Red Hat Enterprise Linux 9

9.0 릴리스 노트

Release Notes for Red Hat Enterprise Linux 9.0

Last Updated: 2022-09-04

Red Hat Enterprise Linux 9 9.0 릴리스 노트

Release Notes for Red Hat Enterprise Linux 9.0

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

법적 공지

초록

릴리스 노트는 Red Hat Enterprise Linux 9.0에서 구현된 개선 사항 및 추가 사항에 대한 고급 기능을 제공하며 이 릴리스의 알려진 문제, 중요한 버그 수정, 기술 프리뷰, 사용되지 않는 기능 및 기타 세부 사항에 대해 설명합니다.

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 CTO Chris Wright의 메시지 를 참조하십시오.

RED HAT 문서에 관한 피드백 제공

문서에 대한 피드백에 감사드립니다. 어떻게 개선할 수 있는지 알려주십시오.

특정 문구에 대한 의견 제출

1. **Multi-page HTML** 형식으로 설명서를 보고 페이지가 완전히 로드된 후 오른쪽 상단 모서리에 **피드백** 버튼이 표시되는지 확인합니다.
2. 커서를 사용하여 주석 처리할 텍스트 부분을 강조 표시합니다.
3. 강조 표시된 텍스트 옆에 표시되는 **피드백 추가** 버튼을 클릭합니다.
4. 의견을 추가하고 **제출** 을 클릭합니다.

Bugzilla를 통해 피드백 제출(등록 필요)

1. Bugzilla 웹 사이트에 로그인합니다.
2. **버전** 메뉴에서 올바른 버전을 선택합니다.
3. **Summary** (요약) 필드에 설명 제목을 입력합니다.
4. **Description** (설명) 필드에 개선을 위한 제안을 입력합니다. 문서의 관련 부분에 대한 링크를 포함합니다.
5. **버그 제출**을 클릭합니다.

1장. 개요

1.1. RHEL 9.0의 주요 변경 사항

보안
암호화 목적으로 **SHA-1** 메시지 다이제스트를 사용하는 것은 RHEL 9에서 더 이상 사용되지 않습니다. SHA-1에서 생성된 다이제스트는 해시 충돌을 찾는 데 따른 문서화된 여러 공격으로 인해 안전하지 않은 것으로 간주되지 않습니다. RHEL 핵심 암호화 구성 요소는 기본적으로 SHA-1을 사용하여 더 이상 서명을 생성하지 않습니다. RHEL 9의 애플리케이션은 보안 관련 사용 사례에서 SHA-1을 사용하지 않도록 업데이트되었습니다.

예외적으로 HMAC-SHA1 메시지 인증 코드와 UUID(Universal Unique Identifier) 값은 SHA-1을 사용하여 계속 생성할 수 있습니다. 이러한 사용 사례는 현재 보안 위험을 초래하지 않기 때문입니다. SHA-1은 Kerberos 및 WPA-2와 같은 중요한 상호 운용성 및 호환성 문제와 관련된 제한된 경우에도 사용할 수 있습니다. 자세한 내용은 FIPS 140-3과 호환되지 않는 암호화를 사용하는 RHEL 애플리케이션 목록을 참조하십시오.

여전히 SHA-1이 필요한 시스템과의 호환성 문제 해결 방법은 다음 KCS 문서를 참조하십시오.

- RHEL 9에서 RHEL 6 시스템에 대한 SSH가 작동하지 않음
- SHA-1로 서명된 패키지를 설치하거나 업그레이드할 수 없습니다.
- 'server-sig-algs' 확장을 지원하지 않는 SSH 서버 및 클라이언트와의 연결에 실패했습니다.

OpenSSL 은 버전 3.0.1로 제공되어 공급자 개념, 새로운 버전 관리 체계, 개선된 HTTP(S) 클라이언트, 새로운 프로토콜 지원, 형식 및 알고리즘 등이 추가되었습니다.

시스템 전체 **암호화 정책이** 최신 보안 기본값을 제공하도록 조정되었습니다.

OpenSSH 는 RHEL 8.5에 배포된 버전 8.0p1과 비교하여 많은 개선 사항, 버그 수정 및 보안 개선 사항을 제공하는 버전 8.7p1로 배포됩니다.

SFTP 프로토콜은 **OpenSSH** 에서 이전에 사용된 SCP/RCP 프로토콜을 대체합니다. SFTP는 보다 예측 가능한 파일 이름 처리를 제공하며 원격 측의 쉘에 의해 **glob(3)** 패턴의 확장이 필요하지 않습니다.

SELinux 정책을 커널, 메모리 오버헤드 및 기타 매개 변수로 로드하는 시간을 포함하여 SELinux 성능이 크게 향상되었습니다. 자세한 내용은 SELinux 블로그의 성능 및 공간 효율성 향상을 참조하십시오.

RHEL 9는 업스트림 버전 1.1에 **fapolicyd** 프레임워크를 제공합니다. 다른 개선 사항 중에서 이제 새로운 **rules.d/** 및 **trust.d/** 디렉토리, **fagenrules** 스크립트 및 **fapolicyd-cli** 명령을 위한 새로운 옵션을 사용할 수 있습니다.

SCAP Security Guide(SSG) 패키지는 버전 0.1.60에 제공되어 delta 맞춤, 업데이트된 보안 프로필 및 기타 개선 사항이 도입되었습니다.

자세한 내용은 4.7절. "보안"를 참조하십시오.

서명에 SHA-1 사용은 DEFAULT 암호화 정책으로 제한됩니다. HMAC를 제외하고 SHA-1은 TLS, DTLS, SSH, IKEv2, DNSSEC 및 Kerberos 프로토콜에서 더 이상 허용되지 않습니다.

시나리오에 SHA-1을 사용하여 기존 또는 타사 암호화 서명을 확인해야 하는 경우 다음 명령을 입력하여 활성화할 수 있습니다.

```
# update-crypto-policies --set DEFAULT:SHA1
```

또는 시스템 전체 암호화 정책을 LEGACY 정책으로 전환할 수 있습니다. **LEGACY** 는 또한 안전하지 않은 다른 많은 알고리즘을 사용할 수 있습니다.

Cyrus SASL은 이제 Berkeley DB 대신 GDBM을 사용하고, NSS(Network Security Services) 라이브러리는 더 이상 신뢰 데이터베이스에 대한 DBM 파일 형식을 지원하지 않습니다.

/etc/selinux/config 파일에서 **SELINUX=disabled** 옵션을 통해 SELinux를 비활성화하는 기능이 커널에서 제거되었습니다. **/etc/selinux/config** 를 통해서만 SELinux를 비활성화하면 정책이 로드되지 않고 SELinux를 사용하도록 설정하여 시스템이 시작됩니다. 시나리오에 SELinux를 비활성화해야 하는 경우 커널 명령줄에 **selinux=0** 매개 변수를 추가합니다.

RHEL 9 및 *RHEL 8의 보안 관련 주요 차이점에 대한 자세한 내용은 RHEL 9 문서 도입* 의 보안 섹션을 참조하십시오. https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/considerations_in_adopting_rhel_9/assembly_security_considerations_in-adopting-rhel-9

네트워킹
새 MultiPath TCP 데몬(mptcpd)을 사용하여 **iproute2** 유틸리티를 사용하지 않고 MultiPath TCP(MPTCP) 엔드포인트를 구성할 수 있습니다. MPTCP 하위 흐름 및 엔드포인트를 영구적으로 만들려면 NetworkManager 디스패치 스크립트를 사용하십시오.

기본적으로 NetworkManager는 이제 키 파일을 사용하여 새 연결 프로필을 저장합니다. **ifcfg** 형식은 계속 지원됩니다.

이 릴리스에 도입된 기능 및 기존 기능의 변경 사항에 대한 자세한 내용은 새 기능인 네트워킹을 참조하십시오.

이제 Wireoctets VPN 기술을 지원되지 않는 기술 프리뷰로 사용할 수 있습니다. 자세한 내용은 기술 프리뷰 - 네트워킹을 참조하십시오.

teamd 서비스 및 **libteam** 라이브러리는 더 이상 사용되지 않습니다. 교체로 네트워크 팀 대신 본딩을 구성합니다.

iptables-nft 및 **ipset** 은 더 이상 사용되지 않습니다. 이러한 패키지에는 **iptables**,**ip6tables**,**ebtables** 및 **arptables** 와 같은 유틸리티가 포함되어 있습니다. **nftables** 프레임워크를 사용하여 방화벽 규칙을 구성합니다.

더 이상 사용되지 않는 기능에 대한 자세한 내용은 더 이상 사용되지 않는 기능 - 네트워킹을 참조하십시오.

network-scripts 패키지가 제거되었습니다. NetworkManager를 사용하여 네트워크 연결을 구성합니다. 더 이상 RHEL의 일부가 아닌 기능에 대한 자세한 내용은 *RHEL 9 문서 채택 시 고려 사항의* 네트워킹 섹션을 참조하십시오.

동적 프로그래밍 언어, 웹 서버 및 데이터베이스 서버
RHEL 9.0에서는 다음과 같은 동적 프로그래밍 언어를 제공합니다.

- Node.js 16
- Perl 5.32
- PHP 8.0
- Python 3.9
- Ruby 3.0

RHEL 9.0에는 다음과 같은 버전 제어 시스템이 포함되어 있습니다.

- **Git 2.31**
- **Subversion 1.14**

다음은 RHEL 9.0과 함께 제공되는 웹 서버입니다.

- **Apache HTTP Server 2.4.51**
- **nginx 1.20**

다음 프록시 캐싱 서버를 사용할 수 있습니다.

- **Varnish Cache 6.6**
- **squid 5.2**

RHEL 9.0에서는 다음과 같은 데이터베이스 서버를 제공합니다.

- **MariaDB 10.5**
- **MySQL 8.0**
- **PostgreSQL 13**
- **Redis 6.2**

자세한 내용은 4.13절. "동적 프로그래밍 언어, 웹 서버 및 데이터베이스 서버" 를 참조하십시오.

컴파일러 및 개발 도구

시스템 툴체인

RHEL 9.0에서 사용할 수 있는 시스템 툴체인 구성 요소는 다음과 같습니다.

- **GCC 11.2.1**
- **glibc 2.34**
- **binutils 2.35.2**

RHEL 9 시스템 툴체인 구성 요소에는 POWER10에 대한 지원이 포함됩니다.

성능 도구 및 디버거

RHEL 9.0에서는 다음과 같은 성능 도구와 디버거를 사용할 수 있습니다.

- **GDB 10.2**
- **Valgrind 3.18.1**
- **SystemTap 4.6**
- **Dyninst 11.0.0**
- **elfutils 0.186**

성능 모니터링 툴

RHEL 9.0에서는 다음과 같은 성능 모니터링 도구를 사용할 수 있습니다.

- PCP 5.3.5
- Grafana 7.5.11

컴파일러 툴 세트
RHEL 9.0에서는 다음과 같은 컴파일러 툴셋을 사용할 수 있습니다.

- LLVM Toolset 13.0.1
- rust Toolset 1.58.1
- Go Toolset 1.17.7

자세한 변경 사항은 4.14절. "컴파일러 및 개발 도구" 에서 참조하십시오.

RHEL 9의 Java 구현
RHEL 9 AppStream 리포지토리에는 다음이 포함됩니다.

- OpenJDK 17 Java 런타임 환경 및 OpenJDK 17 Java 소프트웨어 개발 키트를 제공하는 **java-17-openjdk** 패키지
- OpenJDK 11 Java 런타임 환경 및 OpenJDK 11 Java 소프트웨어 개발 키트를 제공하는 **java-11-openjdk** 패키지
- OpenJDK 8 Java 런타임 환경 및 OpenJDK 8 Java 소프트웨어 개발 키트를 제공하는 **java-1.8.0-openjdk** 패키지

자세한 내용은 OpenJDK 설명서 를 참조하십시오.

Java 툴
RHEL 9.0에서 사용할 수 있는 Java 툴은 다음과 같습니다.

- Maven 3.6
- ant 1.10

자세한 내용은 4.14절. "컴파일러 및 개발 도구"를 참조하십시오.

데스크탑
GNOME 환경은 많은 새로운 기능이 있는 GNOME 3.28에서 GNOME 40으로 업데이트되었습니다.

X.org 디스플레이 서버는 더 이상 사용되지 않으며 향후 주요 RHEL 릴리스에서 제거될 예정입니다. 대부분의 경우 기본 데스크탑 세션은 이제 **Wayland** 세션입니다.

이제 NVIDIA 드라이버를 사용하는 경우 드라이버 구성에서 Wayland를 지원하는 경우 데스크탑 세션은 기본적으로 Wayland 디스플레이 프로토콜을 선택합니다. 이전 RHEL 릴리스에서 NVIDIA 드라이버는 항상 Wayland를 비활성화했습니다.

PipeWire 서비스는 이제 모든 오디오 출력 및 입력을 관리합니다. **pipeWire** 는 일반적인 사용 사례에서 **PulseAudio** 서비스와 전문적인 사용 사례에서 **JACK** 서비스를 대체합니다.

자세한 내용은 4.16절. "데스크탑"를 참조하십시오.

가상화
RHEL 9에서 **libvirt** 라이브러리는 호스트의 개별 가상화 드라이버 세트를 처리하는 모듈식 데몬을 사용합니다. 따라서 리소스 로드 최적화 및 모니터링과 같은 가상화 드라이버와 관련된 다양한 작업을 미세 조정할 수 있습니다.

이제 QEMU 에뮬레이터가 Clang 컴파일러를 사용하여 빌드되었습니다. 이를 통해 RHEL 9 KVM 하이퍼바이저는 여러 고급 보안 및 디버깅 기능을 사용할 수 있습니다. 이러한 기능 중 하나는 safeStack으로, RHEL 9에서 호스팅되는 가상 머신 (VM)을 반환 프로그래밍 (ROP) 기반 공격에 대해 훨씬 더 안전하게 만듭니다.

또한 VMTPM(Virtual Trusted Platform Module)이 완전히 지원됩니다. vTPM을 사용하면 TPM 가상 암호화 프로세서를 VM에 추가하여 암호화 키를 생성, 저장 및 관리하는 데 사용할 수 있습니다.

마지막으로 **virtiofs** 기능이 구현되어 RHEL 9 호스트와 해당 VM 간에 파일을 보다 효율적으로 공유할 수 있습니다.

이 릴리스에 도입된 가상화 기능에 대한 자세한 내용은 4.20절. "가상화" 을 참조하십시오.

1.2. 인플레이스 업그레이드

RHEL 8에서 **RHEL 9**로 인플레이스 업그레이드
현재 지원되는 인플레이스 업그레이드 경로는 다음과 같습니다.

- 다음 아키텍처의 경우 RHEL 8.6에서 RHEL 9.0으로 마이그레이션:
  - 64비트 Intel
  - 64비트 AMD
  - 64비트 ARM
  - IBM POWER 9 (little endian)
  - z13 제외의 IBM Z 아키텍처
- SAP HANA를 사용하는 시스템의 RHEL 8.6에서 RHEL 9.0까지

자세한 내용은 Red Hat Enterprise Linux에 대한 인플레이스 업그레이드 경로를 참조하십시오. 즉각적 업그레이드 수행에 대한 지침은 RHEL 8에서 RHEL 9으로 업그레이드를 참조하십시오.

RHEL 7에서 **RHEL 9**로 인플레이스 업그레이드
RHEL 7에서 RHEL 9로 직접 인플레이스 업그레이드를 수행할 수 없습니다. 그러나 RHEL 7에서 RHEL 8로 내부 업그레이드를 수행한 다음 두 번째 즉각적 업그레이드를 RHEL 9로 업그레이드할 수 있습니다. 자세한 내용은 RHEL 7에서 RHEL 8로 의 업그레이드를 참조하십시오.

1.3. RED HAT CUSTOMER PORTAL 랩

Red Hat 고객 포털 랩은 https://access.redhat.com/labs/ 에서 제공되는 고객 포털 섹션의 툴 세트입니다. Red Hat 고객 포털 랩의 애플리케이션은 성능을 개선하고, 문제를 신속하게 해결하고, 보안 문제를 식별하며, 복잡한 애플리케이션을 신속하게 배포 및 구성하는 데 도움이 될 수 있습니다. 가장 인기있는 애플리케이션 중 일부는 다음과 같습니다.

- registration Assistant
- Kickstart 생성
- Red Hat 제품 인증서
- Red Hat CVE Checker
- 커널 Oops Analyzer

- Red Hat 코드 브라우저
- VNC Configurator
- Red Hat OpenShift Container Platform Update Graph
- Red Hat Satellite 업그레이드 도우미
- JVM 옵션 구성 도구
- 로드 밸런서 구성 도구
- Red Hat OpenShift Data Foundation Supportability 및 Interoperability Checker
- Ansible Automation Platform Upgrade Assistant
- 풀당 Ceph PG(배치 그룹) 계산기

1.4. 추가 리소스

Red Hat Enterprise Linux 9의 **기능 및 제한사항과** 다른 시스템 버전과 비교하면 지식베이스 문서 Red Hat Enterprise Linux 기술 기능 및 제한사항 에서 확인할 수 있습니다.

Red Hat Enterprise Linux **라이프 사이클** 에 대한 정보는 Red Hat Enterprise Linux 라이프 사이클 문서에서 참조하십시오.

패키지 매니페스트 문서에서는 라이센스 및 애플리케이션 호환성 수준을 포함하여 RHEL 9에 대한 **패키지 목록을** 제공합니다.

애플리케이션 호환성 수준은 Red Hat Enterprise Linux 9에서 설명합니다. 애플리케이션 호환성 가이드 문서.

제거된 기능을 포함하여 **RHEL 8과 RHEL 9의 주요 차이점은** RHEL 9 채택 시 고려 사항에 설명되어 있습니다.

RHEL 8 **에서 RHEL 9로 즉각적 업그레이드를 수행하는 방법에 대한 지침은 RHEL 8에서 RHEL 9 로 업그레이드 문서를** 통해 제공됩니다.

Red Hat Insights 서비스는 알려진 기술 문제를 사전 확인, 검사 및 해결할 수 있으며 모든 RHEL 서브스크립션을 통해 사용할 수 있습니다. Red Hat Insights 클라이언트를 설치하고 시스템에 서비스를 등록하는 방법에 대한 자세한 내용은 Red Hat Insights 시작하기 페이지를 참조하십시오.

2장. 아키텍처

Red Hat Enterprise Linux 9.0은 커널 버전 5.14.0과 함께 배포되어 최소 필수 버전에서 다음 아키텍처를 지원합니다.

- AMD 및 Intel 64비트 아키텍처(x86-64-v2)
- 64비트 ARM 아키텍처(ARMv8.0-A)
- IBM Power Systems, Little Endian (POWER9)
- 64-bit IBM Z (z14)

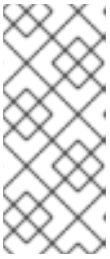
각 아키텍처에 적절한 서브스크립션을 구매해야 합니다. 자세한 내용은 Red Hat Enterprise Linux 시작하기 - 추가 아키텍처 를 참조하십시오.

3장. RHEL 9의 콘텐츠 배포

3.1. 설치

Red Hat Enterprise Linux 9는 ISO 이미지를 사용하여 설치합니다. AMD64, Intel 64비트, 64비트 ARM, IBM Power Systems, IBM Z 아키텍처에서는 두 가지 유형의 ISO 이미지를 사용할 수 있습니다.

- 설치 ISO: BaseOS 및 AppStream 리포지토리가 포함된 전체 설치 이미지로, 추가 리포지토리 없이 설치를 완료할 수 있습니다. 제품 다운로드 페이지에서 **설치 ISO** 를 **Binary DVD** 라고 합니다.



참고

설치 ISO 이미지는 여러 GB 크기로 되어 있으며, 그 결과 광 미디어 형식에 맞지 않을 수 있습니다. 설치 ISO 이미지를 사용하여 부팅 가능한 설치 미디어를 생성할 때 USB 키 또는 USB 하드 드라이브를 사용하는 것이 좋습니다. Image Builder 툴을 사용하여 사용자 지정 RHEL 이미지를 생성할 수도 있습니다. 이미지 빌더에 대한 자세한 내용은 *사용자 지정 RHEL 시스템 이미지* 완료 문서를 참조하십시오.

- 부팅 ISO: 설치 프로그램으로 부팅하는 데 사용되는 최소 부팅 ISO 이미지입니다. 설치 프로그램으로 부팅하는 데 사용하는 최소 부트 ISO 이미지입니다. 리포지토리는 설치 ISO 이미지의 일부입니다. 설치 중에 Red Hat CDN 또는 Satellite에 등록하여 Red Hat CDN 또는 Satellite의 최신 BaseOS 및 AppStream 콘텐츠를 사용할 수도 있습니다.

ISO 이미지 다운로드, 설치 미디어 생성 및 RHEL 설치 완료에 대한 지침은 표준 RHEL 9 설치 수행 문서를 참조하십시오. 자동화된 Kickstart 설치 및 기타 고급 주제 는 고급 RHEL 9 설치 문서를 참조하십시오.

3.2. 리포지토리

Red Hat Enterprise Linux 9는 다음 두 가지 주요 리포지토리를 통해 배포됩니다.

- BaseOS
- AppStream

두 리포지토리 모두 기본 RHEL 설치에 필요하며 모든 RHEL 서브스크립션을 통해 사용할 수 있습니다.

