하십시오. 통지에는 다음과 같은 항목이 포함됩니다:

  - **fenceNodeName** - 차단된 노드의 이름
  - **fenceNodeID** - 차단된 노드의 노드 id
  - **fenceResult** - 차단 작업 결과 (0은 성공, -1은 문제 발생, -2는 정의된 펜싱 방식이 없음)

- **rgmanagerServiceStateChange**

  이 트랩은 클러스터 서비스의 상태가 변경될 때 발생합니다. 통지에는 다음과 같은 항목이 포함됩니다:

  - **rgmanagerServiceName** - 서비스 이름, 서비스 유형 (예: **service:foo** 또는 **vm:foo**) 포함.
  - **rgmanagerServiceState** - 서비스 상태. 이는 트랩의 혼란을 줄이기 위해 **starting** 및 **stopping**과 같은 과도 상태는 제외합니다.
  - **rgmanagerServiceFlags** - 서비스 플래그. 현재 지원되는 플래그는 두 가지입니다: **frozen**은 **clusvcadm -Z**를 사용하여 동결되는 서비스입니다. **partial**은 실패한 리소스가 **non-critical**로 플래그되어 리소스가 실패해도 구성 요소는 전체 서비스에 영향을 미치지 않고 수동으로 다시 시작할 수 있습니다.
  - **rgmanagerServiceCurrentOwner** - 서비스 소유자. 서비스가 실행하지 않는 경우, 이는 **(none)**이 됩니다.
  - **rgmanagerServicePreviousOwner** - 마지막 서비스 소유자. 마지막 서비스 소유자를 알 수 없는 경우, 이는 **(none)**으로 나타납니다.

corosync-nodifyd 데몬은 다음과 같은 트랩을 생성합니다:

- **corosyncNoticesNodeStatus**

  이 트랩은 노드가 클러스터에 참여하거나 탈퇴할 때 발생합니다. 통지에는 다음과 같은 항목이 포함됩니다:

  - **corosyncObjectsNodeName** - 노드 이름
  - **corosyncObjectsNodeID** - 노드 id
  - **corosyncObjectsNodeAddress** - 노드 IP 주소
  - **corosyncObjectsNodeStatus** - 노드 상태 (**joined** 또는 **left**)

- **corosyncNoticesQuorumStatus**

  이 트랩은 쿼럼 상태가 변경되었을 때 발생합니다. 이 통지에는 다음과 같은 항목이 포함됩니다:

  - **corosyncObjectsNodeName** - 노드 이름
  - **corosyncObjectsNodeID** - 노드 id
  - **corosyncObjectsQuorumStatus** - 쿼럼의 새로운 상태 (**quorate** 또는 **NOT quorate**)

- **corosyncNoticesAppStatus**

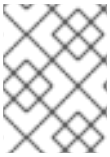
  이 트랩은 클라언트 어플리케이션은 Corosync로 부터 연결 또는 분리되었을 때 발생합니다.

  - **corosyncObjectsNodeName** - 노드 이름
  - **corosyncObjectsNodeID** - 노드 id
  - **corosyncObjectsAppName** - 어플리케이션 이름

- **corosyncObjectsAppStatus** - 어플리케이션의 새로운 상태 (**connected** 또는 **disconnected**)

11장. 클러스터 SAMBA 설정

Red Hat Enterprise Linux 6.2 릴리즈에서 Red Hat 고가용성 애드온은 클러스터 Samba를 active/active 설정을 실행하는 것을 지원합니다. 이를 위해 클러스터의 모든 모드에 CTDB를 설치 및 설정해야 하고 GFS2 클러스터 파일 시스템과 함께 사용합니다.



참고

Red Hat Enterprise Linux 6는 클러스터 Samba를 실행하는 노드를 최대 4개 까지 지원합니다.

다음 부분에서는 예시 시스템을 설정하여 CTDB를 설정하는 방법을 설명합니다. GFS2 파일 시스템 설정에 대한 자세한 내용은 *Global File System 2*에서 참조하십시오. 논리 볼륨을 설정하는 방법은 *LVM (Logical Volume Manager) 관리*에서 참조하십시오.

11.1. CTDB 개요

CTDB는 Samba가 사용하는 TDB 데이터베이스의 클러스터 구현입니다. CTDB를 사용하려면 클러스터 파일 시스템이 사용 가능하고 클러스터에 있는 모든 노드에서 공유되어야 합니다. CTDB는 클러스터 파일 시스템의 최상위에서 클러스터 기능을 제공합니다. Red Hat Enterprise Linux 6.2 릴리즈에서 CTDB는 클러스터 스택을 Red Hat Enterprise Linux 클러스터링이 제공하는 클러스터 스택과 함께 실행합니다. CTDB는 노드 멤버십, 복구/장애 조치, IP 재배치, Samba 서비스를 관리합니다.

11.2. 필요한 패키지

Red Hat 고가용성 애드온 및 Red Hat 장애 복구형 스토리지 애드온을 실행하는데 필요한 표준 패키지 뿐만 아니라 Red Hat Enterprise Linux 클러스터링과 함께 Samba를 실행하려면 다음과 같은 패키지가 필요합니다:

- **`ctdb`**
- **`samba`**
- **`samba-common`**
- **`samba-winbind-clients`**

11.3. GFS2 설정

Red Hat Enterprise Linux 클러스터링으로 Samba를 설정하려면 두 가지의 GFS 파일 시스템이 필요합니다. CTDB 용의 작은 파일 시스템 하나와 Samba 공유를 위한 파일 시스템입니다. 다음 예제에서는 이러한 두 가지 GFS2 파일 시스템을 생성하는 방법을 보여줍니다.

GFS2 파일 시스템을 생성하기 전 각 파일 시스템에 LVM 논리 볼륨을 먼저 생성합니다. LVM 논리 볼륨을 생성하는 방법은 *LAM (Logical Volume Manager) 관리*에서 참조하십시오. 이 예제에서는 다음과 같은 논리 볼륨을 사용합니다:

- **`/dev/csmb_vg/csmb_lv`**는 Samba 공유를 통해 내보낸 사용자 데이터를 보관하고 맞는 크기로 조정합니다. 이 예제에서는 100GB 크기의 논리 볼륨을 생성합니다.
- **`/dev/csmb_vg/ctdb_lv`**는 공유 CTDB 상태 정보를 저장하고 1 GB 크기가 필요합니다.

하나의 클러스터 노드에만 클러스터된 볼륨 그룹 및 논리 볼륨을 생성합니다.

논리 볼륨에 GFS2 파일 시스템을 생성하려면 **mkfs.gfs2** 명령을 실행합니다. 하나의 클러스터 노드에서만 이 명령을 실행합니다.

/dev/csmb_vg/csmb_lv 논리 볼륨에 Samba 공유를 호스팅하는 파일 시스템을 생성하려면 다음 명령을 실행합니다:

```
[root@clusmb-01 ~]# mkfs.gfs2 -j3 -p lock_dlm -t csmb:gfs2
/dev/csmb_vg/csmb_lv
```

매개 변수의 의미는 다음과 같습니다:

-j

파일 시스템에 생성할 저널 수를 지정합니다. 이 예에서는 3 개의 노드로 구성된 클러스터를 사용하므로 노드 당 하나의 저널을 만듭니다.

-p

잠금 프로토콜을 지정합니다 **lock_dlm**은 GFS2가 노드간 통신에 사용하는 잠금 프로토콜입니다.

-t

잠금 테이블 이름을 *cluster_name:fs_name* 형식으로 지정합니다. 이 예제에서 **cluster.conf** 파일에 지정된 클러스터 이름은 **csmb**이며 파일 시스템 이름으로 **gfs2**를 사용합니다.

이 명령의 출력 결과는 다음과 같이 나타납니다:

```
This will destroy any data on /dev/csmb_vg/csmb_lv.
  It appears to contain a gfs2 filesystem.

Are you sure you want to proceed? [y/n] y

Device:
/dev/csmb_vg/csmb_lv
Blocksize:  4096
Device Size  100.00 GB (26214400 blocks)
Filesystem Size: 100.00 GB (26214398 blocks)
Journals:  3
Resource Groups:  400
Locking Protocol:   "lock_dlm"
Lock Table:   "csmb:gfs2"
UUID:
  94297529-ABG3-7285-4B19-182F4F2DF2D7
```

이 예제에서 **/dev/csmb_vg/csmb_lv** 파일 시스템은 모든 노드에 있는 **/mnt/gfs2**에 마운트됩니다. 이 마운트 지점은 **/etc/samba/smb.conf** 파일에 있는 **path =** 옵션과 함께 **share** 디렉토리 위치로 지정한 값과 일치해야 합니다. 자세한 내용은 11.5절. “Samba 설정 ”에 설명되어 있습니다.

/dev/csmb_vg/ctdb_lv 논리 볼륨에 CTDB 상태 정보를 호스팅하기 위해 파일 시스템을 생성하려면 다음 명령을 실행합니다:

```
[root@clusmb-01 ~]# mkfs.gfs2 -j3 -p lock_dlm -t csmb:ctdb_state
/dev/csmb_vg/ctdb_lv
```

이 명령은 **/dev/csmb_vg/csmb_lv**에 파일 시스템을 생성한 경우에 잠금 테이블과 다른 잠금 테이블 이름을 지정하는 것에 유의합니다. 이는 파일 시스템에 사용된 다른 장치의 잠금 테이블 이름과 구분됩니다.

mkfs.gfs2의 출력 결과는 다음과 같이 나타납니다:

```
This will destroy any data on /dev/csmb_vg/ctdb_lv.
  It appears to contain a gfs2 filesystem.

Are you sure you want to proceed? [y/n] y

Device:
/dev/csmb_vg/ctdb_lv
Blocksize:    4096
Device Size   1.00 GB (262144 blocks)
Filesystem Size:  1.00 GB (262142 blocks)
Journals:  3
Resource Groups:  4
Locking Protocol:  "lock_dlm"
Lock Table:   "csmb:ctdb_state"
UUID:
  BCDA8025-CAF3-85BB-B062-CC0AB8849A03
```

예에서 **/dev/csmb_vg/ctdb_lv** 파일 시스템은 모든 노드에 있는 **/mnt/ctdb**에 마운트됩니다. 이러한 마운트 지점은 **/etc/sysconfig/ctdb** 파일에 있는 **CTDB_RECOVERY_LOCK** 옵션을 갖는 **.ctdb.lock** 파일의 위치로 지정된 값과 일치해야 합니다. 이는 11.4절. “CTDB 설정”에 설명되어 있습니다.

11.4. CTDB 설정

CTDB 설정 파일은 **/etc/sysconfig/ctdb**에 있습니다. CTDB가 작동하도록 설정해야 하는 필수 영역은 다음과 같습니다:

- **CTDB_NODES**
- **CTDB_PUBLIC_ADDRESSES**
- **CTDB_RECOVERY_LOCK**
- **CTDB_MANAGES_SAMBA** (활성화해야 함)
- **CTDB_MANAGES_WINBIND** (구성원 서버에서 실행하는 경우 활성화해야 함)

다음 예제에서는 예제 매개 변수와 설정한 CTDB 동작의 필수 필수의 설정 파일을 보여주고 있습니다:

```
CTDB_NODES=/etc/ctdb/nodes
CTDB_PUBLIC_ADDRESSES=/etc/ctdb/public_addresses
CTDB_RECOVERY_LOCK="/mnt/ctdb/.ctdb.lock"
CTDB_MANAGES_SAMBA=yes
CTDB_MANAGES_WINBIND=yes
```

이 매개 변수의 의미는 다음과 같습니다.

CTDB_NODES

클러스터 노드 목록을 포함하는 파일의 위치를 지정합니다.

CTDB_NODES가 참조하는 **/etc/ctdb/nodes** 파일은 다음 예제와 같이 클러스터 노드의 IP 주소를 나열합니다:

```
192.168.1.151
192.168.1.152
192.168.1.153
```

이 예제에서 클러스터/CTDB 통신 및 클라이언트 서비스 모두에 사용되는 각 노드에는 하나의 인터페이스/IP만이 있습니다. 하지만 각 클러스터 노드는 두 개의 네트워크 인터페이스를 보유할 것을 강력히 권장합니다. 이렇게 하면 인터페이스 1 세트는 클러스터/CTDB 통신 전용으로 다른 인터페이스 세트는 공용 클라이언트 액세스 전용으로 사용할 수 있습니다. 클러스터 네트워크에 적절한 IP 주소를 사용하여 **cluster.conf**에 사용되는 호스트 이름/IP 주소가 동일한지 확인하십시오. 마찬가지로 **public_addresses** 파일에 있는 클라이언트 액세스의 공용 네트워크에 적합한 인터페이스를 사용하도록 하십시오.

/etc/ctdb/nodes 파일이 모든 노드에서 완전히 동일하다는 것은 중요합니다. 서로 다른 노드에서 다른 정보를 찾으면 CTDB는 실패하기 때문에 순서가 중요한 것입니다.

CTDB_PUBLIC_ADDRESSES

클러스터에서 내보낸 Samba 공유에 액세스하는데 사용할 수 있는 IP 주소를 나열하는 파일의 위치를 지정합니다. 이는 클러스터 Samba 서버 이름의 DNS에 설정해야 하는 IP 주소이며 CIFS 클라이언트가 연결되는 주소입니다. 클러스터 Samba 서버의 이름을 여러 IP 주소를 갖는 DNS A 레코드 유형으로 설정하고 라운드 로빈 DNS가 클러스터 노드에 걸쳐 클라이언트를 배포하게 합니다.

이 예제에서는 **/etc/ctdb/public_addresses** 파일에 나열된 모든 주소와 함께 라운드 로빈 DNS 항목 **csmb-server**을 설정했습니다. DNS는 클러스터 전체에 라운드 로빈으로 이 항목을 사용하는 클라이언트를 배포하게 됩니다.

각 노드에 있는 **/etc/ctdb/public_addresses** 파일의 내용은 다음과 같습니다:

```
192.168.1.201/0 eth0
192.168.1.202/0 eth0
192.168.1.203/0 eth0
```

이 예제에서는 현재 네트워크에서 사용되지 않는 세 개의 주소를 사용합니다. 실제 설정에서는 대상 클라이언트가 액세스할 수 있는 주소를 선택하십시오.

다른 방법으로 이 예제에서는 총 4 개의 공용 주소를 제외한 3 개의 노드가 있는 클러스터의 **/etc/ctdb/public_addresses** 파일 내용을 보여주고 있습니다. 예에서 IP 주소 198.162.2.1은 노드 0 또는 노드 1 중 하나로 호스트될 수 있으며 이러한 노드 중 최소 하나가 사용 가능한 한 클라이언트가 이 주소를 사용할 수 있습니다. 노드 0과 노드 1 모두에 문제가 발생하는 경우에만 클라이언트가 이 공용 주소를 사용할 수 없게 됩니다. 기타 다른 모든 공용 주소는 각각 하나의 노드에서만 작동할 수 있으므로 각각의 상응하는 노드가 사용 가능한 경우에만 사용할 수 있게 됩니다.

노드 0에 있는 **/etc/ctdb/public_addresses** 파일에는 다음과 같은 내용이 들어 있습니다:

```
198.162.1.1/24 eth0
198.162.2.1/24 eth1
```

노드 1에 있는 **/etc/ctdb/public_addresses** 파일에는 다음과 같은 내용이 들어 있습니다:

```
198.162.2.1/24 eth1
198.162.3.1/24 eth2
```

노드 2에 있는 **/etc/ctdb/public_addresses** 파일에는 다음과 같은 내용이 들어 있습니다:

```
198.162.3.2/24 eth2
```

CTDB_RECOVERY_LOCK

CTDB가 내부에서 복구에 사용할 잠금 파일을 지정합니다. 이 파일은 모든 클러스터 노드에서 액세스할 수 있도록 공용 저장소에 두어야 합니다. 예제에서는 모든 노드에 있는 **/mnt/ctdb**에 마운트된 GFS2 파일 시스템을 사용합니다. 이는 내보내기 되는 Samba 공유를 호스팅하는 GFS2 파일 시스템과 다릅니다. 이 복구 잠금 파일은 분할 브레인 시나리오를 막기 위해 사용됩니다. 새 CTDB 버전 (1.0.112 이상)에서는 이것이 다른 분할 브레인 방지 메카니즘으로 대체되는 한 이 파일을 지정하는 것은 옵션입니다.

CTDB_MANAGES_SAMBA

이를 **yes**로 설정하여 활성화하는 경우 서비스 마이그레이션/장애 조치를 제공하기 위해 필요하다고 생각되므로 CTDB가 Samba 서비스를 시작 및 중지할 수 있게 지정합니다.

CTDB_MANAGES_SAMBA가 활성화되면 다음 명령을 실행하여 **smb** 및 **nmb** 데몬의 **init** 자동 시작을 비활성화해야 합니다:

```
[root@clusmb-01 ~]# chkconfig snb off
[root@clusmb-01 ~]# chkconfig nmb off
```

CTDB_MANAGES_WINBIND

이를 **yes**로 설정하여 활성화하는 경우 필요에 따라 CTDB가 **winbind** 데몬을 시작/중지할 수 있도록 지정합니다. 이는 Windows 도메인이나 활성 디렉토리 보안 모드에서 CTDB를 사용하는 경우에 활성화해야 합니다.

CTDB_MANAGES_WINBIND가 활성화되어 있는 경우, 다음 명령을 실행하여 **winbind** 데몬의 **init** 자동 시작을 비활성화해야 합니다:

```
[root@clusmb-01 ~]# chkconfig windinbd off
```

11.5. SAMBA 설정

이 예제에서 Samba 설정 파일 **smb.conf**은 **/etc/samba/smb.conf**에 있습니다. 이 파일에는 다음과 같은 매개 변수가 들어 있습니다:

```
[global]
 guest ok = yes
 clustering = yes
 netbios name = csmb-server
[csmb]
 comment = Clustered Samba
  public = yes
 path = /mnt/gfs2/share
 writeable = yes
 ea support = yes
```

이 예에서는 **/mnt/gfs2/share**에 있는 **csmb** 이름으로된 공유를 내보내기 합니다. 이는 **/etc/sysconfig/ctdb**의 CTDB 설정 파일에서 **CTDB_RECOVERY_LOCK** 매개 변수로 지정된 **/mnt/ctdb/.ctdb.lock**의 GFS2 공유 파일 시스템과 다릅니다.

