Red Hat Enterprise Linux 6
마이그레이션 계획 가이드

Red Hat Enterprise Linux 6로 마이그레이션하기
엮음 6.1

Red Hat Inc.

Red Hat Enterprise Linux 6 마이그레이션 계획 가이드

Red Hat Enterprise Linux 6로 마이그레이션하기
엮음 6.1

Red Hat Inc.

초록

다음에서는 Red Hat Enterprise Linux 5를 실행하여 Red Hat Enterprise Linux 6로 시스템을 마이그레이션하는 방법에 대해 설명합니다.

차례

머리말

1. 문서화 규정

이 메뉴얼에서는 특정 단어 및 구문을 강조 표시하여 특정 정보 부분에 주의를 집중시키기 위해 문서화 규정을 사용하고 있습니다.

PDF 및 문서 편집에서 이 메뉴얼은 Liberation 글꼴 모음에 있는 서체를 사용합니다. 시스템에 Liberation 글꼴 모음이 설치되어 있을 경우 이는 HTML 편집에서도 사용되지만 설치되어 있지 않을 경우, 다른 동일한 서체로 나타나게 됩니다. 알림: Red Hat Enterprise Linux 5 및 이후 버전에는 기본값으로 Liberation 글꼴 모음이 들어 있습니다.

1.1. 표기 규정

네 가지 표기 규정을 사용하여 특정 단어 및 구문에 주의를 집중시킵니다. 이러한 규정 및 적용 방식은 다음과 같습니다.

고정폭 굵은체

쉘 명령, 파일 이름 및 경로를 포함한 시스템 입력을 강조하기 위해 사용됩니다. 키 및 키 조합을 강조하기 위해 사용되기도 합니다. 예:

> 현재 작업 중인 디렉토리에 있는 **my_next_bestselling_novel** 파일 내용을 확인하려면, 쉘 프롬프트에서 **cat my_next_bestselling_novel** 명령을 입력하고 **Enter** 키를 눌러 명령을 실행합니다.

위에서 파일 이름, 쉘 명령, 키 모두는 고정폭 굵은체로 나타나 있어 내용과 구별될 수 있습니다.

키 조합은 키 조합의 각 부분을 더하기(+) 기호로 연결하여 개별 키과 구별되게 할 수 있습니다. 예:

> **Enter** 키를 눌러 명령을 실행합니다.
>
> **Ctrl+Alt+F2**를 눌러 가상 터미널로 전환합니다.

첫 번째 예에서는 눌러야 하는 특정 키를 강조하고 있습니다. 두 번째 예에서는 키 조합 (동시에 눌러야 하는 세 개의 키 묶음)을 강조하고 있습니다.

소스 코드를 설명해야 할 경우, 문장에서 언급된 클래스 이름, 방식, 기능, 변수 이름 및 반환값은 위와 같이 **고정폭 굵은체**로 나타나게 됩니다. 예:

> 파일 관련 클래스에는 파일 시스템의 경우 **filesystem**, 파일의 경우 **file**, 디렉토리의 경우 **dir**가 포함됩니다. 각각의 클래스에는 자체의 권한 설정이 있습니다.

가변폭 굵은체

이는 프로그램 이름; 대화 상자 텍스트; 레이블된 버튼; 체크 박스 및 라디오 버튼 레이블; 메뉴 제목; 하부 메뉴 제목을 포함하여 시스템에 있는 단어 또는 구문을 나타냅니다. 예:

> 주 메뉴 바에서 **시스템 → 기본설정 → 마우스**를 선택하여 **마우스 기본 설정**을 시작합니다. **버튼** 탭에서, **왼손 잡이 마우스** 체크 상자를 선택하고 **닫기**를 클릭하여 주요 마우스 버튼을 왼쪽에서 오른쪽으로 전환합니다 (왼손 잡이일 경우 보다 적절하게 마우스 사용을 할 수 있게 함).
>
> **gedit** 파일에 특수 문자를 입력하려면 주 메뉴 바에서 **프로그램 → 보조 프로그램 → 문자 표**를 선택합니다. 그 다음으로, **문자 표** 메뉴 바에서 **찾기 → 찾기...**를 선택하고 **찾기** 필드에 문자를 입력하고 **다음**을 클릭합니다. 찾기한 문자가 **문자 표**에 강조 표시되어 나타납니다. 강

조 표시된 문자를 두 번 클릭하여 **복사할 텍스트** 필드에 나타나면 **복사** 버튼을 클릭합니다. 편집 중인 문서로 돌아가 **gedit** 메뉴 바에서 **편집 → 붙여넣기**를 선택합니다.

위의 내용에는 프로그램 이름, 다양한 시스템 메뉴 이름 및 항목; 특정 프로그램 메뉴 이름; GUI 인터페이스에 있는 버튼 및 텍스트가 포함되어 있으며, 텍스트와 구별 가능하도록 모두 가변폭 굵은체로 되어 있습니다.

고정폭 굵은 이텔릭체 또는 ***가변폭 굵은 이텔릭체***

고정폭 굵은체이던 가변폭 굵은체이던지 간에 이텔릭체가 추가될 경우 이는 교체 또는 변경 가능한 텍스트를 나타내는 것입니다. 글자 그대로 입력하지 말아야 할 텍스트나 또는 상황에 따라 변경해야 하는 텍스트의 경우 이텔릭체로 나타냅니다. 예:

ssh를 사용하여 원격 컴퓨터에 연결하려면, 쉘 프롬프트에 **ssh *username@domain.name***을 입력합니다. 원격 컴퓨터가 **example.com**이고 사용자 이름이 john일 경우, **ssh john@example.com**을 입력합니다.

mount -o remount *file-system* 명령은 지정한 파일 시스템을 다시 마운트합니다. 예를 들어, **/home** 파일 시스템을 다시 마운트하려면 **mount -o remount /home** 명령을 사용합니다.

현재 설치된 패키지 버전을 보려면, **rpm -q *package*** 명령을 사용합니다. 그러면 다음과 같은 값이 출력됩니다: ***package-version-release***.

사용자 이름, 도메인 이름, 파일 시스템, 패키지, 버전 및 릴리즈는 굵은 이탤릭체로 표시되는 점에 유의하십시오. 각 단어는 자리 표시자로 명령을 실행할 때 입력하는 텍스트나 도는 시스템에 의해 나타나는 텍스트 중 하나입니다.

작업 제목을 표시하기 위한 기본적인 사용을 제외하고 중요한 새로운 용어를 처음 사용할 때 이텔릭체로 표시합니다. 예:

Publican은 *DocBook* 발행 시스템입니다.

1.2. 인용문 규정

터미널 출력 결과 및 소스 코드 목록은 주위의 문장에서 잘 보이는 위치에 설정됩니다.

터미널로 보내진 출력 결과는**mono-spaced roman**에 설정되어 다음과 같이 나타납니다:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

소스 코드 목록도 **mono-spaced roman**에 설정되지만, 다음과 같이 구문 강조가 추가되어 있습니다:

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                  struct kvm_assigned_pci_dev *assigned_dev)
{
         int r = 0;
         struct kvm_assigned_dev_kernel *match;

         mutex_lock(&kvm->lock);

         match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
                                       assigned_dev->assigned_dev_id);
         if (!match) {
                 printk(KERN_INFO "%s: device hasn't been assigned before, "
                   "so cannot be deassigned\n", __func__);
                 r = -EINVAL;
                 goto out;
         }

         kvm_deassign_device(kvm, match);

         kvm_free_assigned_device(kvm, match);