BaseOS 리포지토리의 콘텐츠는 모든 설치의 기반이 되는 기본 OS 기능의 코어 세트를 제공하는 데 사용됩니다. 이 콘텐츠는 RPM 형식으로 사용 가능하며 이전 RHEL 릴리스와 비슷한 지원 조건이 적용됩니다. 자세한 내용은 지원 범위 세부 정보 문서를 참조하십시오.

AppStream 리포지토리의 콘텐츠에는 다양한 워크로드 및 사용 사례를 지원하는 추가 사용자 공간 애플리케이션, 런타임 언어 및 데이터베이스가 포함됩니다.

또한 CodeReady Linux Builder 리포지토리는 모든 RHEL 서브스크립션을 통해 사용할 수 있습니다. 이는 개발자가 사용할 수 있는 추가 패키지를 제공합니다. CodeReady Linux Builder 리포지토리에 포함된 패키지는 지원되지 않습니다.

RHEL 9 리포지토리 및 제공하는 패키지에 대한 자세한 내용은 패키지 매니페스트 를 참조하십시오.

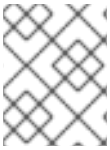
3.3. APPLICATION STREAMS

여러 사용자 공간 구성 요소의 버전은 Application Streams로 제공되며 핵심 운영 체제 패키지보다 자주 업데이트됩니다. 따라서 플랫폼 또는 특정 배포의 기본 안정성에 영향을 주지 않고 RHEL을 사용자 정의할 수 있는 유연성이 향상됩니다.

Application Streams는 친숙한 RPM 형식, 모듈이라는 RPM 형식에 대한 확장, Software Collections로 또는 flatpaks로 사용할 수 있습니다.

각 Application Stream 구성 요소에는 RHEL 9와 동일한 라이프 사이클이 있거나 더 짧습니다. RHEL 라이프 사이클 정보는 Red Hat Enterprise Linux 라이프 사이클을 참조하십시오.

RHEL 9는 기존 **dnf install** 명령을 사용하여 RPM 패키지로 설치할 수 있는 초기 Application Stream 버전을 제공하여 Application Streams 환경을 향상시킵니다.



참고

RPM 형식의 특정 초기 Application Streams는 Red Hat Enterprise Linux 9보다 라이프사이클이 짧습니다.

일부 추가 Application Stream 버전은 향후 마이너 RHEL 9 릴리스에서 라이프 사이클이 짧은 모듈로 배포됩니다. 모듈은 논리 단위, 애플리케이션, 언어 스택, 데이터베이스 또는 툴 세트를 나타내는 패키지 컬렉션입니다. 이러한 패키지는 함께 빌드, 테스트, 릴리스됩니다.

설치하려는 Application Stream 버전을 항상 결정하고 먼저 Red Hat Enterprise Linux Application Stream Lifecycle 을 검토하십시오.

대체 컴파일러 및 컨테이너 툴과 같이 빠른 업데이트가 필요한 콘텐츠는 대체 버전을 동시에 제공하지 않는 롤링 스트림에서 사용할 수 있습니다. 롤링 스트림은 RPM 또는 모듈로 패키징할 수 있습니다.

RHEL 9 및 애플리케이션 호환성 수준에서 사용 가능한 Application Streams에 대한 자세한 내용은 패키지 매니페스트 를 참조하십시오. 애플리케이션 호환성 수준은 Red Hat Enterprise Linux 9에서 설명합니다. 애플리케이션 호환성 가이드 문서.

3.4. YUM/DNF를 통한 패키지 관리

Red Hat Enterprise Linux 9에서는 **DNF에서 소프트웨어 설치를 보장합니다.** Red Hat은 이전 주요 RHEL 버전과의 일관성을 위해 **yum** 용어를 계속 지원합니다. **yum** 대신 **dnf** 를 입력하는 경우 둘 다 호환성을 위한 별칭이므로 명령이 예상대로 작동합니다.

RHEL 8 및 RHEL 9는 **DNF** 를 기반으로 하지만 RHEL 7에서 사용되는 **YUM** 과 호환됩니다.

자세한 내용은 DNF 툴을 사용하여 소프트웨어 관리를 참조하십시오.

4장. 새로운 기능

이 부분에서는 Red Hat Enterprise Linux 9.0에 도입된 새로운 기능 및 주요 개선 사항에 대해 설명합니다.

4.1. 설치 프로그램 및 이미지 생성

Anaconda에서는 Satellite용 Kickstart 설치를 통해 시스템 프로비저닝에 대한 rhsm 지원

이전에는 시스템 프로비저닝이 Red Hat Satellite에 Kickstart 설치를 위한 사용자 지정 **%post** 스크립트에 의존했습니다. 이 **%post** 스크립트는 사용자 지정 Satellite 자체 서명 인증서를 가져와서 시스템을 등록하고, 서브스크립션을 연결하고, 리포지토리에 있는 설치된 패키지를 가져왔습니다.

RHEL 9에서는 시스템 프로비저닝을 위해 **rhsm** 명령에 Satellite 지원이 추가되었습니다. 이제 시스템 등록, RHEL 서브스크립션 연결 및 Satellite 인스턴스에서 설치와 같은 모든 프로비저닝 작업에 **rhsm** 을 사용할 수 있습니다.

(BZ#1951709)

라이센스, 시스템 및 사용자 설정 화면의 구성 설정이 표준 설치 후 비활성화되었습니다.

이전 버전에서는 RHEL 사용자가 **gnome-initial-setup** 및 로그인 화면 전에 Licensing, System (Subscription manager) 및 사용자 설정을 구성했습니다. 이번 업데이트를 통해 사용자 환경을 개선하기 위해 초기 설정 화면이 기본적으로 비활성화되어 있습니다.

사용자 생성 또는 라이센스 디스플레이에 대한 초기 설정을 실행해야 하는 경우 요구 사항에 따라 다음 패키지를 설치합니다.

1. 초기 설정 패키지를 설치합니다.

```
# dnf install initial-setup initial-setup-gui
```

2. 시스템을 재부팅하는 동안 초기 설정을 활성화합니다.

```
# systemctl enable initial-setup
```

3. 시스템을 재부팅하여 초기 설정을 확인합니다.

Kickstart 설치의 경우 패키지 섹션에 **initial-setup-gui** 를 추가하고 **initial-setup** 서비스를 활성화합니다.

```
firstboot --enable
%packages
@^graphical-server-environment
initial-setup-gui
%end
```

(BZ#1878583)

Anaconda는 대화형 설치를 위해 네트워크를 자동으로 활성화합니다.

이전에는 Kickstart 또는 부팅 옵션을 통해 네트워크를 활성화하지 않고 대화형 설치를 수행할 때 사용자가 네트워크에서 수동으로 네트워크를 활성화해야 했습니다. 이번 업데이트를 통해 Anaconda는 사용자가 네트워크 스포크를 방문하여 수동으로 활성화할 필요 없이 자동으로 네트워크를 활성화합니다.

참고

이번 업데이트에서는 **ip=** boot 옵션을 사용하여 Kickstart 설치 및 설치에 대한 설치 환경을 변경하지 않습니다.

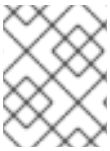
(BZ#1978264)

이미지 빌더에서 파일 시스템 구성 지원

이번 개선된 기능을 통해 credentials에서 사용자 정의 파일 시스템 구성을 지정할 수 있으며 원하는 디스크 레이아웃을 사용하여 이미지를 생성할 수 있습니다. 결과적으로 기본이 아닌 레이아웃을 사용하면 보안 벤치마크의 이점, 기존 설정, 성능 및 디스크 외부 오류로부터 보호 기능을 활용할 수 있습니다.

greater에서 파일 시스템 구성을 사용자 정의하려면 다음 사용자 지정을 설정합니다.

```
[[customizations.filesystem]]
mountpoint = "MOUNTPOINT"
size = MINIMUM-PARTITION-SIZE
```



참고

청사진에 파일 시스템 사용자 지정을 추가한 후 파일 시스템이 LVM 파티션으로 변환됩니다.

(BZ#2011448)

루트 계정을 잠그기 위한 새로운 옵션 및 **암호로 루트 SSH 로그인 허용**

RHEL 그래픽 설치의 루트 암호 구성 화면에 다음과 같은 새로운 옵션이 추가되었습니다.

- root 계정을 잠급니다. 이 옵션을 사용하여 시스템에 대한 root 액세스를 잠급니다.
- 암호로 root SSH 로그인을 허용합니다. 암호 기반 SSH 루트 로그인을 활성화하려면 이 옵션을 사용합니다.

암호 기반 SSH 루트 로그인 을 활성화하려면 설치 프로세스를 시작하기 전에 Kickstart 파일에 다음 행을 추가합니다.

```
%post
echo "PermitRootLogin yes" > /etc/ssh/sshd_config.d/01-permitrootlogin.conf
%end
```

(BZ#1940653)

이미지 빌더에서 부팅 가능한 설치 프로그램 이미지 생성 지원

이번 개선된 기능을 통해 이미지 빌더를 사용하여 루트 파일 시스템이 포함된 **tarball** 파일로 구성된 부팅 가능한 ISO 이미지를 생성할 수 있습니다. 결과적으로 부팅 가능한 ISO 이미지를 사용하여 베어 메탈 시스템에 **tarball** 파일 시스템을 설치할 수 있습니다.

(BZ#2019318)

4.2. EDGE용 RHEL

Edge용 **RHEL**은 기본적으로 **Greenboot** 내장 상태 점검을 지원

이번 업데이트를 통해 이제 Edge **Greenboot** 용 RHEL에는 재부팅하는 동안 하드웨어가 중단되거나 정지되지 않도록 **워치독** 기능에 기본 상태 점검이 포함되어 있습니다. 이를 통해 다음과 같은 기능을 활용할 수 있습니다.

- 이를 통해 **워치독** 하드웨어 사용자가 내장 상태 점검을 쉽게 채택할 수 있습니다.
- 기본 제공 OS 구성 요소에 대한 값을 제공하는 기본 상태 점검 세트
- **워치독** 이 기본 사전 설정으로 제공되므로 이 기능을 쉽게 활성화하거나 비활성화할 수 있습니다.
- 이미 사용 가능한 상태 점검을 기반으로 사용자 정의 상태 점검을 생성하는 기능입니다.

(BZ#2083036)

RHEL 9는 rpm-ostree v2022.2를 제공합니다.

RHEL 9는 여러 버그 수정 및 개선 사항을 제공하는 **rpm-ostree** 버전 v2022.2와 함께 배포됩니다. 주요 변경 사항은 다음과 같습니다.

- 새로운 **--append-if-missing** 및 **--delete-if-present** kargs 플래그를 사용하여 커널 인수를 멱등 방식으로 업데이트할 수 있습니다.
- DNF의 **Count Me** 기능은 이제 모든 리포지토리 쿼리에서 기본적으로 완전히 비활성화되며 해당 **rpm-ostree-countme.timer** 및 **rpm-ostree-countme.service** 유닛에서만 트리거됩니다. countme 를 참조하십시오.
- 후처리 논리는 이제 **user.ima** IMA 확장 특성을 처리할 수 있습니다. xattr **확장** 속성이 발견되면 시스템은 최종 **OSTree** 패키지 콘텐츠의 **security.ima** 로 자동 변환됩니다.
- **treefile** 파일에는 새 **repo-packages** 필드가 있습니다. 이를 사용하여 패키지 집합을 특정 리포지토리에 고정할 수 있습니다.

(BZ#1961324)

RHEL 9에서는 OSTree v2021.2를 제공합니다.

RHEL 9는 **OSTree** 패키지 버전 v2021.2와 함께 배포되어 여러 버그 수정 및 개선 사항을 제공합니다. 주요 변경 사항은 다음과 같습니다.

- 새 ostree-rs-ext 프로젝트에 사용되는 파일을 작성하기 위한 새로운 API로 tarball에서 가져오기를 개선합니다.
- 이제 **rofiles-fuse** 명령에서 xattr **s 확장** 속성을 처리합니다. 참고: **rofiles-adapter는 더** 이상 사용되지 않는 것으로 간주되며 #2281 을 참조하십시오.
- **인트로스펙션** API 및 테스트 개선

(BZ#1961254)

rpm-ostree rebase 툴은 RHEL 8에서 RHEL 9로의 업그레이드를 지원합니다.

이번 개선된 기능을 통해 **rpm-ostree rebase** 툴을 사용하여 RHEL 8 시스템을 RHEL 9로 업그레이드할 수 있습니다. RHEL 8의 최신 업데이트에서 RHEL 9의 최신 업데이트까지 에지 업그레이드를 위해 기본 RHEL 패키지 세트를 완전히 지원합니다.

(BZ#2082306)

4.3. 서브스크립션 관리

subscription-manager syspurpose에서 시스템 용도 명령 병합

이전에는 시스템 용도 특성을 설정하는 두 가지 명령(**syspurpose** 및 **subscription-manager**)이 있었습니다. 하나의 모듈에서 모든 시스템 용도 속성을 통합하기 위해 subscription**-manager 의 모든 애드온,role, 서비스 수준**, 서비스 수준 , **사용** 명령이 새 하위 모듈인 **subscription-manager syspurpose** 로 이동되었습니다.

새 하위 모듈 외부의 기존 **subscription-manager** 명령은 더 이상 사용되지 않습니다. RHEL 9에서 **syspurpose** 명령줄 툴을 제공하는 별도의 패키지(**python3-syspurpose**)가 제거되었습니다.

이번 업데이트에서는 subscription-manager의 단일 명령을 사용하여 모든 시스템 용도 속성을 확인, 설정 및 업데이트할 수 있는 일관된 방법을 제공합니다. 그러면 기존 시스템 용도 명령이 모두 새 하위 명령으로 사용 가능한 동일한 버전으로 대체됩니다. 예를 들어 **subscription-manager 역할 --set SystemRole** 은 **subscription-manager syspurpose 역할 --set SystemRole** 이 됩니다.

새 명령, 옵션 및 기타 속성에 대한 자세한 내용은 **subscription-manager** man 페이지의 **SYSPURPOSE OPTIONS** 섹션을 참조하십시오.

(BZ#1898563)

4.4. 소프트웨어 관리

RHEL 9에서 RPM 4.16 제공

RHEL 9는 RPM 버전 4.16과 함께 배포됩니다. 주요 버그 수정 및 버전 4.14의 개선 사항은 다음과 같습니다.

- 특히 새로운 SPEC 기능:
  - 빠른 매크로 기반 종속성 생성기
  - 동적 빌드 종속성을 생성할 수 있는 **%generate_buildrequires** 섹션
  - 메타(주문되지 않음) 종속성
  - 패키지 빌드에서 병렬 처리 증가
  - 표현식의 기본 버전 비교
  - caret 버전 연산자, 틸드와 반대
  - **%elif**,**%elifos** 및 **%elifarch** 문
  - 선택 사항인 자동 패치 및 소스 번호 지정
  - **%autopatch** 에서 패치 범위를 허용
  - **%patchlist** 및 **%sourcelist** 섹션
  - 빌드 시 헤더 데이터의 UTF-8 검증
- rpm 데이터베이스는 이제 **sqlite** 라이브러리를 기반으로 합니다. 마이그레이션 및 쿼리 목적으로 **BerkeleyDB** 데이터베이스에 대한 읽기 전용 지원이 유지됩니다.

- 이전에 RPM 자체에 빌드된 트랜잭션에서 감사 로그 이벤트를 발행하기 위한 새로운 **rpm-plugin-audit** 플러그인

(JIRA:RHELPLAN-80734)

새로운 **RPM** 플러그인에서 **RPM** 트랜잭션 중 변경 사항에 대해 알립니다.

rpm 패키지 업데이트에는 fapolicyd 프레임워크를 RPM 데이터베이스와 통합하는 새로운 RPM 플러그인이 도입되었습니다. 플러그인은 RPM 트랜잭션 중에 설치 및 변경된 파일에 대해 fapolicyd 에 알립니다. 그 결과 fapolicyd 는 이제 무결성 검사를 지원합니다.

RPM 플러그인은 DNF 트랜잭션으로 제한되지 않고 RPM의 변경 사항도 처리하기 때문에 DNF 플러그인을 대체합니다.

(BZ#1942549)

RPM에서 EdDSA 공개 키 알고리즘을 지원

이번 개선된 기능을 통해 rpm 명령은 EdDSA 공개 키 알고리즘을 사용하여 서명 키를 지원합니다. 결과적으로 EdDSA를 사용하여 생성된 서명 키를 사용하여 패키지 서명 및 확인에 사용할 수 있습니다.

그러나 EdDSA를 사용하여 서명 키를 서명할 수 있지만 RSA는 GnuPG의 기본 공개 키 알고리즘으로 계속 지원됩니다.

(BZ#1962234)

RPM에서 Zstandard (zstd) 압축 알고리즘을 지원

이 향상된 기능을 통해 기본 RPM 압축 알고리즘이 Zstandard(zstd)로 전환되었습니다. 결과적으로 사용자는 대규모 트랜잭션 중에 특히 눈에 띄게 쉬운 패키지 설치의 이점을 누릴 수 있습니다.

(JIRA:RHELPLAN-117903)

새로운 DNF 옵션 exclude_from_weak_autodetect 및 exclude_from_weak

이번 개선된 기능을 통해 기본 DNF 동작이 불필요한 약한 종속성을 설치하지 않습니다. 이 동작을 수정하려면 다음 새 옵션을 사용하십시오.

- exclude_from_weak_autodetect

  활성화된 경우 exclude_from_weak_autodetect 옵션은 시스템에 설치된 패키지의 약한 종속성(Recommends: 또는 Supplements:)을 자동으로 탐지합니다. 결과적으로 이러한 약한 종속성의 공급자는 약한 종속성으로 설치되지 않지만 에서 가져온 경우 일반 종속성으로 설치됩니다. 기본값은 true입니다.

- exclude_from_weak

  활성화하면 exclude_from_weak 옵션은 약한 종속성으로 패키지를 설치할 수 없습니다 (Recommends: 또는 Supplements:). 패키지 이름 또는 글로 패키지를 지정하고 쉼표로 구분할 수 있습니다. 기본값은 [] 입니다.

(BZ#2005305)

RHEL 9에서는 libmodulemd 2.13.0을 제공합니다.

RHEL 9는 libmodulemd 패키지 버전 2.13.0과 함께 배포됩니다. 주요 버그 수정 및 버전 2.9.4에 대한 개선 사항은 다음과 같습니다.

- 모듈의 demodularized 패키지 목록에 대한 지원이 추가되었습니다.
- modulemd-validator 툴의 새로운 --type 옵션을 사용하여 modulemd-packager-v3 문서 검증 지원을 추가했습니다.
- 강제 구문 분석 정수입니다.
- 수정된 다양한 modulemd-validator 문제

(BZ#1984403)

4.5. 쉘 및 명령행 툴

bracketed paste가 기본적으로 bash 에서 활성화됨

bash readline 라이브러리 버전 **8.1**을 사용할 수 있으므로 기본적으로 **bracketed paste** 모드를 활성화합니다. 터미널에 텍스트를 붙여넣을 때 **bash** 는 텍스트를 강조 표시하고 **enter** 키를 눌러 붙여넣은 명령을 실행해야 합니다. 브래킹된 붙여넣기 모드는 실수로 악의적인 명령을 실행하지 않도록 기본 설정입니다.

특정 사용자에 대해 **bracketed paste** 모드를 비활성화하려면 **~/.inputrc** 에 다음 행을 추가합니다.

```
set enable-bracketed-paste off
```

모든 사용자에 대해 **bracketed paste** 모드를 비활성화하려면 **/etc/inputrc** 에 다음 행을 추가합니다.

```
set enable-bracketed-paste off
```

bracketed paste 모드를 비활성화할 때 명령은 **paste**에서 직접 실행되며, **Enter** 키를 눌러 확인할 필요가 없습니다.

(**BZ#2079078**)

RHEL 9에는 **powerpc-utils 1.3.9**가 포함되어 있습니다.

RHEL 9는 **powerpc-utils** 패키지 버전 **1.3.9**를 제공합니다. 주요 버그 수정 및 버전 **1.3.8**에 대한 개선 사항은 다음과 같습니다.

- **drmgr** 에서 로그 크기를 **1MB**로 늘렸습니다.
- 부팅 시 **HCIND** 어레이 크기를 고정합니다.
- **hcnmgr** 의 **HNV** 연결에 **autoconnect-slaves** 를 구현합니다.
- **hcnmgr** 에서 **HNV** 본딩 목록 연결 개선
- **hcnmgr** 의 **util-linux** 에서 **hexdump** 를 사용합니다.

- hcn-init.service 는 NetworkManager로 시작합니다.
- ofpathname 의 다중 경로용 논리적 FC 조회에 대한 고정 OF.
- ofpathname의 파티션을 사용하여 논리 조회에 대한 고정 OF OF OF to logical lookup with partitions in ofpathname.
- 5개 이상의 경로가 있는 다중 경로 장치의 고정 부팅 목록.
- devpart 의 누락된 부분 문자열 추출을 pathname의 l2of_vd()에 추가했습니다.
- 새로운 lpamumascore.
- drmgr 에서 인덱스 작업을 통해 제거했습니다.
- SYS_PATH 의 정의를 l2of_vs() 에서 pathname 의 l2of_scsi() 로 이동했습니다.
- partstat의 보안을 강화하기 위해 -x 옵션을 추가했습니다.
- lparstat 도움말 페이지의 nroff 경고 및 오류를 수정합니다.
- drmgr 에서 NUMA 기반 LMB 제거를 구현했습니다.
- hcnmgr 의 udev 이름을 사용하여pathname 경쟁이 수정되었습니다.
- NetworkManager nmcli 를 사용하여 hcnmgr 에서 본딩 인터페이스 상태를 확인합니다.
- HNV가 없는 경우 NetworkManager nmcli 를 사용하여 부팅 시 본딩 인터페이스를 정리합니다.