예에서 처음으로 이를 마운트하면 **/mnt/gfs2**에 **share** 디렉토리가 생성됩니다. **clustering = yes** 항목은 Samba가 CTDB를 사용하도록 지시합니다. **netbios name = csmb-server** 항목은 모든 노드가 공통적인 NetBIOS 이름을 갖도록 명시적으로 설정합니다. **ea support** 매개 변수는 확장 속성을 사용하고자 할 경우 필요합니다.

smb.conf 설정 파일은 모든 클러스터 노드에서 동일해야 합니다.

Samba는 **net conf** 명령을 사용하여 레지스트리 기반 설정도 제공합니다. 이렇게 하면 클러스터 노드 간 설정 파일을 수동으로 복사하지 않고 클러스터 멤버 간에 설정으로 자동으로 동기화할 수 있게 합니다. **net conf** 명령에 대한 자세한 내용은 **net(8)** man 페이지를 참조하십시오.

11.6. CTDB 및 SAMBA 서비스 시작

클러스터를 시작한 후, 11.3절. “GFS2 설정 ”에서 설명하고 있듯이 생성한 GFS2 파일 시스템을 마운트해야 합니다. Samba **share** 디렉토리에 있는 권한과 클러스터 노드에 있는 사용자 계정은 클라이언트 액세스용으로 설정해야 합니다.

모든 노드에서 다음 명령을 실행하여 **ctdbd** 데몬을 시작합니다. 예제에서는 **CTDB_MANAGES_SAMBA=yes**로 CTDB가 설정되어 있기 때문에 CTDB는 모든 노드에서 Samba 서비스를 시작하여 구성된 모든 Samba 공유를 내보내기합니다.

```
[root@clusmb-01 ~]# service ctdb start
```

Samba를 시작하고, 공유를 내보내기하여 안정화하는데 CTDB는 몇 분 정도 소요될 수 있습니다. **ctdb status**를 실행하면 다음과 같은 CTDB 상태가 표시됩니다:

```
[root@clusmb-01 ~]# ctdb status
Number of nodes:3
pnn:0 192.168.1.151     OK (THIS NODE)
pnn:1 192.168.1.152     OK
pnn:2 192.168.1.153     OK
Generation:1410259202
Size:3
hash:0 lmaster:0
hash:1 lmaster:1
hash:2 lmaster:2
Recovery mode:NORMAL (0)
Recovery master:0
```

모든 노드가 "OK"로 확인되면, 11.7절. “클러스터 Samba 서버 사용 ”에서 설명하고 있듯이 클러스터 Samba 서버를 안전하게 사용할 수 있습니다.

11.7. 클러스터 SAMBA 서버 사용

클라이언트는 **/etc/ctdb/public_addresses** 파일에 지정된 IP 주소 중 하나에 연결하거나 다음과 같이 이전에 설정한 **csmb-server** DNS 항목을 사용하여 내보내기된 Samba 공유에 연결할 수 있습니다:

```
[root@clusmb-01 ~]# mount -t cifs //csmb-server/csmb /mnt/sambashare -o
user=testmonkey
```

또는

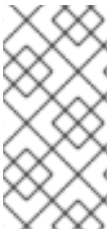
```
[user@clusmb-01 ~]$ smbclient //csmb-server/csmb
```

부록 A. 차단 장치 매개 변수

다음에는 차단 장치 매개 변수 설명이 있는 표가 제공됩니다. **ccs** 명령을 사용하거나 **etc/cluster/cluster.conf**를 편집하여 **luci**로 매개 변수를 설정할 수 있습니다. 각 차단 에이전트의 차단 장치 매개 변수의 전체 목록 및 설명은 각 에이전트의 **man** 페이지에서 참조하십시오.

참고

차단 장치의 **Name** 매개 변수는 Red Hat 고가용성 추가 기능에 의해 사용되는 장치의 임의 이름을 지정합니다. 이는 장치의 DNS 이름과 동일하지 않습니다.



참고

특정 차단 장치에는 **Password Script** 매개 변수 옵션이 있습니다. **Password Script** 매개 변수를 사용하면 차단 장치 암호가 **Password** 매개 변수가 아닌 스크립트에서 공급되도록 지정할 수 있습니다. **Password Script** 매개 변수는 **Password** 매개 변수를 대신하는 것으로 암호가 클러스터 설정 파일에서 (**/etc/cluster/cluster.conf**) 볼 수 없게 합니다.

표 A.1. “차단 장치 요약 ” 에는 차단 장치, 차단 장치와 관련된 차단 장치 에이전트가 나열되어 있고 차단 장치 매개 변수에 대한 문서화된 표에 대한 참조가 있습니다.

표 A.1. 차단 장치 요약

차단 장치	차단 에이전트	매개 변수 설명에 대한 참조
APC 전원 스위치 (telnet/SSH)	fence_apc	표 A.2. “APC 전원 스위치 (telnet/SSH)”
Brocade Fabric 스위치	fence_brocade	표 A.4. “Brocade Fabric 스위치 ”
Cisco MDS	fence_cisco_mds	표 A.5. “Cisco MDS”
Cisco UCS	fence_cisco_ucs	표 A.6. “Cisco UCS”
Dell DRAC 5	fence_drac5	표 A.7. “Dell DRAC 5”
Eaton Network Power Switch (SNMP 인터페이스)	fence_eaton_snmp	표 A.8. “Eaton 네트워크 파워 컨트롤러 (SNMP 인터페이스) (Red Hat Enterprise Linux 6.4 이상)”
Egenera SAN 제어기	fence_egenera	표 A.9. “Egenera SAN 제어기 ”
ePowerSwitch	fence_eps	표 A.10. “ePowerSwitch”
Fence virt	fence_virt	표 A.11. “Fence virt”

차단 장치	차단 에이전트	매개 변수 설명에 대한 참조
Fujitsu Siemens Remoteview Service Board (RSB)	fence_rsb	표 A.12. “Fujitsu Siemens Remoteview Service Board (RSB)”
HP BladeSystem	fence_hpblade	표 A.13. “HP BladeSystem (Red Hat Enterprise Linux 6.4 이상)”
HP iLO/iLO2 (Integrated Lights Out)	fence_ilo	표 A.14. “HP iLO/iLO2 (Integrated Lights Out)”
HP iLO (Integrated Lights Out) MP	fence_ilo_mp	표 A.15. “HP iLO (Integrated Lights Out) MP”
IBM BladeCenter	fence_bladecenter	표 A.16. “IBM BladeCenter”
IBM BladeCenter SNMP	fence_ibmblade	표 A.17. “IBM BladeCenter SNMP”
IBM iPDU	fence_ipdu	표 A.18. “IBM iPDU (Red Hat Enterprise Linux 6.4 이상)”
IF MIB	fence_ifmib	표 A.19. “IF MIB”
Intel Modular	fence_intelmodular	표 A.20. “Intel Modular”
IPMI (Intelligent Platform Management Interface) LAN	fence_ipmilan	표 A.21. “IPMI (Intelligent Platform Management Interface) LAN”
RHEV-M REST API	fence_rhevm	표 A.22. “RHEV-M REST API (RHEV 3.0 이상에 대해 RHEL 6.2 이상)”
SCSI Fencing	fence_scsi	표 A.23. “SCSI Fencing”
VMware 차단 장치 (SOAP 인터페이스)	fence_vmware_soap	표 A.24. “VMware 차단 장치 (SOAP 인터페이스) (Red Hat Enterprise Linux 6.2 이상)”

차단 장치	차단 에이전트	매개 변수 설명에 대한 참조
WTI 전원 스위치	fence_wti	표 A.25. “WTI 전원 스위치 ”

표 A.2. “APC 전원 스위치 (telnet/SSH)” 는 telnet/SSH를 통한 APC의 차단 에이전트인 **fence_apc**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.2. APC 전원 스위치 (telnet/SSH)

luci 필드	cluster.conf 속성	설명
이름	**name**	차단 장치 데몬이 telnet/ssh를 통해 기록하는 클러스터에 연결된 APC 장치의 이름
IP 주소 또는 호스트 이름	**ipaddr**	장치에 할당된 IP 주소 또는 호스트 이름
IP 포트 (옵션)	**ipport**	장치에 연결하기 위해 사용하는 TCP 포트
로그인	**login**	장치에 액세스하기 위해 사용하는 로그인 이름
암호	**passwd**	장치에 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	**passwd_script**	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
전원 대기 시간	**power_wait**	power off 또는 power on 명령 실행 후 대기 시간 (초)
포트	**port**	TCP 포트
스위치 (옵션)	**switch**	여러 데이지 체인 (daisy-chained) 스위치를 사용할 때 노드로 연결하는 APC 스위치의 스위치 번호
SSH 사용	**secure**	장치에 액세스하기 위해 SSH를 사용하는 시스템을 나타냅니다.
SSH 식별 파일로의 경로	**identity_file**	SSH 용 식별 파일

표 A.3. “SNMP를 통한 APC 전원 스위치 ” 에서는 SNMP 프로토콜을 통해 SNP 장치로 로그인할 APC의 차단 에이전트인 **fence_apc_snmp**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.3. SNMP를 통한 APC 전원 스위치

luci 필드	cluster.conf 속성	설명
이름	**name**	차단 장치 데몬이 SNMP 프로토콜을 통해 기록하는 클러스터에 연결된 APC 장치의 이름
IP 주소 또는 호스트 이름	**ipaddr**	장치에 할당된 IP 주소 또는 호스트 이름
UDP/TCP 포트	**udpport**	장치에 연결하는 데 사용할 UDP/TCP 포트. 기본값은 161입니다.
로그인	**login**	장치에 액세스하기 위해 사용하는 로그인 이름
암호	**passwd**	장치에 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	**passwd_script**	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
SNMP 버전	**snmp_version**	사용할 SNMP 버전 (1, 2c, 3). 기본값은 1 입니다.
SNMP 커뮤니티	**community**	SNMP 커뮤니티 문자열. 기본값은 **private**입니다.
SNMP 보안 수준	**snmp_sec_level**	SNMP 보안 수준 (noAuthNoPriv, authNoPriv, authPriv).
SNMP 인증 프로토콜	**snmp_auth_prot**	SNMP 인증 프로토콜 (MD5, SHA).
SNMP 개인 정보 프로토콜	**snmp_priv_prot**	SNMP 개인 정보 프로토콜 (DES, AES).
SNMP 개인 정보 프로토콜 암호	**snmp_priv_passwd**	SNMP 개인 정보 프로토콜 암호
SNMP 개인 정보 프로토콜 스크립트	**snmp_priv_passwd_script**	SNMP 개인 정보 프로토콜에 대한 암호를 제공하는 스크립트입니다. 이를 사용하는 것은 **SNMP 개인 정보 프로토콜 암호** 매개 변수를 대신합니다.
전원 대기 시간	**power_wait**	power off 또는 power on 명령 실행 후 대기 시간 (초)
포트 (콘센트) 번호	**port**	TCP 포트

표 A.4. “Brocade Fabric 스위치 ” 에서는 Brocade FC 스위치의 차단 에이전트인 **fence_brocade**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.4. Brocade Fabric 스위치

luci 필드	cluster.conf 속성	설명
이름	**name**	클러스터에 연결된 Brocade 장치의 이름
IP 주소 또는 호스트 이름	**ipaddr**	장치에 할당된 IP 주소
로그인	**login**	장치에 액세스하기 위해 사용하는 로그인 이름
암호	**passwd**	장치에 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	**passwd_script**	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
포트	**port**	스위치 콘센트 번호

표 A.5. “Cisco MDS”에서는 Cisco MDS의 차단 에이전트인 **fence_cisco_mds**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.5. Cisco MDS

luci 필드	cluster.conf 속성	설명
이름	**name**	SNMP가 활성화되어 있는 Cisco MDS 9000 시리즈 장치 이름
IP 주소 또는 호스트 이름	**ipaddr**	장치에 할당된 IP 주소 또는 호스트 이름
UDP/TCP 포트	**udpport**	장치에 연결하는 데 사용할 UDP/TCP 포트. 기본값은 161입니다.
로그인	**login**	장치에 액세스하기 위해 사용하는 로그인 이름
암호	**passwd**	장치에 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	**passwd_script**	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
포트 (콘센트) 번호	**port**	TCP 포트
SNMP 버전	**snmp_version**	사용할 SNMP 버전 (1, 2c, 3).
SNMP 커뮤니티	**community**	SNMP 커뮤니티 문자열

luci 필드	cluster.conf 속성	설명
SNMP 보안 수준	snmp_sec_level	SNMP 보안 수준 (noAuthNoPriv, authNoPriv, authPriv).
SNMP 인증 프로토콜	snmp_auth_prot	SNMP 인증 프로토콜 (MD5, SHA).
SNMP 개인 정보 프로토콜	snmp_priv_prot	SNMP 개인 정보 프로토콜 (DES, AES).
SNMP 개인 정보 프로토콜 암호	snmp_priv_passwd	SNMP 개인 정보 프로토콜 암호
SNMP 개인 정보 프로토콜 스크립트	snmp_priv_passwd_script	SNMP 개인 정보 프로토콜에 대한 암호를 제공하는 스크립트입니다. 이를 사용하는 것은 **SNMP 개인 정보 프로토콜 암호** 매개 변수를 대신합니다.
전원 대기 시간	power_wait	power off 또는 power on 명령 실행 후 대기 시간 (초)

표 A.6. “Cisco UCS”에서는 Cisco UCS의 차단 장치인 **fence_cisco_ucs**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.6. Cisco UCS

luci 필드	cluster.conf 속성	설명
이름	name	Cisco UCS 장치 이름
IP 주소 또는 호스트 이름	ipaddr	장치에 할당된 IP 주소 또는 호스트 이름
IP 포트 (옵션)	ipport	장치에 연결하기 위해 사용하는 TCP 포트
로그인	login	장치에 액세스하기 위해 사용하는 로그인 이름
암호	passwd	장치에 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	passwd_script	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
SSL 사용	ssl	장치와 통신하기 위해 SSL 연결을 사용합니다.
하위 조직	suborg	하위 조직에 액세스하기 위해 필요한 추가 경로입니다.

luci 필드	cluster.conf 속성	설명
포트 (콘센트) 번호	port	가상 머신 이름입니다.
전원 대기 시간	power_wait	power off 또는 power on 명령 실행 후 대기 시간 (초)

표 A.7. “Dell DRAC 5” 에서는 Dell DRAC 5 용 차단 장치인 fence_drac5에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.7. Dell DRAC 5

luci 필드	cluster.conf 속성	설명
이름	name	DRAC에 할당된 이름입니다.
IP 주소 또는 호스트 이름	ipaddr	DRAC에 할당된 IP 주소 또는 호스트 이름입니다.
IP 포트 (옵션)	ipport	장치에 연결하기 위해 사용하는 TCP 포트
로그인	login	DRAC에 액세스하기 위해 사용하는 로그인 이름입니다.
암호	passwd	DRAC으로의 연결을 인증하기 위해 사용되는 암호입니다.
암호 스크립트 (옵션)	passwd_script	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 Password 매개 변수를 대신합니다.
SSH 사용	secure	장치에 액세스하기 위해 SSH를 사용하는 시스템을 보여줍니다.
SSH 식별 파일로의 경로	identity_file	SSH 용 식별 파일
모듈 이름	module_name	여러 DRAC 모듈이 있을 때 DRAC의 모듈 이름 (옵션)입니다.
명령 프롬프트를 강제	cmd_prompt	사용할 명령 프롬프트입니다. 디폴트값은 ’\$’입니다.
전원 대기 시간	power_wait	power off 또는 power on 명령 실행 후 대기 시간 (초)

표 A.8. “Eaton 네트워크 파워 컨트롤러 (SNMP 인터페이스) (Red Hat Enterprise Linux 6.4 이상)” 에서는 SNMP 네트워크 전원 스위치를 통해 Eaton 용 차단 장치 fence_eaton_snmp에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.8. Eaton 네트워크 파워 컨트롤러 (SNMP 인터페이스) (Red Hat Enterprise Linux 6.4 이상)

luci 필드	cluster.conf 속성	설명
이름	**name**	클러스터에 연결된 Eaton 네트워크 파워 스위치의 이름입니다.
IP 주소 또는 호스트 이름	**ipaddr**	장치에 할당된 IP 주소 또는 호스트 이름
UDP/TCP 포트 (옵션)	**udpport**	장치에 연결하는 데 사용할 UDP/TCP 포트. 기본값은 161입니다.
로그인	**login**	장치에 액세스하기 위해 사용하는 로그인 이름
암호	**passwd**	장치에 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	**passwd_script**	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
SNMP 버전	**snmp_version**	사용할 SNMP 버전 (1, 2c, 3). 기본값은 1 입니다.
SNMP 커뮤니티	**community**	SNMP 커뮤니티 문자열. 기본값은 **private**입니다.
SNMP 보안 수준	**snmp_sec_level**	SNMP 보안 수준 (noAuthNoPriv, authNoPriv, authPriv).
SNMP 인증 프로토콜	**snmp_auth_prot**	SNMP 인증 프로토콜 (MD5, SHA).
SNMP 개인 정보 프로토콜	**snmp_priv_prot**	SNMP 개인 정보 프로토콜 (DES, AES).
SNMP 개인 정보 프로토콜 암호	**snmp_priv_passwd**	SNMP 개인 정보 프로토콜 암호
SNMP 개인 정보 프로토콜 스크립트	**snmp_priv_passwd_script**	SNMP 개인 정보 프로토콜에 대한 암호를 제공하는 스크립트입니다. 이를 사용하는 것은 **SNMP 개인 정보 프로토콜 암호** 매개 변수를 대신합니다.
전원 대기 시간 (초)	**power_wait**	power off 또는 power on 명령 실행 후 대기 시간 (초)
포트 (콘센트) 번호	**port**	물리적 플러그 번호 또는 가상 머신 이름입니다. 이 매개 변수는 항상 필요합니다.

표 A.9. “Egenera SAN 제어기 ” 에서는 Egenera BladeFrame의 차단 장치인 **fence_egenera**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.9. Egenera SAN 제어기

luci 필드	cluster.conf 속성	설명
이름	**name**	클러스터에 연결되는 Egenera BladeFrame 장치의 이름
CServer	**cserver**	장치에 할당된 호스트 이름 (옵션으로 **username@hostname** 형식의 사용자 이름).자세한 정보는 fence_egenera(8) man 페이지를 참조하십시오.
ESH 경로 (옵션)	**esh**	cserver에서 esh 명령으로의 경로 (디폴트 값: /opt/panmgr/bin/esh)
사용자 이름	**user**	로그인 이름입니다. 디폴트 값은 **root**입니다.
lpan	**lpan**	장치의 논리적 프로세스 영역 네트워크 (LPAN)입니다.
pserver	**pserver**	장치의 프로세싱 블레이드 (pserver)의 이름입니다.