out:
         mutex_unlock(&kvm->lock);
         return r;
}
```

1.3. 알림 및 경고

마지막으로, 3 종류의 시각적 스타일을 사용하여 간과될 수 있는 정보에 주의를 집중시킵니다.

참고

알림에서는 현재 작업에 대한 도움말, 지름길 또는 대안적 방법을 제공합니다. 알림 내용을 무시해도 상관없지만 효율적으로 작업할 수 있는 방법을 놓칠 수 있습니다.

중요

중요 상자에서는 현재 세션에만 적용되는 설정을 변경하거나 업데이트를 적용하기 전 다시 시작해야 하는 서비스와 같이 간과하기 쉬운 세부 사항을 제공합니다. 중요 상자를 무시해도 데이터를 손실하게 되지 않지만 문제를 일으킬 수 있습니다.

주의

경고는 무시해서는 안됩니다. 경고를 무시할 경우 대부분 데이터가 손실될 수 있습니다.

2. 피드백을 보내주십시오!

오자를 발견하셨거나, 보다 좋은 메뉴얼을 만들기 위한 제안이 있다면, 언제든지 저희에게 알려 주십시오! **Red Hat Enterprise Linux**에 대한 리포트를 버그질라 (http://bugzilla.redhat.com/)에 제출해 주십시오.

버그 리포트를 작성하실 때, 문서 식별자: doc-*Migration_Guide* 및 버전 번호: **6**를 꼭 기입해 주시기 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. 소개

마이그레이션 계획 가이드 (Migration Planning Guide)는 마이너 Red Hat Enterprise Linux 5 설치 버전을 Red Hat Enterprise Linux 6로 마이그레이션하기 위해 마이그레이션할 때 주요 동작 변경 사항에 대해 문서화되어 있습니다.

이 문서를 통해 Red Hat Enterprise Linux 5와 Red Hat Enterprise Linux 6 간의 제품 변경 사항에 대한 지침을 제공하여 보다 쉽게 Red Hat Enterprise Linux 6를 사용할 수 있도록 합니다. 하지만 이 문서에서는 모든 새로운 기능을 설명하도록 *구성되어 있지 않습니다.* 이는 Red Hat Enterprise Linux 5의 일부분으로 Red Hat Enterprise Linux 6에서 변경되었거나 또는 그 기능이 다른 패키지에 의해 대체된 애플리케이션 또는 구성 요소의 *동작* 변경 사항에 초점을 두고 있습니다.

1.1. Red Hat Enterprise Linux 6

Red Hat Enterprise Linux는 오픈 소스 컴퓨팅 업계 선두의 플랫폼입니다. 이는 서브스크립션 방식으로 판매되며 지속적인 가치를 제공하여 상위 엔터프라이즈 하드웨어 및 소프트웨어 벤더에 의해 인정받고 있습니다. 데스크탑에서 데이터 센터까지 Enterprise Linux는 오픈 소스 기술의 혁신과 진정한 엔터프라이즈급 플랫폼의 안정성을 통합하고 있습니다.

Red Hat Enterprise Linux 6는 Red Hat의 포괄적인 차세대 운영 체제 모음으로 절대적으로 필요한 엔터프라이즈 컴퓨터 설정을 위해 디자인되었으며 상위 엔터프라이즈 소프트웨어 및 하드웨어 벤더에 의해 인정받고 있습니다. 이 릴리즈는 다음과 같은 아키텍처에서 단일 키트로 사용할 수 있습니다:

- i386
- AMD64/Intel64
- System z
- IBM Power (64-비트)

이번 릴리즈에서 Red Hat은 서버, 데스크탑, 종합적인 Red Hat 오픈 소스 경험을 통해 개선된 사항을 제공합니다. 다음은 이번 릴리즈에 포함된 여러 개선 사항 및 새로운 기능의 일부분입니다:

전원 관리

Tuned 의한 적응형 시스템 튜닝, PowerTOP, 전원 관리 (ASPM, ALPM)에 의한 wakeups와 전력 소비를 줄이기 위해 애플리케이션 스택을 통해 Tickless kernel 및 개선

차세대 네트워킹

포괄적인 IPv6 지원 (NFS 4, CIFS, 모바일 지원 [RFC 3775], ISATAP 지원), FCoE, iSCSI 및 새롭게 개선된 mac80211 무선 스택

신뢰성, 가용성 및 서비스 용이성

하드웨어 RAS 기능과 NUMA 아키텍처를 최대한 살리기 위해 업계 협력하여 시스템 레벨의 강화

세분화된 제어 및 관리

CFS (Completely Fair Scheduler) 및 CG (Control Groups)를 통해 커널에서의 개선된 스케줄러 및 자원 관리

확장 가능한 파일 시스템

ext4는 디폴트 파일 시스템이지만, xfs는 견고성, 확장성, 고성능을 제공합니다.

가상화

KVM에는 성능 개선 및 새로운 기능이 포함되어 있으며, sVirt는 게스트 위반에 대해 호스트, VM 및 데이터를 보호합니다. SRIOV 및 NPIV는 실제 장치의 고성능 가상 사용을 공급하며, libvirt는 커널 CG 컨트롤러 기능을 활용합니다.

엔터프라이즈 보안 강화

SELinux에는 사용의 용이성 향상, 애플리케이션 샌드박스, 시스템 서비스 범위의 지속적 확장이 포함되어 있습니다. 반면, SSSD는 오프라인 사용을 위한 캐싱 외에도, 식별 및 인증 서비스를 위한 통합된 액세스를 제공합니다.

개발 및 런타임 지원

시스템 탭 (재컴파일없이 실행중인 커널을 사용하도록 허용), ABRT (간단한 버그 정보 모음), GCC (4.4.3 버전)와 glibc (2.11.1 버전) 및 GDB (7.0.1 버전)의 개선

1.2. 애플리케이션 호환성

이번 Red Hat Enterprise Linux 릴리즈는 종속성을 제공하여 이전 운영 체제 버전에서 실행된 애플리케이션은 계속하여 실행할 수 있습니다. 이를 위해 이번 릴리즈와 이전 버전 간의 변경 가능성이 있는 레거시 인터페이스를 저장하기 위해 이전 버전의 주요 라이브러리가 포함되어 있습니다. 이러한 라이브러리는 C/C++로 작성된 애플리케이션에 대한 의존성을 해결합니다.

Red Hat Enterprise Linux 마이너 릴리즈 사이의 애플리케이션을 다시 테스트 및 재인증할 필요가 없음에 유의하십시오. Red Hat Enterprise Linux 호환성 정책은 릴리스 버전에서 실행하는 애플리케이션이 해당 릴리스의 수명 기간 동안 계속 실행할 수 있음을 보장합니다. 예를 들어, Red Hat Enterprise Linux 6.0에서 인증된 애플리케이션은 Red Hat Enterprise Linux 6.1과 같은 버전에서 완벽한 호환성을 가지게 됩니다.

이러한 호환 패키지에 대한 보다 자세한 내용은 다음 표를 참조하십시오:

표 1.1. 호환성 라이브러리

패키지	설명
compat-db	Berkeley DB 데이터베이스 호환성 라이브러리. Berkeley 데이터베이스 (Berkeley DB)는 전통적인 애플리케이션과 클라이언트/서버 애플리케이션 모두에 내장된 데이터베이스 지원을 제공하는 프로그램 형식 툴킷입니다. 이 패키지에는 이전 버전에 포함되어 있던 여러 Berkeley DB 버전이 들어 있습니다.
compat-expat1	Expat는 스트림 지향 XML 파서입니다. 이 패키지는 이전 버전과의 라이브러리 호환성을 제공합니다.
compat-glibc	glibc는 시스템 호출과 다른 기본적인 기능에 사용되는 C 라이브러리입니다. 이 패키지는 이전 glibc 버전을 필요로 하는 바이너리의 컴파일을 위한 호환성 (및 런타임 라이브러리)을 제공하며, 이를 Red Hat Enterprise Linux 릴리스에서 실행되도록 합니다.
compat-libf2c-34	이 패키지는 Fortran 77 공유 라이브러리의 이전 버전을 제공합니다. 이 버전은 동적으로 링크된 Fortran 77 프로그램을 실행하기 위해 필요합니다.
compat-libgcc-296	2.96 libgcc.a 라이브러리를 포함하고 있으며, 이전 GCC 버전과 호환성을 유지하기 위해 객체 파일을 지원합니다.
compat-libgfortran-41	이 패키지에는 GCC 4.1.x 컴파일 Fortran 애플리케이션과의 호환을 위한 Fortran 95 런타임 라이브러리가 들어 있습니다.
compat-libstdc++-295	GNU 표준 C++ 라이브러리의 2.95 버전과의 호환성을 제공합니다.
compat-libstdc++-296	GNU 표준 C++ 라이브러리의 2.96 버전과의 호환성을 제공합니다.
compat-libstdc++-33	GNU 표준 C++ 라이브러리 3.3 버전과의 호환성을 제공합니다.
compat-libtermcap	이 패키지는 이전 termcap 기반 프로그램에 대해 호환성을 제공합니다.
compat-openldap	OpenLDAP는 LDAP (Lightweight Directory Access Protocol) 애플리케이션 및 개발 도구에 대한 오픈 소스 모음입니다. compat-openldap패키지에는 일부 애플리케이션에서 필요할 수 있는 OpenLDAP 공유 라이브러리의 이전 버전이 포함되어 있습니다.
openssl098e	이 패키지는 일부 SSL 애플리케이션에서 필요할 수 있는 OpenSSL 0.98e를 제공합니다.

2장. 설치

다음 부분에서는 Red Hat Enterprise Linux 6과 Red Hat Enterprise Linux 5 간의 설치 과정의 차이에 대해 설명합니다. 마이그레이션한 Red Hat Enterprise Linux 5 릴리즈 버전에 따라 여기에 나열된 모든 옵션 및 기능이 현재의 환경에 해당하지 않을 수 있습니다. 이는 이미 Red Hat Enterprise Linux 5 환경에서 사용되고 있을 수 있습니다.

2.1. 커널 및 부트 옵션

- Red Hat Enterprise Linux를 설치하기 전 **boot:** 프롬프트에서 **memtest86**을 입력하여 메모리 검사를 수행할 수 있습니다. 이 옵션은 *Anaconda* 시스템 설치 프로그램 대신 독립 메모리 테스팅 소프트웨어 *Memtest86* 을 실행합니다. 일단 실행을 시작하면 **Esc** 키를 누를때 까지 *Memtest86*메모리 테스트 루프가 계속 실행됩니다.
- 모듈 로드 순서를 정의하기 위해 이전 **scsi_hostadapter** 옵션 대신 **rdloaddriver** 커널 매개 변수가 필요합니다.

2.2. 그래픽 설치 프로그램

다음 부분에서는 그래픽 설치 프로그램에서 변경된 사항에 대해 설명합니다.

2.2.1. 장치 및 디스크

- IDE 드라이버 용 i386 및 x86_64 아키텍쳐에서 /dev/hd*X* 장치 이름은 사용되지 않으며 /dev/sd*X*로 변경되었습니다. 이러한 변경 사항은 PPC 아키텍쳐에는 적용되지 않습니다.
- 설치 프로그램이 스마트 어레이 (Smart Array) 카드를 찾지 못할 경우, 설치 프로그램 프롬프트에 **linux isa**를 입력합니다. 이는 필요한 카드를 수동으로 선택할 수 있게 합니다.
- 기존 IDE 드라이버는 장치마다 최대 63개 파티션까지 지원했지만, SCSI 장치는 장치마다 15개 파티션으로 제한하고 있습니다. Anaconda는 Red Hat Enterprise Linux의 다른 부분과 같은 방법으로 새로운 *libata* 드라이버를 사용하므로 설치 또는 업그레이드 중 IDE 디스크에 있는 15개 이상의 파티션을 검색하는 것은 불가능합니다. 15개 이상의 파티션을 가진 시스템을 업그레이드하는 경우 디스크를 LVM (Logical Volume Manager)으로 전환해야 할 것입니다.
- 커널이 스토리지 장치를 처리하는 방법의 변경은 /dev/hd*X* 또는 /dev/sd*X*와 같은 장치 이름이 이전 버전에서 사용된 값과 다를 수 있음을 의미합니다. Anaconda는 파티션 레이블에 의존하여 이러한 문제를 해결합니다. 이러한 레이블이 없을 경우, Anaconda는 이러한 파티션 레이블이 필요하다고 경고하게 됩니다. LVM (Logical Volume Management)과 장치 맵퍼를 사용하는 시스템을 다시 레이블할 필요가 없습니다.
- root 파일 시스템을 포함하여 암호화된 블록 장치에 설치하기 위한 지원이 포함됩니다.
- 모든 IDE RAID 컨트롤러가 지원되지 않습니다. RAID 컨트롤러가 **dmraid**에 의해 지원되지 않는 경우, Linux 소프트웨어 RAID를 설정하여 드라이버를 RAID 어레이에 결합하는 것이 좋습니다. 지원되는 컨트롤러의 경우, 컴퓨터 BIOS에서 RAID 기능을 설정할 수 있습니다.
- Red Hat Enterprise Linux 6에 포함된 GRUB 버전은 ext4를 지원하고 있으므로, Anaconda는 **/boot** 및 root 파티션을 포함하여 파티션에 있는 ext4 파일 시스템 사용을 허용합니다.

2.2.2. 킥스타트

다음 부분에서는 자동 설치 (Kickstart)와 관련하여 변경된 사항에 대해 설명합니다.

2.2.2.1. 동작 변경 사항

- 이전에 **network** 행이 없는 킥스타트 파일은 네트워크 설정을 위해 DHCP를 사용해야 한다고 가정하였습니다. 이는 생략된 다른 모든 행은 설치가 중지되어 입력해야 함을 의미한다는 점에서 킥스타트의 다

른 부분과 조화되지 않습니다. **network** 행이 없는 것은 설치가 중지되어 입력하라는 뜻입니다. 또한, **--bootproto=query** 옵션은 사용 중지되었습니다. 인터럽트없이 DHCP를 계속 사용하려면, **network --bootproto=dhcp**를 킥스타트 파일에 추가합니다.

- 전통적으로 디스크는 킥스타를 통해 장치 노드 이름 (예: **sda**)에서 참조했습니다. Linux 커널은 보다 동적인 방식으로 전환하고 여기서 재부팅 시 장치 이름이 유지된다는 것을 보장하지 않으므로 킥스타트 스크립트에서의 사용이 복잡하게 될 수 있습니다. 장치 이름을 안정적으로 유지하려면 장치 노드 이름 대신 **/dev/disk**에서 아무 항목 중 하나를 사용합니다. 예:

  ```
  part / --fstype=ext4 --onpart=sda1
  ```

  다음중 하나와 유사한 항목을 사용할 수 있습니다:

  ```
  part / --fstype=ext4 --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-
  part1
  part / --fstype=ext4 --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
  ```

  이는 단순한 **sda** 보다 의미있는 디스크에서 참조하기 위한 일관된 방법을 제공합니다. 특히 이는 대규모 스토리지 환경에서 유용합니다.
- 또한 디스크 참조를 위해 쉘과 같은 항목을 사용할 수 있습니다. 이는 대규모 스토리지 환경에서 **clearpart** 및 **ignoredisk** 명령을 보다 쉽게 사용하고자 하는 의도였습니다. 예를 들어, 다음 대신:

  ```
  ignoredisk --drives=sdaa,sdab,sdac
  ```

  다음과 유사한 항목을 사용할 수 있습니다:

  ```
  ignoredisk --drives=/dev/disk/by-path/pci-0000:00:05.0-scsi-*
  ```

- 킥스타트는 이전 버전 보다 더 많은 경우에 있어서 오류 메세지와 함께 정지됩니다. 예를 들어, 존재하지 않는 디스크를 참조하면 설치가 중단되고 오류를 알립니다. 이는 더 심각한 문제로 넘어가기 전에 킥스타트 파일에서 오류를 검색하도록 설정되어 있기 때문입니다. 부작용으로 다른 시스템 구성에 걸쳐 일반적으로 고안된 파일은 보다 자주 실행 실패할 수 있습니다. 이는 상황에 따라 상황별로 처리되어야 합니다.
- 킥스타트 네트워크 정보에 사용되는 **/tmp/netinfo** 파일은 삭제되었습니다. Anaconda는인터페이스 설정에 *NetworkManager*를 사용하고 설정을 **/etc/sysconfig/network-scripts/**에 있는 ifcfg 파일에 저장합니다. 이러한 새로운 위치를 *%pre* 및 *%post* 스크립트에 대한 네트워크 설정 소스로 사용할 수 있습니다.

2.2.2.2. 명령 변경 사항

다음 부분에서는 명령 및 옵션에서의 가장 중요한 변경 사항을 나열합니다:

- **network --device** 옵션은 장치 이름 대신 MAC 주소로 장치를 참조할 수 있습니다. 디스크와 마찬가지로 네트워크 장치 이름은 장치가 검색된 순서에 따라 재부팅 후 변경될 수 있습니다. 킥스타트에서 일관성있게 이름을 지정하려면 다음과 유사한 항목을 사용할 수 있습니다:

  ```
  network --device=00:11:22:33:44:55 --bootproto=dhcp
  ```

- **langsupport**, **key**, **mouse** 명령이 삭제되었습니다. 이 명령을 사용할 경우 구문 오류가 나타납니다. **monitor** 명령도 삭제되었습니다.

  langsupport 대신 해당 그룹을 킥스타트 파일의 **%packages** 섹션에 추가합니다. 예를 들어, 프랑스어 지원을 포함하려면 **@french-support**를 추가합니다.

  설치 중 설치키는 더이상 요청되지 않으므로 **key** 옵션에 대한 대체 옵션이 없습니다. 간단하게 파일에서 이 옵션을 제거합니다.

X는 자동으로 감지되어 설정될 수 있으므로 **mouse** 및 **monitor** 명령이 필요없습니다. 동일한 이유로 **xconfig --resolution=** 명령은 더이상 유효하지 않으며 안전하게 파일에서 제거할 수 있습니다.

- **part --start** 및 **part --end** 명령은 사용 중지되어 아무런 효과를 갖지 않습니다. Anaconda는 특정 부분 경계에는 파티션 생성을 허용하지 않습니다. 보다 엄격한 수준의 파티션이 필요한 경우, **%pre**에 있는 외부 도구를 사용하여 anaconda에게 **part --onpart** 명령으로 기존 파티션을 사용하도록 지시합니다. 그렇지 않으면 특정 크기의 파티션을 만들거나 **--grow**를 사용합니다.
- **%post**에서 수동으로 그룹을 만드는 대신, **group** 명령을 사용하여 이를 만들 수 있습니다. 보다 자세한 내용은 킥스타트 문서를 참조하십시오.
- 디폴트 autopart 알고리즘이 변경되었습니다. 모든 시스템에 대해 autopart는 **/boot** (또는 아키텍쳐에서 필요로 하는 기타 다른 특정한 부트 로더 파티션) 및 스왑을 만듭니다. 최소 50 GB 여유 디스크 공간을 갖는 시스템의 경우 autopart는 적절한 크기의 root 파티션 (**/**)을 생성하고 나머지는 **/home**에 할당됩니다. 그 이하의 여유 공간을 갖는 시스템의 경우 root (**/**)만이 생성됩니다.

  /home 볼륨 생성을 원하지 않을 경우, autopart를 사용하지 않습니다. 대신 **/boot**, 스왑, **/**를 지정하고 필요에 따라 root 볼륨을 확장할 수 있는지 확인합니다.
- Anaconda는 설치 시 육안으로 확인 가능한 장치를 제어하기 위해 새로운 스토리지 필터링 인터페이스를 포함하고 있습니다. 이러한 인터페이스는 기존 **ignoredisk**, **clearpart** **zerombr** 명령에 해당합니다. **ignoredisk**는 옵션이기 때문에 킥스타트 파일에서 이를 제외해도 필터 UI가 설치 시 나타나지 않게 됩니다. 이러한 인터페이스를 사용하고자 할 경우 다음을 추가합니다:

```
ignoredisk --interactive
```

- **/tmp/partition-include** 파일에서의 **--size=1 --grow** 옵션은 더이상 사용할 수 없습니다. 적당한 디폴트 크기를 지정하면 이에 따라 파티션은 확장됩니다.

2.2.2.3. 패키지 변경 사항

이러한 변경 사항은 **%packages** 섹션에 영향을 미칩니다:

- **--ignoreDeps** 및 **--resolveDeps** 인수가 삭제되었습니다. Anaconda는 자동으로 의존성 문제를 해결하지만 의존성을 만족하지 않는 패키지 설치는 생략합니다.
- 모든 기본값을 수락하는 기본 GUI 설치에 있는 것과 동일한 패키지 세트를 킥스타트를 통해 이용하고 싶은 경우 다음을 추가합니다:

```
%packages --default
%end
```

- 옵션으로 여러 아키텍처 설치를 위해 설치하려는 패키지의 아키텍쳐를 지정할 수 도 있습니다. 예:

```
%packages
glibc.i686
%end
```

  이는 x86 *glibc* 패키지를 세트에 추가합니다. 이는 호환성을 이유로 x86 패키지를 필요로 하는 x86-64 시스템에서 유용합니다.
- **%packages** 섹션에 있는 모든 패키지 및 그룹을 감시하고 마이그레이션하는 것은 불가능합니다. 일부 패키지 및 그룹은 삭제되고, 일부는 추가되었으며 일부는 이름이 변경되었습니다. 보다 자세한 내용은 릴리즈 노트를 참조하십시오.

2.2.2.4. 스크립트 변경 사항

이러한 변경 사항은 **%pre**, **%post**, **%traceback** 스크립트 사용에 영향을 줍니다.

- 스크립트 실행 중 오류 로그가 향상되었습니다. 스크립트는 실행 후 더이상 삭제되지 않으므로 이를 확

인할 수 있습니다. 이는 스크립트가 동적으로 생성되는 시스템에서 가장 유용하며 시스템에서 무엇이 실행되고 있는지를 확인할 수 있게 합니다. 또한 stderr 및 stdout 출력은 모든 스크립트에 대해 기록됩니다. 이는 한 가지 중요한 부작용을 가지고 있습니다: 스크립트가 대화식 프로그램을 사용하는 경우, 스크립트의 헤더에 **`--logfile=/dev/tty3`**를 전달해야 합니다. 그렇지 않을 경우, 프로그램과 상호 작용할 수 없습니다.

2.2.2.5. 구문 변경 사항

주요 킥스타트 구문으로 변경하는 경우는 드물지만 주의해야 할 두 가지 중요한 구문 변경 사항이 있습니다:

- **`%include`** 옵션은 파일 이름 외에 URL을 인수로 허용합니다.
- **`%packages`**, **`%post`**, **`%pre`**, **`%traceback`** 섹션은 마지막에 **`%end`** 옵션이 있어야 합니다. 이전에 이러한 섹션은 명시적인 종료 토큰을 가지지 않고 다른 것이 시작하는 위치에서 종료되었습니다. Red Hat Enterprise Linux 6 시작에서 **`%end`**를 사용할 것을 권장합니다. **`%end`** 토큰이 없는 파일은 실행 실패하게 됩니다.

2.2.2.6. 다른점 요약

다음 부분에서는 Red Hat Enterprise Linux 6의 명령과 옵션의 차이점에 대해 열거합니다:

삭제된 명령:

- **`key`**
- **`langsupport`**
- **`mouse`**

사용 중지된 명령:

- **`monitor`**
- **`xconfig --resolution`**

추가된 명령:

- **`fcoe`**
- **`group`**
- **`rescue`**
- **`sshpw`**
- **`updates`**

2.2.2.7. pykickstart

pykickstart 패키지에는 마이그레이션을 쉽게하기 위해 사용할 수 있는 유틸리티가 들어 있습니다. 최신 패키지가 설치되어 있는지 확인합니다. **`ksverdiff`** 명령은 시작 및 종료 구문 버전을 받고 이 두가지 버전에 대해 명령과 옵션에서의 차이를 보고합니다. 이는 새롭거나 사용 중지된 또는 삭제된 명령 및 옵션을 제공합니다. 예:

```
$ ksverdiff --from RHEL5 --to RHEL6

The following commands were removed in RHEL6:
langsupport mouse key

The following commands were deprecated in RHEL6:
monitor

The following commands were added in RHEL6:
sshpw group rescue updates fcoe
 ...
```

ksvalidator 명령을 사용하여 킥스타트 파일의 유효성을 확인할 수 있습니다. 이 명령은 지정된 킥스타트 구문 버전에 대해 파일의 유효성을 검사합니다. 하지만, 설치 시 발생할 수 있는 문제에 대해 알릴 수 없습니다. 예를 들어 **part --ondisk=sdr**를 지정했을 경우 이러한 장치가 존재하지 않습니다. 예시:

```
$ ksvalidator --version RHEL6 my-rhel5-ks.cfg
```

2.2.3. 네트워크 설정

다음 부분에서는 네트워크 설정 관련 그래픽 설치 프로그램에서 변경된 사항에 대해 설명합니다.

- Anaconda는 설치 시 네트워크 설정을 위해 *NetworkManager*를 사용합니다. Anaconda에서의 주요 네트워크 인터페이스 설정 화면은 삭제되었습니다. 필요한 경우 설치 시 사용자가 네트워크 설정 정보를 Anaconda는 설치 중에 네트워크 설정 *NetworkManager*를 사용합니다.Anaconda에 있는 주요 네트워크 인터페이스 설정 화면은 삭제되었습니다. 설치 시 정보가 필요한 경우에만 사용자에게 네트워크 설정 정보를 묻는 메세지가 나타나게 됩니다. 설치 시 사용된 설정은 나중에 사용하기 위해 시스템에 기록됩니다.
- 설치 프로그램을 부팅하기 위해 **boot.iso**를 사용하는 경우 기본 설치 방법이 모두 선택되어 있어도 소스 선택 화면이 나타납니다.
- PXE 부팅 때 설치 미디어에 NFS를 통해 마운트된 .iso 파일을 사용하는 경우 명령행에 **repo=nfs:server:/path/**를 추가합니다. 또한 **install.img** 및 **product.img** 파일을 추출하여 **nfs:server:/path/images/** 디렉토리에 배치해야 합니다. **product.img** 파일에는 다양한 정의와 각종 설치 클래스가 포함되어 있습니다.
- 다수의 네트워크 인터페이스를 갖는 일부 시스템은 시스템 BIOS가 인식한 첫번째 네트워크 인터페이스에 *eth0*를 할당하지 않을 수 있습니다. 이는 설치 프로그램이 본래 PXE가 사용했던 것과 다른 네트워크 인터페이스를 사용 시도하게 하는 원인이 될 수 있습니다. 이를 변경하려면, **pxelinux.cfg/*** 설정 파일에서 다음을 사용합니다:

  ```
  IPAPPEND 2 APPEND
        ksdevice=bootif
  ```

- 이 설정 옵션은 설치 프로그램이 시스템 BIOS와 PXE가 사용하는 것과 동일한 네트워크 인터페이스를 사용하게 합니다. 다음과 같은 옵션을 사용하여 설치 프로그램이 네트워크 스위치에 연결된 첫번째 네트워크 장치를 사용하게 합니다:

  ```
  ksdevice=link
  ```

2.2.4. 제품 서브스크립션 및 콘텐츠 업데이트

Red Hat Enterprise Linux 6에서는 콘텐츠 배포 및 서브스크립션 관리를 통해 보다 유연한 최신의 서비스를 도입하고 있습니다. 다음 부분에서는 콘텐츠 서비스에서의 변경 사항에 대해 설명합니다.

- Red Hat Network 호스트 환경은 제품 및 품질 기반 서브스크립션에서 채널 기반 서브스크립션을 사용하여 업데이트됩니다. 이러한 새로운 인증서 기반 RHN으로 서브스크립션 및 시스템 관리를 위한 클라이언트 도구가 다시 개발되어 새로운 서브스크립션 및 콘텐츠 전송 네트워크 (CDN)와 작동합니다.

  기존의 채널 기반 RHN도 *RHN 클래식*에서 사용할 수 있습니다.

  동일한 플랫폼에서 두 가지 종류의 서브스크립션을 사용할 수 있습니다. 병렬 기술을 사용하여 모든 서브스크립션은 어떠한 방법을 사용해서도 등록 및 관리 가능합니다.

  Satellite 또는 Proxy 서버를 사용하는 환경에서는 기존의 채널 기반 서브스크립션 시스템이 계속 사용되어 시스템을 RHN 클래식으로 등록합니다.
- 새로운 컨텐츠 서버 옵션, Red Hat Network 클래식이 firstboot 마법사에 추가되어 있습니다. 이는 업데이트된 RHN 및 CDN 대신 기존의 채널 기반 RHN을 사용합니다. 디폴트 Red Hat Network 옵션은 인증서 기반의 새로운 Red Hat Network 관리 플랫폼을 사용합니다.
- 인증서 기반 RHN과 RHN 클래식은 상호 운용이 가능합니다. 시스템이 하나의 서비스를 사용하여 등록되면, 다른 서비스는 이를 인식하여 아무 문제가 발생하지 않습니다. 하지만 이러한 서비스는 동시에 작동하지 않으므로 시스템은 하나의 서브스크립션 서비스로 등록해야 합니다. 두 서비스에 모두 등록할 수 없습니다.

  현재 RHN 클래식을 사용하는 시스템에서 새 인증서 기반의 Red Hat Network를 사용하는 시스템으로 직접 마이그레이션할 수 있는 경로는 없습니다. 시스템을 한 서비스에서 다른 서비스로 이동하려는 경우 다음의 두 가지 방법을 사용합니다:
  - **`yum`** 대신 부팅 ISO를 사용하여 시스템을 Red Hat Enterprise Linux 6.1 이상 버전으로 업데이트합니다.
  - 수동으로 RHN 클래식에서 시스템을 제거하고 호스트 기록을 삭제한 후, 해당 시스템을 인증서 기반의 Red Hat Network에 Red Hat 서브스크립션 관리자 도구를 사용하여 등록합니다.
- 클라이언트 도구의 새로운 세트가 되는 Red Hat 서브스크립션 관리자 GUI 및 CLI는 인증서 기반 RHN을 통해 서브스크립션을 관리할 수 있도록 Red Hat Enterprise Linux 6.1에서 제공되고 있습니다. RHN 클래식을 통해 관리하는 시스템을 처리하기 위해 기존 **`rhn_*`** 도구도 사용 가능합니다.

2.3. 텍스트 기반 설치 프로그램

Red Hat Enterprise Linux 6의 텍스트 모드 설치 옵션은 이전 버전에 비해 상당히 간소화되어 있습니다. 텍스트 모드 설치는 이전 설치 과정의 일부였던 복잡한 단계를 생략하고 정리된 간단한 형식을 제공합니다. 다음 부분에서는 텍스트 기반 설치 프로그램 사용시 동작 변경 사항에 대해 설명합니다:

- 현재 Anaconda는 기본적이고 핵심적인 그룹에서만 자동으로 패키지를 선택합니다. 설치 프로세스의 마지막에 시스템이 실행 가능하며, 업데이트 및 새 패키지를 설치할 준비가 되어 있는 지를 확인하기 위해 이러한 패키지면 충분합니다.
- Anaconda는 시스템의 어디에 Red Hat Enterprise Linux을 설치할 지를 지정할 수 있는 이전 버전의 초기 화면을 여전히 제시합니다. 전체 드라이브 사용, 기존 Linux 파티션 삭제, 드라이브의 여유 공간 사용을 선택할 수 있습니다. 현재 Anaconda는 자동으로 파티션 레이아웃을 설정하고 이러한 기본 레이아웃에서 파티션이나 파일 시스템 추가/제거를 요구하지 않습니다. 설치 시 사용자 정의 레이아웃이 필요할 경우, VNC 연결이나 킥스타트 설치를 통해 그래픽 설치를 실행해야 합니다. LVM (logical volume management), 암호화된 파일 시스템, 파일 시스템 크기 조정과 같은 고급 옵션은 아직 그래픽 모드와 킥스타트에서만 사용 가능합니다.
- Anaconda는 텍스트 기반 설치 프로그램에서 자동으로 부트 로더 설정을 실행합니다.
- 킥스타트를 사용한 텍스트 모드 설치는 이전 버전과 동일한 방법으로 수행됩니다. 하지만 패키지 선택, 고급 파티션 설정 및 부트로더 설정은 텍스트 모드에서 자동화되어 있기 때문에 Anaconda는 이 단계에서 필요한 정보를 사용자에게 요청할 수 없습니다. 따라서 킥스타트 파일에 패키지 선택, 파티션 설정 및 부트로서 설정이 포함되어 있는지를 확인해야 합니다. 이 중 하나라도 결여되어 있을 경우, Anaconda는 오류 메세지와 함께 종료됩니다.

3장. 스토리지 및 파일 시스템

3.1. RAID

업그레이드

dmraid 세트에서 **mdraid** 세트로의 업그레이드 작업은 지원되지 않습니다. 이러한 유형의 업그레이드를 시도하면 경고가 표시됩니다. 기존 **mdraid** 세트에서의 업그레이드 및 새로운 **mdraid** 세트를 만들 수는 있습니다.

세트를 업그레이드할 때 새로운 기본 슈퍼 블록이 문제의 원인이 될 수 있습니다. 이러한 새로운 슈퍼 블록 형식 (RAID1 **/boot** 파티션을 만들 때 이외의 모든 장치에서 사용)은 현재 어레이의 시작 부분에 있고 파일 시스템이나 LVM 데이터는 파티션 시작 부분에서 오프셋입니다. 어레이가 실행되지 않을 때, LVM과 파일 시스템 **mount** 명령은 유효한 볼륨이나 파일 시스템 데이터를 갖는 장치를 찾지 못할 수 있습니다. 이는 의도적인 동작으로 RAID1 어레이에 하나의 디스크를 마운트하려면 이러한 디스크가 있는 어레이를 시작하고 나서 어레이를 마운트해야함을 의미합니다. 베어 디스크를 직접 마운트할 수 없습니다. 이러한 변경이 이루어진 이유는 resync가 강제되지 않은 경우 베어 디스크를 직접 마운트하면 어레이를 손상시킬 수 있기 때문입니다.

재부팅하면 RAID 시스템은 어레이에 포함되지 않은 디스크가 호환되지 않은 것으로 간주하고 해당 장치를 어레이에서 분리합니다. 이는 정상적인 동작으로 어레이에 다른 디스크를 다시 추가할 준비가 되었을 경우, **mdadm** 명령을 사용하여 디스크를 어레이에 hot add (실행 중인 시스템에 추가)합니다. 이 때 디스크의 변경 부분 (쓰기 예정 비트맵이 있는 경우) 또는 디스크 전체 (비트맵이 없는 경우) resync가 실행되며 어레이는 다시 동기화됩니다. 이 시점 이후 어레이는 올바르게 구성되었다고 간주하므로 장치는 어레이에서 분리되지 않습니다.

새로운 슈퍼 블록은 **mdraid**라는 어레이의 개념을 지원합니다. 어레이를 구별하기 위해 이전 어레이 열거 방법에 의존하지 않습니다 (예: **/dev/md0**의 다음은 **/dev/md1**). 이제는 어레이에 대해 임의의 이름 (예: **home**, **data**, **opt**)을 선택할 수 있습니다. **--name=opt** 옵션을 사용하여 선택한 이름으로 어레이를 만들 수 있습니다. 어떤 이름을 어레이로 지정해도 **/dev/md/**에서 만들어집니다. (이름으로 전체 경로를 사용하지 않는 한, 어떤 경우든지 경로가 만들어 집니다. 또는 0과 같은 하나의 숫자를 지정하지 않는 한, **mdadm**은 이전 **/dev/md***x* 형식을 사용하여 어레이를 시작합니다). Anaconda 설치 프로그램은 현재 어레이 이름의 선택을 허용하지 않지만, 대신 과거에 어레이가 생성된 방법을 모방하기 위한 방식으로 쉬운 숫자를 사용합니다.

새로운 **mdraid** 어레이는 쓰기 예정 비트맵 (write intent bitmap)을 사용합니다. 이는 시스템이 갑자기 종료되었을 때 어레이의 문제 부분을 식별하여 디스크 전체가 아닌 이 부분만을 다시 동기화합니다. 이는 다시 동기화하는데 필요한 시간을 현저하게 감소시킵니다. 새로 생성된 어레이는 적절한 경우 자동으로 쓰기 예정 비트맵을 추가합니다. 예를 들어, 스왑하는데 사용되는 어레이와 아주 작은 어레이 (예: **/boot** 어레이)는 쓰기 예정 비트맵에서 혜택을 받지 않습니다. 업그레이드를 완료한 후 장치에서 **mdadm --grow** 명령을 통해 기존의 어레이에 쓰기 예정 비트맵을 추가할 수 있지만 쓰기 예정 비트맵의 성능이 조금 저하될 수 있습니다 (65536 비트맵 청크 크기에서 약 3-5% 감소, 하지만 8192와 같은 작은 비트맵 청크 크기에서는 10% 이상으로 증가될 수 있음). 이는 쓰기 예정 비트맵이 어레이에 추가될 경우, 청크 크기를 크게 유지하는 것이 가장 좋다는 것을 의미합니다. 권장 크기는 65536입니다.

3.2. ext4

ext3에서 마이그레이션

ext4를 사용하고자 할 경우 새로 포맷된 파티션으로 시작할 것을 권장합니다. 하지만 레거시 ext3 파티션을 ext4로 변환하려면 **ext4migrate** 부트 옵션과 함께 Red Hat Enterprise Linux 6를 설치할 수 있습니다. 현

재 파티션에 있는 데이터는 확장 기능 및 기타 다른 변경 사항을 사용하지 않기 때문에 이러한 작업을 실행할 경우 ext4 가 제공하는 모든 혜택을 받으실 수 없는 점에 유의하십시오. 반면 새로운 데이터는 이러한 확장 기능을 사용합니다. ext4로 마이그레이션하기 위해 부트 옵션을 전달하는 것은 권장되지 않으며 마이그레이션을 시도하기 전 파일 시스템을 백업할 것을 적극 권장합니다.

동작 변경 사항

Red Hat Enterprise Linux 6에서는 ext4에 대해 완전하게 지원하며 이는 새로운 설치 프로그램의 디폴트 파일 시스템입니다. 다음 부분에서는 이러한 새로운 파일 시스템의 주요 동작 변경 사항에 대해 설명합니다.

- *GRUB* 부트로더에 포함된 버전은 ext4 파티션에 대해 완전한 지원을 제공합니다. 또한 설치 프로그램은 ext4 파티션 상에 어떤 **/boot** 파일 시스템도 배치할 수 있게 합니다.
- *e2fsprogs* 패키지에 포함된 버전은 ext4와 완전하게 호환 가능합니다.
- 일부 경우 ext4 파일 시스템은 *e4fsprogs* 패키지와 함께 Red Hat Enterprise Linux 5.3에서 생성되어 **ext4dev** 파일 시스템 유형을 만듭니다. 이러한 파일 시스템을 개발 버전으로 확인하는 **test_fs** 기능 플래그는 **tune2fs -E ^test_fs** 명령을 사용하여 제거할 수 있습니다. 이러한 파일 시스템이 정상적으로 ext4 파일 시스템으로 인식되도록 하기 위해 이 명령을 사용합니다.

3.3. fusecompress

fusecompress

Fusecompress는 권한이 없는 사용자가 마운트할 수 있는 압축 파일 시스템입니다. Red Hat Enterprise Linux 6에는 몇 가지 버그가 수정되었지만 온디스크 (on-disk) 포맷을 변경하는 업데이트 버전이 들어 있습니다. 기존 fusecompress 파일 시스템을 갖고 있는 사용자는 해당 데이터를 새로운 포맷으로 마이그레이션해야 합니다. 업그레이드하기 전 압축 풀기를 실행하지 않을 경우 *fusecompress_offline1* 패키지가 필요합니다.

3.4. blockdev

blockdev

blockdev --rmpart 명령 옵션은 더이상 지원되지 않습니다. **partx(8)** 및 **delpart(8)** 명령이 이러한 기능을 제공합니다.

4장. 네트워크 설정 및 서비스

4.1. 인터페이스 및 설정

NetworkManager

Red Hat Enterprise Linux 6은 네트워크 인터페이스를 구성할 때 디폴트로 NetworkManager를 사용합니다.

Infiniband

Infiniband 지원 (특히 **openib** 시작 스크립트 및 **openib.conf** 파일)은 Red Hat Enterprise Linux 5에 있는 *openib*에서 제공되었습니다. 이 패키지의 기능을 보다 정확하게 반영하기 위해 Red Hat Enterprise Linux 6에서 패키지의 이름은 변경되었습니다. Infiniband 기능은 *rdma* 패키지에서 배포되고 있습니다. 서비스는 **rdma**라는 이름으로 설정 파일은 **/etc/rdma/rdma.conf**에 있습니다.

4.2. 서비스 초기화

xinetd

Xinetd는 요청시 네트워크 서비스를 시작하기 위해 사용되는 데몬입니다. xinetd의 변경 내용은 오픈 파일 디스크립터의 허용 한도와 관련되어 있습니다:

- 리스닝 메커니즘은 **select()**에서 **poll()**로 변경되었습니다. 이로 인해, xinetd에서 사용되는 오픈 파일 디스크립터의 제한을 변경할 수 있습니다.
- 파일 디스크립터 한계는 서비스 단위 기준으로 변경할 수 있습니다. 이는 **rlimit_files** 지시문을 사용하여 서비스의 설정 파일에서 실행할 수 있습니다. 값은 양의 정수 또는 UNLIMITED입니다.

런레벨

Red Hat Enterprise Linux 6에서는 사용자 지정 런레벨 7, 8, 9는 더이상 지원되지 않으므로 사용할 수 없습니다.

Upstart

Red Hat Enterprise Linux 6에서 *sysvinit* 패키지의 *init*은 이벤트 기반 init 시스템인 *Upstart*로 대체되었습니다. 이 시스템은 부팅시 작업 및 서비스의 시작, 종료시 시스템 중지 및 시스템 실행 시 서비스 모니터링을 담당합니다. Upstart 에 대한 보다 자세한 내용은 **init(8)** man 페이지를 참조하십시오.

Upstart 라고 알려진 프로세스는 **/etc/init** 디렉토리에 있는 파일에 의해 정의됩니다. Upstart는 man 페이지에서 자세히 설명하고 있습니다. 명령 개요는 **init(8)**에 있으며 작업 구문은 **init(5)**에서 설명하고 있습니다.

Upstart는 Red Hat Enterprise Linux 6에서 다음과 같은 동작 변경을 제공합니다.

- **/etc/inittab** 파일은 사용되지 않으며 이는 현재 *initdefault* 행을 통해 디폴트 런레벨 설정을 위한 *작업에서만* 사용됩니다. 다른 설정은 **/etc/init** 디렉토리에 있는 **upstart** 작업에서 실행됩니다.
- 활성 tty 콘솔 수는 **/etc/sysconfig/init**의 *ACTIVE_CONSOLES* 매개 변수에 의해 설정됩니다. 이 변수는 **/etc/init/start-ttys.conf** 작업에 의해 읽을 수 있습니다. 디폴트 값은 *ACTIVE_CONSOLES=/dev/tty[1-6]*으로 이는 tty1 부터 tty6 까지 getty를 시작합니다.
- 직렬 콘솔이 주요 시스템 콘솔인 경우 직렬 getty는 자동으로 설정됩니다. 이전 릴리즈에서 이는 **kudzu**에 의해 실행되어, **/etc/inittab**을 편집하였습니다. Red Hat Enterprise Linux 6에서 주요 직렬 콘솔 설정은 **/etc/init/serial.conf**에 의해 처리됩니다.

- 디폴트값 이외의 직렬 콘솔에서 실행되고 있는 getty를 설정하려면, **/etc/inittab**을 편집하는 대신 Upstart 작업을 작성해야 합니다. 예를 들어, ttyS1의 getty가 요구되는 경우, 다음의 작업 파일 (**/etc/init/serial-ttyS1.conf**)이 작동합니다:

```
# This service maintains a getty on /dev/ttyS1.

start on stopped rc RUNLEVEL=[2345]
stop on starting runlevel [016]

respawn
exec /sbin/agetty /dev/ttyS1 115200 vt100-nav
```

이전 릴리즈와 마찬가지로 getty에서 root 로그인을 허용하려면 ttyS1이 **/etc/securetty**에 있는지 확인해야 합니다.

Upstart로 이동하기 위해 **/etc/shutdown.allow**를 사용하여 컴퓨터를 종료할 수 있는 사용자를 정의하는 것은 더이상 지원되지 않습니다.

4.3. IPTables/Firewalls (방화벽)

IPTables에는 **SECMARK** 대상 모듈이 포함되어 있습니다. 이는 SELinux와 같은 보안 서브 시스템에서 사용하기 위한 패킷과 관련된 보안 표시 값을 설정하는데 사용됩니다.

```
iptables -t mangle -A INPUT -p tcp --dport 80 -j SECMARK --selctx \
system_u:object_r:httpd_packet_t:s0
```

4.4. Apache HTTP 서버

Red Hat Enterprise Linux 6로 마이그레이션 시 주의해야 할 Apache HTTP 서버의 변경 사항은 다음과 같습니다:

- **mod_file_cache**, **mod_mem_cache**, **mod_imagemap** 모듈은 더이상 지원되지 않습니다.
- **Charset=UTF-8** 옵션은 디폴트 **IndexOptions** 지시문에 추가되었습니다. UTF-8 이외의 문자 집합이 있는 디렉토리 목록이 필요한 경우 (예: **mod_autoindex**에 의해 생성된 것 등) 이 옵션을 변경합니다.
- **distcache** 배포되는 세션 캐시는 **mod_ssl**에서 더이상 지원되지 않습니다.
- 프로세스 ID (pid)의 기본 위치는 **/var/run**에서 **/var/run/httpd**로 이동되었습니다.
- *mod_python* 패키지는 업스트림 개발이 중단되었기 때문에 더이상 제공되지 않습니다. Red Hat Enterprise Linux 6에서는 대신 *mod_wsgi*를 제공하고 WSGI 인터페이스를 통해 Python 스크립팅을 지원합니다.

4.5. PHP

PHP 변경 사항은 다음과 같습니다:

- PHP는 버전 5.3으로 업그레이드되었습니다. 호환성 문제로 인해 스크립트 업데이트를 필요로 하는 경우가 있습니다. 자세한 내용은 다음 URL을 참조하십시오:
  - http://php.net/manual/migration52.php
  - http://php.net/manual/migration53.php
- 기본 설정에 다음과 같은 사항이 변경되었습니다 (**/etc/php.ini**):
  - **error_reporting**은 현재 **E_ALL & ~E_DEPRECATED**로 설정되어 있습니다. (이전에는 **E_ALL**로 설정됨)

  - **short_open_tag**는 현재 **Off**로 설정되어 있습니다 (이전에는 **On**으로 설정됨)
  - **variables_order**는 현재 **GPCS**로 설정되어 있습니다 (이전에는 **EGPCS**로 설정됨)
  - **enable_dl**는 현재 **Off**로 설정되어 있습니다 (이전에는 **On**으로 설정됨)
- **mime_magic**, **dbase**, **ncurses** 확장은 더이상 배포되지 않습니다.

4.6. BIND

BIND 설정에 있어서 몇 가지 주요 변경 사항이 있습니다:

- 디폴트 ACL 설정 - Red Hat Enterprise Linux 5에서 디폴트 ACL 설정은 쿼리를 허용하고 모든 호스트에 대해 재귀적 쿼리를 제공합니다. Red Hat Enterprise Linux 6에서 디폴트로 모든 호스트는 권한이 있는 데이터에 대해 쿼리할 수 있지만 로컬 네트워크의 호스트만이 재귀적 쿼리할 수 있습니다.
- 새로운 **allow-query-cache** 옵션 - 이 옵션을 우선하는 **allow-recursion** 옵션은 사용 중지되었습니다. 이는 권한이 없는 데이터 전체 (재귀적 검색 및 root 네임서버 정보 등)를 포함하는 서버 캐시에 대한 액세스를 제어하는데 사용됩니다.
- Chroot 환경 관리 - 비 chroot 환경에서 chroot 환경에 symlink를 생성하기 위해 사용된 **bind-chroot-admin** 스크립트는 사용 중지되어 더 이상 존재하지 않습니다. 대신 설정은 비 chroot 환경에서 직접 관리할 수 있으며 chroot이 없는 경우 init 스크립트는 **named** 시작 시 필요한 파일을 chroot 환경에 자동으로 마운트할 수 있습니다.
- **/var/named** 디렉토리 권한 - **/var/named** 디렉토리에는 더 이상 쓰기 불가능합니다. 쓰기를 할 필요가 있는 모든 영역 파일 (동적 DNS 영역, DDNS 등)은 쓰기 가능한 새로운 디렉토리에 배치해야 합니다: **/var/named/dynamic**
- **dnssec [yes|no]** 옵션은 더이상 존재하지 않습니다 - 글로벌 **dnssec [yes|no]** 옵션은 **dnssec-enable** 및 **dnssec-validation**이라는 새로는 두 개의 옵션으로 분할되었습니다. **dnssec-enable** 옵션은 DNSSEC 지원을 활성화합니다. **dnssec-validation** 옵션은 DNSSEC 인증을 활성화합니다. 재귀적 서버에서 **dnssec-enable**를 "no"로 설정하면 DNSSEC 인증을 실행하는 다른 서버에서 전달자(forwarder)로 이를 사용할 수 없음에 유의합니다. 두 옵션은 디폴트로 'yes'로 설정되어 있습니다.
- **rndc** 관리 유틸리티를 사용하는 경우 **/etc/named.conf**에서 **controls** 명령을 지정할 필요가 없어졌습니다. **named** 서비스가 자동으로 루프백 장치를 통해 제어 연결을 허용하고 **named** 및 **rndc**는 설치 과정에서 생성된 동일한 키를 사용합니다 (**/etc/rndc.key**에 위치함).

기본 설치에서 BIND는 DNSSEC 서명 검증을 사용하여 설치되며, ISC DLV 레지스터를 사용합니다. 이는 ISC DLV 레지스터에 키가 있는 모든 서명된 도메인 (예: gov., se., cz.)은 재귀적 서버에서 암호 확인되고 있음을 의미합니다. 캐시 감염 (cache poisoning)으로 암호 확인 실패시 최종 사용자에게 이러한 위조된/스푸핑 데이터를 제공하지 않습니다. DNSSEC 사용은 광범위하게 구현되고 있는 기능으로 최종 사용자를 위해 인터넷을 보다 안전하게 하기 위해 중요한 단계로 Red Hat Enterprise Linux 6에서 완전 지원되고 있습니다. 앞에서 언급했듯이 DNSSEC 서명 검증은 **/etc/named.conf**에 있는 **dnssec-validation** 옵션을 사용하여 제어합니다.

4.7. NTP

NTP (Network Time Protocol)는 네트워크를 통해 컴퓨터 시스템의 시계를 동기화하는데 사용됩니다. Red Hat Enterprise Linux 6에서 디폴트 설정 파일 **/etc/ntp.conf**는 다음 줄을 주석 처리합니다:

```
#server 127.127.1.0 # local clock
#fudge 127.127.1.0 stratum 10
```

이 설정은 **ntpd**가 NTP 서버 또는 참조 클럭에 동기화되어 있는 경우에만 네트워크 클라이언트에 시간 정보를 전달하게 됨을 의미합니다. **ntpd**가 동기화되어 있지 않을 때에도 이러한 정보를 제공하게 하려면 두

개의 줄을 주석 해제 처리해야 합니다.

ntpd가 **-x** 옵션으로 (**/etc/sysconfig/ntpd** 파일의 OPTIONS에서) 시작되었을 때 또는 **/etc/ntp/step-tickers**에 서버가 지정되어 있는 경우, 서비스가 시작되기 전 이는 **ntpdate** 명령을 실행하지 않습니다. 현재 **ntpd** 서비스에서 독립적으로 활성화될 수 있는 별도의 **ntpdate** 서비스가 있습니다. **ntpdate** 서비스는 디폴트로 비활성화되어 있으며 다른 서비스를 시작하기 전 정확한 시간을 필요로 하는 경우에만 사용되어야 합니다. 그렇지 않을 경우 나중에 **ntpd**로 시간 조절을 실시할 경우 제대로 작동하지 않게 됩니다.

디폴트 NetworkManager 설정으로 이 서비스를 실행하는데 문제가 발생할 수 있습니다. Red Hat Enterprise Linux 운용 가이드에 설명된 대로 **NETWORKWAIT=1**을 **/etc/sysconfig/network**에 추가해야 할 수 있습니다.

4.8. Kerberos (커 베 로 스)

Red Hat Enterprise Linux 6에서 Kerberos 클라이언트 및 서버 (KDC 포함)는 디폴트로 **des-cbc-crc**, **des-cbc-md4**, **des-cbc-md5**, **des-cbc-raw**, **des3-cbc-raw**, **des-hmac-sha1**, **arcfour-hmac-exp** 암호에 대한 키를 사용하지 않습니다. 디폴트로 클라이언트는 이러한 유형의 키를 갖는 서비스에 인증할 수 없습니다.

대부분의 서비스는 키 탭에 새로운 키 집합 (보다 강력한 암호화 문장을 사용하는 키 포함)을 추가할 수 있고 작업 중단 시간이 발생하지 않습니다. 티켓 부여 서비스 키도 kadmin **cpw -keepold** 명령을 사용하여 보다 강력한 암호화 문장을 사용하는 키가 포함된 키 집합을 업데이트할 수 있습니다.

약한 암호화 문장을 계속해서 사용해야 하는 시스템은 임시 해결 방법으로 **/etc/krb5.conf** 파일의 *libdefaults* 섹션에 있는 **allow_weak_crypto** 옵션을 필요로합니다. 이 변수는 디폴트로 *false*로 설정되어 있으므로 이 옵션을 활성화하지 않으면 인증 실패합니다:

```
[libdefaults]
allow_weak_crypto = yes
```

또한 사용 가능한 공유 라이브러리와 애플리케이션에서 지원되는 인증 메커니즘 모두에서 Kerberos IV에 대한 지원이 제거되었습니다. 잠금 정책에 대한 새로운 기능 지원에는 데이터베이스 덤프 형식으로의 변경이 필요합니다. 이전 KDC가 사용할 수 있는 형식에서 데이터베이스를 덤프해야 하는 마스터 KDC는 **-r13** 옵션과 함께 kdb5_util의 **dump** 명령을 실행해야 합니다.

4.9. Mail

4.9.1. Sendmail

Red Hat Enterprise Linux 5의 일부 릴리즈에서 *sendmail* Mail Transport Agent (MTA)는 디폴트로 외부 호스트에서 네트워크 연결을 허용합니다. Red Hat Enterprise Linux 6에서 *sendmail*은 디폴트오 로컬 시스템 (localhost)에서의 연결만을 허용합니다. 원격 호스트에 대해 서버 역할을 하는 *sendmail* 기능을 부여하려면 다음 단계 중 하나를 실행합니다:

- **/etc/mail/sendmail.mc**를 편집하고 네트워크 장치에서 수신하기 위해 **DAEMON_OPTIONS** 행을 변경합니다.
- **/etc/mail/sendmail.mc**에 있는 **DAEMON_OPTIONS** 행을 주석 처리합니다.

이러한 변경 사항을 적용하려면 *sendmail-cf* 패키지를 설치하고 **/etc/mail/sendmail.cf**를 다시 생성합니다. 이를 위해 다음 명령을 실행합니다:

```
su -c 'yum install sendmail-cf'
su -c 'make -C /etc/mail'
```

4.9.2. Exim

Exim은 Red Hat Enterprise Linux 6에서 삭제되었습니다. Postfix는 디폴트이며 권장 MTA입니다.

4.9.3. Dovecot

Dovecot 설정

Dovecot 2.x의 설정이 변경되었습니다. 마스터 설정 파일 **/etc/dovecot.conf**는 **/etc/dovecot/dovecot.conf**로 이동하였으며 Dovecot 설정의 다른 부분은 **/etc/dovecot/conf.d/*.conf**로 이동하였습니다. 대부분의 설정은 동일하며 새로운 버전과 호환됩니다. 하지만 다음 명령을 사용하여 사용자의 설정을 테스트하여 새로운 버전에서 이름 변경되고 삭제되거나 또는 다른 방법으로 변경된 옵션 목록을 나열할 수 있습니다:

```
doveconf [-n] -c /old/dovecot.conf
```

4.10. MySQL®

DBD 드라이버

MySQL DBD 드라이버는 이중 라이센스를 가지고 있으며, 관련 라이센스 문제가 해결되었습니다. 그 결과 *apr-util-mysql* 패키지는 Red Hat Enterprise Linux 6 소프트웨어 리포지터리에 포함되어 있습니다.

4.11. PostgreSQL

데이터베이스 업그레이드

PostgreSQL 8.4 (*postgresql84-** 패키지)를 사용하여 기존 Red Hat Enterprise Linux 5에서 업그레이드하려면 드롭인 대체 용으로 Red Hat Enterprise 6 PostgreSQL 패키지가 실행되어야 합니다.

하지만 PostgreSQL 8.1 (*postgresql-** 패키지) 또는 그 이전 버전을 사용하고 있는 Red Hat Enterprise Linux 5 설치에서 업그레이드하고 있고 보관되어야 하는 기존 데이터베이스 내용이 있는 경우 데이터 형식이 변경되기 때문에 덤프 단계를 실행하고 설명된 절차를 다시 불러와야 합니다. (http://www.postgresql.org/docs/8.4/interactive/install-upgrading.html) Red Hat Enterprise Linux 6로 업그레이드하기 **전에** 덤프 단계를 실행해야 합니다.

기타 변경 사항

PostgreSQL 8.1에서 8.4로 마이그레이션과 관련된 응용 프로그램 호환성 문제는 http://wiki.postgresql.org/wiki/WhatsNew84에서 참조하십시오.

4.12. Squid

Squid는 3.1로 업데이트되었으며 기본 IPv6 지원을 제공합니다. 설정 파일 **/etc/squid/squid.conf**는 크게 단축되고 Squid 3.1의 구성 옵션은 변경되어 일부 이전 버전과의 하향 호환성은 완전하지 않습니다. 설정 및 기타 변경 사항에 대한 자세한 내용은 Squid 3.1 릴리즈 노트를 참조하십시오: http://www.squid-cache.org/Versions/v3/3.1/RELEASENOTES.html.

Squid는 *ncsa_auth* 및 *pam_auth*를 통해 사용자를 인증하는 기능을 제공합니다. 이러한 인증 기능의 권한

은 Red Hat Enterprise Linux 6에서 변경되었습니다. 이전 릴리즈에서는 인증에 필요한 시스템 파일을 액세스하기 위해 상승된 권한이 필요했기 때문에 *ncsa_auth* 및 *pam_auth*에 대해 setuid 플래그가 사용되었습니다. 이번 Red Hat Enterprise Linux 6에서 Squid는 이러한 기능에 대해 setuid 플래그를 필요로 하지 않습니다. 이러한 변경은 setuid 플래그를 실행할 때 존재하는 보안 위험으로 인한 것입니다. 이러한 플래그를 설정하지 않고 정상적인 기능은 유지되고 있습니다.

4.13. Bluetooth

Bluetooth 온디맨드 서비스

Bluetooth 장치를 지원하기 위해, Bluetooth 백그라운드 서비스는 이전 Red Hat Enterprise Linux 버전에서 디폴트로 시작되었습니다. 이번 릴리즈에서 Bluetooth 서비스는 필요한 시점에 온디맨드 형식으로 시작하여 장치 사용 후 30 초 후에 자동으로 정지합니다. 이는 전체적인 초기 시작 시간과 자원 소비를 줄일 수 있습니다.

4.14. Cron

Vixie cron 및 Cronie

Red Hat Enterprise Linux 6에는 *vixie-cron* 대신 *cronie* 패키지가 들어 있습니다. 이 패키지의 주요 차이점은 정기적이 작업 (매일, 매주, 매월)이 실행되는 방식에 있습니다. Cronie는 **/etc/anacrontab** 파일을 사용하며 디폴트로 다음과 같이 나타납니다:

```
# the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=45

# the jobs will be started during the following hours only
START_HOURS_RANGE=3-22

# period in days   delay in minutes   job-identifier      command

1   5  cron.daily nice run-parts /etc/cron.daily
7   25  cron.weekly nice run-parts /etc/cron.weekly
@monthly  45  cron.monthly nice run-parts /etc/cron.monthly
```

이러한 정기적인 작업은 임의 지연을 포함하여 03:00-22:00 시간대에서 매일 한번 실행됩니다. 예를 들어, *cron.daily*는 5 분 강제 지연외에 0-45 분 임의 지연이 있습니다. 4시와 5시 사이에 지연 없이 작업을 실행할 수 있습니다:

```
RANDOM_DELAY=0 # or don't use this option at all

START_HOURS_RANGE=4-5

# period in days   delay in minutes   job-identifier      command
1   0  cron.daily nice run-parts /etc/cron.daily
7   0  cron.weekly nice run-parts /etc/cron.weekly
@monthly  0  cron.monthly nice run-parts /etc/cron.monthly
```

cronie 기능은 다음과 같습니다:

- **/etc/anacrontab**에 있는 작업을 시작하기 위해 임의 지연됩니다.
- 정기적인 작업의 시간 범위는 **/etc/anacrontab**에서 정의할 수 있습니다.
- 각 cron 테이블은 CRON_TZ 변수를 사용하여 자신의 시간대를 정의할 수 있습니다.

- 디폴트로 cron 데몬은 inotify를 사용하여 테이블의 변경 사항을 확인합니다.

cronie 및 *cronie-anacron*에 대한 보다 자세한 내용은 Red Hat Enterprise Linux 운용 가이드를 참조하십시오.

4.15. Logging (로그 처리)

`dateext` 옵션이 디폴트로 **`/etc/logrotate.conf`**에서 활성화되어 있습니다. 이 옵션은 날짜 (YYYYMMDD 형식)를 표시하는 확장을 추가하여 이전 버전의 로그 파일을 보관합니다. 이전에는 번호가 파일에 추가되었습니다.

5장. 명령행 도구

다음 부분에서는 Red Hat Enterprise Linux 6에서 명령행 도구의 동작 변경 사항에 대해 설명합니다.

5.1. Grep

grep 명령의 동작은 대/소문자 문자열 검색과 관련하여 변경되었습니다. [a-z] 형식의 범위 검색을 사용하는 것은 *LC_COLLATE* 변수에 의존합니다.

LC_COLLATE=C을 설정하여 이전 동작을 저장하고 이 명령으로 범위 검색을 실행할 때 올바른 결과를 얻을 수 있습니다. 하지만, Red Hat Enterprise Linux 6에서 권장되는 범위 검색 방식은 [[:lower:]],[[:upper:]] 형식을 사용하는 것입니다.

이러한 변경 사항은 출력 결과에 크게 영향을 미칠 수 있으므로, 올바른 결과를 지속적으로 얻기 위해 스크립트 및 프로세스를 확인해야 합니다.

5.2. Sed

-i 옵션을 지정한 *sed* 명령은 읽기 전용 파일의 내용을 삭제하고 다른 보호된 파일을 삭제할 수 있습니다. 파일에 대한 권한은 파일에 대한 작업을 정의하지만, 디렉토리에 대한 권한은 디렉토리의 파일 목록에 대한 작업을 정의합니다. 이러한 이유로 **sed**는 읽기 전용 디렉토리에 있는 쓰기 가능한 파일에 **-i** 옵션의 사용을 허가하지 않고, 해당 파일에서**-i** 옵션을 사용하면 심볼릭 링크 또는 하드 링크를 삭제합니다.

5.3. Pcre

pcre 패키지는 7.8 버전으로 업데이트되었습니다. 이는 다음과 같은 동작 변경 사항을 포함합니다:

- UTF-8 검사가 이번에는 RFC 2279 대신 RFC 3629를 참조합니다. 이로 인해 이를 허용하는 문자열이 보다 제한적일 수 있습니다. 예를 들어, UTF-8 문자의 서수는 0x0010FFFF로 제한됩니다:

  ```
  $ echo -ne "\x00\x11\xff\xff" | recode UCS-4-BE..UTF8 | pcregrep --utf-8 '.'
  pcregrep: pcre_exec() error -10 while matching this line:
  ```

  RFC에 대한 자세한 내용은 http://tools.ietf.org/html/rfc3629#section-12에서 참조하십시오.
- PCRE의 이전 버전에 의해 컴파일되었던 저장된 패턴을 다시 컴파일해야 합니다. 이는 외부 메모리(예: 파일)에 미리 컴파일된 PCRE 표현을 나열하고 나중에 로드하는 응용 프로그램에 영향을 미칩니다. 이는 주로 대규모 스팸 필터와 같이 성능 상의 이유로 실행됩니다.

5.4. Shells (쉘)

쉘 바이너리 파일의 위치가 변경되었습니다. 예를 들어, **bash** 및 **ksh** 바이너리는 더이상 **/usr/bin**에 없습니다. 이러한 두 바이너리는 **/bin**에 있습니다. 바이너리의 새 위치를 가리키도록 스크립트를 업데이트해야 합니다.

5.5. Nautilus

nautilus-open-terminal 패키지는 오른쪽 클릭 **Open Terminal** 옵션을 제공하여 현재 디렉토리에 새 터미널 창을 엽니다. 이전에는 **Desktop**에서 이 옵션을 선택하면 새 터미널 창은 기본값으로 사용자의 홈 디렉토리에 설정되어 있었습니다. Red Hat Enterprise Linux 6에서는 기본값 동작은 Desktop 디렉토리에서 열립니다. (예: **~/Desktop/**) 이전 동작을 활성화하려면, 다음 명령을 사용하여 **desktop_opens_home_dir** GConf boolean을 true로 설정합니다:

```
gconftool-2 -s /aps/nautilus-open-terminal/desktop_opens_dir --type=bool true
```

6장. 데스크탑

- Red Hat Enterprise Linux 6에서 GUI 콘솔은 tty7에서 tty1로 이동하였습니다.

GDM 설정

현재 여러 GDM 설정은 GConf에서 관리하고 있습니다.

GDM 기본 Greeter는 단순 Greeter라고 부르며 GConf를 통해 설정됩니다. 기본값은 *gdm-simple-greeter.schemas* 파일의 GConf에 저장됩니다. **gconftool2** 또는 **gconf-editor**를 사용하여 이러한 값을 수정할 수 있습니다. Greeter에는 다음과 같은 옵션이 있습니다:

- /apps/gdm/simple-greeter/banner_message_enable

  ```
  false (boolean)
  ```

  배너 텍스트 메시지가 표시되는지 여부를 제어합니다.
- /apps/gdm/simple-greeter/banner_message_text

  ```
  NULL (string)
  ```

  greeter 창에서 볼 수 있도록 배너의 텍스트 메시지를 지정합니다.
- /apps/gdm/simple-greeter/logo_icon_name

  ```
  computer (string)
  ```

  greeter 로고에 사용되는 테마의 아이콘 이름을 설정합니다.
- /apps/gdm/simple-greeter/disable_restart_buttons

  ```
  false (boolean)
  ```

  로그인 창에서 다시 시작 버튼을 표시할지 여부를 제어합니다.
- /apps/gdm/simple-greeter/wm_use_compiz