(BZ#1873868)

RHEL 9는 opal-prd 6.7.1과 함께 배포됩니다.

opal-prd 패키지 버전 6.7.1은 이전에 사용 가능한 버전 6.6.3에 비해 다음과 같은 주요 버그 수정 및 개선 사항을 제공합니다.

- xscom OPAL 호출으로 인한 수정된 xscom 오류 로깅 문제.
- DEBUG 빌드로 가능한 교착 상태를 수정했습니다.
- core/platform 에서 fast-reboot 가 실패하면 full_reboot 에 대한 대체 기능.
- core/cpu 에서 next_ungarded_ primary 를 수정했습니다.
- SBE(self-Boot Engine)에서 속도 제한 타이머 요청 및 타이머 상태 개선

(BZ#1869560)

RHEL 9는 lsvpd 1.7.12를 제공합니다.

RHEL 9는 lsvpd 패키지 버전 1.7.12와 함께 배포됩니다. 주요 버그 수정 및 버전 1.7.11의 개선 사항은 다음과 같습니다.

- sysvpd.에 UUID 속성을 추가했습니다.
- NVMe 펌웨어 버전이 개선되었습니다.
- 고정 PCI 장치 제조업체에서 논리를 구문 분석합니다.
- lsvpd 설정 파일에 절을 추가했습니다.

(BZ#1869564)

ppc64-diag 버전 2.7.7 사용 가능

ppc64-diag 패키지 버전 2.7.7은 RHEL 9에서 제공됩니다. 주요 버그 수정 및 버전 2.7.6에 대한 개선 사항은 다음과 같습니다.

- 단위 테스트 케이스 개선
- sysvpd.에 UUID 속성을 추가했습니다.
- rtas_errd 서비스는 Linux 컨테이너에서 실행되지 않습니다.
- 더 이상 사용되지 않는 로깅 옵션은 systemd 서비스 파일에서 더 이상 사용할 수 없습니다.

(BZ#1869567)

RHEL 9에는 Fetchmail 6.4.24포함

RHEL 9는 fetchmail 패키지 버전 6.4.24와 함께 배포됩니다. fetchmail 은 원격 이메일 검색 및 전송 유틸리티입니다.

자세한 내용은 다음을 참조하십시오.

- /usr/share/doc/fetchmail/NEWS 파일
- fetchmail(1) 매뉴얼 페이지
- 설정을 변경해야 하는 경우 SSL 관련 정보에 대한 /usr/share/doc/fetchmail/README.SSL 파일입니다.

(BZ#1999276)

RHEL 9에는 Eigen 3.4포함

RHEL 9는 eigen3 패키지 버전 3.4와 함께 배포됩니다. Eigen 3.4 는 linear algebra를 위한 C++ 템플릿 라이브러리로, 이제 POWER10 매트릭스 다이렉션 지원 지침을 지원합니다.

결과적으로 Eigen 3.4 사용자는 POWER10 시스템에서 최적화된 선형 아지브라 계산을 수행할 수 있습니다.

(BZ#2032423)

RHEL 9에서는 cdrskin 패키지를 도입합니다.

RHEL 9에는 CD, DVD 또는 BD 미디어에서 데이터를 구울 수 있는 cdrskin 패키지가 도입되었습니다. cdrskin 패키지는 RHEL 9에서 사용할 수 없는 wodim 패키지에서 cd records 실행 파일을 대체합니다.

cdrskin 패키지에는 다음이 포함됩니다.

- 광기 미디어에서 데이터를 비우기, 포맷 및 화상
- CD의 다중 세션입니다.
- 덮어 쓰기 가능한 DVD+RW, DVD-RW, DVD-RAM, BD-RE에 대한 ISO-9660 멀티세션을 에뮬레이션했습니다.

cdrskin 패키지는 cdrskin 바이너리에 대한 심볼릭 링크로 cd history 명령을 제공하므로 사용자 스크립트를 변경할 필요가 없습니다. 전체 기능 세트는 cdrskin(1) 매뉴얼 페이지를 참조하십시오.

(BZ#2015861)

redhat.rhel_backend Ansible 컬렉션은 RHEL 9 릴리스에서 지원됩니다.

이번 업데이트에서는IPMI(Intelligent Platform Management Interface) Ansible 모듈을 지원합니다. IPMI 는 BMC(Baseboard Management Controller) 장치와 통신하는 관리 인터페이스 세트의 사양입니다. IPMI 모듈 - ipmi_power 및 ipmi_boot -는 ansible-collection- redhat-rhel_ceilometer 패키지를 설치하여 액세스할 수 있는 redhat. rhel_backend 컬렉션에서 사용할 수 있습니다.

(BZ#2023381)

RHEL 9에는 util-linux-core 패키지가 도입되었습니다.

RHEL 9는 util-linux 패키지 외에도 설치된 패키지의 크기가 중요한 기능(예: buildroot, 특정 컨테이너 및 부팅 이미지)인 시나리오를 위한 util-linux-core 하위 패키지를 제공합니다.

util-linux-core 하위 패키지에는 Linux 시스템을 부팅하는 데 필요한 util-linux 유틸리티의 제한된 하위 집합이 포함됩니다(예: mount 유틸리티).

util-linux-core 하위 패키지에는 외부 종속 항목이 포함되어 있지 않습니다. 예를 들어 PAM 라이브러리의 종속으로 인해 로그인 유틸리티를 사용할 수 없습니다.

설치와 같은 표준 사용 사례의 경우 표준 util-linux 패키지를 사용합니다. util-linux 패키지는 util-linux-core 에 따라 달라집니다. 즉, util-linux -core를 설치하면util-linux-core 가 자동으로 설치됩니다.

(BZ#2079313)

4.6. 인프라 서비스

s-nail 은 mailx를 대체합니다.

s-nail 메일 처리 시스템이 mailx 유틸리티를 대체했습니다. s-nail 유틸리티는 mailx 와 호환되며 다양한 새로운 기능을 추가합니다. mailx 패키지는 더 이상 업스트림에서 유지 관리되지 않습니다.

(BZ#1940863)

tuned 2.18 사용 가능

RHEL 9는 TuneD 버전 2.18과 함께 배포됩니다. 버전 2.16에 대한 주요 변경 사항은 다음과 같습니다.

- net plugin: txqueuelen 튜닝 지원이 추가되었습니다.
- disk plugin: NVMe 디스크 튜닝에 대한 지원 추가

- tuned-gui 버그 수정.

(BZ#2003838)

RHEL 9는 mod_security_crs 3.3을 제공합니다.

RHEL 9는 mod_security_crs 패키지 버전 3.3과 함께 배포됩니다. 주요 버그 수정 및 개선 사항은 다음과 같습니다.

- 도입 된 libinjection.
- 파일 이름에 ~ 로 끝나는 백업 파일 차단
- 새로운 LDAP 삽입 및 HTTP 분할 규칙이 추가되었습니다.
- 확장 기능을 제한하기 위해 .swp 가 추가되었습니다.
- 공격 분류를 위해 Common Attack Pattern Enumeration and Classification (CAPEC) 태그를 추가했습니다.
- Nuclei , WFuzz 및 ffuf 취약점 스캐너를 감지할 수 있는 지원이 추가되었습니다.
- 소문자로 개선된 변수 (modsec3 동작 수정)
- Unix RCE가 초기화되지 않은 변수, 문자열 연결 및 글러빙 패턴을 통해 기술을 우회하기 위한 지원을 추가했습니다.
- 오래된 규칙 태그를 제거합니다. WASCTC,OWASP_TOP_10,OWASP_AppSensor/RE1, OWASP_CRS/FO/BAR. OWASP_CRS 및 attack-type 은 여전히 mod_security_crs 패키지에 포함되어 있습니다.
- crs-setup.conf 변수 tx.allowed_request_content_type 형식이 다른 변수와 같이 변경되었

습니다. 변수가 재정의된 경우 새 구분자를 위해 **crs-setup.conf** 파일의 예제를 참조하십시오.

(BZ#1947962)

RHEL 9에서 chrony 4.1제공

RHEL 9는 **chrony** 버전 **4.1**과 함께 배포됩니다. 주요 버그 수정 및 버전 **3.5**에 대한 개선 사항은 다음과 같습니다.

- **Network Time Security (NTS)** 인증 지원이 추가되었습니다. 자세한 내용은 **chrony**의 **Network Time Security (NTS)** 개요를 참조하십시오.
- 기본적으로 인증되지 않은 **NTP(Network Time Protocol)** 소스는 인증되지 않은 **NTP** 소스를 통해 신뢰할 수 있습니다. 원래 동작을 복원하려면 **chrony.conf** 파일에 **autselectmode ignore** 인수를 추가합니다.
- **RIPEMD** 키 - **RMD128,RMD160,RMD256,RMD320** 과의 인증 지원을 더 이상 사용할 수 없습니다.
- **NTPv4** 패킷에서 표준이 아닌 긴 **MAC**에 대한 지원은 더 이상 제공되지 않습니다. **chrony 2.x, MD5/SHA1** 키를 사용하는 경우 버전 **3** 옵션으로 **chrony** 를 구성해야 합니다.

또한 다음은 **RHEL 8** 버전의 **chrony** 와 다릅니다.

- **seccomp** 필터는 기본적으로 활성화됩니다(**-F 2** 는 **/etc/sysconfig/chronyd**에 설정되어 있음). **seccomp** 필터는 **mailonchange** 지시문과 충돌합니다. **/etc/chrony.conf** 에 **mailonchange** 지시문이 있는 경우 **/etc/sysconfig/chronyd** 에서 **-F 2** 설정을 제거하십시오.

(BZ#1961131)

4.7. 보안

시스템 전체 암호화 정책은 더 안전합니다.

이번 업데이트를 통해 최신 보안 기본값을 제공하도록 시스템 전체 암호화 정책이 조정되었습니다.

- 모든 정책에서 **TLS 1.0, TLS 1.1, DTLS 1.0, RC4, Camellia, DSA, 3DES, FFDHE-1024**를 비활성화했습니다.

- **LEGACY**에서 최소 **RSA** 키 크기 및 최소 **Diffie-Hellman** 매개변수 크기를 늘립니다.

- **SHA-1**을 사용하는 **TLS** 및 **SSH** 알고리즘은 해시 기반 메시지 인증 코드**(HMAC)**에서 **SHA-1**을 사용합니다.

시나리오에서 일부 비활성화된 알고리즘 및 암호를 활성화해야 하는 경우 사용자 지정 정책 또는 하위 정책을 사용합니다.

(BZ#1937651)

RHEL 9는 **OpenSSL 3.0.1**을 제공합니다.

RHEL 9는 업스트림 버전 **3.0.1**의 **openssl** 패키지를 제공하며 여기에는 이전 버전에 비해 많은 개선 사항 및 버그 수정이 포함되어 있습니다. 주요 변경 사항은 다음과 같습니다.

- 새 공급자 개념을 추가했습니다. 공급자는 알고리즘 모음이며 다양한 애플리케이션에 대해 다양한 공급자를 선택할 수 있습니다.

- 다음 형식으로 새 버전 관리 체계를 도입했습니다. ***<major>.<minor>.<patch>***.

- 인증서 관리 프로토콜**(CMP, RFC 4210)**, 인증서 요청 메시지 형식**(CRMF)** 및 **HTTP** 전송 **(RFC 6712)**에 대한 지원이 추가되었습니다.

- **GET** 및 **POST**, 리디렉션, 일반 및 **ASN.1-encoded** 콘텐츠, 프록시 및 시간 초과를 지원하는 **HTTP(S)** 클라이언트를 도입했습니다.

- 새로운 **Key Derivation Function API (EVP_KDF)** 및 **Message Authentication Code API (EVP_MAC)**가 추가되었습니다.

- **enable-ktls** 구성 옵션으로 컴파일을 통해 **Linux Kernel TLS(KTLS)** 지원이 추가되었습니다.

- CAdES-BES 서명 확인 지원이 추가되었습니다.
- CMS API에 CAdES 서명 스키마 및 속성 지원(RFC 5126)이 추가되었습니다.
- 예를 들면 다음과 같은 새로운 알고리즘에 대한 지원이 추가되었습니다.
  - KDF 알고리즘 "SINGLE" 및 "SSH".
  - MAC 알고리즘 "GMAC" 및 "KMAC".
  - KEM 알고리즘 "RSASVE".
  - 암호화 알고리즘 "AES-SIV"
- AES_GCM를 사용하여 AuthEnvelopedData 콘텐츠 유형 구조(RFC 5083)를 추가했습니다.
- PKCS12_create() 함수를 사용하여 PKCS #12 생성의 기본 알고리즘이 최신 PBKDF2 및 AES 기반 알고리즘으로 변경되었습니다.
- 새 일반 추적 API를 추가했습니다.

(BZ#1990814)

OpenSSL에 공급자 포함

RHEL 9에 포함된 버전 3.0.1의 OpenSSL 툴킷은 공급자 개념을 추가했습니다. 공급자는 알고리즘 모음이며 다양한 애플리케이션에 대해 다양한 공급자를 선택할 수 있습니다. OpenSSL에는 현재 기본 공급자인 기본 , 기본 ,fips,legacy, null 이 포함되어 있습니다.

기본적으로 OpenSSL은 RSA, DSA, DH, CAMELLIA, SHA-1 및 SHA-2와 같이 일반적으로 사용되는 알고리즘을 포함하는 기본 공급자를 로드하고 활성화합니다.

FIPS 플래그가 커널에 설정되면 OpenSSL은 FIPS 공급자를 자동으로 로드하고 FIPS 승인 알고리즘만 사용합니다. 따라서 OpenSSL을 FIPS 모드로 수동으로 전환할 필요가 없습니다.

시스템 수준에서 다른 공급자로 변경하려면 openssl.cnf 구성 파일을 편집합니다. 예를 들어 시나리오에 레거시 공급자를 사용해야 하는 경우 해당 섹션의 주석을 제거합니다.

주의

공급자를 명시적으로 활성화하면 기본 공급자의 암시적 활성화가 재정의되고 예를 들어 OpenSSH 제품군에서 시스템에 원격으로 액세스할 수 없게 될 수 있습니다.

각 공급자에 포함된 알고리즘에 대한 자세한 내용은 관련 도움말 페이지를 참조하십시오. 예를 들어 기존 공급자에 대한 OSSL_PROVIDER-legacy(7) 도움말 페이지입니다.

(BZ#2010291)

OpenSSL random bit 생성기가 CPACF 지원

openssl 패키지 릴리스에서는 OpenSSL NIST SP800-90A 호환 AES 기반 AES(DRBG)의 CP Cryptographic Functions(CPACF)에 대한 지원이 추가되었습니다.

(BZ#1871147)

OpenSSL-spkac 은 이제 SHA-1 및 SHA-256로 서명된 SPKAC 파일을 만들 수 있습니다.

openssl-spkac 유틸리티에서 이제 MD5와 다른 해시로 서명된SPKAC(공개 키 및 챌린지) 파일을 만들 수 있습니다. 이제 SHA-1 및 SHA-256 해시로 서명된 SPKAC 파일을 생성하고 확인할 수 있습니다.

(BZ#1970388)

RHEL 9 provides openCryptoki 3.17.0

RHEL 9는 openCryptoki 버전 3.17.0과 함께 배포됩니다. 주요 버그 수정 및 버전 3.16.0에 대한 개선

사항은 다음과 같습니다.

- p11sak 유틸리티는 키를 나열하는 새로운 기능을 추가합니다.
- openCryptoki 는 다음을 지원합니다.
  - OpenSSL 3.0.
  - 이벤트 알림.
  - ICA 토큰의 소프트웨어 대체
- 하드웨어 암호화 어댑터가 활성화되면 WebSphere Application Server가 더 이상 시작되지 않습니다.

RHEL 9에는 RHEL에 고유한 추가 패치가 포함된 OpenSSL이 포함되어 있습니다. 시스템이 Federal Information Processing Standards (FIPS) 모드에 있는 경우 OpenSSL은 FIPS 공급자 및 기본 공급자를 자동으로 로드하여 애플리케이션이 FIPS 공급자를 사용하도록 강제 적용합니다. 따라서 RHEL 9의 openCryptoki 동작은 업스트림과 다릅니다.

- 승인되지 않은 메커니즘이 여전히 사용 가능한 것으로 표시되더라도, OpenSSL의 크립토 작업(soft 토큰 및 ICA 토큰 소프트웨어 대체) 구현에 의존하는 토큰은 이제 FIPS 승인 메커니즘만 지원합니다.
- openCryptoki 는 두 가지 다른 토큰 데이터 형식, 즉 비FIPS 승인 알고리즘(예: DES 및 SHA1)을 사용하는 이전 데이터 형식과 FIPS 승인 알고리즘만 사용하는 새로운 데이터 형식을 지원합니다.

  FIPS 공급자는 FIPS 승인 알고리즘만 사용할 수 있으므로 이전 데이터 형식이 더 이상 작동하지 않습니다.

중요

RHEL 9에서 **openCryptoki** 를 작동하려면 시스템에서 **FIPS** 모드를 활성화하기 전에 새 데이터 형식을 사용하도록 토큰을 마이그레이션합니다. 이전 데이터 형식이 **openCryptoki 3.17** 의 기본값이므로 이 작업이 필요합니다. 이전 토큰 데이터 형식을 사용하는 기존 **openCryptoki** 설치는 시스템을 **FIPS** 사용이 변경될 때 더 이상 작동하지 않습니다.

openCryptoki 와 함께 제공되는 **pkcstok_migrate** 유틸리티를 사용하여 토큰을 새 데이터 형식으로 마이그레이션할 수 있습니다. **pkcstok_migrate** 는 마이그레이션 중에**FIPS**가 승인되지 않은 알고리즘을 사용합니다. 따라서 시스템에서 **FIPS** 모드를 활성화하기 전에 이 툴을 사용합니다. 자세한 내용은 **FIPS** 준수 마이그레이션 **- pkcstok_migrate** 유틸리티를 참조하십시오.

(BZ#1869533)

GnuTLS 버전 **3.7.3**으로 제공

RHEL 9에서는 업스트림 버전 **3.7.3**으로 **gnutls** 패키지가 제공됩니다. 이는 이전 버전에 비해 많은 개선 사항 및 버그 수정을 제공합니다.

- **FIPS 140-3** 명시적인 지표를 위한 **API**를 도입했습니다.
- **PKCS#12** 파일을 내보낼 수 있는 강화된 기본값입니다.
- 초기 데이터의 고정 타이밍 **(0** 라운드트 데이터, **0-RTT)** 교환.
- **certutil** 툴은 **CSR**(인증서 서명 요청)에 서명할 때 인증 기관**(CA)**에서 더 이상 **CRL(Certificate Revocation List)** 배포 지점을 상속하지 않습니다.

(BZ#2033220)

RHEL 9에서 **NSS 3.71** 제공

RHEL 9는 **NSS(Network Security Services)** 라이브러리 버전 **3.71**과 함께 배포됩니다. 주요 변경 사항은 다음과 같습니다.

- 레거시 DBM 데이터베이스 형식에 대한 지원이 완전히 제거되었습니다. NSS는 RHEL 9에서 SQLite 데이터베이스 형식만 지원합니다.

- PKCS #12 암호화 암호는 이제 PBE-SHA1-RC2-40 및 PBE-SHA1-2DES 대신 PBKDF2 및 SHA-256 알고리즘과 함께 AES-128-CBC를 사용합니다.

(BZ#2008320)

NSS는 1023비트보다 짧은 RSA 키를 더 이상 지원하지 않습니다.

NSS(Network Security Services) 라이브러리의 업데이트는 모든 RSA 작업의 최소 키 크기를 128에서 1023비트로 변경합니다. 즉, NSS는 더 이상 다음 기능을 수행하지 않습니다.

- RSA 키는 1023bit 미만의 비트를 생성합니다.

- RSA 키가 1023비트보다 짧은 경우 RSA 서명을 서명하거나 확인합니다.

- 1023비트보다 짧은 RSA 키를 사용하여 값을 암호화하거나 해독합니다.

(BZ#2099438)

OpenSSH는 8.7p1에서 배포됩니다.

RHEL 9에는 버전 8.7p1에 OpenSSH 가 포함되어 있습니다. 이 버전은 RHEL 8.5에 배포된 OpenSSH 버전 8.0p1에 비해 많은 개선 사항 및 버그 수정을 제공합니다.

새로운 기능

- 이전에 사용된 SCP/RCP 프로토콜 대신 SFTP 프로토콜을 사용한 전송 지원. SFTP는 보다 예측 가능한 파일 이름 처리를 제공하며 원격 측의 쉘에 의해 glob(3) 패턴의 확장이 필요하지 않습니다.

  SFTP 지원은 기본적으로 활성화되어 있습니다. 사용자의 시나리오에서 SFTP를 사용할 수 없거나 호환되지 않는 경우 -O 플래그를 사용하여 원래 SCP/RCP 프로토콜을 강제로 사용할 수

있습니다.

- file/function/line 패턴 목록을 통해 최대 디버그 로깅을 강제 적용할 수 있는 LogVerbose 구성 지시문.