표 A.10. “ePowerSwitch”에서는 ePowerSwitch의 차단 에이전트인 **fence_eps**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.10. ePowerSwitch

luci 필드	cluster.conf 속성	설명
이름	**name**	클러스터에 연결되는 ePowerSwitch 장치의 이름입니다.
IP 주소 또는 호스트 이름	**ipaddr**	장치에 할당된 IP 주소 또는 호스트 이름
로그인	**login**	장치에 액세스하기 위해 사용하는 로그인 이름
암호	**passwd**	장치에 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	**passwd_script**	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
숨겨진 페이지의 이름	**hidden_page**	장치의 숨겨진 페이지 이름입니다.
포트 (콘센트) 번호	**port**	물리적 플러그 번호 또는 가상 머신 이름입니다.

표 A.11. “Fence virt”에서는 Fence virt 차단 장치의 차단 에이전트인 **fence_virt**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.11. Fence virt

luci 필드	cluster.conf 속성	설명
이름	**name**	Fence virt 차단 장치의 이름입니다.
직렬 장치	**serial_device**	호스트에서 직렬 장치는 각 도메인의 설정 파일에 맵핑되어야 합니다. 자세한 내용은 **fence_virt.conf** man 페이지를 참조하십시오. 이 필드가 지정되어 있는 경우, **fence_virt** 차단 장치 에이전트가 직렬 모드로 실행되는 요인이 됩니다. 값을 지정하지 않으면 **fence_virt** 차단 장치 에이전트가 VM 채널 모드로 실행되는 요인이 됩니다.
직렬 매개 변수	**serial_params**	직렬 매개 변수입니다. 디폴트 값은 115200, 8N1입니다.
VM 채널 IP 주소	**channel_address**	채널 IP입니다. 디폴트 값은 10.0.2.179입니다.
포트 또는 도메인 (사용 권장되지 않음)	**port**	차단할 가상 머신 (도메인 UUID 또는 이름)입니다.
	ipport	채널 포트입니다. 디폴트 값은 1229이며 이는 **luci**로 차단 장치를 설정할 때 사용되는 값입니다.

표 A.12. “Fujitsu Siemens Remoteview Service Board (RSB)” 에서는 Fujitsu-Siemens RSB의 차단 에이전트인 **fence_rsb**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.12. Fujitsu Siemens Remoteview Service Board (RSB)

luci 필드	cluster.conf 속성	설명
이름	**name**	차단 장치로 사용할 RSB의 이름입니다.
IP 주소 또는 호스트 이름	**ipaddr**	장치에 할당된 호스트 이름입니다.
로그인	**login**	장치에 액세스하기 위해 사용하는 로그인 이름
암호	**passwd**	장치에 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	**passwd_script**	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
TCP 포트	**ipport**	telnet 서비스가 수신하는 포트 번호입니다. 디폴트 값은 3172입니다.

표 A.13. “HP BladeSystem (Red Hat Enterprise Linux 6.4 이상)” 에서는 HP BladeSystem의 차단 에이전트인 **fence_hpblade**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.13. HP BladeSystem (Red Hat Enterprise Linux 6.4 이상)

luci 필드	cluster.conf 속성	설명
이름	**name**	클러스터에 연결된 HP Bladesystem 장치에 할당된 이름
IP 주소 또는 호스트 이름	**ipaddr**	HP BladeSystem 장치에 할당된 IP 주소 및 호스트이름입니다.
IP 포트 (옵션)	**ipport**	장치에 연결하기 위해 사용하는 TCP 포트
로그인	**login**	HP BladeSystem 장치에 액세스하기 위해 사용되는 로그인 이름입니다. 이 매개 변수는 필수 사항입니다.
암호	**passwd**	차단 장치로의 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	**passwd_script**	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
명령 프롬프트를 강제	**cmd_prompt**	사용할 명령 프롬프트입니다. 디폴트값은 '\$'입니다.
포트가 없는 경우 오류를 발생하는 대신 OFF를 반환	**missing_as_off**	포트가 없는 경우 오류를 발생하는 대신 OFF를 반환합니다.
전원 대기 시간 (초)	**power_wait**	power off 또는 power on 명령 실행 후 대기 시간 (초)
SSH 사용	**secure**	장치에 액세스하기 위해 SSH를 사용하는 시스템을 보여줍니다.
SSH 식별 파일로의 경로	**identity_file**	SSH 용 식별 파일

표 A.14. "HP iLO/iLO2 (Integrated Lights Out)" 에서는 HP iLO 장치의 차단 에이전트인 **fence_ilo**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.14. HP iLO/iLO2 (Integrated Lights Out)

luci 필드	cluster.conf 속성	설명
이름	**name**	HP iLO가 지원되는 서버의 이름입니다.
IP 주소 또는 호스트 이름	**ipaddr**	장치에 할당된 IP 주소 또는 호스트 이름

luci 필드	cluster.conf 속성	설명
IP 포트 (옵션)	ipport	장치에 연결하는 데 사용할 TCP 포트입니다.
로그인	login	장치에 액세스하기 위해 사용하는 로그인 이름
암호	passwd	장치에 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	passwd_script	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
전원 대기 시간	power_wait	power off 또는 power on 명령 실행 후 대기 시간 (초)

표 A.15. "HP iLO (Integrated Lights Out) MP" 에서는 HP iLO MP 장치의 차단 에이전트인 **fence_ilo_mp**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.15. HP iLO (Integrated Lights Out) MP

luci 필드	cluster.conf 속성	설명
이름	name	HP iLO가 지원되는 서버의 이름입니다.
IP 주소 또는 호스트 이름	ipaddr	장치에 할당된 IP 주소 또는 호스트 이름
IP 포트 (옵션)	ipport	장치에 연결하는 데 사용할 TCP 포트입니다.
로그인	login	장치에 액세스하기 위해 사용하는 로그인 이름
암호	passwd	장치에 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	passwd_script	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
SSH 사용	secure	장치에 액세스하기 위해 SSH를 사용하는 시스템을 보여줍니다.
SSH 식별 파일로의 경로	identity_file	SSH 용 식별 파일입니다.
명령 프롬프트를 강제	cmd_prompt	사용할 명령 프롬프트입니다. 디폴트 값은 'MP>', 'hpiLO->'입니다.
전원 대기 시간	power_wait	power off 또는 power on 명령 실행 후 대기 시간 (초)

표 A.16. "IBM BladeCenter"에서는 IBM BladeCenter의 차단 에이전트인 **fence_bladecenter**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.16. IBM BladeCenter

luci 필드	cluster.conf 속성	설명
이름	**name**	클러스터에 연결된 IBM BladeCenter 장치의 이름입니다.
IP 주소 또는 호스트 이름	**ipaddr**	장치에 할당된 IP 주소 또는 호스트 이름
IP 포트 (옵션)	**ipport**	장치에 연결하는 데 사용할 TCP 포트입니다.
로그인	**login**	장치에 액세스하기 위해 사용하는 로그인 이름
암호	**passwd**	장치에 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	**passwd_script**	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
전원 대기 시간	**power_wait**	power off 또는 power on 명령 실행 후 대기 시간 (초)
SSH 사용	**secure**	장치에 액세스하기 위해 SSH를 사용하는 시스템을 나타냅니다.
SSH 식별 파일로의 경로	**identity_file**	SSH 용 식별 파일

표 A.17. “IBM BladeCenter SNMP”에서는 SNMP를 통해 IBM BladeCenter의 차단 장치인 **fence_ibmblade**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.17. IBM BladeCenter SNMP

luci 필드	cluster.conf 속성	설명
이름	**name**	클러스터에 연결된 IBM BladeCenter SNMP 장치의 이름입니다.
IP 주소 또는 호스트 이름	**ipaddr**	장치에 할당된 IP 주소 또는 호스트 이름
UDP/TCP 포트 (옵션)	**udpport**	장치에 연결하는 데 사용하는 UDP/TCP 포트입니다. 디폴트 값은 161입니다.
로그인	**login**	장치에 액세스하기 위해 사용하는 로그인 이름
암호	**passwd**	장치에 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	**passwd_script**	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.

luci 필드	cluster.conf 속성	설명
SNMP 버전	snmp_version	사용할 SNMP 버전 (1, 2c, 3). 기본값은 1 입니다.
SNMP 커뮤니티	community	SNMP 커뮤니티 문자열
SNMP 보안 수준	snmp_sec_level	SNMP 보안 수준 (noAuthNoPriv, authNoPriv, authPriv).
SNMP 인증 프로토콜	snmp_auth_prot	SNMP 인증 프로토콜 (MD5, SHA).
SNMP 개인 정보 프로토콜	snmp_priv_prot	SNMP 개인 정보 프로토콜 (DES, AES).
SNMP 개인 정보 프로토콜 암호	snmp_priv_passwd	SNMP 개인 정보 프로토콜 암호입니다.
SNMP 개인 정보 프로토콜 스크립트	snmp_priv_passwd_script	SNMP 개인 정보 프로토콜에 대한 암호를 제공하는 스크립트입니다. 이를 사용하는 것은 **SNMP 개인 정보 프로토콜 암호** 매개 변수를 대신합니다.
전원 대기 시간	power_wait	power off 또는 power on 명령 실행 후 대기 시간 (초)
포트	port	물리적 플러그 번호 또는 가상 머신 이름입니다.

표 A.18. "IBM iPDU (Red Hat Enterprise Linux 6.4 이상)" 에서는 SNMP 장치를 통해 iPDU 용 차단 에이전트 **fence_ipdu**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.18. IBM iPDU (Red Hat Enterprise Linux 6.4 이상)

luci 필드	cluster.conf 속성	설명
이름	name	차단 장치 데몬이 SNMP 프로토콜을 통해 기록하는 클러스터에 연결된 IBM iPDU 장치의 이름
IP 주소 또는 호스트 이름	ipaddr	장치에 할당된 IP 주소 또는 호스트 이름
UDP/TCP 포트	udpport	장치에 연결하는 데 사용할 UDP/TCP 포트. 기본값은 161입니다.
로그인	login	장치에 액세스하기 위해 사용하는 로그인 이름
암호	passwd	장치에 연결을 인증하기 위해 사용되는 암호

luci 필드	cluster.conf 속성	설명
암호 스크립트 (옵션)	passwd_script	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
SNMP 버전	snmp_version	사용할 SNMP 버전 (1, 2c, 3). 기본값은 1 입니다.
SNMP 커뮤니티	community	SNMP 커뮤니티 문자열. 기본값은 private입니다.
SNMP 보안 수준	snmp_sec_level	SNMP 보안 수준 (noAuthNoPriv, authNoPriv, authPriv).
SNMP 인증 프로토콜	snmp_auth_prot	SNMP 인증 프로토콜 (MD5, SHA).
SNMP 개인 정보 프로토콜	snmp_priv_prot	SNMP 개인 정보 프로토콜 (DES, AES).
SNMP 개인 정보 프로토콜 암호	snmp_priv_passwd	SNMP 개인 정보 프로토콜 암호
SNMP 개인 정보 프로토콜 스크립트	snmp_priv_passwd_script	SNMP 개인 정보 프로토콜에 대한 암호를 제공하는 스크립트입니다. 이를 사용하는 것은 **SNMP 개인 정보 프로토콜 암호** 매개 변수를 대신합니다.
전원 대기 시간	power_wait	power off 또는 power on 명령 실행 후 대기 시간 (초)
포트	port	TCP 포트

표 A.19. “IF MIB”에서는 IF-MIB 장치의 차단 장치인 fence_ifmib에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.19. IF MIB

luci 필드	cluster.conf 속성	설명
이름	name	클러스터에 연결된 IF MIB 장치의 이름
IP 주소 또는 호스트 이름	ipaddr	장치에 할당된 IP 주소 또는 호스트 이름
UDP/TCP 포트 (옵션)	udpport	장치에 연결하는 데 사용할 UDP/TCP 포트. 기본값은 161입니다.

luci 필드	cluster.conf 속성	설명
로그인	login	장치에 액세스하기 위해 사용하는 로그인 이름
암호	passwd	장치에 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	passwd_script	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
SNMP 버전	snmp_version	사용할 SNMP 버전 (1, 2c, 3). 기본값은 1 입니다.
SNMP 커뮤니티	community	SNMP 커뮤니티 문자열
SNMP 보안 수준	snmp_sec_level	SNMP 보안 수준 (noAuthNoPriv, authNoPriv, authPriv).
SNMP 인증 프로토콜	snmp_auth_prot	SNMP 인증 프로토콜 (MD5, SHA).
SNMP 개인 정보 프로토콜	snmp_priv_prot	SNMP 개인 정보 프로토콜 (DES, AES).
SNMP 개인 정보 프로토콜 암호	snmp_priv_passwd	SNMP 개인 정보 프로토콜 암호
SNMP 개인 정보 프로토콜 스크립트	snmp_priv_passwd_script	SNMP 개인 정보 프로토콜에 대한 암호를 제공하는 스크립트입니다. 이를 사용하는 것은 **SNMP 개인 정보 프로토콜 암호** 매개 변수를 대신합니다.
전원 대기 시간	power_wait	power off 또는 power on 명령 실행 후 대기 시간 (초)
포트	port	물리적 플러그 번호 또는 가상 머신 이름입니다.

표 A.20. “Intel Modular”에서는 Intel Modular의 차단 에이전트인 **fence_intelmodular**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.20. Intel Modular

luci 필드	cluster.conf 속성	설명
이름	name	클러스터에 연결된 Intel Modular 장치의 이름입니다.
IP 주소 또는 호스트 이름	ipaddr	장치에 할당된 IP 주소 또는 호스트 이름

luci 필드	cluster.conf 속성	설명
로그인	login	장치에 액세스하기 위해 사용하는 로그인 이름
암호	passwd	장치에 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	passwd_script	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
SNMP 버전	snmp_version	사용할 SNMP 버전 (1, 2c, 3). 기본값은 1 입니다.
SNMP 커뮤니티	community	SNMP 커뮤니티 문자열. 기본값은 **private**입니다.
SNMP 보안 수준	snmp_sec_level	SNMP 보안 수준 (noAuthNoPriv, authNoPriv, authPriv).
SNMP 인증 프로토콜	snmp_auth_prot	SNMP 인증 프로토콜 (MD5, SHA).
SNMP 개인 정보 프로토콜	snmp_priv_prot	SNMP 개인 정보 프로토콜 (DES, AES).
SNMP 개인 정보 프로토콜 암호	snmp_priv_passwd	SNMP 개인 정보 프로토콜 암호
SNMP 개인 정보 프로토콜 스크립트	snmp_priv_passwd_script	SNMP 개인 정보 프로토콜에 대한 암호를 제공하는 스크립트입니다. 이를 사용하는 것은 **SNMP 개인 정보 프로토콜 암호** 매개 변수를 대신합니다.
전원 대기 시간	power_wait	power off 또는 power on 명령 실행 후 대기 시간 (초)
포트	port	물리적 플러그 번호 또는 가상 머신 이름입니다.

표 A.21. “IPMI (Intelligent Platform Management Interface) LAN” 에서는 LAN을 통한 IPMI의 차단 에이전트인 **fence_ipmilan**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.21. IPMI (Intelligent Platform Management Interface) LAN

luci 필드	cluster.conf 속성	설명
이름	name	클러스터에 연결되는 IPMI LAN 장치의 이름입니다.
IP 주소 또는 호스트 이름	ipaddr	장치에 할당된 IP 주소 또는 호스트 이름

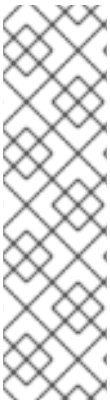
luci 필드	cluster.conf 속성	설명
로그인	**login**	주어진 IPMI 포트에 power on/off 명령을 실행할 수 있는 사용자 로그인 이름입니다.
암호	**passwd**	IPMI 포트로의 연결 인증에 사용되는 암호입니다.
암호 스크립트 (옵션)	**passwd_script**	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
인증 유형	**auth**	IPMI LAN 인증 유형: **none**, **password**, 또는 **md5**.
Lanplus 사용	**lanplus**	**True** 또는 **1**. 비어있는 경우, 값은 **False**가 됩니다.
사용할 Ciphersuite	**cipher**	IPMIv2 lanplus 연결에 사용할 원격 서버 인증, 무결성 및 암호화 알고리즘입니다.
권한 수준	**privlvl**	IPMI 장치에 있는 권한 수준입니다.

표 A.22. “RHEV-M REST API (RHEV 3.0 이상에 대해 RHEL 6.2 이상)” 에서는 RHEV-M REST API의 차단 에이전트인 **fence_rhevm**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.22. RHEV-M REST API (RHEV 3.0 이상에 대해 RHEL 6.2 이상)

luci 필드	cluster.conf 속성	설명
이름	**name**	RHEV-M REST API 차단 장치의 이름입니다.
IP 주소 또는 호스트 이름	**ipaddr**	장치에 할당된 IP 주소 또는 호스트 이름
IP 포트 (옵션)	**ipport**	장치에 연결하는 데 사용할 TCP 포트
로그인	**login**	장치에 액세스하기 위해 사용하는 로그인 이름
암호	**passwd**	장치에 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	**passwd_script**	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
SSL 사용	**ssl**	장치와 통신하기 위해 SSL 연결을 사용합니다.
전원 대기 시간	**power_wait**	power off 또는 power on 명령 실행 후 대기 시간 (초)
포트	**port**	물리적 플러그 번호 또는 가상 머신 이름입니다.

표 A.23. “SCSI Fencing”에서는 SCSI 영구적 저장소의 차단 장치인 **fence_scsi**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.



참고

차단 장치 방식으로 SCSI 영구적 저장소 사용에는 다음과 같은 제한 사항이 지원됩니다:

- SCSI 차단 장치를 사용할 때 클러스터에 있는 모든 노드는 동일한 장치에 등록하여 등록된 모든 장치에서 각 노드가 다른 노드의 등록키를 제거할 수 있도록 해야 합니다.
- 클러스터 볼륨에 사용되는 장치는 파티션이 아니라 전체 LUN이여야 합니다. SCSI 영구적 저장소는 전체 LUN에서 작동하므로 액세스는 개별적 파티션이 아닌 각각의 LUN에 제어할 수 있습니다.