  ```
  false (booleans)
  ```

  metacity 대신 compiz를 창 관리자로 사용할 지에 대한 여부를 제어합니다.

플러그인도 GConf를 사용하여 비활성화할 수 있습니다. 예를 들어, 사운드 플러그인을 비활성화하려면 다음의 키를 설정 해제합니다: **/apps/gdm/simple-greeter/settings-manager-plugins/sound/active**.

7장. 보안 및 인증

다음 부분에서는 SELinux, SSSD, LDAP,체크섬 (Checksums), PAM 등의 보안 및 인증을 위한 변경 사항에 대해 다룹니다.

7.1. SELinux

현재 **sshd** 데몬은 제한된 서비스입니다.

7.2. SSSD

SSSD (System Security Services Daemon)는 *공급자 (provider)*라고도 하는 원격 확인 및 인증 메커니즘에 대한 액세스를 제공합니다. SSSD를 사용하면 공급자가 SSSD 백 엔드로 플러그인 가능하기 때문에 로컬 및 네트워크 식별 및 인증 소스를 추출하여 모든 식별 종류의 데이터 공급자가 연결할 수 있게 합니다. *도메인 (domain)*은 사용자 정보가 포함된 데이터베이스로 공급자 식별 정보의 소스로 작동할 수 있습니다. 여러 식별 공급자가 지원되므로 두 개 이상의 식별 서버가 별도의 사용자 네임스페이스로 작동하게 할 수 있습니다. 수집된 정보는 표준 PAM과 NSS 인터페이스를 통해 프론트엔드의 애플리케이션에서 사용할 수 있습니다.

SSSD는 서비스 모음처럼 작동하고 이를 사용하는 애플리케이션에서 독립적입니다. 따라서 더이상 이러한 애플리케이션이 원격 도메인에 연결할 필요가 없으며 어떤 도메인을 사용하고 있는지를 알 필요가 없습니다. ID 및 그룹 구성원 정보의 전체적인 로컬 캐싱은 ID의 출처 (예: LDAP, NIS, IPA, DB, Samba 등)에 관계없이 운용 가능하며 향상된 성능을 제공하여 오프라인 작업시 또는 온라인 인증을 사용할 수 없을 때에도 인증을 수행할 수 있게 합니다. 또한 SSSD는 동일한 유형 (예: 여러 LDAP 공급자)의 여러 공급자군의 사용을 가능하게 하여 이러한 다른 공급자에 의해 정규화된 도메인 인증 요청을 해결하게 합니다. 보다 자세한 내용은 Red Hat Enterprise Linux 6 운용 가이드를 참조하십시오.

7.3. LDAP

OpenLDAP

OpenLDAP 서비스에 필요한 설정은 Red Hat Enterprise Linux 6에서 변경되었습니다. 이전 버전에서 **slapd**는 **/etc/openldap/slapd.conf** 파일을 통해 설정되었습니다. 현재 Red Hat Enterprise Linux 6에서 **slapd** 설정은 사전 정의된 스키마 및 DIT (Directory Information Tree)와 함께 특정 LDAP 디렉토리 (**/etc/openldap/slapd.d/**)에 저장됩니다. 설정 스키마에 대한 보다 자세한 내용은 openldap.org에서 확인하십시오. 다음 부분에서는 새 디렉토리와 작동하기 위해 이전 설정 파일을 변환하는 방법에 대한 예제를 자세히 설명합니다:

7.3.1. slapd 설정 변환

예에서 기존 **slapd** 설정에서 변환하기 위한 파일은 **/etc/openldap/slapd.conf**에 배치되어 있고 OpenLDAP 구성을 위한 새로운 디렉토리는 **/etc/openldap/slapd.d/**에 위치하고 있다고 가정합니다.

- 새 **/etc/openldap/slapd.d/** 디렉토리의 내용을 삭제합니다:

```
# rm -rf /etc/openldap/slapd.d/*
```

- **slaptest**를 실행하여 설정 파일의 유효성을 확인하고 새 설정 디렉토리를 지정합니다:

```
slaptest -f /etc/openldap/slapd.conf -F /etc/openldap/slapd.d
```

- 새 디렉토리에서 권한을 설정합니다:

```
chown -R ldap:ldap /etc/openldap/slapd.d
```

```
chmod -R 000 /etc/openldap/slapd.d
```

```
chmod -R u+rwX /etc/openldap/slapd.d
```

- 서비스가 새로운 설정 디렉토리 내에서 작동하는지 확인한 후, 이전 설정 파일을 삭제합니다:

```
rm -rf /etc/openldap/slapd.conf
```

7.4. Checksums

Red Hat Enterprise Linux는 이전 보다 더 많은 장소에서 데이터 검증 및 인증을 위해 취약한 암호 SHA-1 및 MD5 알고리즘에서 업그레이드하여 SHA-256 digest 알고리즘을 사용합니다.

7.5. PAM (Pluggable Authentication Modules)

PAM 서비스에 대한 일반 설정은 **/etc/pam.d/system-auth-ac** 파일에 저장됩니다.

인증 모듈은 이번 추가 PAM 설정 파일에 기록됩니다 :**/etc/pam.d/password-auth-ac**, **etc/pam.d/smartcard-auth-ac**, **/etc/pam.d/fingerprint-auth-ac**

sshd에 대한 PAM 모듈과 **ftpd**와 같은 기타 다른 원격 서비스는 **/etc/pam.d/system-auth** 대신 Red Hat Enterprise Linux 6에서 **/etc/pam.d/password-auth** 파일을 포함합니다.

7.6. 시스템 사용자

정적으로 할당된 UID/GID 번호의 한계 (**/usr/share/doc/setup-*/uidgid** 파일에 있는 *setup* 패키지에 의해 정의)는 100에서 (Red Hat Enterprise Linux 3, 4, 5에서) 200으로 (Red Hat Enterprise Linux 6에서) 증가했습니다. 이러한 변경은 동적으로 또는 정적으로 할당된 100-200의 UID/GID를 갖는 시스템에 영향을 미치며 일부 애플리케이션의 설치 및 실행에 장애가 발생할 수 있습니다.

UID/GID의 동적 할당은 Red Hat Enterprise Linux 6에서 499 이하의 범위입니다. *setup* 패키지에 의한 강제 없이 정적 시스템 사용자 생성의 경우 300 또는 그 이상의 범위에서 UID/GID를 사용하는 것이 좋습니다.

8장. 커널

8.1. 커널

dracut 도구 대신 *mkinitrd*를 사용합니다. 또한 커널 모듈 관리에서 **/etc/modprobe.conf**는 더이상 기본 값으로 사용할 수 없지만 수동으로 생성할 경우 사용할 수 있습니다. 다음의 *dracut* 도구 사용의 예를 참조하십시오.:

```
# mv /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r)-old.img
# dracut --force /boot/initramfs-$(uname -r).img $(uname -r)
```

9장. 패키지 및 드라이버 변경 사항

포함된 패키지 및 시스템 드라이버 목록은 Red Hat Enterprise Linux 릴리즈에서 정기적으로 변경됩니다. 이는 다음과 같은 이유로 변경됩니다: 패키지 및 드라이버가 새로운 기능을 제공하기 위해 운영 체제에 추가되거나 업데이트됩니다. 또는 기간 만료된 패키지 및 드라이버는 삭제됩니다. 패키지 및 드라이버에 대한 업스트림 프로젝트는 더이상 관리되지 않거나 하드웨어 벤더에 의해 지원되지 않는 특정 하드웨어 패키지 및 드라이버는 삭제될 수 있습니다.

다음 부분에서는 Red Hat Enterprise Linux 6에서 새롭게 업데이트되었거나 삭제 또는 사용되지 않는 패키지 및 드라이버를 나열합니다.

9.1. 시스템 설정 도구 변경 사항

system-config-bind

system-config-bind 도구는 사용되지 않으며 대체 도구없이 삭제되었습니다. Red Hat Enterprise Linux 6에서 **named.conf** 파일을 통해 네임 서버 설정을 수동으로 편집할 것을 권장합니다. 총괄적인 BIND 문서는 *bind* 패키지의 일부로 **/usr/share/doc/bind-x.y.z**에 설치되어 있습니다. 또한 설정 예제는 **/usr/share/doc/bind-x.y.z/sample** 디렉토리에 있습니다. 하지만 이전 버전에서 *system-config-bind* 도구는 표준 BIND 설정을 생성하므로 사용자의 환경에 따라 이전 설정 파일을 올바른 위치로 이동하여 충분한 테스트를 거쳐 Red Hat Enterprise Linux 6에 있는 BIND 버전으로 마이그레이션할 수 있습니다.

system-config-boot

system-config-boot 도구는 GRUB 부트로더의 그래픽 설정을 허용했습니다. Red Hat Enterprise Linux 6에서 이는 사용되지 않고 대체 도구없이 삭제되었습니다. 대부분의 사용자에게 기본 GRUB 설정이면 충분하지만, 수동 변경이 필요한 경우 **/boot/grub** 디렉토리에 있는 **grub.conf**에서 부트 설정을 액세스하여 변경할 수 있습니다. GRUB 설정에 대한 문서는 GRUB 홈페이지 - http://www.gnu.org/software/grub/에서 보실 수 있습니다.

system-config-cluster

system-config-cluster 도구는 사용되지 않아 교체없이 제거되었습니다. *ricci* 및 *luci* (*Conga* 프로젝트에서) 사용을 권장합니다.

system-config-display

system-config-display 도구는 지원되는 데스크탑 (GNOME 및 KDE)에 있는 *XRandr* 설정 도구로 대체되었습니다. 디스플레이 관리는 다음의 메뉴 옵션 중 하나에 의해 동적으로 구현되기 때문에 기본 X 서버 설치에는 명시적 설정 파일 (**xorg.conf**)이 없습니다:

GNOME: **시스템 → 기본 설정 → 디스플레이**

KDE: **시스템 설정 → 컴퓨터 관리 → 디스플레이**

알림: 명령행 유틸리티 (**xrandr**)도 디스플레이 설정에 사용할 수 있습니다. 자세한 내용은 **xrandr --help** 명령이나 또는 **man xrandr** 명령을 사용하여 man 페이지에서 볼 수 있습니다.

system-config-httpd

system-config-httpd 도구는 사용되지 않아 대체 도구없이 삭제되었습니다. 사용자는 수동으로 웹 서버를 설정해야 합니다. 설정은 **/etc/httpd** 디렉토리에서 실행할 수 있습니다. 주요 설정 파일은 **/etc/httpd/conf/httpd.conf**에 있습니다. 이 파일은 대부분의 서버 설정에 대한 파일에 주석문으로 문서화되어 있지만, 필요한 경우 Apache 웹 서버 문서가 *httpd-manual* 패키지에 들어 있습니다.

system-config-lvm

system-config-lvm 도구는 사용되지 않습니다. 사용자는 *gnome-disk-util* 또는 *lvm* 도구를 통해 논리 볼륨 관리를 수행합니다.

system-config-netboot

system-config-netboot 도구는 사용되지 않아 교체없이 제거되었습니다. Red Hat Satellite 사용을 권장합니다.

system-config-network

system-config-network 도구는 최신의 강력한 네트워크 설정 도구인 *NetworkManager*로 대체되었습니다. *NetworkManager-applet* (nm-applet)은 기본적으로 지원되는 두 데스크탑 환경에 설치되어 있으며 시스템 트레이 패널 영역에 위치하고 있습니다. 보다 자세한 내용은 NetworkManager 홈페이지에서 확인하십시오: http://projects.gnome.org/NetworkManager/.

system-config-nfs

system-config-nfs 도구는 사용되지 않아 교체없이 제거되었습니다. 사용자는 수동으로 NFS 서버 설정을 수행합니다.

system-config-rootpassword

system-config-rootpassword 도구는 강력한 사용자 관리 및 설정 도구인 *system-config-users* 도구로 대체되었습니다. root 암호는 기본 설정 (Preferences) 대화 상자의 **"시스템 사용자 및 그룹 숨기기(Hide system users and groups)"** 옵션을 체크 해제하여 *system-config-users* 도구에서 설정할 수 있습니다. root 사용자는 주요 목록에 나열되고 암호는 다른 사용자와 마찬가지로 수정될 수 있습니다.

system-config-samba

system-config-samba 도구는 사용되지 않아 교체 없이 제거되었습니다. 사용자는 수동으로 SMB 서버 설정을 수행합니다.

system-config-securitylevel

system-config-securitylevel 도구는 삭제되었습니다. 대신 사용자는 *system-config-firewall* 도구를 사용합니다.

system-config-soundcard

system-config-soundcard 도구는 제거되었습니다. 사운드 카드 검색 및 설정은 자동으로 수행됩니다.

system-config-switchmail

system-config-switchmail 도구는 사용되지 않아 대체 도구없이 제거되었습니다. Postfix는 Red Hat Enterprise Linux 6에서 선호되는 디폴트 MTA (Mail Transfer Agent)입니다. 다른 MTA를 사용하는 경우, 특정 설정 파일 및 기술에 따라 이를 수동으로 설정해야 합니다.

9.2. Bash (Bourne-Again Shell)

Red Hat Enterprise Linux 6에는 Bash 4.1 버전이 디폴트 쉘로 포함되어 있습니다. 다음 부분에서는 이번 버전에서 소개하는 호환성 문제에 대해 설명합니다.

- Bash-4.0 및 이후 버전은 중괄호 확장 (brace expansion)을 통해 변경 사항 없이 통과시키기 위해 프로세스 치환 (process substitution) 구성을 허용하여 내용의 확장은 개별적으로 지정해야 하며 각 프로세스 치환은 별도로 입력해야 합니다.

- Bash-4.0 및 이후 버전은 Posix가 지정하는 것과 마찬가지로 SIGCHLD가 wait 내부 명령(builtin)을 인터럽트할 수 있도록 하여 모든 자식을 기다리기 위해 'wait'을 사용하는 경우 더이상 SIGCHLD trap은 기존 자식마다 호출되지 않습니다.
- Bash-4.0 및 이후 버전에서는 $() 명령 치환의 closing delimiter를 검색하기 위해 Posix 규칙을 따르므로 이는 이전 버전처럼 동작하지 않습니다. 하지만 명령 치환을 평가하기 위해 서브 쉘을 생성하기 전 더 많은 구문 오류 및 해석 오류를 감지하게 됩니다.
- 프로그램 가능한 완료 코드는 명령행을 단어로 나눌 때 쉘 메타 문자 집합이 아닌 readline과 같은 구분 문자 집합을 사용하므로 프로그램 가능한 완료 코드와 readline은 일관되어야 합니다.
- read 내부 명령(builtin)이 만료되면, 지정된 변수에 입력 read를 할당 시도합니다. 이때 입력이 충분하지 않을 경우 변수는 빈 문자열로 설정되는 원인이 될 수 있습니다. 이전 버전에서는 문자의 read를 삭제했습니다.
- Bash-4.0 및 이후 버전에서 파이프라인에 있는 명령 중 하나가 명령 목록을 실행하고 있는 동안에 SIGINT에 의해 종료되었을 때 쉘은 인터럽트를 받은것 처럼 동작합니다.
- Bash-4.0 및 이후 버전에서는 **`set -e`** 옵션의 처리 방법을 변경하여 파이프라인이 실패한 경우에도 (실패한 파이프라인의 마지막 명령이 간단한 명령이 아닌 경우에도) 쉘이 존재하도록 합니다. 이는 Posix가 지정하는 것과는 다릅니다. 이 부분의 기준을 업데이트하는 작업이 진행 중입니다. Bash-4.0의 동작은 출시 시점에 합의점을 취할 것입니다.
- Bash-4.0 및 이후 버전에서는 **`"."`**가 시스템 PATH에 존재하지 않는 경우에도 **`. (source)`** 내부 명령이 파일 이름 인수에 대한 현재 디렉토리를 검색하는 원인되었던 Posix 모드의 버그가 수정되었습니다. 이 경우, Posix는 쉘이 PWD 변수 보기를 할 수 없다고 말합니다.
- Bash-4.1은 **`[[`** 명령에 대한 연산자를 사용하여 문자열을 비교할 때 현재의 로케일을 사용합니다. 이는 **`compatNN`** *shopt* 옵션 중 하나를 설정하여 이전 동작으로 전환될 수 있습니다.

정규 표현식

이미 언급된 사항 이외에, 조건 연산자 =~와 일치하는 정규 표현식의 패턴 인수에 인용 부호를 사용하면 정규 표현식 일치가 작동 중단의 원인이 될 수 있습니다. 이는 모든 아키텍처에서 발생합니다. 3.2 이전 버전의 *bash*에서는 정규 표현식 인수를 **[[** 명령의 =~ 연산자에 인용하는 효과가 지정되지 않았습니다. 실제적인 효과는 패턴 인수 이중 인용 부호가 특수 패턴 문자를 인용하기 위해 백슬래시 (backslash)를 필요로 하여 이는 이중 인용 부호 문자 확장에 의해 실행되는 백슬래시 처리를 방해하여 == 쉘 패턴과 일치하는 연산자가 인용한 문자를 처리하는 방법과 일치하지 않았습니다.

bash 3.2 버전에서 =~ 연산자에 대해 단일 및 이중 인용부호가 있는 문자열 인수에서 내부적으로 문자를 인용하는 쉘이 변경되었습니다. 이는 정규 표현식 처리 *(`.', `[', `\', `(', `), `*', `+', `?', `{', `|', `^', `$')* 에서 중요한 문자의 특정한 의미를 억제하고 이를 그대로 일치시키도록 강제합니다. 이는 == 패턴과 일치하는 연산자가 패턴 인수의 인용 부분을 다루는 방법과 일치합니다.

인용 문자열 인수 처리가 변경되고 나서, 여러 문제가 발생하였습니다. 그 중 중요한 문제는 패턴 인수의 공백 및 *bash* 3.1과 *bash* 3.2 사이에서의 서로 다른 인용된 문자열 처리 방법에 있습니다. 이 두 문제는 패턴을 유지하기 위해 쉘 변수를 사용하여 해결할 수 있습니다. **[[** 명령의 모든 피연산자에서 쉘 변수를 확장할 때 단어 분할이 실행되지 않기 때문에, 이는 변수를 지정할 때 패턴에 인용 부호를 붙이고 공백이 포함되어 있을 수 있는 단일 문자열 값을 확장하는 기능을 제공합니다. 첫 번째 문제는 백슬래시 또는 다른 인용 메커니즘을 사용하여 패턴에 있는 공백을 제거하여 해결될 수 있습니다.

Bash 4.0은 *shopt* 내부 명령(builtin)의 여러 옵션에 의해 제어되는 *호환성 수준 (compatibility level)*의 개념을 도입합니다. *compat31* 옵션이 활성화되어 있을 경우, bash는 =~ 연산자의 오른쪽 따옴표에 대해 3.1 동작으로 전환합니다.

9.3. 기타 다른 패키지 변경 사항

업데이트된 패키지

다음 표에는 Red Hat Enterprise Linux 6에서 업데이트된 패키지 및 주요 변경 사항에 대한 설명이 나열되어 있습니다.

표 9.1. 업데이트된 패키지

업데이트된 패키지	설명
OProfile	OProfile은 0.9.5 버전으로 업데이트되었습니다. 이 새로운 버전에는 Intel Atom과 i7 프로세서, AMD Family 11h 프로세서,AMD Family 10h의 IBS (Instruction Based Sampling) 기능이 포함되어 있습니다.
quota, edquota, setquota	사용자 이름 또는 사용자 ID를 인수로 받도록 합니다. 인수가 숫자로 표시되는 경우 이는 사용자 ID로 간주되며 그렇지 않을 경우 이는 ID로 자동 변환됩니다. 사용자 이름이 숫자로만 구성되는 경우 문제가 될 수 있으므로 유의하십시오. **quota** 패키지가 업데이트되었습니다. **quota**, **edquota**, **setquota** 등의 유틸리티에서 사용자 이름을 ID로 강제 변환한 **-x** 인수는 삭제되었습니다. 현재 이 기능은 **--always-resolve** 옵션으로 제공됩니다.
module-init-tools	**/etc/modprobe.conf** 디폴트로 존재하지 않습니다. 수동으로 생성할 경우 계속 사용할 수 있습니다.

사용 중지된 패키지

다음 표에서는 Red Hat Enterprise Linux 6에서 사용 중지 (삭제)된 패키지 및 대체된 내용을 나열합니다.

표 9.2. 사용 중지된 패키지

사용 중지된 패키지	대체 내용
aspell	hunspell. aspell은 빌드 종속성으로만 제공됩니다. 맞춤법 검사가 필요한 애플리케이션은 hunspell을 사용합니다.
beecrypt	NSS/OpenSSL
crash-spu-commands	없음. 특정 Cell 패키지는 더 이상 포함되지 않습니다.
dhcpv6/dhcpv6-client	dhcp/dhclient 바이너리에는 IPv6 지원 기능이 내장되어 있습니다.
elfspe2	없음. 특정 Cell 패키지는 더 이상 포함되지 않습니다.
exim	Postfix
gnbd	대신 iSCSI 사용을 권장합니다.
gnome-vfs	gvfs
ipsec-tools	Openswan
kmod-gnbd	대신 iSCSI 사용을 권장합니다.
lam	openmpi
libspe2	없음. 특정 Cell 패키지는 더 이상 포함되지 않습니다.
libspe2-devel	없음. 특정 Cell 패키지는 더 이상 포함되지 않습니다.
linuxwacom	xorg-x11-drv-wacom
mod_python	WSGI 인터페이스를 사용하는 mod_wsgi는 Python 스크립트 작성 대신으로 사용될 수 있습니다.
mkinitrd	dracut
nss_ldap	nss-pam-ldapd, pam_ldap
openmotif-2.2	openmotif-2.3
spu-tools	없음. 특정 Cell 패키지는 더 이상 포함되지 않습니다.
switchdesk	세션 관리는 세션관리를 지원하는 두 개의 세션 관리자, GDM과 KDM에 의해 실행됩니다.
syslog	rsyslog
SysVinit	upstart
vixie-cron	cronie

사용되지 않는 패키지

- qt3
- GFS1
- gcj - 성능상의 이유로 Red Hat Enterprise Linux 6에 포함되어 있으나 gcj는 차후 릴리즈에 포함되지 않을 예정입니다.

9.4. 드라이버 변경 사항

다음 부분에서는 Red Hat Enterprise Linux 6의 드라이버 변경 사항에 대해 설명합니다. 현재 모든 드라이버

는 디폴트로 initramfs로 로드됨에 유의하십시오.

사용 중지된 드라이버

- aic7xxx_old
- atp870u
- cpqarray
- DAC960
- dc395x
- gdth
- hfs
- hfsplus
- megaraid
- net/tokenring/
- paride
- qla1280
- sound/core/oss
- sound/drivers/opl3/*
- sound/pci/nm256

사용되지 않는 드라이버

- aacraid
- aic7xxx
- i2o
- ips
- megaraid_mbox
- mptlan
- mptfc
- sym53c8xx

사용 중지된 커널 구성 요소

- NBD - Network Block Device는 Red Hat Enterprise Linux 6에서 iSCSI로 대체합니다.
- HFS - Apple 파일 시스템 지원은 Red Hat Enterprise Linux 6에서 사용 중지되었습니다.
- Tux - Web Server 가속기는 Red Hat Enterprise Linux 6에서 사용 중지되었습니다.
- 비 PAE x86 커널 - 이전 Red Hat Enterprise Linux 버전에는 i686 아키텍처에 대해 여러 커널이 포함되어 있었습니다: PAE있는 커널과 PAE없는 커널. 비 PAE 하드웨어가 볼륨에서 판매된 이래 오랜 기간이 지났습니다. 이로 인해 Red Hat Enterprise Linux 6에서 PAE를 포함하는 단일 커널만이 존재합니다.
- 선행 (Anticipatory) I/O 스케줄러는 사용되지 않으며 Red Hat Enterprise Linux 6에 존재하지 않습니다. 이는 CFQ (Completely Fair Queueing) I/O 스케줄러로 대체되어 2006년 이후 Linux 커널에서 기본 I/O 스케줄러가 되었습니다. 선행 I/O 스케줄러를 사용하는 사용자는 CFQ로 작업 부하를 테스트하고 성능 상의 문제가 있을 경우 버그 보고할 것을 권장합니다. 이는 CFQ가 테스트된 모든 작업 부하에서 선행 I/O 스케줄러와 동등한 성능을 실행하기 위함이며, Red Hat은 예상에서 벗어나는 상태 (outliers)가 없다는 것을 보증하기 어렵습니다.

9.5. 라이브러리의 변경

32 비트 라이브러리는 Red Hat Enterprise Linux 6에서 디폴트로 설치되지 않습니다. **/etc/yum.conf**에

서 **multilib_policy=all**을 설정하여 이를 변경할 수 있으며 이는 시스템 전체 정책으로 multilib 정책을 설정하게 됩니다.

개정 내역

고침 6.1-39.402	**Fri Oct 25 2013**	**Rüdiger Landmann**
Rebuild with Publican 4.0.0		
고침 6.1-39.8	**2012-07-18**	**Anthony Towns**
Rebuild for Publican 3.0		
고침 6.1-39	**Wed May 18 2011**	**Scott Radvan**
6.1 릴리즈 리뷰		