- 새로운 sshd_config PerSourceMaxStartups 및 PerSourceNetBlockSize 지시문을 사용하여 클라이언트 주소 기반 속도 제한. 이는 글로벌 MaxStartups 제한보다 더 세밀한 제어를 제공합니다.

- HostbasedAcceptedAlgorithms 키워드는 이제 키 유형별로 필터링하지 않고 서명 알고리즘을 기반으로 필터링합니다.

- sshd 데몬에 glob 패턴을 사용하여 추가 구성 파일을 포함할 수 있는 sshd_config 키워드 포함

- FIDORegistryLogin이 지정한 Universal 2nd factor (U2F) 하드웨어 인증자를 지원합니다. U2F/FIDO는 웹 사이트 인증에 널리 사용되는 저렴한 이중 인증 하드웨어에 대한 오픈 표준입니다. OpenSSH 에서 FIDO 장치는 ecdsa-sk 및 ed25519-sk 및 해당 인증서 유형에 의해 지원됩니다.

- 각 사용에 대해 PIN이 필요한 FIDO 키를 지원합니다. 새 verify-required 옵션과 함께 ssh-keygen 을 사용하여 이러한 키를 생성할 수 있습니다. PIN-required 키가 사용되면 사용자에게 서명 작업을 완료하기 위한 description이 표시됩니다.

- 이제 authorized_keys 파일에서 새 verify-required 옵션을 지원합니다. 이 옵션을 사용하려면 서명하기 전에 사용자 존재 여부에 대한 토큰 확인을 선언하려면 FIDO 서명이 필요합니다. FIDO 프로토콜은 사용자 확인을 위해 여러 가지 방법을 지원하며 OpenSSH는 현재 PIN 검증만 지원합니다.

- FIDO webauthn 서명을 확인하는 지원이 추가되었습니다. webauthn 은 웹 브라우저에서 FIDO 키를 사용하기 위한 표준입니다. 이 서명은 일반 FIDO 서명과 약간 다른 형식이므로 명시적인 지원이 필요합니다.

버그 수정

- ClientAliveCountMax=0 키워드의 명확한 의미 체계. 이제 첫 번째 활성 테스트와 관계없이 첫 번째 활성 테스트 후 연결을 즉시 종료합니다.

보안

- XMSS 키 유형의 개인 키 구문 분석 코드에서 악용 가능한 정수 오버플로 버그가 수정되었습니다. 이 키 유형은 아직 실험적이며 이에 대한 지원은 기본적으로 컴파일되지 않습니다. 이식 가능한 OpenSSH에는 사용자용 autoconf 옵션이 없으므로 이를 활성화할 수 있습니다.

- Spectre, Meltdown 및 Rambleed와 같은 메모리 사이드 채널 공격과 관련된 RAM의 미사용 개인 키 보호 기능이 추가되었습니다. 이 릴리스에서는 임의의 데이터로 구성된 비교적 큰 "이전 키"에서 파생된 대칭 키와 함께 사용하지 않는 경우 개인 키를 암호화합니다(현재 16KB).

(BZ#1952957)

OpenSSH에서 기본적으로 비활성화되어 있는 로케일 전달

컨테이너 및 가상 머신과 같은 작은 이미지에서 C.UTF-8 로케일을 사용하면 기존 en_US.UTF-8 로케일을 사용하여 크기를 줄이고 성능을 향상시킵니다.

대부분의 배포에서는 기본적으로 로케일 환경 변수를 보내고 서버 측에서 허용합니다. 그러나 이는 C 또는 C.UTF-8 이외의 로케일을 glibc-langpack-en 또는 glibc-all-langpacks 패키지가 설치되어 있지 않은 서버로 사용한 클라이언트의 SSH를 통해 로그인하면 사용자 환경에 성능이 저하되었습니다. 특히 UTF-8 형식의 출력이 손상되어 일부 도구가 작동하지 않거나 자주 경고 메시지가 전송되었습니다.

이번 업데이트를 통해 OpenSSH에서 로케일 전달은 기본적으로 꺼집니다. 이렇게 하면 클라이언트가 소규모 로케일만 지원하는 설치를 최소화하여 서버에 연결하는 경우에도 로케일을 실행 가능하게 유지합니다.

(BZ#2002734)

OpenSSH는 U2F/FIDO 보안 키 지원

이전에는 하드웨어에 저장된 OpenSSH 키는 PKCS #11 표준을 통해서만 지원되었으며 SSH에서 다른 보안 키를 사용하는 것이 제한적이었습니다. U2F/FIDO 보안 키에 대한 지원은 업스트림으로 개발되었으며 이제 RHEL 9에서 구현되었습니다. 이로 인해 PKCS #11 인터페이스와 관계없이 SSH 내에서 보안 키의 유용성이 향상됩니다.

(BZ#1821501)

버전 4.6에서 제공되는 libreswan

RHEL 9에서는 업스트림 버전 4.6에서 Libreswan이 제공됩니다. 이 버전은 많은 버그 수정 및 향상된 기능, 특히 인터넷 키 교환 버전 2 (IKEv2)와 함께 사용되는 라벨이있는 IPsec에 대한 개선 사항을 제공합니다.

(BZ#2017355)

Libreswan은 기본적으로 IKEv1 패키지를 허용하지 않습니다.

인터넷 키 교환 v2 (IKEv2) 프로토콜은 현재 널리 배포되기 때문에 Libreswan은 더 이상 기본적으로 IKEv1 패킷을 지원하지 않습니다. IKEv2는 공격에 대해 더 안전한 환경과 복원력을 제공합니다. 시나리오가 IKEv1을 사용해야 하는 경우 ikev1-policy=accept 옵션을 /etc/ipsec.conf 설정 파일에 추가하여 활성화할 수 있습니다.

(BZ#2039877)

RHEL 9는 stunnel 5.62를 제공합니다.

RHEL 9는 stunnel 패키지 버전 5.62와 함께 배포됩니다. 주요 버그 수정 및 개선 사항은 다음과 같습니다.

- FIPS 모드의 시스템에서 stunnel 은 이제 항상 FIPS 모드를 사용합니다.
- NO_TLSv1.1,NO_TLSv1.2, NO_TLSv1.3 옵션의 이름이 NO_TLSv1_1,NO_TLSv1_2, NO_TLSv1_3 로 각각 변경되었습니다.
- 새로운 서비스 수준 sessionResume 옵션은 세션 재개를 활성화 및 비활성화합니다.
- LDAP는 이제 protocol 옵션을 사용하여 stunnel 클라이언트에서 지원됩니다.
- 이제 Bash 완료 스크립트를 사용할 수 있습니다.

(BZ#2039299)

RHEL 9는 nettle 3.7 . 3을 제공합니다.

RHEL 9는 nettle 패키지 3.7.3 버전을 제공하며 여러 버그 수정 및 개선 사항을 제공합니다. 주요 변경 사항은 다음과 같습니다.

- 예를 들어 Ed448,SHAKE256,AES-XTS,SIV-CMAC 와 같은 새로운 알고리즘 및 모드를 지원합니다.
- 기존 알고리즘에 대한 아키텍처별 최적화를 추가합니다.

(BZ#1986712)

RHEL 9에서는 p11-kit 0.24를 제공합니다.

RHEL 9는 0.24 버전의 p11-kit 패키지를 제공합니다. 이 버전에서는 여러 버그 수정 및 향상된 기능을 제공합니다. 특히 신뢰할 수 없는 인증 기관을 저장하기 위한 하위 디렉터리 이름이 blocklist 로 변경되었습니다.

(BZ#1966680)

Cyrus-sasl 은 Berkeley DB 대신 GDBM을 사용합니다.

이제 Cyrus-sasl 패키지가 libdb 종속성 없이 빌드되고, the sasldb 플러그인은 Berkeley DB 대신 GDBM 데이터베이스 형식을 사용합니다. 이전 Berkeley DB 형식으로 저장된 기존 SASL(Simple Authentication and Security Layer) 데이터베이스를 마이그레이션하려면 다음 구문으로 Cyrus bdb2current 도구를 사용합니다.

```
cyrusbdb2current <sasldb_path> <new_path>
```

(BZ#1947971)

RHEL 9의 SELinux 정책은 현재 커널에 대한 최신 업데이트입니다.

SELinux 정책에는 커널의 일부이기도 하는 새로운 권한, 클래스 및 기능이 포함되어 있습니다. 따라서 SELinux는 커널에서 제공하는 모든 가능성을 활용할 수 있습니다. 특히 SELinux는 권한을 부여하는 데 필요한 세분성을 개선하여 이후의 보안 이점을 제공합니다. 또한 MLS 정책에 따라 일부 시스템이 정책에

알 수 없는 권한이 포함되어 있는 경우 일부 시스템이 시작되지 않기 때문에 **MLS SELinux** 정책으로 시스템을 실행할 수 있습니다.

(BZ#1941810, BZ#1954145)

기본 **SELinux** 정책에서는 텍스트 재배치 라이브러리가 있는 명령을 허용하지 않음

selinuxuser_execmod 부울은 이제 설치된 시스템의 보안 공간을 개선하기 위해 기본적으로 꺼져 있습니다. 따라서 **SELinux** 사용자는 라이브러리 파일에 **textrel_shlib_t** 레이블이 없으면 텍스트 재배치가 필요한 라이브러리를 사용하여 명령을 입력할 수 없습니다.

(BZ#2055822)

OpenSCAP 버전 **1.3.6**에서 제공됩니다.

RHEL 9에는 버그 수정 및 개선 사항을 제공하는 버전 **1.3.6**에 **OpenSCAP**이 포함되어 있습니다.

- **--local-files** 옵션을 사용하여 검사 중에 다운로드하는 대신 원격 **SCAP** 소스 데이터 스트림 구성 요소의 로컬 사본을 제공할 수 있습니다.
- **OpenSCAP**에서는 여러 **--rule** 인수를 수락하여 명령줄에서 여러 규칙을 선택합니다.
- **--skip-rule** 옵션을 사용하여 일부 규칙 평가를 건너뛸 수 있습니다.
- **OSCAP_PROBE_MEMORY_USAGE_RATIO** 환경 변수를 사용하여 **OpenSCAP** 프로브에서 사용하는 메모리를 제한할 수 있습니다.
- **OpenSCAP**에서 **OSBuild Blueprint**를 수정 유형으로 지원합니다.

(BZ#2041782)

OSCAP Anaconda 애드온에서 새로운 애드온 이름 지원

이번 개선된 기능을 통해 **OSCAP Anaconda** 애드온 플러그인의 **Kickstart** 파일의 레거시 **org_fedora_oscap** 애드온 이름과 달리 새로운 **com_redhat_oscap** 애드온 이름을 사용할 수 있습니다.

예를 들어 Kickstart 섹션은 다음과 같이 구성할 수 있습니다.

```
%addon com_redhat_oscap
  content-type = scap-security-guide
%end
```

OSCAP Anaconda 애드온은 현재 레거시 애드온 이름과 관련이 있지만, 기존 애드온 이름에 대한 지원은 향후 주요 RHEL 버전에서 제거됩니다.

(BZ#1893753)

CVE OVAL 피드

이번 업데이트를 통해 Red Hat은 CVE OVAL 피드를 압축 형식으로 제공합니다. 더 이상 XML 파일로 사용할 수 없지만 대신 EgressIP 2 형식으로 되어 있습니다. RHEL9의 피드 위치도 이러한 변경 사항을 반영하도록 업데이트되었습니다. 타사 SCAP 스캐너에는 압축된 콘텐츠를 참조하는 내용이 표준화되지 않기 때문에 압축된 피드를 사용하는 검사 규칙에 문제가 있을 수 있습니다.

(BZ#2028435)

SCAP Security Guide 버전 0.1.60

RHEL 9에는 버전 0.1.60에 scap-security-guide 패키지가 포함되어 있습니다. 이 버전에서는 버그 수정 및 개선 사항을 제공합니다.

- PAM 스택 강화 규칙은 이제 authselect 를 구성 도구로 사용합니다.
- SCAP Security Guide는 이제 ImageStreamTag 프로파일에 대한 delta 맞춤 파일을 제공합니다. 이 맞춤형 파일은 DISA의 자동화된 organization과 SSG 자동화 콘텐츠 간의 차이점을 나타내는 프로필을 정의합니다.

(BZ#2014561)

RHEL 9.0에서 지원되는 SCAP 보안 가이드 프로필

RHEL 9.0에 포함된 SCAP 보안 가이드 컴플라이언스 프로필을 사용하면 시스템을 발행 조직의 권장 사항으로 시스템을 강화할 수 있습니다. 결과적으로 관련 수정 사항 및 SCAP 프로필을 사용하여 필요한 강화 수준에 따라 RHEL 9 시스템의 규정 준수를 구성하고 자동화할 수 있습니다.

프로파일 이름	프로파일 ID	정책 버전
정보 시스템 (ANSSI) BP-028의 보안 수준을 위한 프랑스어 국가국	**xccdf_org.ssgproject.content_profile_anssi_bp28_enhanced**	1.2
정보 시스템(ANSSI) BP-028 고가용성을 위한 프랑스어 국가국	**xccdf_org.ssgproject.content_profile_anssi_bp28_high**	1.2
정보 시스템 (ANSSI) BP-028 인테리어 수준을 위한 프랑스 국가국	**xccdf_org.ssgproject.content_profile_anssi_bp28_intermediary**	1.2
정보 시스템 (ANSSI) BP-028 Minimal Level용 프랑스 국가국	**xccdf_org.ssgproject.content_profile_anssi_bp28_minimal**	1.2
[DRAFT] CIS Red Hat Enterprise Linux 9 벤치마크 for Level 2 - 서버	**xccdf_org.ssgproject.content_profile_cis**	초안[a]
[DRAFT] CIS Red Hat Enterprise Linux 9 벤치마크 for Level 1 - 서버	**xccdf_org.ssgproject.content_profile_cis_server_l1**	DRAFT[a]
[DRAFT] CIS Red Hat Enterprise Linux 9 벤치마크 for Level 1 - Workstation	**xccdf_org.ssgproject.content_profile_cis_workstation_l1**	DRAFT[a]
[DRAFT] CIS Red Hat Enterprise Linux 9 벤치마크 for Level 2 - Workstation	**xccdf_org.ssgproject.content_profile_cis_workstation_l2**	DRAFT[a]
[DRAFT] 비독점 정보 시스템 및 조직 (NIST 800-171)에서 분류되지 않은 정보	**xccdf_org.ssgproject.content_profile_cui**	r2
ACS(Austricial Security Center) Essential Eight	**xccdf_org.ssgproject.content_profile_e8**	버전이 지정되지 않음
건강 강화 및 책임법 (HIPA)	**xccdf_org.ssgproject.content_profile_hipaa**	버전이 지정되지 않음
austrie Security Center (ACSC) ISM 공식	**xccdf_org.ssgproject.content_profile_ism_o**	버전이 지정되지 않음
[DRAFT] 보호 프로필 일반 용도 운영 체제	**xccdf_org.ssgproject.content_profile_ospp**	4.2.1

프로파일 이름	프로파일 ID	정책 버전
PCI-DSS v3.2.1 Red Hat Enterprise Linux 9에 대한 제어 기준	**xccdf_org.ssgproject.content_profile_pci-dss**	3.2.1
[DRAFT] Red Hat Enterprise Linux 9의 DISAHeader	**xccdf_org.ssgproject.content_profile_stig**	초안[b]
[DRAFT] Red Hat Enterprise Linux 9용 GUI를 통한 DISA STIG	**xccdf_org.ssgproject.content_profile_stig_gui**	DRAFT[b]

[a] CIS는 RHEL 9의 공식 벤치마크를 아직 게시하지 않았습니다.

[b] DISA는 아직 RHEL 9에 대한 공식 벤치마크를 게시하지 않았습니다.

주의

자동 수정을 통해 시스템이 작동하지 않을 수 있습니다. 테스트 환경에서 먼저 수정을 실행합니다.

(BZ#2045341,BZ#2045349,BZ #2045368 ,BZ#2045368,BZ#2045 381 ,BZ#2045381,BZ#2045 393 ,BZ#2045393,BZ#2045403)

RHEL 9에서는 fapolicyd 1.1을 제공합니다.

RHEL 9는 fapolicyd 패키지 버전 1.1과 함께 배포됩니다. 주요 개선 사항은 다음과 같습니다.

- **허용 및 거부 규칙이 포함된 파일의 /etc/fapolicyd/rules.d/ 디렉터리는 /etc/fapolicyd/fapolicyd.rules 파일을 대체합니다. 이제 fagenrules 스크립트는 이 디렉터리의 모든 구성 요소 규칙 파일을 /etc/fapolicyd/ compiled.rules 파일에 병합합니다. 자세한 내용은 새로운 fagenrules(8) 매뉴얼 페이지를 참조하십시오.**

- **RPM 데이터베이스 외부에 파일을 안정적으로 표시하기 위한 /etc/fapolicyd/fapolicyd.trust 파일 외에도 신뢰할 수 있는 파일 목록을 더 많은 파일로 분리하는 기능을 지원하는 새로운 /etc/fapolicyd/trust.d 디렉터리를 사용할 수 있습니다. 이러한 파일에 --trust-file 지시문과 함께 fapolicyd-cli -f 하위 명령을 사용하여 파일에 대한 항목을 추가할 수도 있습니다. 자세한 내용은 fapolicyd-cli(1) 및 fapolicyd.trust(13) 매뉴얼 페이지를 참조하십시오.**

- 이제 **fapolicyd trust** 데이터베이스는 파일 이름의 공백을 지원합니다.

- 이제 **fapolicyd** 는 파일을 신뢰 데이터베이스에 추가할 때 실행 파일에 올바른 경로를 저장합니다.

(BZ#2032408)

rsyslog에는 고성능 작업을 위한 **mmfields** 모듈 및 **CEF**가 포함되어 있습니다.

rsyslog에는 이제 **mmfields** 모듈을 제공하는 **rsyslog-mmfields** 하위 패키지가 포함되어 있습니다. 이는 **property replacer** 필드 추출을 사용하는 대신, 속성 교체기와 달리 모든 필드가 한 번에 추출되어 구조화된 데이터 부분 내에 저장됩니다. 결과적으로 필드 기반 로그 형식(예: **Common Event Format**)을 처리하는 데 **mmfields** 를 사용할 수 있으며 많은 수의 필드가 필요하거나 특정 필드를 재사용할 수 있습니다. 이러한 경우 **mmfields** 는 기존 **Rsyslog** 기능보다 성능이 향상됩니다.

(BZ#2027971)

별도의 **rsyslog-logrotate** 패키지에 포함된 **logrotate**

logrotate 구성은 기본 **rsyslog** 패키지와 새로운 **rsyslog-logrotate** 패키지로 분리되었습니다. 불필요한 종속성을 설치하지 않도록 로그 회전이 필요하지 않은 경우와 같이 특정 최소한의 환경에서 유용합니다.

(BZ#1992155)

sudo 가 **Python** 플러그인 지원

RHEL 9에 포함된 **sudo** 프로그램 버전 **1.9**를 사용하면 **Python**에서 **sudo** 플러그인을 작성할 수 있습니다. 이렇게 하면 **sudo** 를 보다 정확하게 특정 시나리오에 맞게 개선할 수 있습니다.

자세한 내용은 **sudo_plugin_python(8)** 매뉴얼 페이지를 참조하십시오.

(BZ#1981278)

libseccomp 버전 **2.5.2**에서 제공

RHEL 9.0에서는 업스트림 버전 **2.5.2**에서 **libseccomp** 패키지를 제공합니다. 이 버전은 이전 버전에

비해 많은 버그 수정 및 개선 사항을 제공합니다.

- Linux의 syscall 테이블을 버전 v5.14-rc7 로 업데이트했습니다.
- 알림 파일 설명자를 가져오기 위해 get_notify_fd() 함수를 Python 바인딩에 추가했습니다.
- 모든 아키텍처에 대한 다중화된 syscall 처리가 하나의 위치로 통합되었습니다.
- PPC(PowerPC) 및 MIPS 아키텍처에 다중화된 syscall 지원이 추가되었습니다.
- 커널 내에서EC COMP_IOCTL_NOTIF_ID_VALID 작업의 의미를 변경했습니다.
- SECCOMP_IOCTL_NOTIF_ID_VALID 의 이전 커널 및 새로운 사용을 지원하도록 libseccomp 파일 설명자 알림 논리를 변경했습니다.
- seccomp_load() 가 한 번만 호출할 수 있는 버그를 수정했습니다.
- 필터에 _NOTIFY 작업이 있는 경우 알림 fd 처리만 요청하도록 알림을 변경했습니다.
- seccomp_add_rule(3) manpage에 SCMP_ACT_NOTIFY 에 대한 문서를 추가했습니다.
- 유지 관리자의 GPG 키를 명확히 했습니다.

(BZ#2019887)

Clevis에서 SHA-256지원

이번 개선된 기능을 통해 Clevis 프레임워크는 RFC 7638 에서 권장하는 대로 SHA-256 알고리즘을 JSON 웹 키(JWK) 지문의 기본 해시로 지원합니다. 이전 지문(SHA-1)은 계속 지원되므로 이전에 암호화된 데이터를 해독할 수 있습니다.

(BZ#1956760)

4.8. 네트워킹

이제 커널에서 diag 모듈을 사용할 수 있습니다.