표 A.23. SCSI Fencing

luci 필드	cluster.conf 속성	설명
이름	**name**	SCSI 차단 장치의 이름입니다.
노드 이름		
현재 작업을 위한 키		(노드 이름 덮어쓰기)

표 A.24. “VMware 차단 장치 (SOAP 인터페이스) (Red Hat Enterprise Linux 6.2 이상)” 에서는 SOAP API를 통한 VMWare의 차단 에이전트인 **fence_vmware_soap**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.24. VMware 차단 장치 (SOAP 인터페이스) (Red Hat Enterprise Linux 6.2 이상)

luci 필드	cluster.conf 속성	설명
이름	**name**	가상 머신 차단 장치의 이름입니다.
IP 주소 또는 호스트 이름	**ipaddr**	장치에 할당된 IP 주소 또는 호스트 이름
IP 포트 (옵션)	**ipport**	장치에 연결하는 데 사용할 TCP 포트
로그인	**login**	장치에 액세스하기 위해 사용하는 로그인 이름
암호	**passwd**	장치에 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	**passwd_script**	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.

luci 필드	cluster.conf 속성	설명
구분 기호	**separator**	작업 목록에 의해 생성되는 CSV 용 구분 기호입니다. 기본값은 콤마 (,)입니다.
전원 대기 시간	**power_wait**	power off 또는 power on 명령 실행 후 대기 시간 (초)
VM 이름	**port**	인벤토리 경로 형식의 가상 머신 이름입니다 (예: /datacenter/vm/Discovered_virtual_machine/myMachine).
VM UUID	**uuid**	차단 할 가상 머신의 UUID입니다.
SSL 사용	**ssl**	장치와 통신하기 위해 SSL 연결을 사용합니다.

표 A.25. “WTI 전원 스위치 ”에서는 WTI 네트워크 전원 스위치의 차단 에이전트인 **fence_wti**에 의해 사용되는 차단 장치 매개 변수를 나열합니다.

표 A.25. WTI 전원 스위치

luci 필드	cluster.conf 속성	설명
이름	**name**	클러스터에 연결되는 WTI 전원 스위치의 이름입니다.
IP 주소 또는 호스트 이름	**ipaddr**	장치에 할당된 IP 또는 호스트 이름 주소입니다.
IP 포트 (옵션)	**ipport**	장치에 연결하기 위해 사용하는 TCP 포트
로그인	**login**	장치에 액세스하기 위해 사용하는 로그인 이름
암호	**passwd**	장치에 연결을 인증하기 위해 사용되는 암호
암호 스크립트 (옵션)	**passwd_script**	차단 장치에 액세스하기 위한 암호를 제공하는 스크립트입니다. 이는 **Password** 매개 변수를 대신합니다.
포트	**port**	물리적 플러그 번호 또는 가상 머신 이름입니다.
명령 프롬프트를 강제	**cmd_prompt**	사용할 명령 프롬프트입니다. 디폴트 값은 다음과 같습니다: ['RSM>', '>MPC', 'IPS>', 'TPS>', 'NBB>', 'NPS>', 'VMR>']
전원 대기 시간	**power_wait**	power off 또는 power on 명령 실행 후 대기 시간 (초)
SSH 사용	**secure**	장치에 액세스하기 위해 SSH를 사용하는 시스템을 나타냅니다.
SSH 식별 파일로의 경로	**identity_file**	SSH 용 식별 파일

부록 B. HA 리소스 매개 변수

다음에서는 HA 리소스 매개 변수에 대해 설명합니다. **ccs** 명령을 사용하거나 **etc/cluster/cluster.conf** 파일을 편집하여 **luci**로 매개 변수를 설정할 수 있습니다. 표 B.1. “HA 리소스 요약 ”에는 리소스, 해당 리소스 에이전트, 매개 변수 설명이 포함된 다른 표에 대한 참조가 나열되어 있습니다. 리소스 에이전트에 대한 보다 자세한 내용은 클러스터 노드의 **/usr/share/cluster**에서 참조하십시오.

여기에서 설명된 리소스 에이전트 이외에 **/usr/share/cluster** 디렉토리에는 리소스 그룹 **service.sh**의 더미 (dummy) OCF 스크립트가 들어 있습니다. 이 스크립트에 포함된 매개 변수에 관한 보다 자세한 내용은 **service.sh** 스크립트에서 참조하십시오.

cluster.conf 요소 및 속성에 대한 전체 목록과 설명은 **/usr/share/cluster/cluster.rng**에 있는 클러스터 스키마 및 **/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html**에 있는 주석 스키마를 참조하십시오. (예: **/usr/share/doc/cman-3.0.12/cluster_conf.html**).

표 B.1. HA 리소스 요약

리소스	리소스 에이전트	매개 변수 설명에 대한 참조
Apache	apache.sh	표 B.2. “Apache 서버 ”
Condor 인스턴스	condor.sh	표 B.3. “Condor 인스턴스 ”
파일 시스템	fs.sh	표 B.4. “파일 시스템 ”
GFS2 파일 시스템	clusterfs.sh	표 B.5. “GFS2”
IP 주소	ip.sh	표 B.6. “IP 주소 ”
HA LVM	lvm.sh	표 B.7. “HA LVM”
MySQL	mysql.sh	표 B.8. “MySQL”
NFS 클라이언트	nfsclient.sh	표 B.9. “NFS 클라이언트 ”
NFS 내보내기	nfsexport.sh	표 B.10. “NFS 내보내기 ”
NFS 서버	nfsserver.sh	표 B.11. “NFS 서버”
NFS/CIFS 마운트	netfs.sh	표 B.12. “NFS/CIFS 마운트 ”
Open LDAP	openldap.sh	표 B.13. “Open LDAP”
Oracle 10g/11g 페일오버 인스턴스	oracledb.sh	표 B.14. “Oracle 10g/11G 페일오버 인스턴스 ”
Oracle 10g 페일오버 인스턴스	orainstance.sh	표 B.15. “Oracle 10g 페일오버 인스턴스 ”

리소스	리소스 에이전트	매개 변수 설명에 대한 참조
Oracle 10g 리스너	oralistener.sh	표 B.16. “Oracle 10g 리스너”
PostgreSQL 8	postgres-8.sh	표 B.17. “PostgreSQL 8”
SAP 데이터베이스	SAPDatabase	표 B.18. “SAP 데이터베이스 ”
SAP 인스턴스	SAPInstance	표 B.19. “SAP 인스턴스 ”
Samba	samba.sh	표 B.20. “Samba 서버 ”
스크립트	script.sh	표 B.21. “스크립트 ”
Sybase ASE	ASEHAagent.sh	표 B.22. “Sybase ASE 페일오버 인스턴스 ”
Tomcat 6	tomcat-6.sh	표 B.23. “Tomcat 6”
가상 머신	vm.sh	표 B.24. “가상 머신 ” 알림: 호스트 클러스터가 가상 시스템을 지원하는 경우 **luci**는 이를 가상 서비스로 표시합니다.

표 B.2. Apache 서버

luci 필드	`cluster.conf` 속성	설명
이름	**`name`**	Apache 서비스 이름
서버 Root	**`server_root`**	기본값은 **`/etc/httpd`**입니다.
설정 파일	**`config_file`**	Apache 설정 파일을 지정합니다. 기본값은 **`/etc/httpd/conf`**입니다.
httpd 옵션	**`httpd_options`**	**`httpd`**에 대한 다른 명령행 옵션
종료 대기 시간 (초)	**`shutdown_wait`**	서비스 종료까지의 정확한 대기 시간 (초)을 지정합니다.

표 B.3. Condor 인스턴스

필드	luci 필드	`cluster.conf` 속성
인스턴스 이름	**`name`**	Condor 인스턴스에 고유한 이름을 지정합니다.

필드	luci 필드	cluster.conf 속성
Condor 하부 시스템 유형	**type**	이 인스턴스의 Condor 하부 시스템 유형을 지정합니다: **schedd**, **job_server**, **query_server**

표 B.4. 파일 시스템

luci 필드	cluster.conf 속성	설명
이름	**name**	파일 시스템 리소스에 대한 이름을 지정합니다.
파일 시스템 유형	**fstype**	지정하지 않으면, **mount**가 파일 시스템 유형을 결정합니다.
마운트 지점	**mountpoint**	파일 시스템 계층에서 이 파일 시스템을 마운트하기 위한 경로
장치, FS 레이블, 또는 UUID	**device**	파일 시스템 리소스와 연관된 장치를 지정합니다. 이는 블록 장치, 파일 시스템 레이블, 또는 파일 시스템의 UUID가 될 수 있습니다.
마운트 옵션	**options**	마운트 옵션으로 파일 시스템을 마운트하는 데 사용하는 옵션입니다. 이는 파일 시스템 고유 정보일 수 있습니다. 지원되는 마운트 옵션은 ***mount(8)*** man 페이지를 참조하십시오.
파일 시스템 ID (옵션)	**fsid**	 **참고** ***File System ID***는 NFS 서비스에 의해서만 사용됩니다. 새 파일 시스템 리소스를 만들때, 필드를 공백으로 남겨둘 수 있습니다. 필드를 공백으로 두면 설정 매개 변 수가 커밋된 후, 파일 시스템 ID를 자동으로 할당합니다. 파일 시스템 ID를 명시적으로 지정할 필요가 있을 경우, 이 필드에 지정합니다.
마운트 해제 강제	**force_unmount**	활성화하면 파일 시스템 마운트 해제를 강제합니다. 기본 설정은 ***disabled***입니다. ***Force Unmount***는 마운트 포인트를 사용하는 모든 프로세스를 종료하고 마운트를 해재합니다.
fsck 강제	**force_fsck**	활성화하면, 마운트하기전 파일 시스템에서 **fsck**를 실행하게 합니다. 기본 설정은 ***disabled***입니다.
NFS 데몬 및 lockd 문제 해결 방법을 사용 (Red Hat Enterprise Linux 6.4 이상)	**nfsrestart**	파일 시스템이 NFS를 통해 내보내기되고 마운트 해제하는데 문제가 발생할 수 있는 경우 (종료시나 서비스의 재배치시) 이 옵션을 설정하면 마운트 해제 작업 전에 모든 파일 시스템 참조를 드롭하게 됩니다. 이 옵션을 설정하려면 **Force unmount** 옵션을 활성화해야 하며 **NFS Server** 리소스와 함께 사용 할 수 없습니다. 파일 시스템 마운트 해제가 어려워지기 때문에 최후의 수단으로 이 옵션을 설정하도록 합니다.

luci 필드	cluster.conf 속성	설명
빠른 상태 검사 사용	**quick_status**	활성화된 경우 빠른 상태 검사를 수행합니다.
마운트 해제 실패 시 호스트 노드 재부팅	**self_fence**	활성화된 경우 파일 시스템 마운트 해제에 문제가 발생하면 노드를 다시 시작합니다. **filesystem** 리소스 에이전트는 1, **yes**, **on**, **true**로 이 매개 변수를 활성화하며 0, **no**, **off**, **false**로 이를 비활성화합니다. 기본값 설정은 ***disabled***입니다.

표 B.5. GFS2

luci 필드	cluster.conf 속성	설명
이름	**name**	파일 시스템 리소스의 이름
마운트 지점	**mountpoint**	파일 시스템 리소스가 마운트되는 경로입니다.
장치, FS 레이블, 또는 UUID	**device**	파일 시스템 리소스에 관련된 장치 파일입니다.
파일 시스템 유형	**fstype**	**luci**에서 GFS2로 설정합니다
마운트 옵션	**options**	마운트 옵션
파일 시스템 ID (옵션)	**fsid**	 **참고** ***File System ID***는 NFS 서비스에 의해서만 사용됩니다. 새 GFS2 리소스를 만들 때, 이 필드를 공백으로 남겨둘 수 있습니다. 필드를 공백으로 두면 설정 매개변수가 커밋된 후에 파일 시스템 ID를 자동으로 할당합니다. 파일 시스템 ID를 명시적으로 지정할 필요가있을 경우, 이 필드에 지정합니다.
마운트 해제 강제	**force_unmount**	활성화할 경우, 파일 시스템의 마운트 해제를 강제합니다. 기본 설정은 ***disabled***입니다. ***Force Unmount***는 마운트 포인트를 사용하는 모든 프로세스를 종료하여 마운트를 해제합니다. GFS2 리소스의 마운트 포인트는 ***Force Unmount***가 *활성화*로 되어 있지 않을 경우 서비스 중지 시 마운트 포인트가 마운트 해제되지 *않습니다.*

luci 필드	cluster.conf 속성	설명
NFS 데몬 및 lockd 문제 해결 방법을 사용 (Red Hat Enterprise Linux 6.4 이상)	**nfsrestart**	파일 시스템이 NFS를 통해 내보내기되고 마운트 해제하는데 문제가 발생할 수 있는 경우 (종료시나 서비스의 재배치시) 이 옵션을 설정하면 마운트 해제 작업 전에 모든 파일 시스템 참조를 드롭하게 됩니다. 이 옵션을 설정하려면 **Force unmount** 옵션을 활성화해야 하며 **NFS Server** 리소스와 함께 사용 할 수 없습니다. 파일 시스템 마운트 해제가 어려워지기 때문에 최후의 수단으로 이 옵션을 설정하도록 합니다.
마운트 해제 실패 시 호스트 노드 재부팅	**self_fence**	활성화되어 있고 파일 시스템 마운트 해제에 실패할 경우 노드는 즉시 다시 시작됩니다. 일반적으로 이는 force-unmount 지원과 함께 사용되지만 필수 사항은 아닙니다. **GFS2** 리소스 에이전트는 1, **yes**, **on**, **true**로 이러한 매개 변수를 활성화하며 0, **no**, **off**, **false**로 이를 비활성화합니다.

표 B.6. IP 주소

luci 필드	cluster.conf 속성	설명
IP 주소, 넷마스크 비트	**address**	리소스의 IP 주소 (및 옵션으로 넷마스크 비트). 넷 마스크 비트 또는 네트워크 접두사 길이는 슬래시로 구분하여 주소 뒤에 옵니다. 이는 CIDR 표기법을 준수합니다 (예: 10.1.1.1/8). 이는 가상 IP 주소입니다. IPv4 및 IPv6 주소는 각 IP 주소의 NIC 링크 모니터링으로 지원됩니다.
연결 상태 검사	**monitor_link**	이를 활성화하면 해당 IP 주소가 바인딩되어 있는 NIC 연결이 없는 경우 상태 검사를 실패하게 합니다.
고정 라우트에 업데이트 비활성화	**disable_rdisc**	RDISC 프로토콜을 사용하여 라우팅 업데이트를 비활성화합니다.
IP 주소 삭제 후 슬립 상태 시간 (초)	**sleeptime**	슬립 상태에 있게 할 시간 (초)을 지정합니다.

표 B.7. HA LVM

luci 필드	cluster.conf 속성	설명
이름	**name**	이 LVM 리소스의 고유한 이름입니다.
볼륨 그룹 이름	**vg_name**	관리되는 볼륨 그룹을 설명하는 이름입니다.
논리 볼륨 이름 (옵션)	**lv_name**	관리되는 논리 볼륨의 이름. 이 매개 변수는 관리되는 볼륨 그룹에 여러 개의 논리 볼륨이있는 경우 옵션입니다.

luci 필드	cluster.conf 속성	설명
LVM 태그를 제거할 수 없는 경우 노드 차단	**self_fence**	LVM 태그를 제거할 수 없는 경우 노드를 차단합니다. LVM 리소스 에이전트는 1 또는 **yes**로 이러한 매개 변수를 활성화하며 0 또는**no**로 이를 비활성화합니다.

표 B.8. MySQL

luci 필드	cluster.conf 속성	설명
이름	**name**	MySQL 서버 리소스의 이름을 지정합니다.
설정 파일	**config_file**	설정 파일을 지정합니다. 기본값은 **/etc/my.cnf**입니다.
수신 주소	**listen_address**	MySQL 서버의 IP 주소를 지정합니다. IP 주소가 제시되어 있지 않은 경우에는 서비스의 첫 번째 IP 주소가 사용됩니다.
mysqld 옵션	**mysqld_options**	**httpd**에 대한 다른 명령행 옵션
시작 대기 시간 (초)	**startup_wait**	정확한 서비스 시작 종료까지의 대기 시간 (초)을 지정합니다.
종료 대기 시간 (초)	**shutdown_wait**	서비스 종료까지의 정확한 대기 시간 (초)을 지정합니다.

표 B.9. NFS 클라이언트

luci 필드	cluster.conf 속성	설명
이름	**name**	리소스 트리에서 클라이언트를 참조하는데 사용되는 심볼릭 이름입니다. 이는 ***Target*** *옵션과 동일하지 않습니다.*
대상 호스트 이름, 와일드 카드, Netgroup	**target**	마운트 하려는 서버입니다. 호스트 이름, 와일드 카드 (IP 주소 또는 호스트 이름 기반), 또는 내보내려는 호스트를 정의하는 netgroup을 사용하여 지정할 수 있습니다.
NFS 클라이언트의 복구 허용	**allow_recover**	복구 허용합니다.
옵션	**options**	이 클라이언트를 위한 옵션 목록을 정의합니다 – 예를 들면, 추가 클라이언트는 액세스 권한을 갖음. 자세한 정보는 ***exports*** *(5)* man 페이지, *일반 옵션*을 참조하십시오.

표 B.10. NFS 내보내기

luci 필드	cluster.conf 속성	설명
이름	**name**	리소스의 기술적인 이름입니다. NFS 내보내기 리소스는 NFS 데몬이 실행되고 있는지를 확인합니다. 이는 완전히 다시 사용할 수 있는 일반적인것으로 하나의 NFS 내보내기 리소스만 필요합니다.  **참고** NFS 내보내기 리소스를 다른 NFS 리소스와 구별할 수 있도록 명확하게 이름을 지정합니다.

표 B.11. NFS 서버

luci 필드	cluster.conf 속성	설명
이름	**name**	NFS 서버 리소스를 설명하는 이름입니다. NFS 서버 리소스는 NFSv4 파일 시스템을 클라이언트로 내보내기할 때 유용합니다. NFSv4 작동 방식 때문에 한 번에 하나의 NFSv4 리소스만 서버에 존재할 수 있습니다. 또한 각 클러스터 노드에 NFS 로컬 인스턴스를 사용할 때 NFS 서버 리소스를 사용할 수 없습니다.

표 B.12. NFS/CIFS 마운트

luci 필드	cluster.conf 속성	설명
이름	**name**	NFS 또는 CIFS 마운트의 심볼릭 이름입니다.  **참고** 이 리소스는 클러스터 서비스가 NFS 클라이언트로 설정되는 경우에만 필요합니다.
마운트 지점	**mountpoint**	파일 시스템 리소스가 마운트되는 경로입니다.
호스트	**host**	NFS/CIFS 서버의 IP 주소 또는 호스트 이름입니다.
NFS 내보내기 디렉토리 이름 또는 CIFS 공유	**export**	NFS 내보내기 디렉토리 이름 또는 CIFS 공유 이름

luci 필드	cluster.conf 속성	설명
파일 시스템 유형	**fstype**	파일 시스템 유형: • ***NFS*** – 기본 NFS 버전 사용을 지정합니다. 이는 기본 설정입니다. • ***NFS v4*** – NFSv4 프로토콜 사용을 지정합니다. • ***CIFS*** – CIFS 프로토콜 사용을 지정합니다.
마운트 해제 강제	**force_unmount**	***Force Unmount***가 활성화되어 있을 경우, 서비스가 중지되었을 때 클러스터는 파일 시스템을 사용하는 모든 프로세스를 종료합니다. 파일 시스템을 사용하는 모든 프로세스를 종료하면 파일 시스템을 해제합니다. 그렇지 않을경우, 마운트 해제는 실패하고 서비스를 다시 시작합니다.
이동 작업 중지 시 파일 시스템을 마운트 해제하지 않음	**no_unmount**	활성화되어 있을 경우 파일 시스템은 중지 또는 이동 작업시 마운트 해제되지 않음이 지정됩니다.
옵션	**options**	마운트 옵션입니다. 마운트 옵션의 목록을 지정합니다. 아무것도 지정하지 않으면 파일 시스템은 **-o sync**로 마운트됩니다.

표 B.13. Open LDAP

luci 필드	cluster.conf 속성	설명
이름	**name**	로깅 및 다른 목적을 위한 서비스 이름을 지정합니다.
설정 파일	**config_file**	설정 파일에 절대 경로를 지정합니다. 기본값은 **/etc/openldap/slapd.conf**입니다.
URL 목록	**url_list**	기본값은 **ldap:///**입니다.
slapd 옵션	**slapd_options**	**slapd**의 다른 명령행 옵션
종료 대기 시간 (초)	**shutdown_wait**	서비스 종료까지의 정확한 대기 시간 (초)을 지정합니다.

표 B.14. Oracle 10g/11G 페일오버 인스턴스

luci 필드	cluster.conf 속성	설명
Oracle 인스턴스의 인스턴스 이름 (SID)	**name**	인스턴스 이름
Oracle 사용자 이름	**user**	이는 Oracle AS 인스턴스를 실행하는 Oracle 사용자의 사용자 이름입니다.
Oracle 애플리케이션 홈 디렉토리	**home**	이는 Oracle (사용자가 아닌 응용 프로그램)의 홈 디렉토리입니다. Oracle 설치시 설정됩니다.
Oracle 설치 유형	**type**	Oracle 설치 유형. 기본값: **10g** 데이터베이스 인스턴스 및 리스너 전용 **base**, 데이터베이스, 리스너, Enterprise 관리자, ISQL*Plus:**base-em** (또는 **10g**), 또는 Internet Application Server (infrastructure):**ias** (또는 **10g-ias**).
가상 호스트 이름 (옵션)	**vhost**	Oracle 10g 설치 호스트 이름과 일치하는 가상 호스트 이름입니다. oracledb 리소스 시작/중지시 호스트 이름은 이 호스트이름으로 일시적으로 변경됨에 유의하십시오. 따라서 oracledb 리소스를 자체 서비스의 일부로만 설정해야 합니다.

표 B.15. Oracle 10g 페일오버 인스턴스

luci 필드	cluster.conf 속성	설명
Oracle 인스턴스의 인스턴스 이름 (SID)	**name**	인스턴스 이름
Oracle 사용자 이름	**user**	Oracle 인스턴스가 Oracle 사용자로 실행하는 사용자 이름입니다.
Oracle 애플리케이션 홈 디렉토리	**home**	이는 Oracle (사용자가 아닌 응용 프로그램)의 홈 디렉토리입니다. Oracle 설치시 설정됩니다.
Oracle 리스너 목록 (옵션, 공백으로 구분)	**listeners**	데이터베이스 인스턴스로 시작되는 Oracle 리스너 목록입니다. 리스너 이름은 공백으로 구분합니다. 기본값은 공백으로 리스너를 비활성화합니다.
잠금 파일로의 경로 (옵션)	**lockfile**	Oracle이 실행되고 있는지에 대한 여부를 확인하기 위해 사용되는 lockfile의 위치입니다. 기본값 위치는 **/tmp** 아래입니다.

표 B.16. Oracle 10g 리스너

luci 필드	cluster.conf 속성	설명
리스너 이름	name	리스너 이름입니다.
Oracle 사용자 이름	user	Oracle 인스턴스가 Oracle 사용자로 실행하는 사용자 이름입니다.
Oracle 애플리케이션 홈 디렉토리	home	이는 Oracle (사용자가 아닌 응용 프로그램)의 홈 디렉토리입니다. Oracle 설치시 설정됩니다.

표 B.17. PostgreSQL 8

luci 필드	cluster.conf 속성	설명
이름	name	로깅 및 다른 목적을 위한 서비스 이름을 지정합니다.
설정 파일	config_file	설정 파일로의 절대 경로를 정의합니다. 기본값은 /var/lib/pgsql/data/postgresql.conf입니다.
Postmaster 사용자	postmaster_user	root로 데이터베이스 서버를 실행할 수 없기대문에 이를 실행하는 사용자입니다. 기본값은 postgres입니다.
Postmaster 옵션	postmaster_options	postmaster에 대한 다른 명령행 옵션
종료 대기 시간 (초)	shutdown_wait	서비스 종료까지의 정확한 대기 시간 (초)을 지정합니다.

표 B.18. SAP 데이터베이스

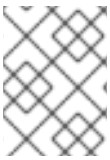
luci 필드	cluster.conf 속성	설명
SAP 데이터베이스 이름	SID	고유한 SAP 시스템 식별자를 지정합니다. 예: P01.
SAP 실행 가능 디렉토리	DIR_EXECUTABLE	sapstartsrv 및 sapcontrol에 대한 정규화된 경로를 지정합니다.
데이터베이스 유형	DBTYPE	다음의 데이터베이스 유형 중 하나를 지정합니다: Oracle, DB6, ADA.
Oracle 리스너 이름	NETSERVICENAME	Oracle TNS 청취자 이름을 지정합니다.

luci 필드	cluster.conf 속성	설명
ABAP 스택이 설치되어 있지 않음, Java 스택만 설치됨	**DBJ2EE_ONLY**	SAP 데이터베이스에 ABAP 스택이 설치되어 있지 않은 경우, 이 매개 변수를 활성화합니다.
애플리케이션 수준 모니터링	**STRICT_MONITORING**	애플리케이션 수준 모니터링을 활성화합니다.
자동 복구 시작	**AUTOMATIC_RECOVER**	자동 복구 시작을 활성화 또는 비활성화합니다
Java SDK로의 경로	**JAVE_HOME**	Java SDK로의 경로입니다.
JDBC 드라이버의 파일 이름	**DB_JARS**	JDBC 드라이버의 파일 이름입니다.
Pre-Start 스크립트로의 경로	**PRE_START_USEREXIT**	pre-start 스크립트로의 경로입니다.
Post-Start 스크립트로의 경로	**POST_START_USEREXIT**	post-start 스크립트로의 경로입니다.
Pre-Stop 스크립트로의 경로	**PRE_STOP_USEREXIT**	pre-stop 스크립트로의 경로입니다.
Post-Stop 스크립트로의 경로	**POST_STOP_USEREXIT**	post-stop 스크립트로의 경로입니다.
J2EE 인스턴스 부트스트랩 디렉토리	**DIR_BOOTSTRAP**	J2EE 인스턴스 부트스트랩 디렉토리의 정규화된 경로입니다. 예: **/usr/sap/P01/J00/j2ee/cluster/bootstrap**.
J2EE 보안 저장 경로	**DIR_SECSTORE**	J2EE 보안 저장 디렉토리의 정규화된 경로입니다. 예: **/usr/sap/P01/SYS/global/security/lib/tools**.

표 B.19. SAP 인스턴스

luci 필드	cluster.conf 속성	설명
SAP 인스턴스 이름	**InstanceName**	정규화된 SAP 인스턴스 이름입니다. 예: P01_DVEBMGS00_sapp01ci.

luci 필드	cluster.conf 속성	설명
SAP 실행 가능 디렉토리	**DIR_EXECUTABLE**	**sapstartsrv** 및 **sapcontrol**로의 정규화된 경로입니다.
SAP START 프로파일이 포함된 디렉토리	**DIR_PROFILE**	SAP START 프로파일로의 정규화된 경로입니다.
SAP START 프로파일 이름	**START_PROFILE**	SAP START 프로파일의 이름을 지정합니다.
시작 상태를 확인하기 전 까지 대기 시간 (초)	**START_WAITTIME**	시작 상태를 확인하기 전 까지 대시 시간 (초)을 지정합니다 (J2EE-Addin의 경우 기다리지 않음).
자동 복구 시작을 활성화	**AUTOMATIC_RECOVER**	자동 복구 시작을 활성화 또는 비활성화합니다
Pre-Start 스크립트로의 경로	**PRE_START_USEREXIT**	pre-start 스크립트로의 경로입니다.
Post-Start 스크립트로의 경로	**POST_START_USEREXIT**	post-start 스크립트로의 경로입니다.
Pre-Stop 스크립트로의 경로	**PRE_STOP_USEREXIT**	pre-stop 스크립트로의 경로입니다.
Post-Stop 스크립트로의 경로	**POST_STOP_USEREXIT**	post-stop 스크립트로의 경로입니다.



참고

표 B.20. “Samba 서버 ” 에 대해 클러스터 서비스를 생성하거나 편집할 때 Samba-service 리소스를 직접 서비스에 연결하는 것이지 서비스 내의 리소스에 연결하는 것이 *아닙니다*.

표 B.20. Samba 서버

luci 필드	cluster.conf 속성	설명
이름	**name**	Samba 서버의 이름을 지정합니다.
설정 파일	**config_file**	Samba 설정 파일

luci 필드	cluster.conf 속성	설명
smbd의 다른 명령행 옵션	**smbd_options**	smbd의 다른 명령행 옵션입니다.
nmbd의 다른 명령행 옵션	**nmbd_options**	nmbd의 다른 명령행 옵션입니다.
종료 대기 시간 (초)	**shutdown_wait**	서비스 종료까지의 정확한 대기 시간 (초)을 지정합니다.

표 B.21. 스크립트

luci 필드	cluster.conf 속성	설명
이름	**name**	사용자 정의 사용자 스크립트의 이름을 지정합니다. 스크립트 리소스를 통해 표준 LSB 호환 init 스크립트가 클러스터 서비스를 시작하는데 사용할 수 있습니다.
스트립트 파일로의 전체 경로	**file**	이 사용자 지정 스크립트가 배치되어 있는 위치의 경로를 입력합니다 (예: **/etc/init.d/*userscript***).

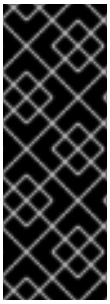
표 B.22. Sybase ASE 페일오버 인스턴스

luci 필드	cluster.conf 속성	설명
인스턴스 이름	**name**	Sybase ASE 리소스의 인스턴스 이름을 지정합니다.
ASE 서버 이름	**server_name**	HA 서비스에 대해 설정된 ASE 서버의 이름
SYBASE 홈 디렉토리	**sybase_home**	Sybase 제품의 홈 디렉토리
로그인 파일	**login_file**	로그인-암호 쌍이 포함된 로그인 파일의 전체 경로
인터페이스 파일	**interfaces_file**	ASE 서버 시작 및 액세스하는 데 사용되는 인터페이스 파일의 전체 경로
SYBASE_ASE 디렉토리 이름	**sybase_ase**	ASE 제품이 설치되는 sybase_home 하의 디렉토리 이름
SYBASE_OCS 디렉토리 이름	**sybase_ocs**	OCS 제품이 설치되는 sybase_home 하의 디렉토리 이름입니다. 예, ASE-15_0

luci 필드	cluster.conf 속성	설명
Sybase 사용자	sybase_user	ASE 서버를 실행할 수 있는 사용자
시작 시간 제한 (초)	start_timeout	시작 시간 제한 값입니다.
종료 시간 제한 (초)	shutdown_timeout	종료 시간 제한 값입니다.
Deep Probe 시간 제한	deep_probe_timeout	deep probe를 실행하는 동안 서버가 응답하지 않는다는 것을 확인하기 전에 ASE 서버의 응답을 기다리는 최대 시간 (초)

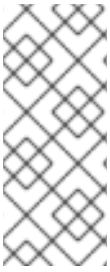
표 B.23. Tomcat 6

luci 필드	cluster.conf 속성	설명
이름	name	로깅 및 다른 목적을 위한 서비스 이름을 지정합니다.
설정 파일	config_file	설정 파일의 절대 경로를 지정합니다. 기본값은 /etc/tomcat6/tomcat6.conf입니다.
종료 대기 시간 (초)	shutdown_wait	서비스 종료까지의 올바른 대기 시간 (초)을 지정합니다. 기본값은 30입니다.



중요

표 B.24. "가상 머신 "에 대해 가상 머신 리소스로 사용자 클러스터를 설정할 때 rgmanager 도구를 사용하여 가상 머신을 시작 및 중지해야 합니다. virsh를 사용하여 컴퓨터를 시작하면 가상 머신이 여러 위치에서 실행되어 가상 머신에 있는 데이터가 손상될 수 있습니다. 클러스터 및 비 클러스터 도구 모두를 사용하여 관리자가 가상 머신을 실수로 "중복 시작"하는 위험을 줄이기 위해 시스템을 설정하는 방법에 대한 내용은 2.14절. "클러스터 환경에서 가상 머신 설정 "에서 참조하십시오.

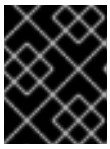


참고

가상 머신 리소스는 다른 클러스터 리소스와는 다르게 설정됩니다. luci를 사용하여 가상 머신 리소스를 설정하려면 서비스 그룹을 클러스터에 추가한 후 리소스를 서비스에 추가합니다. Virtual Machine을 리소스 유형으로 선택하고 가상 머신 리소스 매개 변수를 입력합니다. ccs를 사용하여 가상 머신을 설정하는 내용은 5.12절. "가상 머신 리소스 "에서 참조하십시오.

표 B.24. 가상 머신

luci 필드	cluster.conf 속성	설명
서비스 이름	**name**	가상 머신 이름을 지정합니다. **luci** 인터페이스를 사용할 때 이를 서비스 이름으로 지정해야 합니다.
자동으로 서비스 시작	**autostart**	활성화되어 있을 경우, 가상 시스템은 클러스터가 정원을 구성한 후 자동으로 시작됩니다. 이 매개 변수가 *비활성화*되어 있는 경우 가상 머신은 클러스터가 정원을 구성한 후에도 자동으로 시작되지 *않습니다*. 가상 머신은 ***disabled*** (비활성화) 상태로 저장됩니다.
단독 실행	**exclusive**	활성화되어 있을 경우, 가상 머신은 다른 노드에서 전용으로 실행되도록 재배치될 수 있습니다. 즉, 다른 가상 머신이 실행하지 않는 노드에서 실행됩니다. 가상 머신 전용 실행을 위한 노드를 사용할 수 없는 경우 가상 머신은 실패 후 다시 시작하지 않습니다. 또한 다른 가상 머신은 ***Run exclusive*** 로 이 가상 머신을 실행하는 노드에 자동으로 재배치되지 않습니다. 수동으로 시작하거나 또는 재배치 작업을 하여 이옵션을 무시할 수 있습니다.
페일오버 도메인	**domain**	가상 시스템이 실패한 상황에서 시행하는 클러스터 구성원의 목록을 정의합니다.
복구 정책	**recovery**	***Recovery policy*** 는 다음과 같은 옵션을 제공합니다: • ***Disable*** – 가상 머신이 실패하면, 이를 비활성화합니다. • ***Relocate*** – 가상 머신을 다른 노드에서 다시 시작하려고 합니다. 즉, 현재 노드에서 다시 시작하려고 시도하지 않습니다. • ***Restart*** – 가상 머신을 다른 노드로 재배치하기 전 (기본값) 가상 머신을 로컬 (현재 노드)에서 다시 시작 시도합니다. • ***Restart-Disable*** – 서비스 실패 시 이는 다시 시작됩니다. 하지만 다시 시작한 서비스가 실패할 경우, 서비스는 클러스터에 있는 다른 호스트로 이동하지 않고 비활성화됩니다.
재시작 옵션	**max_restarts**, **restart_expire_time**	서비스의 복구 정책으로 **다시 시작 (Restart)** 또는 **다시 시작-비활성화 (Restart-Disable)**를 선택한 경우, 서비스를 이동 또는 비활성화하기 전까지 다시 시작 실패의 최대 횟수를 지정할 수 있으며 다시 시작을 잊어버린 후 시간을 초 단위로 지정할 수 있습니다.
마이그레이션 유형	**migrate**	***live*** 또는 ***pause***의 마이그레이션 유형을 지정합니다. 기본 설정은 ***live***입니다.
마이그레이션 맵핑	**migration_mapping**	마이그레이션을 위한 대체 인터페이스를 지정합니다. 예를 들어 노드 상의 가상 머신 마이그레이션을 위해 사용된 네트워크 주소가 클러스터 통신을 위해 사용된 노드 주소와 다를 경우에 이를 지정할 수 있습니다. 다음을 지정하면 가상 머신을 **member**에서 **member2**로 마이그레이션할 때 실제로 **target2**로 마이그레이션하는 것으로 표시됩니다. 유사하게 **member2**에서 **member**로 마이그레이션할 때 **target**을 사용하여 마이그레이션할 수 있습니다. **member:target,member2:target2**

luci 필드	cluster.conf 속성	설명
상태 프로그램	**status_program**	가상 머신 존재 확인을 위한 표준 검사 이외에 실행할 상태 프로그램입니다. 지정되어 있는 경우 상태 프로그램은 1 분 마다 한 번씩 실행됩니다. 이를 사용하여 가상 머신 내의 중요한 서비스의 상태를 확인할 수 있습니다. 예를 들어, 가상 머신이 웹 서버를 실행하면 상태 프로그램은 웹 서버가 활성화되어 실행되고 있는지를 확인할 수 있습니다. 이러한 상태 검사가 실패하면 (0 이외의 값이 반환될 경우), 가상 머신은 복구됩니다. 가상 머신을 시작한 후, 가상 머신 리소스 에이전트는 정기적으로 상태 프로그램을 호출하고 반환전 성공적 반환 코드 (0) 결과를 기다립니다. 이러한 대기 시간은 5 분 후에 만료됩니다.
VM 생성에 사용되는 xmlfile로의 경로	**xmlfile**	**libvirt** 도메인 정의가 포함된 **libvirt** XML 파일로의 완전 경로
VM 설정 파일 경로	**path**	가상 머신 리소스 에이전트 (**vm.sh**)가 가상 머신 설정 파일을 검색하는 콜론으로 구분된 경로의 사양. 예: **/mnt/guests/config:/etc/libvirt/qemu**  중요 경로는 *절대로* 가상 머신 설정 파일을 직접 포인트해서는 안됩니다.
VM 스냅샷 디렉토리로의 경로	**snapshot**	가상 머신 이미지가 저장되는 스냅샷 디렉토리로의 경로입니다.
하이퍼바이저 URI	**hypervisor_uri**	하이퍼바이저 URI (일반적으로 자동).
마이그레이션 URI	**migration_uri**	마이그레이션 URI (일반적으로 자동).
마이그레이션 도중 ssh를 통한 터널 데이터	**tunnelled**	마이그레이션 도중 ssh를 통한 터널 데이터입니다.