이제 diag 모듈이 커널 이미지에 포함됩니다. 이번 업데이트를 통해 ss 명령을 사용할 때 diag 모듈을 더 이상 동적으로 로드할 필요가 없습니다. 이를 통해 커널 모듈의 고객 정책에 관계없이 네트워킹 문제를 보다 효과적으로 디버깅할 수 있습니다. 커널에 포함된 모듈:

```
CONFIG_INET_DIAG
CONFIG_INET_RAW_DIAG
CONFIG_INET_TCP_DIAG
CONFIG_INET_UDP_DIAG
CONFIG_INET_MPTCP_DIAG
CONFIG_NETLINK_DIAG
CONFIG_PACKET_DIAG
CONFIG_UNIX_DIAG
```

(BZ#1948340)

새로운 코어 및 IPv4 관련 네트워킹 sysctl 커널 매개변수

RHEL 9.0 커널은 이전 RHEL 버전에 비해 다음과 같은 새로운 core 및 IPv4 네트워킹 sysctl 매개변수를 제공합니다.

- **net.core.devconf_inherit_init_net**
- **net.core.gro_normal_batch**
- **net.core.high_order_alloc_disable**
- **net.core.netdev_unregister_timeout_secs**
- **net.ipv4.fib_multipath_hash_fields**

- **net.ipv4.fib_notify_on_flag_change**
- **net.ipv4.fib_sync_mem**
- **net.ipv4.icmp_echo_enable_probe**
- **net.ipv4.ip_autobind_reuse**
- **net.ipv4.nexthop_compat_mode**
- **net.ipv4.raw_l3mdev_accept**
- **net.ipv4.tcp_comp_sack_slack_ns**
- **net.ipv4.tcp_migrate_req**
- **net.ipv4.tcp_mtu_probe_floor**
- **net.ipv4.tcp_no_ssthresh_metrics_save**
- **net.ipv4.tcp_reflect_tos**

이러한 매개변수에 대한 자세한 내용은 **kernel-doc** 패키지를 설치하고 다음 파일을 참조하십시오.

- **/usr/share/doc/kernel-doc-<version>/Documentation/admin-guide/sysctl/net.rst**
- **/usr/share/doc/kernel-doc-<version>/Documentation/networking/ip-sysctl.rst**

(BZ#2068532)

영역 간에 패킷을 전송할 때 firewalld 에서 변경된 동작

영역 기반 방화벽에서 패킷은 하나의 영역만 입력합니다. 암시적 패킷 전송은 개념 위반이며 트래픽 또는 서비스가 예기치 않게 허용할 수 있습니다. Red Hat Enterprise Linux 9에서 firewalld 서비스는 더 이상 서로 다른 두 영역 간의 암시적 패킷 전송을 허용하지 않습니다.

이 변경 사항에 대한 자세한 내용은 기술 문서 간에 패킷을 전송할 때 firewalld 의 변경된 동작을 참조하십시오.

(BZ#2029211)

기본적으로 intra-zone 전달이 활성화되었습니다.

firewalld 내부 영역 전달 기능을 사용하면 firewalld 영역 내의 인터페이스 또는 소스 간에 트래픽을 전달할 수 있습니다. RHEL 9.0부터는 이 기능이 기본적으로 활성화되어 있습니다. firewall-cmd 유틸리티의 --add-forward 옵션을 사용하여 특정 영역에 대해 intra-zone 전달을 활성화합니다. firewall-cmd --list-all 명령은 영역 전달이 활성화되었는지 또는 비활성 상태인지 표시합니다.

```
# firewall-cmd --list-all
public (active)
  ...
  forward: no
```

(BZ#2089193)

Nmstate를 더 포괄적으로 만들기

Red Hat은 인식적인 언어를 사용하기 위해 최선을 다하고 있습니다. 보다 포괄적 수용을 위한 오픈 소스 용어 교체 의 이 이니셔티브에 대한 자세한 내용을 참조하십시오. 따라서 nmstate API의 슬레이브 용어가 port 라는 이름으로 교체되었습니다.

(BZ#1969941)

NetworkManager는 IBM Z의 rd.znet_ifname 커널 옵션에 설정된 인터페이스 이름을 지원

이번 개선된 기능을 통해 NetworkManager는 네트워크에서 Red Hat Enterprise Linux를 설치하거나 부팅할 때 rd.znet 및 rd.znet_ifname 커널 명령 줄 옵션을 해석합니다. 결과적으로 기본 채널 대신 하위 채널로 식별되는 네트워크 인터페이스의 이름을 지정할 수 있습니다.

(BZ#1980387)

hostapd 패키지가 RHEL 9.0에 추가되었습니다.

이번 릴리스에서는 **RHEL**이 **hostapd** 패키지를 제공합니다. 그러나 **Red Hat**은 이더넷 네트워크에서 **802.1X** 인증 도구로 **RHEL** 호스트를 설정하는 경우에만 **hostapd** 를 지원합니다. **Wi-Fi** 액세스 포인트 또는 유선**-Fi** 네트워크의 인증자와 같은 다른 시나리오는 지원되지 않습니다.

FreeRADIUS 백엔드를 사용하여 **RHEL**을 **802.1X** 인증자로 구성하는 방법에 대한 자세한 내용은 **FreeRADIUS** 백엔드와 함께 **hostapd**를 사용하여 **LAN** 클라이언트에 대한 **802.1x** 네트워크 인증 서비스 설정을 참조하십시오.

(BZ#2019830)

버전 1.18.2에서 제공되는 ModemManager

RHEL 9.0에서는 업스트림 버전 **1.18.2**의 **ModemManager** 패키지를 제공합니다. 이 버전에는 이전 버전에 대한 버그 수정 및 개선 사항이 포함되어 있습니다. 특히 다음과 같습니다.

- **5G** 기능이 있는 장치의 기능 및 모드 처리 개선
- 추가 장치 지원

주요 변경 사항 전체 목록은 업스트림 릴리스 노트를 참조하십시오.

- https://github.com/freedesktop/ModemManager/blob/7a85bc243bc1be9f720ae1bda92e9eba7b277213/NEWS#L185
- https://github.com/freedesktop/ModemManager/blob/7a85bc243bc1be9f720ae1bda92e9eba7b277213/NEWS#L417

(BZ#1996716)

NetworkManager를 사용하여 본딩 포트의 **queue_id** 를 변경할 수 있습니다.

본딩의 **NetworkManager** 포트는 이제 **queue_id** 매개 변수를 지원합니다. **eth1** 이 본딩 인터페이스의 포트라고 가정하면 다음을 사용하여 본딩 포트에 대해 **queue_id** 를 활성화할 수 있습니다.

```
# nmcli connection modify eth1 bond-port.queue-id 1
# nmcli connection up eth1
```

이 옵션을 사용해야 하는 네트워크 인터페이스는 모든 인터페이스에 대해 적절한 우선순위가 설정될 때까지 여러 호출으로 구성해야 합니다. 자세한 내용은 **kernel-doc** 패키지에서 제공하는 **/usr/share/docs/kernel-doc-_<version>/Documentation/networking/bonding.rst** 파일을 참조하십시오.

(BZ#1949127)

최신 **NetworkManager**를 사용하여 차단, 금지 및 연결할 수 없는 경로 유형 지원

커널은 공통 유니캐스트,브로드캐스트, 로컬 경로 유형 외에 여러 경로 유형을 지원합니다. 또한 사용자는 **NetworkManager**의 연결 프로필에서 블랙홀 을 구성하여 고정 경로 유형을금지하고 연결할 수 있습니다. **NetworkManager**는 프로필이 활성화되면 프로필을 추가합니다.

(BZ#2060013)

RoCE Express Adapters가 개선된 인터페이스 이름 지정 스키마 사용

이러한 향상된 기능을 통해 **RDMA over Converged Ethernet (RoCE) Express** 어댑터는 예측 가능한 인터페이스 이름 지정 체계와 **z-system (zPCI)** 커넥터의 **Peripheral Communication Interface**를 사용합니다. 이 이름 지정 체계에서 **RHEL**은 **UID**(사용자 **ID**) 또는 함수 식별자(**FID**)를 사용하여 고유한 이름을 생성합니다. 고유한 **UID**가 없는 경우 **RHEL**은 **FID**를 사용하여 이름 지정 체계를 설정합니다.

(BZ#2091653)

4.9. 커널

RHEL 9.0의 커널 버전

Red Hat Enterprise Linux 9.0은 커널 버전 **5.14.0-70**과 함께 배포됩니다.

(BZ#2077836)

Red Hat은 기본적으로 권한 있는 사용자에 대해서만 모든 **RHEL** 버전에서 **eBPF** 를 활성화합니다.

eBPF(extended Berkeley Packet Filter)는 복잡한 기술로, 사용자가 **Linux** 커널 내에서 사용자 지정 코드를 실행할 수 있습니다. 따라서 **eBPF** 코드는 검증기 및 기타 보안 메커니즘을 통과해야 합니다. **CVE(Common Vulnerabilities and Exposures)** 인스턴스가 있으며, 이 코드의 버그는 인증되지 않은 작업을 위해 오용될 수 있었습니다. 이 위험을 완화하기 위해 **Red Hat**은 권한 있는 사용자에 대해서만 모든 **RHEL** 버전에서 기본적으로 **eBPF** 를 활성화했습니다. 커널 명령줄 매개변수 **unprivileged_bpf_disabled=0** 을 사용하여 권한이 없는 사용자에 대해 **eBPF** 를 활성화할 수 있습니다. 권한이 없는_bpf_disabled=0 을 적용하면 **Red Hat** 지원에서 커널을 분배하고 시스템이 보안 위험에 노출됩니다. 또한 **Red Hat**은 기능이 **CAP_SYS_ADMIN** 과 동일한 것처럼 **CAP_BPF** 기능을 통해 프로세스를 처리할 것을 권장하고 있습니다.

커널 명령줄 매개 변수를 적용하는 방법에 대한 자세한 내용은 커널 명령줄 매개변수 구성을 참조하십시오.

(BZ#2091643)

Red Hat은 마이너 릴리스에서 커널 기호만 보호

Red Hat은 보호된 커널 기호를 사용하여 커널 모듈을 컴파일하는 경우에만 **EUS (Extended Update Support)** 릴리스 내에서 향후 모든 업데이트에서 커널 모듈이 계속 로드되도록 합니다. **RHEL 9**의 마이너 릴리스 간에 **ABI(**커널 애플리케이션 바이너리 인터페이스**)**는 보장되지 않습니다.

(BZ#2059183)

RHEL 9 베타 커널 신뢰할 수 있는 **SecureBoot** 인증서로 서명

이전 버전에서는 **RHEL** 베타 릴리스에서는 사용자가 **MOK(Machine Owner Key)** 기능을 사용하여 별도의 베타 공개 키를 등록해야 했습니다. **RHEL 9 Beta**부터 커널은 신뢰할 수 있는 **SecureBoot** 인증서로 서명되므로 사용자는 **UEFI Secure Boot**가 활성화된 시스템에서 베타 버전을 사용하기 위해 별도의 베타 공개 키를 등록할 필요가 없습니다.

(BZ#2002499)

RHEL 9에서 **cgroup-v2** 가 기본적으로 활성화되어 있습니다

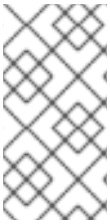
제어 그룹 버전 **2(cgroup-v2)** 기능은 제어 그룹 관리를 간소화하는 단일 계층 구조 모델을 구현합니다. 또한 한 번에 단일 제어 그룹의 멤버만 프로세스가 될 수 있습니다. **systemd** 와의 심도 있는 통합으로

RHEL 시스템에서 리소스 제어를 구성할 때 최종 사용자 환경이 향상됩니다.

새로운 기능의 개발은 대부분 **cgroup-v2** 에 대해 수행되며, **cgroup-v1** 에서는 누락된 일부 기능이 있습니다. 마찬가지로 **cgroup-v1** 에는 **cgroup-v2** 에서 누락된 일부 레거시 기능이 포함되어 있습니다. 또한 제어 인터페이스는 다릅니다. 따라서 **cgroup-v1**에 직접 종속된 타사 소프트웨어가 **cgroup-v 2** 환경에서 제대로 실행되지 않을 수 있습니다.

cgroup-v1 을 사용하려면 커널 명령줄에 다음 매개변수를 추가해야 합니다.

```
systemd.unified_cgroup_hierarchy=0
systemd.legacy_systemd_cgroup_controller
```



참고

cgroup-v1 및 **cgroup-v2** 는 커널에서 완전히 활성화됩니다. 커널 관점에서는 기본 제어 그룹 버전이 없으며 시작 시 마운트할 **systemd** 에 의해 결정됩니다.

(BZ#1953515)

커널 변경으로 인해 타사 커널 모듈에 영향을 줄 수 있음

5.9 이전 커널 버전의 **Linux** 배포판에서는 **GPL** 함수를 비**GPL** 기능으로 내보내기가 지원되었습니다. 결과적으로 **shim** 메커니즘을 통해 독점적인 기능을 **GPL** 커널 기능과 연결할 수 있었습니다. 이번 릴리스에서 **RHEL** 커널은 **shim** 을 다시 사용하여 **GPL**을 시행하는 **RHEL**의 기능을 향상시키는 업스트림 변경 사항을 통합합니다.

중요

파트너 및 **ISV**(독립 소프트웨어 벤더)는 **RHEL 9**의 초기 버전으로 커널 모듈을 테스트해야 합니다.

(BZ#1960556)

64비트 **ARM** 아키텍처에는 **RHEL 9**의 **4 KB** 페이지 크기가 있습니다.

Red Hat은 **Red Hat Enterprise Linux 9**에서 **64**비트 **ARM** 아키텍처에 대해 **4KB**의 실제 메모리 크기를 선택했습니다. 이 크기는 대부분의 **ARM** 기반 시스템에 존재하는 워크로드 및 메모리 양과 잘 연결됩

니다. 대규모 페이지 크기를 효율적으로 사용하려면 대규모 페이지 옵션을 사용하여 대규모 데이터 세트로 더 많은 메모리 또는 워크로드를 처리합니다.

대규모 페이지에 대한 자세한 내용은 시스템 상태 모니터링 및 성능 관리를 참조하십시오.

(BZ#1978382)

strace 유틸리티에서 이제 **SELinux** 컨텍스트 불일치를 올바르게 표시합니다.

strace 의 기존 **--secontext** 옵션이 **mismatch** 매개변수를 사용하여 확장되었습니다. 이 매개변수를 사용하면 불일치 시 실제 컨텍스트와 함께 예상 컨텍스트를 출력할 수 있습니다. 출력은 이중 느낌표(**!!**), 먼저 실제 컨텍스트, 다음 예상 컨텍스트로 구분됩니다. 아래 예제에서 전체,**mismatch** 매개변수는 컨텍스트의 사용자가 일치하지 않기 때문에 실제 컨텍스트와 함께 예상되는 전체 컨텍스트를 출력합니다. 그러나 독방 불일치 를 사용하는 경우 컨텍스트의 유형 부분만 확인합니다. 컨텍스트의 유형이 일치하므로 예상 컨텍스트가 인쇄되지 않습니다.

```
[...]
$ strace --secontext=full,mismatch -e statx stat /home/user/file
statx(AT_FDCWD, "/home/user/file"
[system_u:object_r:user_home_t:s0!!unconfined_u:object_r:user_home_t:s0], ...

$ strace --secontext=mismatch -e statx stat /home/user/file
statx(AT_FDCWD, "/home/user/file" [user_home_t:s0], ...
```

SELinux 컨텍스트 불일치로 인해 **SELinux**와 관련된 액세스 제어 문제가 발생하는 경우가 많습니다. 시스템 호출 추적에 출력되는 불일치는 **SELinux** 컨텍스트 정확성의 검사를 훨씬 신속하게 제공할 수 있습니다. 시스템 호출 추적은 액세스 제어 검사와 관련하여 특정 커널 동작을 설명할 수도 있습니다.

(BZ#2038965)

perf-top 이제 특정 열로 정렬할 수 있습니다.

이번 업데이트를 통해 **perf-top** 시스템 프로파일링 툴을 사용하면 임의의 이벤트 열에 따라 샘플을 정렬할 수 있습니다. 이전에는 그룹의 여러 이벤트가 샘플링된 경우 첫 번째 열에 의해 이벤트가 정렬되었습니다. 샘플을 정렬하려면 **--group-sort-idx** 명령줄 옵션을 사용하고 숫자 키를 눌러 일치하는 데이터 열로 테이블을 정렬합니다. 열 번호 지정은 **0** 에서 시작됩니다.

(BZ#1851933)

새로운 패키지: **jigawatts**

checkpoint/Restore In Userspace(CRIU)는 프로세스를 체크포인트 및 복원할 수 있는 Linux 유틸리티입니다. jigawatts 패키지에는 Java 라이브러리가 포함되어 있으며 Java 애플리케이션에서 CRIU 메커니즘의 사용 편의성을 개선하는 것을 목표로 합니다.

(BZ#1972029)

trace-cmd reset 명령에 새 동작이 있습니다.

이전에는 trace-cmd reset 명령에서 tracing_on 구성을 0으로 재설정했습니다. trace-cmd reset 의 새 동작은 tracing_on 을 기본값인 1로 재설정하는 것입니다.

(BZ#1933980)

RHEL 9에서 extended Berkeley Packet Filter가 지원됨

eBPF(Extended Berkeley Packet Filter) 는 제한된 기능 세트에 대한 액세스 권한이 있는 제한된 샌드박스 환경에서 커널 공간에서 코드를 실행할 수 있는 커널 내 가상 머신입니다. 가상 머신은 특수 어셈블리와 유사한 코드를 실행합니다.

eBPF 바이트 코드가 먼저 커널에 로드됩니다. 그런 다음 바이트 코드가 실시간 컴파일을 사용하여 네이티브 머신 코드로 확인 및 변환됩니다. 마지막으로 가상 시스템은 코드를 실행합니다.

Red Hat은 eBPF 가상 시스템을 사용하는 다양한 구성 요소를 제공합니다. RHEL 9에서는 다음과 같은 구성 요소가 포함됩니다.

- BCC(BPF Compiler Collection) 패키지는 eBPF 를 사용하여 Linux 운영 체제의 I/O 분석, 네트워킹 및 모니터링을 위한 툴을 제공합니다.
- BCC 툴 패키지에 제공된 툴과 유사한 툴을 개발할 수 있는 BCC 라이브러리입니다.
- bpftrace 추적 언어.
- bpfdevelopment 및 bpf trace 와 같은 관련 애플리케이션에 중요한 libbpf 패키지입니다.
  - libbpf 라이브러리의 XDP 및 AF_XDP API 부분은 지원되지 않으며 향후 릴리스에서 제거될 수 있습니다.

- 트래픽 제어(tc)를 위한 eBPF(eBPF) 기능은 커널 네트워크 데이터 경로 내에서 프로그래밍 가능한 패킷 처리를 가능하게 합니다.

- 커널 네트워킹 스택이 처리되기 전에 수신된 패킷에 대한 액세스를 제공하는 e XDP(데이터 경로) 기능입니다. Red Hat은 libxdp 라이브러리를 통해 사용되는 경우에만 XDP 를 지원합니다.

- XDP 기능에 대한 사용자 공간 지원 유틸리티가 포함된 xdp-tools 패키지는 AMD64 및 Intel64 CPU 아키텍처에서 지원됩니다. xdp-tools 패키지는 다음을 포함합니다.

  - libxdp 라이브러리입니다.

  - XDP 프로그램을 로드하기 위한 X dp-loader 유틸리티입니다.

  - 패킷 필터링을 위한 xdp-filter 프로그램 예.

  - XDP 가 활성화된 네트워크 인터페이스에서 패킷을 캡처하기 위한 xdpdump 유틸리티입니다. xdpdump 유틸리티는 현재 AMD64 및 Intel64 CPU 아키텍처에서만 지원됩니다. 기술 프리뷰로 다른 아키텍처에서 사용할 수 있습니다.

- eXpress Data Path(XDP) 경로를 사용자 공간에 연결하기 위한 AF_XDP 소켓입니다.

(BZ#2070506)

RHEL 9에서는 크래시 유틸리티 버전 8.0.0을 제공합니다.

RHEL 9는 크래시 유틸리티 버전 8.0.0과 함께 배포됩니다. 버그 수정 및 주요 개선 사항은 다음과 같습니다.

- add-symbol-file 명령에 새 offset 매개변수를 추가합니다. 이 지원을 통해 kaslr_offset 을 gdb 로 설정하는 데 도움이 됩니다.

- **gdb-7.6** 을 **gdb-10.2** 로 업그레이드합니다.

(BZ#1896647)

makedumpfile 에서 향상된 **zstd** 압축 기능 지원

이번 개선된 기능을 통해 **makedumpfile** 에 **Zstandard (zstd)** 압축 기능이 추가되어 높은 압축률을 제공합니다. 이러한 개선은 특히 대규모 메모리 시스템에 도움이 됩니다.

이제 **zstd** 압축 기능은 이전 압축 비율과 같이 **vmcore** 덤프 크기와 압축 시간 소비 간에 균형을 유지합니다. 결과적으로 개선된 압축 메커니즘으로 이제 허용되는 적절한 압축 시간을 사용하여 더 작은 **vmcore** 파일을 생성합니다.

좋은 압축 비율은 시스템을 사용하는 방법과 **RAM**에 저장된 데이터 유형에 따라 달라집니다.

(BZ#1988894)

Intel Xeon 확장 가능 서버 프로세서에서 **numatop** 활성화

numatop 는 **NUMA** 시스템에서 실행되는 프로세스와 스레드의 동작을 추적 및 분석하고 **NUMA** 관련 성능 병목 현상을 식별할 수 있는 지표를 표시하는 툴입니다.

numatop 는 **Intel** 성능 카운터 샘플링 기술을 사용하고 성능 데이터를 **Linux** 시스템 런타임 정보와 연결하여 프로덕션 시스템에 대한 분석을 제공합니다.

(BZ#1874125)

kexec_file_load 가 **RHEL 9**의 기본 옵션으로 추가되었습니다.

이번 업데이트에서는 **64**비트 **ARM** 아키텍처에 대해 **kexec_file_load** 시스템 호출을 추가합니다. **kdump** 에 대한 커널 **kexec loader**를 제공합니다. 이전에는 보안 부팅 옵션이 활성화되면 커널이 서명되지 않은 커널 이미지를 로드하지 못했습니다. **kdump** 메커니즘은 먼저 보안 부팅이 활성화되었는지 여부를 탐지한 다음 실행할 부팅 인터페이스를 선택합니다. 결과적으로 보안 부팅이 활성화된 상태에서 서명되지 않은 커널이 로드되지 않고 **kexec_file_load()** 가 지정되어 있었습니다.

이번 업데이트에서는 설명된 시나리오에서 문제가 해결되고 서명되지 않은 커널이 올바르게 작동합니다.

(BZ#1895232)

makedumpfile 에 예상 **vmcore** 크기를 얻기 위한 개선된 옵션 포함

이 구현을 통해 **makedumpfile** 유틸리티에 현재 실행 중인 커널의 덤프 크기에 대한 추정치를 출력하는 데 도움이 되는 다음 옵션이 포함되어 있습니다.

- **--dry-run** 은 다른 옵션으로 지정된 모든 작업을 수행하지만 출력 파일은 작성하지 않습니다.
- **--show-stats** 는 보고서 메시지를 출력합니다. 이는 **--message-level** 옵션에 제공된 수준에서 비트 **4**를 활성화하는 대안입니다.

다음 예제에서는 **--dry-run** 및 **--show-stats** 사용량을 보여줍니다.

```
$ makedumpfile --dry-run --show-stats -l --message-level 7 -d 31 /proc/kcore dump.dummy
```

덤프 파일 크기는 패닉 시 시스템 상태에 따라 다를 수 있으며 옵션에 의해 제공되는 추정치는 실제 상태와 다를 수 있습니다.

(BZ#1958452)

kexec-tools 패키지는 **RHEL 9**의 기본 **crashkernel** 메모리 예약 값을 지원합니다.

kexec-tools 패키지에서 기본 **crashkernel** 메모리 예약 값을 유지 관리합니다. **kdump** 서비스는 기본 값을 사용하여 각 커널에 대해 크래시커널 메모리를 예약합니다. 이 구현을 통해 시스템에 사용 가능한 메모리가 **4GB** 미만인 경우 **kdump** 에 대한 메모리 할당도 향상됩니다.

기본 **crashkernel** 값을 쿼리하려면 다음을 수행합니다.

```
$ kdumpctl get-default-crashkernel
```

기본 **crashkernel** 값으로 예약된 메모리가 시스템에 충분하지 않은 경우 **crashkernel** 매개변수를 늘립니다.

부팅 명령줄의 **crashkernel=auto** 옵션은 **RHEL 9** 이상 릴리스에서 더 이상 지원되지 않습니다.

자세한 내용은 **/usr/share/doc/kexec-tools/crashkernel-howto.txt** 파일을 참조하십시오.

(BZ#2034490)

RHEL 9에서 코어 예약이 지원됩니다.

코어 스케줄링 기능을 사용하면 서로 신뢰해서는 안 되는 작업이 동일한 **CPU** 코어를 공유하지 않도록 방지할 수 있습니다. 마찬가지로, 사용자는 **CPU** 코어를 공유할 수 있는 작업 그룹을 정의할 수 있습니다.

이러한 그룹을 지정할 수 있습니다.

- 일부 **SMT(Cross-Symmetric Multithreading)** 공격을 완화하여 보안을 강화하기 위해
- 전체 코어가 필요한 작업을 격리합니다. 예를 들어 실시간 환경의 작업 또는 단일 지침, 여러 데이터(**SIMD**) 처리와 같은 특정 프로세서 기능에 의존하는 작업의 경우

자세한 내용은 코어 스케줄링 을 참조하십시오.

(JIRA:RHELPLAN-100497)

기본적으로 제한 **iommu** 모드를 사용하여 **64**비트 **ARM** 아키텍처에서 성능이 향상되었습니다.

이 업그레이드를 통해 **64**비트 **ARM** 아키텍처는 기본적으로 시스템 메모리 관리 단위(**SMMU**)에 지연 직접 메모리 액세스(**DMA**) 도메인을 사용합니다. 상당한 성능 이점을 가져오는 동안 주소 **unmap**과 **SMMU**의**TLB(Translation Lookaside Buffer)** 플러시 사이에 창을 도입할 수 있습니다. 이전 버전에서 **64**비트 **ARM** 아키텍처는 엄격한 **DMA** 도메인을 기본값으로 구성하여 **4KB** 페이지 크기로 인해 성능이 저하되었습니다.

strict DMA 도메인 모드를 사용해야 하는 경우 커널 명령줄을 사용하여 **iommu.strict=1** 모드를 지정합니다. 엄격한 **DMA** 도메인을 사용하면 **64**비트 **ARM** 아키텍처에서 성능이 저하될 수 있습니다.

(BZ#2050415)

kernel-rt 소스 트리가 RHEL 9.0 트리로 업데이트되었습니다.

최신 **Red Hat Enterprise Linux** 커널 소스 트리를 사용하도록 **kernel-rt** 소스가 업데이트되었습니다. 실시간 패치 세트도 최신 업스트림 버전 **v5.15-rt19**로 업데이트되었습니다. 이러한 업데이트는 여러 버그 수정 및 개선 사항을 제공합니다.

(BZ#2002474)

hv_24x7 및 hv_gpci PMUs의 CPU 핫플러그 지원

이번 업데이트를 통해 **PMU** 카운터는 **CPU**의 핫 플러그에 올바르게 반응합니다. 결과적으로 **hv_gpci** 이벤트 카운터가 비활성화된 **CPU**에서 실행되는 경우 계산은 다른 **CPU**로 리디렉션됩니다.

(BZ#1844416)

POWERPC hv_24x7 중첩 이벤트에 대한 메트릭을 사용할 수 있습니다.

POWERPC hv_24x7 중첩 이벤트에 대한 메트릭을 이제 **perf** 에 사용할 수 있습니다. 이러한 메트릭은 여러 이벤트를 집계하여 **Perf** 카운터에서 얻은 값과 **CPU**가 워크로드를 얼마나 효과적으로 처리할 수 있는지에 대한 이해를 제공합니다.

(BZ#1780258)

RHEL 9에서 DVDMA 드라이버가 도입되었습니다.

recordsDMA 드라이버를 사용하면 **RDMA** 가능 **Intel®** 네트워크 장치에서 **RDMA** 기능을 사용할 수 있습니다. 이 드라이버에서 지원하는 장치는 다음과 같습니다.

- **Intel®** 이더넷 컨트롤러 **E810**
- **Intel®** 이더넷 네트워크 어댑터 **X722**

RHEL 9는 **X722 Internet Wide-area RDMA** 프로토콜 **(iWARP)** 장치에 대해 업데이트된 **Intel®** 이더넷 프로토콜 **(IRDMA)**을 제공합니다. **RHEL 9**에는 **iWARP** 및 **RDMA over Converged Ethernet(RoCEv2)**을 지원하는 새로운 **E810** 장치가 도입되었습니다. **recordsDMA** 모듈은 **X722**에 대한 레거시 **i40iw** 모듈을 교체하고 **i40iw**에 대해 정의된 **ABI(Application Binary Interface)**를 확장합니다. 변경 사항은 기존 **X722 RDMA-Core** 공급자**(libi40iw)**와 역호환됩니다.

- X722 장치는 iWARP 및 보다 제한된 구성 매개변수 세트만 지원합니다.
- E810 장치는 다음과 같은 RDMA 및 혼잡 관리 기능을 지원합니다.
  - iWARP 및 RoCEv2 RDMA 전송
  - PFC(priority Flow Control)
  - 명시적인 Congestion 알림 (ECN)

(BZ#1874195)

커널 본딩 모듈의 새 매개변수: lacp_active

RHEL 9에는 bonding 커널 모듈의 lacp_active 매개변수가 도입되었습니다. 이 매개변수는 지정된 간격으로 링크 집계 제어 프로토콜 데이터 단위(LACECDHE) 프레임을 보낼지 여부를 지정합니다. 옵션은 다음과 같습니다.

- (기본값) - 구성된 lacp_rate 매개변수와 함께 LACECDHE 프레임을 보낼 수 있습니다.
- off - LACECDHE 프레임이 "speak to"로 작동합니다.

LACECDHE 상태 프레임은 연결을 초기화하거나 바인딩 해제해도 전송됩니다.

(BZ#1951951)

4.10. 부트 로더

부트 로더 구성 파일은 CPU 아키텍처 전반에서 통합됩니다.

GRUB 부트 로더에 대한 설정 파일이 지원되는 모든 CPU 아키텍처의 /boot/grub2/ 디렉토리에 저장됩니다. 이전에 UEFI 시스템에서 기본 설정 파일로 사용된 GRUB의 /boot/efi/EFI/redhat/grub.cfg 파일인 /boot/grub2/grub.cfg 파일을 로드하면 됩니다.

이러한 변경은 GRUB 설정 파일의 레이아웃을 단순화하고 사용자 환경을 개선하며 다음과 같은 주요 이점을 제공합니다.

- EFI 또는 레거시 BIOS를 사용하여 동일한 설치를 부팅할 수 있습니다.
- 모든 아키텍처에 동일한 문서 및 명령을 사용할 수 있습니다.
- GRUB 설정 도구는 더 이상 심볼릭 링크에 의존하지 않고 플랫폼별 사례를 처리할 필요가 없기 때문에 더욱 강력합니다.
- GRUB 설정 파일 사용은COSA(CoreOS Assembler) 및 OSBuild에서 생성된 이미지와 정렬됩니다.
- GRUB 설정 파일 사용은 다른 Linux 배포판과 일치합니다.

(JIRA:RHELPLAN-101246)

4.11. 파일 시스템 및 스토리지

일관된 사용자 환경을 위해 Samba 유틸리티의 옵션의 이름이 변경 및 제거되었습니다.

일관된 명령줄 인터페이스를 제공하도록 Samba 유틸리티가 개선되었습니다. 이러한 개선 사항에는 이름 변경 및 제거된 옵션이 포함됩니다. 따라서 업데이트 후 문제를 방지하려면 Samba 유틸리티를 사용하는 스크립트를 검토하고 필요한 경우 업데이트합니다.

Samba 4.15에는 Samba 유틸리티가 다음과 같은 변경 사항이 추가되었습니다.

- 이전에는 Samba 명령줄 유틸리티에서 알 수 없는 옵션을 자동으로 무시했습니다. 예기치 않은 동작을 방지하기 위해 이제 유틸리티에서 알 수 없는 옵션을 일관되게 거부합니다.
- 이제 몇 가지 명령줄 옵션에 해당 KubeMacPool .conf 변수가 있어 기본값을 제어할 수 있습니다. 유틸리티의 도움말 페이지를 참조하여 명령줄 옵션에 ClusterRole .conf 변수 이름이 있는지 확인합니다.

- 기본적으로 **Samba** 유틸리티는 이제 표준 오류(**stderr**)에 기록합니다. 이 동작을 변경하려면 **--debug-stdout** 옵션을 사용합니다.

- **--client-protection=off|sign|encrypt** 옵션이 공통 구문 분석기에 추가되었습니다.

- 다음 옵션의 이름이 모든 유틸리티에서 변경되었습니다.

  - **--kerberos to --use-kerberos=required|desired|off**

  - **--krb5-ccache to --use-krb5-ccache=*CCACHE***

  - **--scope to --netbios-scope=*SCOPE***

  - **--use-winbind-ccache**에 **--use-ccache**

- 다음 옵션이 모든 유틸리티에서 제거되었습니다.

  - **-e** 및 **--encrypt**

  - **-C --use-winbind-ccache**에서 제거

  - **- netbios-scope**에서 제거됨

  - **- s** 및 **--signing**

- 중복 옵션을 방지하려면 다음 유틸리티에서 특정 옵션이 제거되거나 이름이 변경되었습니다.

  - **ndrdump:-l** 은 **--load-dso**에 더 이상 사용할 수 없습니다.

- net:-l 은 --long을 더 이상 사용할 수 없습니다.
- sharesec:-V 는 더 이상 --viewsddl에 사용할 수 없습니다.
- ClusterRolecquotas:--user 의 이름이 --quota-user로 변경되었습니다.
- nmbd:--log-stdout 의 이름이 --debug-stdout로 변경되었습니다.
- smbd: --log-stdout has been renamed to --debug-stdout
- winbindd:--log-stdout 의 이름이 --debug-stdout로 변경되었습니다.

(BZ#2065646)

RHEL 9의 NFS 클라이언트와 서버의 변경 사항

- RHEL 9.0 NFS 서버와 클라이언트는 안전하지 않은 GSS Kerberos 5 암호화 유형 des-cbc-crc 를 더 이상 지원하지 않습니다.
- NFS 클라이언트는 더 이상 UDP 전송을 사용한 파일 시스템 마운트를 지원하지 않습니다.

(BZ#1952863)

이제 GFS2 파일 시스템이 포맷 버전 1802로 생성됩니다.

RHEL 9의 GFS2 파일 시스템은 포맷 버전 1802로 생성됩니다. 이를 통해 다음 기능을 사용할 수 있습니다.

- 신뢰할 수 있는 네임스페이스(" trusted.* xattrs")의 확장 속성은 gfs2 및 gfs2-utils 에서 인식합니다.
- rgrplvb 옵션은 기본적으로 활성화되어 있습니다. 이를 통해 gfs2 는 업데이트된 리소스 그룹

데이터를 **DLM** 잠금 요청에 연결할 수 있으므로, 노드에서 디스크로부터 리소스 그룹 정보를 업데이트할 필요가 없습니다. 이는 경우에 따라 성능이 향상됩니다.

새 형식 버전으로 생성된 파일 시스템은 이전 **RHEL** 버전과 **fsck.gfs2** 유틸리티의 이전 버전에서 마운트할 수 없습니다.

사용자는 **-o format=1801** 옵션과 함께 **mkfs.gfs2** 명령을 실행하여 이전 형식 버전으로 파일 시스템을 생성할 수 있습니다.

사용자는 마운트 해제된 파일 시스템에서 **tunegfs2 -r 1802** *장치*를 실행하는 이전 파일 시스템의 형식 버전을 업그레이드할 수 있습니다. 형식 버전을 다운그레이드하는 것은 지원되지 않습니다.

(BZ#1616432)

RHEL 9에서는 **nvml** 패키지 버전 **1.10.1**를 제공합니다.

RHEL 9.0에서는 **nvml** 패키지를 버전 **1.10.1**로 업데이트합니다. 이번 업데이트에는 정전 시 잠재적인 데이터 손상 버그가 포함되어 있는 기능 및 수정 사항이 추가되었습니다.

(BZ#1874208)

exFAT 파일 시스템에 대한 지원이 추가되었습니다.

RHEL 9.0에서는 확장 가능 파일 할당 테이블(**exFAT**) 파일 시스템을 지원합니다. 이제 마운트, 포맷 및 일반적으로 이 파일 시스템을 사용할 수 있습니다. 이 파일 시스템은 일반적으로 플래시 메모리에서 기본적으로 사용됩니다.

(BZ#1943423)

tutorialctl 명령을 사용하여 **SunRPC** 연결 정보를 표시

이번 업데이트를 통해**RuntimeConfig ctl** 명령을 사용하여 시스템의 **SunRPC objects**에 대한 **SunRPC sysfs** 파일에 수집된 정보를 표시할 수 있습니다. **SunRPC** 네트워크 계층에서 **sysfs** 파일 시스템을 통해 오브젝트를 표시, 제거 및 설정할 수 있습니다.

(BZ#2059245)

LVM의 장치 세트 제한

기본적으로 **RHEL 9**의 **LVM**은 명시적으로 선택한 장치만 사용합니다. 새 명령 **lvmdevices** 및 **vgimportdevices** 를 사용하여 특정 장치를 선택합니다. **pvcreate,vgcreate, vgextend** 명령을 사용하면 아직 선택되지 않은 경우 **lvm** 용 새 장치를 간접적으로 선택합니다. **LVM**은 이러한 명령 중 하나를 사용하여 시스템을 선택할 때까지 시스템에 연결된 장치를 무시합니다. **lvm** 명령은 선택한 장치 목록을 장치 파일 **/etc/lvm/devices/system.devices** 에 저장합니다. **lvm.conf** 필터 또는 다른 명령줄 구성 필터가 새 장치 파일 기능을 활성화할 때 작동하지 않습니다. **devices** 파일을 제거하거나 비활성화하는 경우 **LVM**은 연결된 모든 장치에 필터를 적용합니다. 이 기능에 대한 자세한 내용은 **lvmdevices(8)** 매뉴얼 페이지를 참조하십시오.

(BZ#1749513)

nvme_tcp.ko 가 포함된 **NVMe/TCP** 호스트

nvme_tcp.ko 커널 모듈이 포함된 **NVMe(Nonvolatile Memory Express) over TCP/IP** 네트워크 **(NVMe/TCP)** 스토리지가 이제 완전하게 지원됩니다. **nvmet_tcp.ko** 모듈이 있는 **NVMe/TCP** 대상은 **RHEL 9.0**에서 유지 관리되지 않는 상태로 사용할 수 있습니다.

(BZ#2054441)

4.12. 고가용성 및 클러스터

이제 **resource-stickiness** 리소스 **meta-attribute**가 새로 생성된 클러스터의 경우 **0** 대신 **1**로 설정됩니다.

이전 버전에서는 **resource-stickiness** 리소스 **meta-attribute**의 기본값이 새로 생성된 클러스터에 대한 기본값인 **0**이 있었습니다. 이제 이 **meta-attribute**가 **1**로 설정됩니다.

0의 고착성을 사용하면 클러스터는 필요에 따라 리소스를 다른 노드 간에 균형을 맞추기 위해 리소스를 이동할 수 있습니다. 그러면 관련 리소스가 시작되거나 중지되면 리소스가 이동될 수 있습니다. 긍정적인 고착성을 사용하면 리소스를 우선적으로 유지하고 다른 상황이 고착성을 능가하는 경우에만 이동해야 합니다. 이로 인해 새로 추가된 노드는 관리자의 개입 없이 해당 노드에 리소스를 할당하지 못할 수 있습니다. 두 접근 모두 잠재적으로 예기치 않은 동작이 있지만 대부분의 사용자는 고정성을 선호하는 것이 좋습니다. 이 메타**-attribute**의 기본값은 이 기본 설정을 반영하도록 **1**로 변경되었습니다.

새로 생성된 클러스터만 이 변경의 영향을 받으므로 기존 클러스터에 대한 동작이 변경되지 않습니다. 클러스터의 이전 동작을 선호하는 사용자는 리소스 기본값에서 **resource-stickiness** 항목을 삭제할 수 있습니다.

(BZ#1850145)

자동 비활성화를 제어하는 새로운 LVM 볼륨 그룹 플래그

LVM 볼륨 그룹은 이제 볼륨 그룹에서 생성한 논리 볼륨이 시작 시 자동으로 활성화되는지 여부를 제어하는 setautoactivation 플래그를 지원합니다. 클러스터에서 Pacemaker에서 관리할 볼륨 그룹을 생성할 때 볼륨 그룹에 대해 이 플래그를 n 명령으로 설정하여 가능한 데이터 손상을 방지합니다. Pacemaker 클러스터에 기존 볼륨 그룹이 사용된 경우, KnativeServing change --setautoactivation n 으로 플래그를 설정합니다.

(BZ#1899214)

새로운 pcs resource status display 명령

pcs 리소스 상태 및 pcs stonith 상태 명령에서 다음 옵션을 지원합니다.

- pcs resource status node= node_id 명령과 pcs stonith status node=*node_id*명령을 사용하여 특정 노드에 구성된 리소스 상태를 표시할 수 있습니다. 이 명령을 사용하여 클러스터 및 원격 노드의 리소스 상태를 표시할 수 있습니다.
- pcs resource status resource_id 및 pcs stonith status *resource_id* 명령을 사용하여 단일 리소스의 상태를 표시할 수 있습니다.
- pcs resource status tag_id 및 pcs stonith status *tag_id*명령을 사용하여 지정된 태그를 사용하여 모든 리소스의 상태를 표시할 수 있습니다.

(BZ#1290830, BZ#1285269)

pcs resource safe-disable 명령의 새로운 축소된 출력 표시 옵션

pcs resource safe-disable 및 pcs resource disable --safe 명령으로 오류 보고 후 긴 시뮬레이션 결과를 출력합니다. 이제 해당 명령에 --brief 옵션을 지정하여 오류를 출력할 수 있습니다. 이제 오류 보고서에는 항상 영향을 받는 리소스의 리소스 ID가 포함됩니다.