부록 C. HA 리소스 동작

다음에서는 HA 리소스의 일반적인 동작을 설명합니다. 이는 HA 서비스 설정에 대해 부수적인 정보를 제공하기 위함입니다. **luci**를 사용하거나 **/etc/cluster/cluster.conf**를 편집하여 매개 변수를 설정할 수 있습니다. HA 리소스 매개 변수에 대한 설명은 부록 B. *HA 리소스 매개 변수*에서 참조하십시오. 리소스 에이전트에 대한 자세한 내용은 클러스터 노드의 **/usr/share/cluster**에서 참조하십시오.

참고

이 부분의 내용을 완전하게 이해하려면, 리소스 에이전트와 클러스터 설정 파일 **/etc/cluster/cluster.conf**을 세부적으로 이해할 필요가 있을 수 도 있습니다.

HA 서비스는 일관된 엔티티로 구성된 클러스터 리소스 그룹으로 클라이언트에 전문화된 서비스를 제공합니다. HA 서비스는 클러스터 구성 파일 **/etc/cluster/cluster.conf** (각 클러스터 노드)에서 리소스 트리로 표시됩니다. 클러스터 구성 파일에서 각 리소스 트리는 각 리소스와 리소스의 특성 그리고 리소스 트리에서 다른 리소스와의 관계 (부모, 자식, 형제 관계)를 지정하는 XML 표현입니다.

참고

HA 서비스는 계층 트리 형식으로 조직화된 리소스로 구성되어 있기 때문에 서비스는 *리소스 트리* 또는 *리소스 그룹*이라고 부릅니다. 두가지 모두 *HA 서비스*와 동의어입니다.

각 리소스 트리의 root에는 특수한 리소스 유형 – *서비스 리소스*가 있습니다. 다른 유형의 리소스는 서비스의 나머지 부분을 구성하고 그 특성을 결정합니다. HA 서비스 설정은 서비스 리소스 생성, 종속적 클러스터 리소스 생성 및 서비스의 계층적 제한에 따라 일관된 엔티티로 조직화하는 것으로 이루어져 있습니다.

이는 다음과 같은 부분으로 구성되어 있습니다:

- C.1절. “리소스 간의 부모, 자식, 형제 관계 ”
- C.2절. “형제 시작 순서 및 리소스 자식 순서 ”
- C.3절. “상속, <리소스> 블록 및 리소스 재사용 ”
- C.4절. “장애 복구 및 독립적 하위 트리 ”
- C.5절. “서비스 및 리소스 순서 디버그 및 테스트 ”

참고

클러스터 설정 파일 **/etc/cluster/cluster.conf**의 다음에 나타나는 예시는 설명 목적으로만 사용됩니다.

C.1. 리소스 간의 부모, 자식, 형제 관계

클러스터 서비스는 **rgmanager**의 제어 하에 실행되는 통합된 엔티티입니다. 서비스에 있는 모든 리소스는 동일한 노드에서 실행됩니다. **rgmanager**의 관점에서 클러스터 서비스는 시작, 중지 또는 재배치할 수 있는 하나의 엔티티입니다. 클러스터 서비스에서 리소스의 계층에 따라 리소스를 시작하거나 중지되는 순서가 결정됩니다. 계층 레벨은 부모, 자식, 형제로 구성되어 있습니다.

예 C.1. “foo 서비스 리소스 계층 ” 에서는 *foo* 서비스 리소스 트리의 예시를 보여주고 있습니다. 예에서 리소스 간의 관계는 다음과 같습니다:

- **fs:myfs** (<fs name="myfs" ...>) 와 **ip:10.1.1.2** (<ip address="10.1.1.2 .../>)는 형제입니다.
- **fs:myfs** (<fs name="myfs" ...>)는 **script:script_child** (<script name="script_child"/>)의 부모입니다.
- **script:script_child** (<script name="script_child"/>)는 **fs:myfs** (<fs name="myfs" ...>)의 자식입니다.

예 C.1. foo 서비스 리소스 계층

```
<service name="foo" ...>
    <fs name="myfs" ...>
        <script name="script_child"/>
    </fs>
    <ip address="10.1.1.2" .../>
</service>
```

리소스 트리에서 다음 규칙이 부모/자식 관계에 적용됩니다:

- 부모는 자식 전에 시작됩니다.
- 부모가 멈추기 전에 먼저 자식을 완전히 중지해야 합니다.
- 양호한 상태에 있다고 간주되는 리소스의 경우, 리소스의 모든 자식은 양호한 상태여야 합니다.

C.2. 형제 시작 순서 및 리소스 자식 순서

서비스 리소스는 다음과 같은 자식 리소스에 대해 자식 유형 속성 지정 여부에 따라 자식 리소스 시작 순서 및 중지 순서를 결정합니다:

- 자식 유형 속성 (*유형화된 (typed)* 자식 리소스)을 지정 – 서비스 리소스가 자식 리소스에 대해 자식 유형 속성을 지정할 경우, 자식 리소는 *유형화 (typed)*됩니다. 자식 유형 속성은 명시적으로 자식 리소스의 시작 및 중지 순서를 결정합니다.
- 자식 유형 속성 (*유형화되지 않은* 자식 리소스)을 *지정하지 않음* – 서비스 리소스가 자식 리소스에 대해 자식 유형 속성을 *지정하지 않은* 경우, 자식 리소스는 *유형화되지 않은* 상태입니다. 이 경우 서비스 리소스는 유형화되지 않은 자식 리소스의 시작 순서 및 중지 순서를 명시적으로 제어하지 않습니다. 하지만 유형화되지 않은 자식 리소스는 **/etc/cluster/cluster.conf**에 있는 해당 순서에 따라 시작 및 중지됩니다. 또한 유형화되지 않은 자식 리소스는 모든 유형화된 자식 리소스가 시작된 후에 시작하고 유형화된 자식 리소스 중 하나가 중지되기 전에 중지합니다.

참고

정의된 *자식 리소스 유형* 순서를 구현하기 위한 유일한 리소스는 서비스 리소스입니다.

유형화된 자식 리소스의 시작 및 중지 순서에 대한 자세한 내용은 C.2.1절. “유형화된 자식 리소스 시작 및 중지 순서 ”에서 참조하십시오. 유형화되지 않은 자식 리소스의 시작 및 중지 순서에 대한 자세한 내용은 C.2.2절. “유형화되 않은 자식 리소스의 시작 및 중지 순서 ” 에서 참조하십시오.

C.2.1. 유형화된 자식 리소스 시작 및 중지 순서

유형화된 자식 리소스의 경우, 자식 리소스에 대한 유형 속성은 1에서 100까지의 숫자를 사용하여 각 리소스 유형의 시작 순서 및 중지 순서를 정의합니다. 하나의 값은 시작 용으로 다른 하나의 값은 중지용으로

됩니다. 낮은 번호의 리소스 유형은 보다 먼저 시작 또는 중지합니다. 예를 들어, 표 C.1. "자식 리소스 유형 시작 및 중지 순서 "에서는 각 리소스 유형에 대한 시작 및 중지 값을 보여주고 있으며, 예 C.2. "리소스 시작 및 중지 값: 서비스 리소스 에이전트에서 발췌, **service.sh** "에서는 서비스 리소스 에이전트 **service.sh**에 나타나는 대로 시작 값과 중지 값을 보여주고 있습니다. 서비스 리소스의 경우 모든 LVM 자식 군이 먼저 시작하여 파일 시스템 자식 군, 모든 스크립트 자식 군의 순서로 시작합니다.

표 C.1. 자식 리소스 유형 시작 및 중지 순서

리소스	자식 유형	시작 순서 값	중지 순서 값
LVM	lvm	1	9
파일 시스템	fs	2	8
GFS2 파일 시스템	clusterfs	3	7
NFS 마운트	netfs	4	6
NFS 내보내기	nfsexport	5	5
NFS 클라이언트	nfsclient	6	4
IP 주소	ip	7	2
Samba	smb	8	3
스크립트	script	9	1

예 C.2. 리소스 시작 및 중지 값: 서비스 리소스 에이전트에서 발췌, service.sh

```
<special tag="rgmanager">
    <attributes root="1" maxinstances="1"/>
    <child type="lvm" start="1" stop="9"/>
    <child type="fs" start="2" stop="8"/>
    <child type="clusterfs" start="3" stop="7"/>
    <child type="netfs" start="4" stop="6"/>
    <child type="nfsexport" start="5" stop="5"/>
    <child type="nfsclient" start="6" stop="4"/>
    <child type="ip" start="7" stop="2"/>
    <child type="smb" start="8" stop="3"/>
    <child type="script" start="9" stop="1"/>
</special>
```

리소스 유형에서 순서는 클러스터 설정 파일 **/etc/cluster/cluster.conf**에 있는 대로 저장되어 있습니다. 예를 들어, 예 C.3. "리소스 유형의 순서 " 에서 유형화된 자식 리소스의 시작 및 중지 순서를 고려해 봅시다.

예 C.3. 리소스 유형의 순서

```
<service name="foo">
  <script name="1" .../>
  <lvm name="1" .../>
  <ip address="10.1.1.1" .../>
  <fs name="1" .../>
  <lvm name="2" .../>
</service>
```

C.2.1.1. 유형화된 자식 리소스 시작 순서

예 C.3. "리소스 유형의 순서 "에서 리소스는 다음의 순서로 시작됩니다:

1. **lvm:1** – 이는 LVM 리소스입니다. 모든 LVM 리소스가 먼저 시작됩니다. **lvm:1** (**<lvm name="1" .../>**)은 **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에서 나열된 첫번째 LVM 리소스이기 때문에 LVM 리소스에서 처음으로 시작되는 LVM 리소스입니다.

2. **lvm:2** – 이는 LVM 리소스입니다. 모든 LVM 리소스가 먼저 시작됩니다. **lvm:2** (**<lvm name="2" .../>**)는 **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에서 **lvm:1** 다음에 나열되어 있기 때문에 **lvm:1** 다음에 시작됩니다.

3. **fs:1** – 이는 파일 시스템 리소스입니다. *foo* 서비스에 다른 파일 시스템 리소스가 있을 경우, **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에 있는 목록에 나열된 순서대로 시작할 수 있습니다.

4. **ip:10.1.1.1** – 이는 IP 주소 리소스입니다. *foo* 서비스에 다른 IP 주소 리소스가 있는 경우, **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에 있는 목록에 나열된 순서대로 시작될 수 있습니다.

5. **script:1** – 이는 스크립트 리소스입니다. *foo* 서비스에 다른 스크립트 리소스가 있는 경우 **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에 있는 목록에 나열된 순서대로 시작될 수 있습니다.

C.2.1.2. 유형화된 자식 리소스 중지 순서

예 C.3. "리소스 유형의 순서 "에서 리소스는 다음의 순서로 중지됩니다:

1. **script:1** – 이는 스크립트 리소스입니다. *foo* 서비스에 다른 스크립트 리소스가 있는 경우, **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에 나열된 순서에서 역순으로 이를 중지할 수 있습니다.

2. **ip:10.1.1.1** – 이는 IP 주소 리소스입니다. *foo* 서비스에 다른 IP 주소 리소스가 있는 경우, **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에 나열된 순서에서 역순으로 이를 중지할 수 있습니다.

3. **fs:1** – 이는 파일 시스템 리소스입니다. *foo* 서비스에 다른 파일 시스템 리소스가 있을 경우, **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에 나열된 순서에서 역순으로 이를 중지할 수 있습니다.

4. **lvm:2** – 이는 LVM 리소스입니다. 모든 LVM 리소스는 마지막으로 중지됩니다. **lvm:2** (**<lvm name="2" .../>**)는 **lvm:1** 이전에 중지합니다. 리소스 유형 그룹에 있는 리소스는 **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에 나열된 순서에서 역순으로 중지됩니다.

5. **lvm:1** – 이는 LVM 리소스입니다. 모든 LVM 리소느는 마지막으로 중지됩니다. **lvm:1** (**<lvm name="1" .../>**)은 **lvm:2** 이후에 중지됩니다. 리소스 유형의 그룹에 있는 리소스는 **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에 나열된 순서에서 역순으로 중지됩니다.

C.2.2. 유형화되 않은 자식 리소스의 시작 및 중지 순서

형식화되지 않은 자식 리소스는 추가적 고려 사항이 필요합니다. 형식화되지 않은 자식 리소스의 경우 시작 순서 및 중지 순서는 서비스 리소스에 의해 명시적으로 지정되지 않습니다 대신 시작 순서 및 중지 순서는 **/etc/cluster/cluster.conf**에 있는 자식 리소스의 순서에 따라 결정됩니다. 또한 형식화되지 않은 자식 리소스는 모든 유형의 자식 리소스 이후에 시작되고 모든 유형의 자식 리소스 이전에 중지됩니다.

예를 들어, 예 C.4. "서비스에서 유형화되지 않은/유형화된 자식 리소스 " 에서 유형화되지 않은 자식 리소스의 시작 및 중지 순서를 고려해 봅시다.

예 C.4. 서비스에서 유형화되지 않은/유형화된 자식 리소스

```
<service name="foo">
  <script name="1" .../>
  <nontypedresource name="foo"/>
  <lvm name="1" .../>
  <nontypedresourcetwo name="bar"/>
  <ip address="10.1.1.1" .../>
  <fs name="1" .../>
  <lvm name="2" .../>
</service>
```

C.2.2.1. 유형화되지 않은 자식 리소스 시작 순서

예 C.4. "서비스에서 유형화되지 않은/유형화된 자식 리소스 " 에서 자식 리소스는 다음의 순서로 시작됩니다:

1. **lvm:1** – 이는 LVM 리소스입니다. 모든 LVM 리소스가 먼저 시작됩니다. **lvm:1** (**<lvm name="1" .../>**)은 **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에서 나열된 첫번째 LVM 리소스이기 때문에 LVM 리소스에서 처음으로 시작되는 LVM 리소스입니다.

2. **lvm:2** – 이는 LVM 리소스입니다. 모든 LVM 리소스가 먼저 시작됩니다. **lvm:2** (**<lvm name="2" .../>**)는 **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에서 **lvm:1** 다음에 나열되어 있기 때문에 **lvm:1** 다음에 시작됩니다.

3. **fs:1** – 이는 파일 시스템 리소스입니다. *foo* 서비스에 다른 파일 시스템 리소스가 있을 경우, **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에 있는 목록에 나열된 순서대로 시작할 수 있습니다.

4. **ip:10.1.1.1** – 이는 IP 주소 리소스입니다. *foo* 서비스에 다른 IP 주소 리소스가 있는 경우, **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에 있는 목록에 나열된 순서대로 시작될 수 있습니다.

5. **script:1** – 이는 스크립트 리소스입니다. *foo* 서비스에 다른 스크립트 리소스가 있는 경우 **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에 있는 목록에 나열된 순서대로 시작될 수 있습니다.

6. **nontypedresource:foo** – 이는 유형화되지 않은 리소스입니다. 이는 유형화되지 않은 리소스

이기 때문에 유형화된 리소스가 시작된 후에 시작합니다. 또한 서비스 리소스에서 순서는 다른 유형화되지 않은 리소스 **nontypedresourcetwo:bar** 전으로 되기 때문에 **nontypedresourcetwo:bar** 이전에 시작합니다. (유형화되지 않은 리소는 서비스 리소스에 나타나는 순서대로 시작됩니다.)

7. **nontypedresourcetwo:bar** – 이는 유형화되지 않은 리소스입니다. 이는 유형화되지 않은 리소스이기 때문에 유형화된 리소스가 시작된 후에 시작합니다. 또한 서비스 리소스에서 순서는 다른 유형화되지 않은 리소스 **nontypedresource:foo** 이후로 되기 때문에 **nontypedresource:foo** 이후에 시작됩니다. (유형화되지 않은 리소스는 서비스 리소스에 나타나는 순서로 시작됩니다.)

C.2.2.2. 유형화되지 않은 자식 리소스 중지 순서

예 C.4. “서비스에서 유형화되지 않은/유형화된 자식 리소스 ” 에서 자식 리소스는 다음의 순서로 중지됩니다:

1. **nontypedresourcetwo:bar** – 이는 유형화되지 않은 리소스입니다. 유형화되지 않은 리소스이기 때문에 유형화된 리소스가 중지되기 전에 중지합니다. 또한 서비스 리소스에서 순서는 다른 유형화되지 않은 리소스 **nontypedresource:foo** 이후가 되기 때문에 **nontypedresource:foo** 이전에 중지됩니다. (유형화되지 않은 리소스는 서비스 리소스에 나타나는 순서의 역순으로 중지됩니다.)

2. **nontypedresource:foo** – 이는 유형화되지 않은 리소스입니다. 유형화되지 않은 리소스이기 때문에, 유형화된 리소스가 중지되기 전에 중지합니다. 또한 서비스 리소스에서 순서는 다른 유형화되지 않은 리소스 **nontypedresourcetwo:bar** 전에 있기 때문에 **nontypedresourcetwo:bar** 이후에 중지됩니다. (유형화되지 않은 리소스는 서비스 리소스에 나타나는 순서의 역순으로 중지됩니다.)

3. **script:1** – 이는 스크립트 리소스입니다. *foo* 서비스에 다른 스크립트 리소스가 있는 경우, **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에 나열된 순서에서 역순으로 이를 중지할 수 있습니다.

4. **ip:10.1.1.1** – 이는 IP 주소 리소스입니다. *foo* 서비스에 다른 IP 주소 리소스가 있는 경우, **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에 나열된 순서에서 역순으로 이를 중지할 수 있습니다.