(BZ#1909901)

다른 모든 리소스를 재시작하지 않고 SCSI 펜싱 장치를 업데이트하는 새로운 pcs 명령

pcs stonith update 명령으로 SCSI 펜싱 장치를 업데이트하면 stonith 리소스가 실행 중인 동일한 노드에서 실행 중인 모든 리소스가 다시 시작됩니다. 새로운 pcs stonith update-scsi-devices 명령을 사용하면 다른 클러스터 리소스를 재시작하지 않고도 SCSI 장치를 업데이트할 수 있습니다.

(BZ#1872378)

클러스터 노드의 하위 집합에서 펜싱하도록 워치독 전용 SBD를 구성하는 기능

이전에는 워치독 전용 SBD 구성을 사용하려면 클러스터의 모든 노드가 SBD를 사용해야 했습니다. 이로 인해 일부 노드에서 이를 지원하는 클러스터에서 SBD를 사용하지 못했지만 다른 노드(원격 노드)에는 다른 유형의 펜싱이 필요했습니다. 사용자는 새로운 fence_watchdog 에이전트를 사용하여 워치독 전용 SBD 설정을 구성할 수 있으므로 일부 노드에서는 펜싱에 워치독 전용 SBD만 사용하고 기타 노드에는 다른 펜싱 유형을 사용할 수 있습니다. 클러스터에는 이러한 장치 하나만 있을 수 있으며 워치독 이라는 이름을 지정해야 합니다.

(BZ#1443666)

내부 오류에 대한 자세한 Pacemaker 상태 표시

Pacemaker에서 어떤 이유로든 리소스 또는 펜스 에이전트를 실행할 수 없는 경우(예: 에이전트가 설치되지 않았거나 내부 시간 초과가 있는 경우 Pacemaker 상태가 이제 내부 오류에 대한 자세한 종료 이유가 표시됩니다.

(BZ#1470834)

pcmk_delay_base 매개변수는 다른 노드에 대해 다른 값을 사용할 수 있습니다.

펜스 장치를 구성할 때 pcmk_delay_base 매개변수를 사용하여 다른 노드에 대해 다른 값을 지정할 수 있습니다. 이를 통해 노드마다 다른 지연과 함께 2-노드 클러스터에서 단일 펜스 장치를 사용할 수 있습니다. 이렇게 하면 각 노드에서 다른 노드를 동시에 펜싱하려고 시도하는 상황을 방지할 수 있습니다. 다른 노드의 다른 값을 지정하려면 pcmk_host_map과 유사한 구문을 사용하여 호스트 이름을 해당 노드의 지연 값에 매핑합니다. 예를 들어 node1:0;node2:10s는 node1을 펜싱할 때 no delay를 사용하고, node2를 펜싱할 때 10초 지연을 사용합니다.

(BZ#1082146)

pcmk_host_map 값 내의 특수 문자 지원

pcmk_host_map 속성은 값 앞에 백슬래시(\)를 사용하여 pcmk_host_map 값 내부의 특수 문자를 지원합니다. 예를 들어 호스트 별칭에 공백을 포함하도록 pcmk_host_map="node3:plug\ 1" 을 지정할 수

있습니다.

(BZ#1376538)

OpenShift의 새로운 펜싱 에이전트

Red Hat OpenShift Virtualization에서 **RHEL High Availability**와 함께 **fence_kubevirt** 펜싱 에이전트를 사용할 수 있습니다. **fence_kubevirt** 에이전트에 대한 자세한 내용은 **fence_kubevirt(8)** 도움말 페이지를 참조하십시오.

(BZ#1977588)

pcs cluster setup 명령의 로컬 모드 버전이 완전히 지원됩니다.

기본적으로 **pcs cluster setup** 명령은 모든 구성 파일을 클러스터 노드와 자동으로 동기화합니다. **pcs cluster setup** 명령으로 **--corosync-conf** 옵션을 완전히 지원합니다. 이 옵션을 지정하면 명령이 로컬 모드로 전환됩니다. 이 모드에서 **pcs** 명령줄 인터페이스는 **corosync.conf** 파일을 생성하여 다른 노드와 통신하지 않고 로컬 노드의 지정된 파일에 저장합니다. 이를 통해 스크립트에 **corosync.conf** 파일을 생성하고 스크립트를 통해 해당 파일을 처리할 수 있습니다.

(BZ#2008558)

리소스 이동 후 위치 제한 조건 자동 제거

pcs resource move 명령을 실행하면 리소스에 제약 조건이 추가되어 현재 실행 중인 노드에서 실행되지 않습니다. 기본적으로 명령이 생성하는 위치 제한 조건은 리소스가 이동되면 자동으로 제거됩니다. 이 경우 리소스가 원래 노드로 다시 이동하는 것은 아니며, 해당 시점에서 리소스를 실행할 수 있는 리소스는 처음에 리소스를 구성한 방법에 따라 달라집니다. 리소스를 이동하고 결과 제한 조건을 그대로 유지하려면 **pcs resource move-with-contraint** 명령을 사용합니다.

(BZ#2008575)

OCF Resource Agent API 1.1 표준용 **pcs suppport**

pcs 명령줄 인터페이스에서 **OCF 1.1** 리소스 및 **STONITH** 에이전트를 지원합니다. 이러한 지원의 구현의 일환으로 모든 에이전트의 메타데이터는 에이전트가 **OCF 1.0** 또는 **OCF 1.1** 에이전트인지에 관계없이 **OCF** 스키마를 준수해야 합니다. 에이전트의 메타데이터가 **OCF** 스키마를 준수하지 않는 경우 **pcs** 는 에이전트를 잘못된 것으로 간주하고 **--force** 옵션을 지정하지 않는 한 에이전트의 리소스를 생성하거나 업데이트하지 않습니다. 에이전트를 나열하는 **pcs d Web UI** 및 **pcs** 명령으로 목록에서 잘못된 메타데이터가 있는 에이전트를 생략합니다.

(BZ#2018969)

pcs에서 암호 해독 및 역할 이름으로 Unpromoted 를 허용합니다.

pcs 명령줄 인터페이스에서 ensures d 및 Unpromoted 역할이 Pacemaker 구성에 지정됩니다. 이러한 역할 이름은 이전 RHEL 릴리스에서 Master 및 Slave Pacemaker 역할과 동등한 기능이며, 구성 표시 및 도움말 페이지에 표시되는 역할 이름입니다.

(BZ#2009455)

pcsd Web UI의 업데이트된 버전

Pacemaker/Corosync 클러스터를 생성하고 구성하는 그래픽 사용자 인터페이스인 pcsd Web UI가 업데이트되었습니다. 업데이트된 웹 UI는 다른 Red Hat 웹 애플리케이션에 사용되는 PatternFly 프레임워크로 구축된 개선된 사용자 환경과 표준화된 인터페이스를 제공합니다.

(BZ#1996067)

4.13. 동적 프로그래밍 언어, 웹 서버 및 데이터베이스 서버

RHEL 9의 Python

Python 3.9 는 RHEL 9의 기본 Python 구현입니다. Python 3.9 는 BaseOS 리포지토리의 비표준 python3 RPM 패키지로 배포되고 일반적으로 기본적으로 설치됩니다. Python 3.9 는 RHEL 9의 전체 라이프 사이클 동안 지원됩니다.

Python 3 의 추가 버전은 AppStream 리포지토리를 통해 라이프사이클이 짧은 RPM 패키지로 배포되며 동시에 설치할 수 있습니다.

python 명령(/usr/bin/python) 및 기타 Python- 관련 명령(예: pip)은 버전이 없는 형식으로 사용할 수 있으며 기본 Python 3.9 버전을 가리킵니다.

Python 2 는 RHEL 9와 함께 배포되지 않습니다.

RHEL 9의 Python 에 대한 자세한 내용은 Python 소개를 참조하십시오.

(BZ#1941595, JIRA:RHELPLAN-80598)

Node.js 16 사용 가능

RHEL 9는 JavaScript 프로그래밍 언어로 빠르고 확장 가능한 네트워크 애플리케이션을 구축하기 위한 소프트웨어 개발 플랫폼인 LS(Long Term Support) 버전 16을 제공합니다.

Node.js 14 대신 Node.js 16 의 주요 변경 사항은 다음과 같습니다.

- V8 엔진이 버전 9.4로 업그레이드되었습니다.
- npm 패키지 관리자가 버전 8.3.1로 업그레이드되었습니다.
- 새로운 timer Promises API는 Promise 오브젝트를 반환하는 대체 타이머 함수 세트를 제공합니다.
- Node.js 는 이제 OpenSSL 3.0 과 호환됩니다.
- Node.js 는 이제 새로운 실험적인 Web Streams API와 실험적인 ECMAScript 모듈 (ESM) 로더 API를 제공합니다.

Node.js 16 은 RPM 패키지로 쉽게 설치할 수 있는 이 애플리케이션 스트림의 초기 버전입니다. Node.js 16 은 RHEL 9보다 라이프사이클이 짧습니다. 자세한 내용은 Red Hat Enterprise Linux Application Streams 라이프 사이클 문서를 참조하십시오. 추가 Node.js 버전은 RHEL 9의 향후 마이너 릴리스에서 라이프사이클이 짧은 모듈로 제공됩니다.

(BZ#1953491)

RHEL 9에서는 Ruby 3.0을 제공합니다.

RHEL 9는 Ruby 2.7 에 비해 여러 성능 향상, 버그 및 보안 수정, 새로운 기능을 제공하는 Ruby 3.0.3 과 함께 배포됩니다.

주요 개선 사항은 다음과 같습니다.

- 동시성 및 병렬 기능:
  - 스레드 안전한 병렬 실행을 제공하는 **actor-model** 추상화는 실험적인 기능으로 제공됩니다.
  - 파이버 스케줄러 는 실험적인 기능으로 도입되었습니다. 파이버 스케줄러 는 기존 코드를 변경하지 않고 경량의 동시성을 활성화하는 차단 작업을 차단합니다.
- 정적 분석 기능:
  - **Ruby** 프로그램의 구조를 설명하는 **RBS** 언어가 도입되었습니다. **RBS** 에서 작성된 유형 정의를 구문 분석하기 위해**rbs gem**이 추가되었습니다.
  - **Ruby** 코드의 유형 분석 툴인 **TypeProf** 유틸리티가 도입되었습니다.
- 케이스/인 표현식과 패턴 일치는 더 이상 실험적이지 않습니다.
- 실험적 기능인 한 줄 패턴 일치는 재설계되었습니다.
- 패턴 찾기가 실험적 기능으로 추가되었습니다.

다음과 같은 성능 개선이 구현되었습니다.

- 긴 코드를 **Interactive Ruby Shell(IRB)** 에 붙여넣는 것이 훨씬 빨라졌습니다.
- 시간 측정을 위해 **measure** 명령이 **records B** 에 추가되었습니다.

기타 주요 변경 사항은 다음과 같습니다.

- 이제 키워드 인수가 다른 인수와 분리되어 있습니다.

- 사용자가 설치한 **gems**의 기본 디렉터리는 **$HOME/.local/share/gem/** 입니다. **$HOME/.gem/** 디렉토리는 **$HOME/.gem/** 디렉토리에 이미 있는 경우가 아니면 **$HOME/.local/share/gem/**입니다.

Ruby 3.0 은 **RPM** 패키지로 쉽게 설치할 수 있는 이 **Application Stream**의 초기 버전입니다. 추가 **Ruby** 버전은 **RHEL 9**의 향후 마이너 릴리스에서 라이프사이클이 짧은 모듈로 제공됩니다.

(JIRA:RHELPLAN-80758)

RHEL 9에는 **Perl 5.32**도입

RHEL 9에는 버전 **5.30**에 비해 여러 버그 수정 및 개선 사항을 제공하는 **Perl 5.32** 가 포함되어 있습니다.

주요 개선 사항은 다음과 같습니다.

- **Perl** 은 이제 유니코드 버전 **13.0**을 지원합니다.

- **qr quote-like Operator**가 향상되었습니다.

- **POSIX::mblen(), mbtowc,** 및 **wctomb** 함수는 이제 시프트 주 로캘에서 작동하고 **C99**에서 스레드로부터 안전하며, 스레드가 스레드 스레드가 로캘이 있는 플랫폼에서 실행될 때 스레드로부터 안전합니다. 이제 길이 매개 변수는 선택 사항입니다.

- 새로운 실험적 **isa infix** 연산자는 지정된 개체가 지정된 클래스의 인스턴스인지 또는 클래스에서 파생되는지를 테스트합니다.

- 알파 주장은 더 이상 실험적이지 않습니다.

- 스크립트 실행은 더 이상 실험적이지 않습니다.

- 이제 기능 검사 속도가 빨라집니다.

- Perl 은 이제 최적화 전에 컴파일된 패턴을 덤프할 수 있습니다.

Perl 5.32 는 RPM 패키지로 쉽게 설치할 수 있는 이 Application Stream의 초기 버전입니다. 추가 Perl 버전은 RHEL 9의 향후 마이너 릴리스에서 라이프사이클이 짧은 모듈로 제공됩니다.

(JIRA:RHELPLAN-80759)

RHEL 9에는 PHP 8.0포함

RHEL 9는 PHP 8.0 과 함께 배포되었으며 버전 7.4에 비해 여러 버그 수정 및 개선 사항을 제공합니다.

주요 개선 사항은 다음과 같습니다.

- 새로운 명명된 인수는 순서가 독립적이고 자체 문서화되며 필수 매개변수만 지정할 수 있습니다.

- 새로운 특성을 사용하면 PHP의 네이티브 구문에 구조화된 메타데이터를 사용할 수 있습니다.

- 새로운 통합 유형을 사용하면 유형 조합에 대해 PHPDoc 주석 대신 런타임에서 검증된 네이티브 통합 유형 선언을 사용할 수 있습니다.

- 이제 매개 변수 유효성 검사가 실패하는 경우 내부 함수가 경고 대신 오류 예외를 더 일관되게 발생시킵니다.

- 새로운 Just-In-Time 컴파일 엔진은 애플리케이션 성능을 크게 향상시킵니다.

- PHP의 Xdebug 디버깅 및 생산성 확장 기능이 버전 3으로 업데이트되었습니다. 이 버전에서는 Xdebug 2 에 비해 기능 및 구성의 주요 변경 사항이 추가되었습니다.

PHP 8.0 은 RPM 패키지로 쉽게 설치할 수 있는 이 Application Stream의 초기 버전입니다. 추가 PHP 버전은 RHEL 9의 향후 마이너 릴리스에서 라이프사이클이 짧은 모듈로 제공됩니다.

자세한 내용은 PHP 스크립팅 언어 사용을 참조하십시오.

(BZ#1949319)

RHEL 9에서는 Git 2.31 및 Git LFS 2.13을 제공합니다.

RHEL 9는 RHEL 8에서 사용할 수 있는 버전 2.27에 비해 여러 개선사항 및 성능 향상을 제공하는 Git 2.31 과 함께 배포됩니다. 주요 변경 사항은 다음과 같습니다.

- 이제 git status 명령에서 스파스 체크 아웃의 상태를 보고합니다.
- 이제 git archive 명령과 함께 --add-file 옵션을 사용하여 트리-ish 식별자의 스냅샷에 추적되지 않은 파일을 포함할 수 있습니다.
- clone.defaultremotename 구성 변수를 사용하여 소스 원격 리포지토리의 닉네임을 사용자 지정할 수 있습니다.
- git format-patch 명령으로 생성된 출력 파일 이름의 최대 길이를 구성할 수 있습니다. 이전에는 길이 제한이 64바이트였습니다.
- 더 이상 사용되지 않는 PCRE1 라이브러리 지원이 제거되었습니다.

또한 Git Large File Storage (LFS) 확장 버전 2.13을 사용할 수 있습니다. RHEL 8에서 배포된 2.11 버전에 비해 개선 사항은 다음과 같습니다.

- Git LFS 는 이제 SHA-256 리포지토리를 지원합니다.
- Git LFS 는 이제 socks5h 프로토콜을 지원합니다.

- git lfs install 및 git lfs uninstall 명령에 새로운 --worktree 옵션을 사용할 수 있습니다.
- git lfs migrate import 명령에 새로운 --above 매개변수를 사용할 수 있습니다.

(BZ#1956345, BZ#1952517)

RHEL 9의 Subversion 1.14

RHEL 9는 Subversion 1.14 와 함께 배포됩니다. Subversion 1.14 는 RPM 패키지로 쉽게 설치할 수 있는 이 Application Stream의 초기 버전입니다. 추가 Subversion 버전은 RHEL 9의 향후 마이너 릴리스에서 라이프사이클이 짧은 모듈로 제공됩니다.

(JIRA:RHELPLAN-82578)

Apache HTTP Server에서 주요 변경 사항

RHEL 9.0에서는 Apache HTTP Server 버전 2.4.51을 제공합니다. 버전 2.4.37에 대한 주요 변경 사항은 다음과 같습니다.

- Apache HTTP Server Control Interface (apachectl):
  - 이제 apachectl 상태 출력에 대해 systemctl pager가 비활성화되어 있습니다.
  - 이제 추가 인수를 전달하면 경고를 제공하는 대신 apachectl 명령이 실패합니다.
  - 이제 apachectl normal-stop 명령이 즉시 반환됩니다.
  - 이제 apachectl configtest 명령은 SELinux 컨텍스트를 변경하지 않고 httpd -t 명령을 실행합니다.
  - RHEL의 apachectl(8) 도움말 페이지는 이제 업스트림 apachectl 과 완전히 다른 문서를 사용합니다.

- **Apache eXtenSion 툴(apxs):**
  - **/usr/bin/apxs** 명령은 더 이상 **httpd** 패키지를 빌드할 때 컴파일러 최적화 플래그를 적용으로 사용하거나 노출하지 않습니다. 이제 **/usr/lib64/httpd/build/vendor-apxs** 명령을 사용하여 **httpd** 를 빌드하는 데 사용된 것과 동일한 컴파일러 플래그를 적용할 수 있습니다. **vendor-apxs** 명령을 사용하려면 먼저 **redhat-rpm-config** 패키지를 설치해야 합니다.
- **Apache** 모듈:
  - **mod_lua** 모듈이 이제 별도의 패키지로 제공됩니다.
  - **Apache HTTP Server**의 새로운 **mod_jk** 커넥터는 **Apache JServ Protocol(AJP)**를 사용하여 웹 서버를 **Apache Tomcat** 및 기타 백엔드와 연결하는 모듈입니다.
  - 새로운 **mod_proxy_cluster** 모듈은 통신 채널을 사용하여 로드 밸런서의 요청을 애플리케이션 서버 노드 세트 중 하나로 전달하는 **httpd** 기반 로드 밸런서를 제공합니다. 애플리케이션 서버 노드는 이 연결을 사용하여 **MCMP(Mod-Cluster Management Protocol)**라는 사용자 지정 **HTTP** 방법 세트를 통해 서버 측 부하 분산 요인 및 라이프사이클 이벤트를 로드 밸런서로 다시 전송합니다. 이 추가 피드백 채널을 통해 **mod_proxy_cluster** 는 다른 로드 밸런싱 솔루션에서 찾을 수 없는 수준의 인텔리전스를 제공할 수 있습니다. 이 모듈을 사용하려면 백엔드 서버에 **ModCluster** 클라이언트를 설치해야 성공적으로 통신할 수 있습니다.
- 구성 구문 변경 사항:
  - **mod_access_compat** 모듈에서 제공하는 더 이상 사용되지 않는 **Allow** 지시문에서 주석 **(#** 문자)이 이제 자동으로 무시되는 대신 구문 오류를 트리거합니다.
- 기타 변경 사항:
  - 이제 커널 스레드 **ID**가 오류 로그 메시지에서 직접 사용되어 정확하고 간결하게 만듭니다.
  - 많은 사소한 개선 사항 및 버그 수정

- 모듈 작성자는 다양한 새 인터페이스를 사용할 수 있습니다.

RHEL 8 이후 httpd 모듈 API에 대한 이전 버전과 호환되지 않는 변경 사항은 없습니다.

Apache HTTP Server 2.4는 RPM 패키지로 쉽게 설치할 수 있는 이 Application Stream의 초기 버전입니다.

자세한 내용은 Apache HTTP 웹 서버 설정을 참조하십시오.

(JIRA:RHELPLAN-68364, BZ#1931976, JIRA:RHELPLAN-80725)

RHEL 9에서 Nginx 1.20 사용 가능

RHEL 9에는 nginx 1.20 웹 및 프록시 서버가 포함되어 있습니다. 이번 릴리스에서는 버전 1.18에 비해 여러 버그 수정, 보안 수정, 새로운 기능 및 개선 사항이 포함되어 있습니다.

새로운 기능:

- Nginx 는 이제 OCSP(Online Certificate Status Protocol)를 사용하여 클라이언트 SSL 인증서 검증을 지원합니다.
- 이제 Nginx는 최소 사용 가능한 공간에 따라 캐시 정리를 지원합니다. 이 지원은 proxy_cache_path 지시문의 min_free 매개변수로 구현됩니다.
- 변수의 값을 설정할 수 있는 새로운 ngx_stream_set_module 모듈이 추가되었습니다.
- nginx 용으로 외부 동적 모듈을 빌드하기 위한 RPM 매크로 및 nginx 소스 코드를 포함하여 필요한 모든 파일을 제공하는 새로운 nginx-mod-devel 패키지가 추가되었습니다.

향상된 지시문:

- 이제 ssl_conf_command 및 ssl_reject_handshake 와 같은 여러 새 지시문을 사용할 수 있습니다.