5. **fs:1** – 이는 파일 시스템 리소스입니다. *foo* 서비스에 다른 파일 시스템 리소스가 있을 경우, **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에 나열된 순서에서 역순으로 이를 중지할 수 있습니다.

6. **lvm:2** – 이는 LVM 리소스입니다. 모든 LVM 리소스는 마지막으로 중지됩니다. **lvm:2** (**<lvm name="2" .../>**)는 **lvm:1** 이전에 중지합니다. 리소스 유형 그룹에 있는 리소스는 **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에 나열된 순서에서 역순으로 중지됩니다.

7. **lvm:1** – 이는 LVM 리소스입니다. 모든 LVM 리소느는 마지막으로 중지됩니다. **lvm:1** (**<lvm name="1" .../>**)은 **lvm:2** 이후에 중지됩니다. 리소스 유형의 그룹에 있는 리소스는 **/etc/cluster/cluster.conf**의 *foo* 서비스 부분에 나열된 순서에서 역순으로 중지됩니다.

C.3. 상속, <리소스> 블록 및 리소스 재사용

일부 리소스는 부모 리소스에서 값을 상속하여 혜택을 얻습니다. 이는 일반적으로 NFS 서비스의 사례에서 볼 수 있습니다. 예 C.5. “리소스 재사용과 상속을 위한 NFS 서비스 설정 ” 에서는 리소스 재사용 및 상속을 위해 설정되는 전형적인 NFS 서비스 설정을 보여주고 있습니다.

예 C.5. 리소스 재사용과 상속을 위한 NFS 서비스 설정

```
    <resources>
        <nfsclient name="bob" target="bob.example.com"
options="rw,no_root_squash"/>
        <nfsclient name="jim" target="jim.example.com"
options="rw,no_root_squash"/>
        <nfsexport name="exports"/>
    </resources>
    <service name="foo">
        <fs name="1" mountpoint="/mnt/foo" device="/dev/sdb1"
fsid="12344">
            <nfsexport ref="exports">  <!-- nfsexport's path and fsid
attributes
                                           are inherited from the
mountpoint &
                                           fsid attribute of the
parent fs
                                           resource -->
                <nfsclient ref="bob"/> <!-- nfsclient's path is
inherited from the
                                           mountpoint and the fsid
is added to the
                                           options string during
export -->
                <nfsclient ref="jim"/>
            </nfsexport>
        </fs>
        <fs name="2" mountpoint="/mnt/bar" device="/dev/sdb2"
fsid="12345">
            <nfsexport ref="exports">
                <nfsclient ref="bob"/> <!-- Because all of the critical
data for this
                                           resource is either
defined in the
                                           resources block or
inherited, we can
                                           reference it again! -->
                <nfsclient ref="jim"/>
            </nfsexport>
        </fs>
        <ip address="10.2.13.20"/>
    </service>
```

서비스가 평면적일 경우 (즉, 부모/자식 관계가 없는 경우), 다음과 같이 설정되어야 합니다:

- 서비스는 네 개의 **nfsclient** 리소스를 필요로 합니다 – 파일 시스템 마다 1 개 (파일 시스템에서 총 2개), 그리고 대상 시스템 마다 1개 (대상 시스템에서 총 2 개).
- 서비스는 각 **nfsclient** 마다 내보내기 경로 및 파일 시스템 ID를 지정해야 하며, 이는 설정에서 오류를 일으킬 가능성이 있습니다.

예 C.5. "리소스 재사용과 상속을 위한 NFS 서비스 설정 " 에서는 NFS 클라이언트 리소스 *nfsclient:bob* 및 *nfsclient:jim*이 한 번 정의되고 있습니다. 마찬가지로 NFS 내보내기 리소스 *nfsexport:exports*는 한 번만 정의되고 있습니다. 리소스에 필요한 모든 속성은 부모 리소스에서 상속됩니다. 상속된 속성은 동적 (그리고 서로 충돌하지 않음)이기 때문에 이러한 리소스를 다시 사용 가능합니다 – 이것이 리소스 블록에서 정의되는 이유입니다. 일부 리소스를 여러 위치에서 설정하는 것은 실용적이지 않을 수 있습니다. 예를 들어, 여러 위치에서 파일 시스템 리소스를 설정하면 하나의 파일 시스템을 두 개의 노드에 마운트하는 결과를 일으킬 수 있어 문제가 발생하는 원인이 됩니다.

C.4. 장애 복구 및 독립적 하위 트리

대부분의 엔터프라이즈 환경에서 서비스 장애 복구를 위한 일반적인 작업은 서비스의 구성 요소 중 하나에 문제가 발생한 경우 전체 서비스를 다시 시작하는 것입니다. 예를 들어 예 C.6. "*foo* 서비스의 정상적인 장애 복구 "에서는 이 서비스에 정의된 스크립트 중 하나가 실패한 경우, 정상적인 절차로 서비스를 다시 시작 (또는 서비스 복구 정책에 따라 재배치 또는 비활성화)합니다. 하지만 일부 경우 서비스의 특정 부분이 중요하지 않다고 간주될 수 있습니다. 이는 일반적인 복구 작업을 시도하기 전 서비스의 문제가 있는 부분 만을 다시 시작해야 할 수 있습니다. 이를 위해 ***__independent_subtree*** 속성을 사용할 수 있습니다. 예를 들어 예 C.7. "***__independent_subtree*** 속성을 사용하여 *foo* 서비스 장애 복구 " 에서 ***__independent_subtree*** 속성은 다음과 같은 작업을 위해 사용되고 있습니다:

- script:script_one가 실패할 경우, script:script_one, script:script_two, script:script_three를 다시 시작합니다.
- script:script_two가 실패할 경우, script:script_two 만을 다시 시작합니다.
- script:script_three가 실패할 경우, script:script_one, script:script_two, script:script_three를 다시 시작합니다.
- script:script_four가 실패할 경우, 모든 서비스를 다시 시작합니다.

예 C.6. *foo* 서비스의 정상적인 장애 복구

```
<service name="foo">
      <script name="script_one" ...>
          <script name="script_two" .../>
      </script>
      <script name="script_three" .../>
</service>
```

예 C.7. ***__independent_subtree* 속성을 사용하여 *foo* 서비스 장애 복구**

```
<service name="foo">
      <script name="script_one" __independent_subtree="1" ...>
          <script name="script_two" __independent_subtree="1" .../>
          <script name="script_three" .../>
      </script>
      <script name="script_four" .../>
</service>
```

일부 경우 서비스의 한 구성 요소가 실패한 경우 전체 서비스를 비활성화하지 않고 서비스의 다른 구성 요소를 사용하는 다른 서비스에 영향을 미치지 않도록 해당 문제 구성 요소만 비활성화하고자 할 수 있습니다. Red Hat Enterprise Linux 6.1 릴리즈에서 독립 하위 트리를 중요하지 않은 것으로 지정하는 ***__independent_subtree="2"*** 속성을 사용하여 이를 수행할 수 있습니다.

참고

단일 참조 리소스에서만 중요하지 않은 플래그를 사용합니다. 중요하지 않은 플래그는 리소스 트리의 모든 레벨에 있는 모든 리소스와 작동하지만 서비스와 가상 머신을 정의하고 있을 때에는 최상위에서 사용할 수 없습니다.

Red Hat Enterprise Linux 6.1 릴리스에서 독립 하위 트리에 대해 리소스 트리에서 노드 당 최대 다시 시작 횟수와 다시 시작 만료 시간을 설정할 수 있습니다. 이러한 임계값을 설정하려면 다음 속성을 사용합니다:

- **__max_restarts**는 다시 시작을 포기하기 전 까지 허용되는 최대 다시 시작 횟수를 설정합니다.
- **__restart_expire_time**은 다시 시작이 더이상 시도되지 않는 시간을 (초 단위) 설정합니다.

C.5. 서비스 및 리소스 순서 디버그 및 테스트

rg_test 유틸리티를 사용하여 리소스 순서와 서비스를 디버깅 및 테스트할 수 있습니다. **rg_test**는 명령행 유틸리티로 쉘 또는 터미널에서 실행할 수 있는 **rgmanager** 패키지에 의해 제공됩니다 (이는 **Conga**에서는 사용할 수 없습니다). 표 C.2. "**rg_test** 유틸리티 요약"에서는 **rg_test** 유틸리티의 동작과 구문을 요약하고 있습니다.

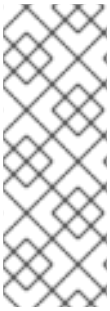
표 C.2. rg_test 유틸리티 요약

동작	구문
rg_test를 이해하는 리소스 규칙 보기	**rg_test rules**
오류 또는 중복 리소스 에이전트에 대한 설정 (및 /usr/share/cluster) 테스트	**rg_test test /etc/cluster/cluster.conf**
서비스 시작 및 중지 순서 표시	시작 순서 표시: **rg_test noop /etc/cluster/cluster.conf start service *servicename*** 중지 순서 표시: **rg_test noop /etc/cluster/cluster.conf stop service *servicename***

동작	구문
서비스를 명시적으로 시작하거나 중지	 **중요** 이는 하나의 노드에서만 실행하고 항상 rgmanager에서 먼저 서비스를 비활성화 합니다. 서비스 시작: **rg_test test /etc/cluster/cluster.conf start service *servicename*** 서비스 중지: **rg_test test /etc/cluster/cluster.conf stop service *servicename***
2 개의 cluster.conf 파일간에 리소스 트리 델타를 계산하고 표시	**rg_test delta *cluster.conf file 1* *cluster.conf file 2*** 예: **rg_test delta /etc/cluster/cluster.conf.bak /etc/cluster/cluster.conf**

부록 D. 클러스터 서비스 리소스 검사 및 페일 오버 시간 제한

이 부록에서는 **rgmanager**가 클러스터 리소스의 상태를 모니터링하는 방법과 상태 검사 간격을 수정하는 방법에 대해 설명합니다. 또한 동작에 대한 시간 제한으로 인해 서비스가 실패하는 것을 나타내는 **__enforce_timeouts** 서비스 매개 변수에 대해 설명합니다.



참고

이 부록에 있는 내용을 충분히 이해하려면 리소스 에이전트와 클러스터 설정 파일 **/etc/cluster/cluster.conf**에 대해 상세히 알고 있어야 합니다. **cluster.conf** 요소 및 속성에 대한 전체 목록 및 설명은 **/usr/share/cluster/cluster.rng**에 있는 클러스터 스키마와 **/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html** (예: **/usr/share/doc/cman-3.0.12/cluster_conf.html**)에 있는 주석 스키마에서 참조하십시오.

D.1. 리소스 상태 검사 간격 수정

rgmanager는 전체 서비스가 아닌 개별 리소스 상태를 확인합니다. 10 초 마다 rgmanager는 리소스 트리를 검색하여 "상태 검사" 간격을 통과한 리소스를 찾습니다.

각 리소스 에이전트는 정지적 상태 검사 사이의 기간을 지정합니다. 각 리소스는 특별한 **<action>** 태그를 사용하여 **cluster.conf** 파일에 명시적으로 덮어쓰기되지 않는 한 이러한 시간 제한 값을 사용합니다.

<cman two_node="1" expected_votes="1"/>

이 태그는 **cluster.conf** 파일에 있는 리소스 자체의 특별한 자식입니다. 예를 들어 상태 검사 간격을 덮어쓰기하고자 하는 파일 시스템 리소스가 있을 경우 다음과 같이 **cluster.conf** 파일에 파일 시스템 리소스를 지정할 수 있습니다:

```
<fs name="test" device="/dev/sdb3">
  <action name="status" depth="*" interval="10" />
  <nfsexport...>
  </nfsexport>
</fs>
```

일부 에이전트는 검사의 여러 "강도"를 제공합니다. 예를 들어 일반적인 파일 시스템 상태 검사 (강도 0)는 파일 시스템이 올바를 위치에 마운트되어 있는지를 확인합니다. 보다 집중적인 검사는 강도 10으로 파일 시스템에서 파일을 읽을 수 있는지에 대한 여부를 확인합니다. 강도 20의 상태 검사는 파일 시스템에 쓰기를 할 수 있는지를 확인합니다. 여기의 예에서 **depth**는 ***로 설정되어 있습니다. 이는 이 값이 모든 강도에 대해 사용되어야 함을 나타냅니다. 결과적으로 10 초 마다 리소스 에이전트에 의해 가장 높게 지정된 강도 (이 경우 20)로 **test** 파일 시스템을 확인합니다.

D.2. 리소스 시간 제한 강제

리소스를 시작, 중지, 장애 조치하기 위한 시간 제한은 없습니다. 일부 리소스는 시작 또는 중지에 불확실하게 오래 걸립니다. 불행히도 중지 오류 (시간 제한 포함)가 발생하면 서비스가 작동 불가능 (실패 상태)하게 됩니다. 필요에 따라 **cluster.conf** 파일에 있는 참조에 **__enforce_timeouts="1"**을 추가하여 개별 서비스의 각 리소스에 시간 제한을 강제로 활성화할 수 있습니다.

다음 예에서는 **netfs** 리소스의 **__enforce_timeouts** 속성 구성으로 설정된 클러스터 서비스를 보여 줍니다. 이러한 속성이 설정된 상태에서 복구 과정에서 NFS 파일 시스템을 마운트 해제하는데 30 초 이상이 걸리는 경우 동작은 시간 초과되어 서비스가 실패 상태로 됩니다.

```
</screen>
<rm>
  <failoverdomains/>
  <resources>
    <netfs export="/nfstest" force_unmount="1" fstype="nfs"
host="10.65.48.65"
           mountpoint="/data/nfstest" name="nfstest_data"
options="rw,sync,soft"/>
  </resources>
  <service autostart="1" exclusive="0" name="nfs_client_test"
recovery="relocate">
    <netfs ref="nfstest_data" __enforce_timeouts="1"/>
  </service>
</rm>
```

부록 E. 명령행 도구 요약

표 E.1. "명령행 도구 요약"에서는 고가용성 추가 기능 설정 및 관리를 위한 우선적인 명령행 도구에 대해 요약하고 있습니다. 명령 및 변수에 대한 자세한 내용은 각각의 명령행 도구 man 페이지를 참조하십시오.

표 E.1. 명령행 도구 요약

명령행 도구	사용 대상	목적
ccs_config_dump – 클러스터 설정 덤프 도구	클러스터 인프라	**ccs_config_dump**는 실행중인 설정의 XML 출력을 생성합니다. 일부 하부 시스템은 설정에 기본 정보를 저장하거나 설정하기 때문에 때때로 실행 중인 설정은 파일에 저장된 설정과 다를 수 있습니다. 일반적으로 이러한 값은 설정 디스크 버전에 존재하지 않지만 클러스터가 런타임에 정상적으로 작동하기 위해 필요합니다. 이 도구에 대한 자세한 내용은 ccs_config_dump(8) man 페이지를 참조하십시오.
ccs_config_validate – 클러스터 설정 유효성 도구	클러스터 인프라	**ccs_config_validate**는 각 노드의 **/usr/share/cluster/cluster.rng**에 위치한 스키마 **cluster.rng**에 대해 **cluster.conf**의 유효성을 검사합니다. 이 도구에 대한 자세한 내용은 ccs_config_validate(8) man 페이지를 참조하십시오.
clustat – 클러스터 상태 유틸리티	고가용성 서비스 관리 구성 요소	**clustat** 명령은 클러스터의 상태를 표시합니다. 이는 멤버쉽 정보, 쿼터 표시, 모든 설정된 사용자 서비스의 상태를 나타냅니다. 이 도구에 대한 자세한 내용은 clustat(8) man 페이지를 참조하십시오.
clusvcadm – 클러스터 사용자 서비스 관리 유틸리티	고가용성 서비스 관리 구성 요소	**clusvcadm** 명령을 사용하여 클러스터에서 고가용성 서비스를 활성화, 비활성화, 재배치 또는 다시 시작할 수 있습니다. 이 도구에 대한 자세한 내용은 clusvcadm(8) man 페이지를 참조하십시오.
cman_tool – 클러스터 관리 도구	클러스터 인프라	**cman_tool**은 CMAN 클러스터 관리자를 관리하는 프로그램입니다. 이는 클러스터에 가입, 탈퇴, 노드 종료할 수 있게하거나 또는 클러스터에 있는 노드의 예상되는 쿼터 수를 변경할 수 있게 합니다. 이 도구에 대한 자세한 내용은 cman_tool(8) man 페이지를 참조하십시오.
fence_tool – 차단 도구	클러스터 인프라	**fence_tool**은 차단 도메인을 가입 및 탈퇴에 사용되는 프로그램입니다. 이 도구에 대한 자세한 내용은 fence_tool(8) man 페이지를 참조하십시오.

부록 F. 고가용성 LVM (HA-LVM)

Red Hat 고가용성 애드온은 장애 조치 설정의 고가용성 LVM 볼륨 (HA-LVM)을 지원합니다. 이는 Clustered Logical Volume Manager (CLVM)에 의해 활성화된 active/active 설정과 다릅니다. CLVM은 컴퓨터의 클러스터가 공유 스토리지를 관리할 수 있게 하는 LVM의 클러스터링 확장 모음입니다.

CLVM 또는 HA-LVM은 배포된 애플리케이션이나 서비스의 필요에 따라 사용하는 것이 좋습니다.

- 애플리케이션이 동시에 여러 컴퓨터에서 실행하도록 조정된 경우 CLVM을 사용해야 합니다. 특히 하나 이상의 클러스터 노드가 활성 노드 사이에서 공유되는 스토리지로의 액세스를 필요로할 경우, CLVM을 사용해야 합니다. CLVM은 논리 볼륨이 설정되는 동안 물리 스토리지로의 액세스를 잠금하여 사용자가 공유 스토리지에 논리 볼륨을 설정하게 하며, 공유 스토리지를 관리하기 위해 클러스터화된 잠금 서비스를 사용합니다. CLVM 및 LVM 설정의 일반적인 내용은 *LVM 관리*에서 참조하십시오.

- 스토리지에 액세스하는 단일 노드만 언제든지 한번에 활성화되는 active/passive (장애 조치) 설정에서 애플리케이션을 최적으로 실행하려면 HA-LVM (High Availability Logical Volume Management) 에이전트를 사용해야 합니다.

대부분의 애플리케이션은 다른 인스턴스와 동시에 실행하도록 설계 또는 최적화되어 있지 않기 때문에 active/passive 설정에서 보다 더 제대로 실행됩니다. 클러스트된 논리 볼륨에서 클러스터를 인식하지 않는 애플리케이션을 실행하도록 선택하면 논리 볼륨이 미러될 경우 성능이 저하될 수 있습니다. 이는 이러한 인스턴스에 논리 불륨 자체에 해당하는 클러스터 통신 오버헤드가 있기 때문입니다. 클러스터 인식 애플리케이션은 클러스터 파일 시스템 및 클러스터 인식 논리 불륨에 의해 생기는 성능 저하를 넘는 높은 성능을 제공할 수 있어야 합니다. 이는 일부 애플리케이션 및 워크로드의 경우 다른 경우 보다 쉽게 수행할 수 있습니다. 클러스터의 요구 사항을 지정하고 active/active 클러스터를 최적화하기 위한 별도의 작업이 도움이 되는지는 두 LVM 변형 사이에서 선택하는 방법에 따라 다릅니다. 대부분의 사용자는 HA-LVM을 사용하여 최적의 HA 결과를 얻을 수 있습니다.

HA-LVM 및 CLVM은 LVM 메타데이터와 논리 볼륨의 손상을 방지한다는 점에서 유사합니다. 그렇지 않을 경우 여러 시스템을 중복 변경할 수 있는 경우에 발생합니다. HA-LVM은 논리 불륨 단독으로 활성화할 수 있는 즉 한 번에 하나의 컴퓨터에서만 활성화되는 제한을 부과합니다. 이는 로컬 (비 클러스터화) 구현된 스토리지 드라이버만 사용되는 것을 의미합니다. 이러한 방법으로 조정된 클러스터 오버헤드를 방지함으로써 성능이 향상됩니다. CLVM은 이러한 제한을 부과하지 않습니다. 즉 사용자는 클러스터의 모든 컴퓨터에 있는 논리 볼륨을 자유롭게 활성화할 수 있습니다. 이렇게 하면 클러스트 인식 스토리지 드라이버 사용을 강제하게 되므로 클러스터 인식 파일 시스템 및 애플리케이션이 상단에 배치됩니다.

HA-LVM은 단독으로 논리 볼륨을 활성화하는 명령을 수행하는 두 가지 방법 중 하나를 사용하도록 설정될 수 있습니다.

- 권장되는 방법은 CLVM을 사용하지만 단독으로 논리 불륨만을 활성화하는 것입니다. 이로 인한 장점은 손쉬운 설정 및 관리적 실수 (사용 중인 논리 볼륨을 삭제하는 등)를 방지할 수 있다는 것입니다. CLVM을 사용하려면 `clvmd` 데몬을 포함하여 고가용성 애드온 및 장애 복구형 스토리지 애드온 소프트웨어를 실행하고 있어야 합니다.

  이 방법을 사용하여 HA-LVM을 설정하는 절차는 F.1절. “CLVM으로 HA-LVM 페일오버 설정 (권장)”에 설명되어 있습니다.

- 두 번째 방법은 로컬 컴퓨터 잠금 및 LVM 태그를 사용하는 것입니다. 이 방법의 장점은 LVM 클러스터 패키지가 필요하지 않다는 것입니다. 하지만 이를 설정하는데는 관련된 더 많은 단계가 있고 관리자가 비활성 클러스터 노드에서 논리 볼륨을 실수로 삭제하는것을 방지할 수 없습니다. 이러한 방법을 사용하여 HA-LVM을 설정하는 절차는 F.2절. “태그로 HA-LVM 페일 오버 설정 ” 에서 설명하고 있습니다.

F.1. CLVM으로 HA-LVM 페일오버 설정 (권장)

(권장되는 CLVM 변형을 사용하여) HA-LVM 페일오버를 구성하려면 다음 단계를 수행합니다:

1. 시스템이 CLVM을 지원하도록 설정되어 있는지 확인합니다. 다음과 같은 요구 사항이 필요합니다:
   - CLVM 논리 볼륨이 미러되면 **cmirror** 패키지를 포함하여 고가용성 애드온 및 장애 복구형 스토리지 애드온이 설치됩니다.
   - **/etc/lvm/lvm.conf** 파일의 글로벌 섹션에 있는 **locking_type** 매개 변수는 "3"으로 설정되어 있습니다.
   - **clvmd** 데몬을 포함한 고가용성 애드온 및 장애 복구형 애드온 소프트웨어를 실행하고 있어야 합니다. CLVM 미러링의 경우 **cmirrord** 서비스도 시작해야 합니다.
2. 다음 예제와 같이 표준 LVM 및 파일 시스템 명령을 사용하여 논리 볼륨 및 파일 시스템을 생성합니다.

```
# pvcreate /dev/sd[cde]1

# vgcreate -cy shared_vg /dev/sd[cde]1

# lvcreate -L 10G -n ha_lv shared_vg

# mkfs.ext4 /dev/shared_vg/ha_lv

# lvchange -an shared_vg/ha_lv
```

   LVM 논리 볼륨 생성에 대한 보다 자세한 내용은 *LVM 관리*에서 참조하십시오.

3. **/etc/cluster/cluster.conf** 파일을 편집하여 새로 생성된 논리 볼륨을 서비스 중 하나에 있는 리소스로 포함합니다. 다른 방법으로 **Conga** 또는 **ccs** 명령을 사용하여 클러스터에 대해 LVM 및 파일 시스템 리소스를 설정할 수 있습니다. 다음은 클러스터 리소스로 CLVM 논리 볼륨을 설정하는 **/etc/cluster/cluster.conf** 파일의 리소스 관리자 섹션에 대한 예입니다:

```
<rm>
   <failoverdomains>
       <failoverdomain name="FD" ordered="1" restricted="0">
          <failoverdomainnode name="neo-01" priority="1"/>
          <failoverdomainnode name="neo-02" priority="2"/>
       </failoverdomain>
   </failoverdomains>
   <resources>
       <lvm name="lvm" vg_name="shared_vg" lv_name="ha-lv"/>
       <fs name="FS" device="/dev/shared_vg/ha-lv" force_fsck="0"
force_unmount="1" fsid="64050" fstype="ext4" mountpoint="/mnt"
options="" self_fence="0"/>
   </resources>
   <service autostart="1" domain="FD" name="serv"
recovery="relocate">
       <lvm ref="lvm"/>
       <fs ref="FS"/>
   </service>
</rm>
```

F.2. 태그로 HA-LVM 페일 오버 설정

/etc/lvm/lvm.conf 파일에 있는 태그를 사용하여 HA-LVM 페일오버를 설정하려면 다음과 같은 단계를 수행합니다:

1. **/etc/lvm/lvm.conf** 파일의 글로벌 섹션에 있는 **locking_type** 매개 변수가 '1'로 설정되어 있는지 확인합니다.
2. 다음 예제와 같이 표준 LVM 및 파일 시스템 명령을 사용하여 논리 볼륨 및 파일 시스템을 생성합니다.

```
# pvcreate /dev/sd[cde]1

# vgcreate shared_vg /dev/sd[cde]1

# lvcreate -L 10G -n ha_lv shared_vg

# mkfs.ext4 /dev/shared_vg/ha_lv
```

   LVM 논리 볼륨 생성에 대한 보다 자세한 내용은 *LVM 관리*에서 참조하십시오.

3. **/etc/cluster/cluster.conf** 파일을 편집하여 새로 생성된 논리 볼륨을 서비스 중 하나에 있는 리소스로 포함합니다. 다른 방법으로 **Conga** 또는 **ccs** 명령을 사용하여 클러스터에 대해 LVM 및 파일 시스템 리소스를 설정할 수 있습니다. 다음은 클러스터 리소스로 CLVM 논리 볼륨을 설정하는 **/etc/cluster/cluster.conf** 파일의 리소스 관리자 섹션에 대한 예입니다:

```
<rm>
   <failoverdomains>
       <failoverdomain name="FD" ordered="1" restricted="0">
          <failoverdomainnode name="neo-01" priority="1"/>
          <failoverdomainnode name="neo-02" priority="2"/>
       </failoverdomain>
   </failoverdomains>
   <resources>
       <lvm name="lvm" vg_name="shared_vg" lv_name="ha_lv"/>
       <fs name="FS" device="/dev/shared_vg/ha_lv" force_fsck="0"
force_unmount="1" fsid="64050" fstype="ext4" mountpoint="/mnt"
options="" self_fence="0"/>
   </resources>
   <service autostart="1" domain="FD" name="serv"
recovery="relocate">
       <lvm ref="lvm"/>
       <fs ref="FS"/>
   </service>
</rm>
```

참고

볼륨 그룹에 여러 논리 불륨이 있을 경우 **lvm** 리소스에 있는 논리 볼륨 이름 **(lv_name)**은 비워 두두거나 지정하지 않도록 합니다. 또한 HA-LVM 설정에서 볼륨 그룹은 단일 서비스만 사용할 수 있다는 점에 유의하십시오.

4. **/etc/lvm/lvm.conf** 파일에 있는 **volume_list** 필드를 편집합니다. **/etc/cluster/cluster.conf** 파일에 나열되어 있듯이 root 볼륨 그룹 이름 및 호스트 이름은 **@**을 앞에 붙여 입력합니다. 여기에 포함된 호스트 이름은 원격 호스트 이름이 아닌 **lvm.conf** 파일을 편집하는 시스템입니다. 이 문자열은 **cluster.conf** 파일에 있는 노드 이름과 일치 *해야 함에* 유의하십시오. 다음은 **/etc/lvm/lvm.conf** 파일의 예제 항목입니다:

```
volume_list = [ "VolGroup00", "@neo-01" ]
```

이 태그는 공유 VG 또는 LV를 활성화하는데 사용됩니다. HA-LVM을 사용하여 공유되는 볼륨 그룹 이름을 포함하지 *않도록 합니다*.

5. 모든 클러스터 노드에 있는 **initrd** 장치를 업데이트합니다:

```
# dracut -H -f /boot/initramfs-$(uname -r).img $(uname -r)
```

6. 모든 노드를 재부팅하여 올바른 **initrd** 장치가 사용 중인지 확인합니다.

부록 G. 고친 과정

고침 5.0-25.1.400 **2013-10-31** **Rüdiger Landmann**
Rebuild with publican 4.0.0

고침 5.0-25.1 **Thu Apr 18 2013** **Chester Cheng**
XML 소스 5.0-25 버전과 번역 파일을 동기화

고침 5.0-25 **Mon Feb 18 2013** **Steven Levine**
6.4 GA 릴리즈 버전

고침 5.0-23 **Wed Jan 30 2013** **Steven Levine**
문제 해결: 901641
iptables 규칙 수정 및 명확화

고침 5.0-22 **Tue Jan 29 2013** **Steven Levine**
문제 해결: 788636
`ccs` 명령을 통한 RRP 설정을 문서화

문제 해결: 789010
`cluster.conf` 파일에서 RRP 설정을 문서화

고침 5.0-20 **Fri Jan 18 2013** **Steven Levine**
문제 해결: 894097
VLAN 태그를 사용하지 않도록 한다는 조언을 삭제

문제 해결: 845365
본딩 모드 0과 2가 지원됨을 기재

고침 5.0-19 **Thu Jan 17 2013** **Steven Levine**
문제 해결: 896234
클러스터 노드 참조 용어를 명확히함

고침 5.0-16 **Mon Nov 26 2012** **Steven Levine**
6.4 베타 릴리즈 버전

고침 5.0-15 **Wed Nov 20 2012** **Steven Levine**

문제 해결: 838988
파일 시스템 리소스 에이전트의 nfsrestart 속성을 문서화

문제 해결: 843169
IBM iPDU 차단 에이전트를 문서화

문제 해결: 846121
Eaton Network Power Controller (SNMP 인터페이스) 차단 에이전트를 문서화

문제 해결: 856834
HP Bladesystem 차단 에이전트를 문서화

문제 해결: 865313
NFS 서버 리소스 에이전트를 문서화

문제 해결: 862281
ccs 명령을 이전 설정이 덮어쓰기할 것인지를 명확히함

문제 해결: 846205
igmp 구성 요소의 **iptables** 방화벽 필터링을 문서화

문제 해결: 857172
luci에서 사용자를 제거하는 기능을 문서화

문제 해결: 857165
IPMI 차단 에이전트의 권한 수준 매개 변수를 문서화

문제 해결: 840912
리소스 매개 변수 테이블의 형식 문제 해결

문제 해결: 849240, 870292
설치 절차를 명확히함

문제 해결: 871165
IP 주소 리소스 에이전트에 있는 IP 주소 매개 변수 설명을 명확히함

문제 해결: 845333, 869039, 856681
약간의 오타 수정 및 기술적으로 약간 모호한 부분을 명확히함

고침 5.0-12 **Thu Nov 1 2012** **Steven Levine**
새로 지원되는 차단 에이전트 추가

고침 5.0-7 **Thu Oct 25 2012** **Steven Levine**
의미를 재정의하는 섹션을 추가

고침 5.0-6 **Tue Oct 23 2012** **Steven Levine**
참여 후 대기 시간의 기본값을 수정

고침 5.0-4 **Tue Oct 16 2012** **Steven Levine**
NFS 서버 리소스에 대한 설명을 추가

고침 5.0-2 **Thu Oct 11 2012** **Steven Levine**
Conga 설명 업데이트

고침 5.0-1 **Mon Oct 8 2012** **Steven Levine**
ccs 의미론을 명확히함

고침 4.0-5 **Fri Jun 15 2012** **Steven Levine**

6.3 GA 릴리즈 버전

고침 4.0-4 **Tue Jun 12 2012** **Steven Levine**

문제 해결: #830148
luci의 예시 포트 번호의 일관성 확인

고침 4.0-3 **Tue May 21 2012** **Steven Levine**

문제 해결: #696897
차단 장치 매개 변수 및 리소스 매개 변수 테이블에 cluster.conf 매개 변수 정보를 추가

문제 해결: #811643
다른 컴퓨터에서 luci 데이터베이스를 복원하는 방법을 추가

고침 4.0-2 **Wed Apr 25 2012** **Steven Levine**

문제 해결: #815619
GFS2 파일 시스템과 함께 UDP Unicast 사용에 관한 경고를 삭제

고침 4.0-1 **Fri Mar 30 2012** **Steven Levine**

문제 해결: #771447, 800069, 800061
Red Hat Enterprise Linux 6.3 버전과 일관성을 유지하기 위해 luci 문서를 업데이트

문제 해결: #712393
RGManager의 애플리케이션 코어를 캡처하는 정보를 추가

문제 해결: #800074
`condor` 리소스 에이전트를 문서화

문제 해결: #757904
luci 설정 백업 및 복구를 문서화

문제 해결: #772374
클러스터에서 가상 머신 관리에 대한 부분을 추가

문제 해결: #712378
HA-LVM 설정에 대한 문서를 추가

문제 해결: #712400
디버그 옵션을 문서화

문제 해결: #751156
새 `fence_ipmilan` 매개 변수에 대해 문서화

문제 해결: #721373
클러스터 재시작에 필요한 설정 변경 사항을 문서화

고침 3.0-5 **Thu Dec 1 2011** **Steven Levine**

Red Hat Enterprise Linux 6.2 GA 릴리즈

문제 해결: #755849
monitor_link 매개 변수 예제를 수정

고침 3.0-4 **Mon Nov 7 2011** **Steven Levine**

문제 해결: #749857
RHEV-M REST API 차단 장치에 대한 문서를 추가

고침 3.0-3 **Fri Oct 21 2011** **Steven Levine**

문제 해결: #747181, #747182, #747184, #747185, #747186, #747187, #747188, #747189, #747190, #747192
Red Hat Enterprise Linux 6.2 QE 문서 검토시 발견된 오타 및 애매한 표현을 수정

고침 3.0-2 **Fri Oct 7 2011** **Steven Levine**

문제 해결: #743757
문제 해결 부분에서 지원되는 본딩 모드에 대한 참조를 수정

고침 3.0-1 **Wed Sep 28 2011** **Steven Levine**

Red Hat Enterprise Linux 6.2 베타 릴리즈 초기 버전

문제 해결: #739613
사용 가능한 차단 장치 및 서비스를 표시하기 위해 새 **ccs** 옵션 지원을 문서화

문제 해결: #707740
Conga 인터페이스 업데이트 및 Conga를 관리하기 위한 사용자 권한 설정 지원에 대해 문서화

문제 해결: #731856
/etc/sysconfig/luci 파일을 사용한 luci 설정 지원에 대해 문서화

문제 해결: #736134
UDPU 전송 지원을 문서화

문제 해결: #736143
클러스터 Samba에 대한 지원을 문서화

문제 해결: #617634
luci가 작동하는 IP 주소 만을 설정하는 방법을 문서화

문제 해결: #713259
fence_vmware_soap 에이전트에 대한 지원을 문서화

문제 해결: #721009
지원되는 필수 문서에 대한 링크를 제공

문제 해결: #717006
iptables 방화벽을 통한 멀티캐스트 트래픽을 허용하는 내용을 제공

문제 해결: #717008
클러스터 서비스 상태 확인 및 장애 복구 시간 제한에 내용을 제공

문제 해결: #711868
autostart에 대해 상세하게 설명

문제 해결: #728337
ccs 명령을 사용하여 **vm** 리소스를 추가하는 절차를 문서화

문제 해결: #725315, #733011, #733074, #733689
일부 오타 수정

고침 2.0-1 **Thu May 19 2011** **Steven Levine**

Red Hat Enterprise Linux 6.1 초기 개정

문제 해결: #671250
SNMP 트랩 지원을 문서화

문제 해결: #659753
ccs 명령을 문서화

문제 해결: #665055
업데이트된 디스플레이와 기능 지원을 반영하기 위해 Conga 문서 업데이트

문제 해결: #680294
ricci 에이전트의 액세스 암호의 필요성을 문서화

문제 해결: #687871
문제 해결에 대한 부분 추가

문제 해결: #673217
오타 수정

문제 해결: #675805
cluster.conf 스키마에 대한 참조를 HA 리소스 매개 변수 표에 추가

문제 해결: #672697
현재 지원되는 펜싱 장치가 포함된 차단 장치 매개 변수 표를 업데이트

문제 해결: #677994
fence_ilo 차단 에이전트 매개 변수에 대한 정보 수정

문제 해결: #629471
2 노드 클러스터에 있는 합의 값 설정에 대한 기술 노트 추가

문제 해결: #579585
Red Hat 고가용성 추가 기능 소프트웨어 업그레이드에 있는 내용 업데이트

문제 해결: #643216
문서 전체에 걸친 사소한 문제를 명료화

문제 해결: #643191
luci 문서에 대한 개선 및 수정 제공

문제 해결: #704539
가상 머신 리소스 매개 변수표 업데이트

고침 1.0-1 **Wed Nov 10 2010** **Paul Kennedy**
Red Hat Enterprise Linux 6 초기 릴리즈

색인

Symbols

A

B

C

T

V

W