- proxy_cookie_flags 지시문은 이제 변수를 지원합니다.

HTTP/2 지원 개선:

- 이제 ngx_http_v2 모듈에 lingering_close,lingering_time,lingering_timeout 지시문이 포함됩니다.
- HTTP/2의 연결 처리가 HTTP/1.x와 일치합니다. nginx 1.20 에서 제거된 http2_recv_timeout,http2_idle_timeout, http2_max_requests 지시문 대신 keepalive_timeout 및 keepalive_requests 지시문을 사용합니다.

Nginx 1.20 은 RPM 패키지로 쉽게 설치할 수 있는 이 Application Stream의 초기 버전입니다. 추가 nginx 버전은 RHEL 9의 향후 마이너 릴리스에서 라이프사이클이 짧은 모듈로 제공됩니다.

자세한 내용은talPod Autoscaler 설정 및 구성을 참조하십시오.

(BZ#1953639, BZ#1991720)

RHEL 9의 Varnish Cache 6.6

RHEL 9에는 고성능 HTTP 역방향 프록시인 Varnish Cache 6.6 이 포함되어 있습니다.

6.0 버전부터 주요 변경 사항은 다음과 같습니다.

- varnishlog와 같은 로그 처리 툴의 성능 향상
- 통계 정확도 개선
- 캐시 조회의 여러 최적화

- 다양한 구성 변경
- 다양한 개선 사항 및 버그 수정

Varnish Cache 6 는 RPM 패키지로 쉽게 설치할 수 있는 이 애플리케이션 스트림의 초기 버전입니다.

(BZ#1984185)

RHEL 9 소개 Squid 5

RHEL 9는 웹 클라이언트를 위한 고성능 프록시 캐싱 서버인 Squid 5.2 와 함께 배포되어 FTP, Gopher 및 HTTP 데이터 오브젝트를 지원합니다. 이 릴리스에서는 버전 4에 대한 여러 버그 수정, 보안 수정, 새로운 기능 및 향상된 기능을 제공합니다.

새로운 기능:

- Squid 는 Happyballballs (HE) 알고리즘을 사용하여 책임을 향상시킵니다.
  - Squid 는 이제 즉시 수신된 IP 주소를 사용합니다. 전달을 요청하려면 모든 잠재적인 전달 대상이 완전히 해결되기를 기다리는 대신 필요합니다.
  - 새로운 지시문을 사용할 수 있습니다. happy_eyeballs_connect_gap,happy_eyeballs_connect_limit, happy_eyeballs_connect_timeout 지시문.
  - dns_v4_first 지시문이 제거되었습니다.
- 이제 squid 에서 CDN(Content Delivery Networks)의 루프 탐지용 소스로 CDN-Loop 헤더를 사용합니다.
- squid 는 SSL 충돌에 대한 피어링 지원을 제공합니다.
-

새로운 ICAP(Internet Content Adaptation Protocol) 추적 기능을 사용할 수 있으므로 ICAP 에이전트가 메시지 본문 후에 메시지를 안정적으로 보낼 수 있습니다.

설정 옵션 변경 사항:

- mark_client_packet 설정 옵션이 clientside_mark 를 대체했습니다.
- shared_transient_entries_limit 구성 옵션이 collapsed_forwarding_shared_entries_limit.

squid 5 는 RPM 패키지로 쉽게 설치할 수 있는 이 애플리케이션 스트림의 초기 버전입니다.

자세한 내용은 Squid 캐싱 프록시 서버 구성을 참조하십시오.

(BZ#1990517)

RHEL 9의 MariaDB 10.5

RHEL 9는 MariaDB 10.5 를 제공합니다. MariaDB 10.5 는 RPM 패키지로 쉽게 설치할 수 있는 이 애플리케이션 스트림의 초기 버전입니다. 추가 MariaDB 버전은 RHEL 9의 향후 마이너 릴리스에서 라이프 사이클이 짧은 모듈로 제공됩니다.

자세한 내용은 MariaDB 사용을 참조하십시오.

(BZ#1971248)

RHEL 9에는 MySQL 8.0이 포함되어 있습니다.

RHEL 9는 MySQL 8.0 과 함께 배포됩니다. MySQL 8.0 은 RPM 패키지로 쉽게 설치할 수 있는 이 Application Stream의 초기 버전입니다. MySQL 8.0 의 라이프 사이클은 RHEL 9보다 짧습니다. 자세한 내용은 Red Hat Enterprise Linux Application Streams 라이프 사이클 문서를 참조하십시오.

사용법에 대한 자세한 내용은 MySQL 사용을 참조하십시오.

(JIRA:RHELPLAN-78673)

RHEL 9에서는 PostgreSQL 13을 제공합니다.

PostgreSQL 13 은 RHEL 9에서 사용할 수 있습니다. PostgreSQL 13 은 RPM 패키지로 쉽게 설치할 수 있는 이 Application Stream의 초기 버전입니다. 추가 PostgreSQL 버전은 RHEL 9의 향후 마이너 릴리스에서 라이프사이클이 짧은 모듈로 제공됩니다.

자세한 내용은 PostgreSQL 사용을 참조하십시오.

(JIRA:RHELPLAN-78675)

RHEL 9의 Redis 6.2

RHEL 9는 RHEL 8에서 사용할 수 있는 버전 6.0에 비해 다양한 버그 및 보안 수정 및 개선 사항을 제공하는 Redis 6.2 와 함께 배포됩니다.

특히 Redis 서버 구성 파일은 이제 전용 디렉터리 /etc/redis/redis.conf 및 /etc/redis/sentinel.conf 에 있습니다. RHEL 8 버전에서 이러한 파일의 위치는 각각 /etc/redis.conf 및 /etc/redis-sentinel.conf 입니다.

Redis 6 은 RPM 패키지로 쉽게 설치할 수 있는 이 Application Stream의 초기 버전입니다.

(BZ#1959756)

새로운 패키지: perl-Module-Signature

RHEL 9에는 perl-Module-Signature Perl 모듈이 도입되었습니다. 이 새로운 모듈을 사용하면 cpan 에 대한 서명 검사를 활성화하여 CVE-2020-16156을 완화할 수 있습니다. 자세한 내용은 perl-App-cpanminus 및 CVE-2020-16156에서 CVE-2020-16154를 완화하는 방법을 참조하십시오.

(BZ#2039361)

4.14. 컴파일러 및 개발 도구

RHEL 9는 IBM POWER10 프로세서 지원

Linux 커널에서 시스템 툴체인(GCC, binutils, glibc)을 통해 IBM의 최신 POWER 프로세서인 POWER10에 대한 지원을 포함하도록 Red Hat Enterprise Linux 9가 업데이트되었습니다. RHEL 9는 POWER10에서 워크로드를 지원할 준비가 되어 있으며 향후 릴리스에서도 개선 사항이 제공됩니다.

(BZ#2027596)

GCC 11.2.1 사용 가능

RHEL 9는 GCC 버전 11.2.1과 함께 배포됩니다. 주요 버그 수정 및 개선 사항은 다음과 같습니다.

일반 개선 사항

- GCC는 이제 DWARF 버전 5 디버깅 형식입니다.
- 진단에 표시된 열 번호는 기본적으로 실제 열 번호와 다중 열 문자를 나타냅니다.
- 스트라이 라인 코드 벡터라이저는 벡터화 시 전체 기능을 고려합니다.
- 동일한 변수를 비교하는 일련의 조건 표현식은 각 변수에 비교 표현식이 포함된 경우 switch 문으로 변환할 수 있습니다.
- 최적화 개선 사항:
  - -fipa-modref 옵션으로 제어되는 새로운 IPA-modref 패스는 함수 호출의 부작용을 추적하고 포인트 간 분석의 정확도를 향상시킵니다.
  - -fipa-icf 옵션에 의해 제어되는 동일한 코드 폴딩 패스는 통합 함수의 수를 늘리고 컴파일 타임 메모리 사용을 줄이기 위해 크게 개선되었다.
- 링크 타임 최적화 개선 사항:
  - Link-time optimization (LTO)를 사용하면 컴파일러가 링크 시간에 중간 표현을 사용하여 프로그램의 모든 번역 단위에서 다양한 최적화를 수행할 수 있습니다. 자세한 내용은 링크

시간 최적화를 참조하십시오.

- 최대 메모리 사용을 줄이기 위해 연결 중에 메모리 할당이 향상되었습니다.

- IDE에서 새로운 GCC_EXTRA_DIAGNOSTIC_OUTPUT 환경 변수를 사용하여 빌드 플래그를 조정하지 않고 머신에서 읽을 수 있는 "고정 힌트"를 요청할 수 있습니다.

- -fanalyzer 옵션으로 실행되는 정적 분석기는 다양한 버그 수정 및 개선 사항으로 크게 개선되었습니다.

언어별 개선 사항

C 제품군

- C 및 C++ 컴파일러는 OpenMP 구조에서 비rectangular 루프 중첩과 OpenMP 5.0 사양의 할당기 루틴을 지원합니다.

- 속성:
  - 새로운 no_stack_protector 속성은 스택 보호(-fstack-protector)로 조정되지 않아야 하는 기능을 표시합니다.
  - 개선된 malloc 속성은 할당기 및 deallocator API 쌍을 식별하는 데 사용할 수 있습니다.

- 새 경고:
  - -Wall 옵션으로 사용하도록 설정된 -Wsizeof-array-div 는 첫 번째 값이 배열에 적용되는 경우 두 크기의 연산자에 대해 경고하고 divisor가 배열 요소의 크기와 같지 않습니다.
  - -Wstringop-overread 는 기본적으로 활성화되어 있는 배열의 끝 부분을 인수로 전달한 문자열 함수에 대한 호출에 대해 경고합니다.

- 개선된 경고:
  - **-Wfree-nonheap-object** 는 동적 메모리 할당 함수에서 반환되지 않은 포인터를 사용하여 배치 함수를 처리하는 더 많은 호출 인스턴스를 감지합니다.
  - **-Wmaybe-uninitialized** 는 포인터와 초기화되지 않은 메모리에 대한 참조를 **const-qualified** 인수를 사용하는 함수에 대한 전달을 진단합니다.
  - **-Wuninitialized** 는 동적으로 할당된 메모리에서 초기화되지 않은 읽기를 감지합니다.

C

- 향후 **ISO C2X** 버전의 새로운 기능은 **-std=c2x** 및 **-std=gnu2x** 옵션을 사용하여 지원됩니다. 예를 들면 다음과 같습니다.
  - 표준 속성이 지원됩니다.
  - **__has_c_attribute preprocessor operator**가 지원됩니다.
  - 레이블은 선언 전과 복합 문 끝에 나타날 수 있습니다.**Labels can appear before declarations and at the end of a compound statement.**

C++

- 기본 모드는 **-std=gnu++17** 로 변경됩니다.
- **C++** 라이브러리 **libstdc++** 는 이제 **C++17** 지원이 개선되었습니다.
- 몇 가지 새로운 **C++20** 기능이 구현됩니다. **C++20** 지원은 실험적이라는 점에 유의하십시오.

기능에 대한 자세한 내용은 C++20 언어 기능을 참조하십시오.

- C++ 프런트 엔드에는 향후 C++23 초안 기능에 대한 실험적인 지원이 있습니다.

- 새 경고:

  - -Wctad-maybe-unsupported, 기본적으로 비활성화됨 - deduction guide 없이 클래스 템플릿 인수 deduction을 수행하는 방법에 대해 경고합니다.

  - -Wrange-loop-construct 는 -Wall 에서 활성화하여 범위 기반 루프가 불필요하고 리소스가 비효율적인 사본을 생성하는 경우 경고합니다.

  - -Wall 에서 활성화한 -Wmismatched-new-delete.NET Framework에서 반환한 포인터를 new 또는 기타 일치하지 않는 할당 함수에서 반환된 포인터를 사용하여 삭제에 대해 경고합니다.

  - -Wvexing-parse 는 기본적으로 활성화되어 있는 구문 분석 규칙에 대해 경고합니다. 선언이 변수 정의처럼 보이는 경우 C++ 언어를 함수 선언으로 해석해야 합니다.

아키텍처별 개선 사항

64비트 ARM 아키텍처

- hasv8-R 아키텍처는 -march=armv8-r 옵션을 통해 지원됩니다.

- GCC는 추가, 제거, 곱셈 및 복잡한 숫자에 대한 누적 및 뺀 변형을 수행하는 작업을 자동 수정할 수 있습니다.

AMD 및 Intel 64비트 아키텍처

- Intel AVX-VNNI에 대한 새로운 RuntimeClass 확장 지원이 추가되었습니다. -mavxvnni 컴파일러 전환은 AVX-VNNIhieras를 제어합니다.

- znver3 코어를 기반으로 하는 AMD CPU는 새로운 -march=znver3 옵션을 통해 지원됩니다.

- x86-64 psABI 부록 에 정의된 세 가지 마이크로 아키텍처 수준은 새로운 -march=x86-64-v2,-march=x86-64-v3, -march=x86-64-v4 옵션으로 지원됩니다.

IBM Z 아키텍처

- GCC 11.2.1 기본값은 IBM z14 프로세서입니다.

IBM Power Systems

- GCC 11.2.1 기본값은 IBM POWER9 프로세서입니다.

- GCC 컴파일러는 새로운 -mcpu=power10 명령줄 옵션을 사용하여 POWER10 명령을 지원합니다.

(BZ#1986836, BZ#1870016, BZ#1870025, BZ#1870028, BZ#2019811, BZ#2047296)

glibc 최적화 데이터를 캡처하기 위한 새로운 명령

새로운 ld.so --list-diagnostics 명령은 IFUNC 선택 및 glibc -hwcaps 구성과 같은 glibc 최적화 결정에 영향을 미치는 데이터를 단일 머신에서 읽을 수 있는 파일로 캡처합니다.

(BZ#2023422)

binutils의 주요 변경 사항

RHEL 9에는 binutils 의 다음과 같은 변경 사항이 추가되었습니다.

- binutils 에서는 이제 Intel의 AMX/TMUL 명령어 세트를 지원하므로 이 새로운 기능을 사용할 수 있는 애플리케이션에 대한 성능이 향상되었습니다.

- 어셈블러, 링커 및 기타 바이너리 유틸리티에서 **POWER10** 지침을 지원합니다.

(BZ#2030554, BZ#1870021)

sched_getcpu 구현은 이제 **64**비트 **ARM** 아키텍처 및 기타 아키텍처에서의 성능을 개선하기 위해 **rseq (restartable sequences)**를 선택적으로 사용할 수 있습니다.

64비트 **ARM** 아키텍처에서 **sched_getcpu** 의 이전 구현은 **getcpu** 시스템 호출을 사용하므로 대부분의 병렬 알고리즘에서 효율적으로 사용하기에는 속도가 너무 느립니다. 다른 아키텍처에서는 **vDSO(virtual dynamic shared object)** 가속을 사용하여 이 문제를 해결합니다. **rseq** 를 사용하여 **sched_getcpu** 를 구현하면 **64**비트 **ARM** 아키텍처의 성능이 크게 향상됩니다. 다른 아키텍처에서는 약간의 개선이 표시됩니다.

rseq 를 사용하도록 **sched_getcpu** 를 구성하려면 **GLIBC_TUNABLES=glibc.pthread.rseq=1** 환경 변수를 설정합니다.

```
# GLIBC_TUNABLES=glibc.pthread.rseq=1
# export GLIBC_TUNABLES
```

(BZ#2024347)

업데이트된 성능 도구 및 디버거

RHEL 9.0에서는 다음과 같은 성능 도구와 디버거를 사용할 수 있습니다.

- **GDB 10.2**
- **Valgrind 3.18.1**
- **SystemTap 4.6**
- **Dyninst 11.0.0**
- **elfutils 0.186**

(BZ#2019806)

IBM POWER10의 GDB에서 DAWR 기능 개선

RHEL 9는 향상된 DAWR 기능을 제공하는 GDB 10.2와 함께 배포됩니다. IBM POWER10 프로세서에서 GDB에서 새로운 하드웨어 감시 기능을 사용할 수 있습니다. 예를 들어 새로운 DAWR/DAWRX 레지스터 세트가 추가되었습니다.

(BZ#1870029)

GDB는 IBM POWER10에서 새로운 접두사로 지정된 명령을 지원

GDB 10.2는 POWER10에 대한 사전 요구 사항을 완전히 지원하며, 여기에는 8바이트 접두사가 지정된 지침이 포함됩니다. RHEL 8.4에서는 GDB는 4바이트 명령만 지원했습니다.

(BZ#1870031)

RHEL 9는 boost 1.75.0을 제공합니다.

RHEL 9는 boost 패키지 버전 1.75.0과 함께 배포됩니다. 주요 버그 수정 및 버전 1.67.0에 대한 개선 사항은 다음과 같습니다.

- Boost.Signals 라이브러리가 제거되어 header-only Boost.Signals2 구성 요소로 교체되었습니다.
- boost-jam 패키지의 bjam 도구는 boost- b2 패키지의 b2로 교체되었습니다.
- 새 라이브러리:
  - Boost.Contracts
  - boost.HOF
  - Boost.YAP

- boost.Safe nmerics
- Boost.Outcome
- Boost.Histogram
- Boost.Variant2
- Boost.Nowide
- Boost.StaticString
- Boost.STL_Interfaces
- Boost.JSON
- boost.LEAF
- Boost.PFR

(BZ#1957950)

RHEL 9에서는 LLVM Toolset 13.0.1을 제공합니다.

RHEL 9는 LLVM Toolset 버전 13.0.1과 함께 배포됩니다. 주요 버그 수정 및 버전 12.0.1의 개선 사항은 다음과 같습니다.

- Clang은 이제 C++에서 C++ 및 __attribute__((musttail)) 에서 문 속성 [[clang::musttail]]을 사용하여 보장된 tail 호출을 지원합니다.
-

이제 Clang에서는 코드에서 예약된 식별자를 사용할 때 개발자에게 경고하는 -Wreserved-identifier 경고를 지원합니다.

- 이제 Clang의 -Wshadow 플래그도 새도 구조화된 바인딩을 확인합니다.
- Clang의 -Wextra 도 -Wnull-pointer-subtraction 을 의미합니다.
- Clang은 이제 C++에서 C++ 및 __attribute__((musttail)) 에서 문 속성 [[clang::musttail]] 을 사용하여 보장된 tail 호출을 지원합니다.

RHEL 9에서는 llvm-toolset 을 RPM 패키지로 쉽게 설치할 수 있습니다.

(BZ#2001107)

CMake 3.20.2에서 주요 변경 사항

RHEL 9는 CMake 3.20.2와 함께 배포됩니다. 버전 3.20.2 이하가 필요한 프로젝트에서 CMake를 사용하려면 cmake_minimum_required(버전 3.20.2) 명령을 사용합니다.

주요 변경 사항은 다음과 같습니다.

- C++23 컴파일러 모드는 이제 CXX_STANDARD,CUDA_STANDARD,OBJCXX_STANDARD, 또는 컴파일 기능 함수의 cxx_std_23 메타 기능을 사용하여 대상 속성을 사용하여 지정할 수 있습니다.
- CUDA 언어 지원을 통해 이제 NVIDIA CUDA 컴파일러가 심볼릭 링크가 될 수 있습니다.
- Intel oneAPI NextGen LLVM 컴파일러는 IntelLLVM 컴파일러 ID에서 지원됩니다.
- CMake는 이제 안 드 로이드 NDK의 툴체인 파일과 병합하여 안 드 로이드에 대한 교차 컴파일을 지원합니다.
- cmake(1) 를 실행하여 프로젝트 빌드 시스템을 생성할 때 하이픈으로 시작하는 알 수 없는 명

령줄 인수가 거부됩니다.

새로운 기능 및 더 이상 사용되지 않는 기능에 대한 자세한 내용은 CMake 릴리스 노트 를 참조하십시오.

(BZ#1957948)

RHEL 9에서는 Go 1.17.7을 제공합니다.

RHEL 9는 Go Toolset 버전 1.17.7과 함께 배포됩니다. 주요 버그 수정 및 버전 1.16.7에 대한 개선 사항은 다음과 같습니다.

- 슬라이스를 배열 포인터로 변환하는 옵션이 추가되었습니다.
- //go:build 행에 대한 지원을 추가했습니다.
- amd64의 함수 호출 성능 개선
- 함수 인수는 스택 추적에서 더 명확하게 포맷됩니다.
- 폐쇄를 포함하는 기능은 인라인화될 수 있습니다.
- x509 인증서 구문 분석에서 리소스 사용량 감소.

RHEL 9에서는 go-toolset 을 RPM 패키지로 쉽게 설치할 수 있습니다.

(BZ#2014087)

OpenSSL 3에서 FIPS 모드 지원

이제 FIPS 모드에서 Go의 경우 OpenSSL 3 라이브러리를 사용할 수 있습니다.

(BZ#1984110)

RHEL 9에서는 Rust Toolset 1.58.1을 제공합니다.

RHEL 9는 Rust Toolset 버전 1.58.1과 함께 배포됩니다. 주요 버그 수정 및 버전 1.54.0에 대한 개선 사항은 다음과 같습니다.

- Rust 컴파일러는 이제 2021년 언어 버전을 지원하고, 클로저트 캡처, 배열용 IntoIterator, 새로운 Cargo 기능 확인자 등을 제공합니다.
- 새로운 사용자 지정 프로필에 대한 Cargo 지원이 추가되었습니다.
- eight deduplicates 컴파일러 오류.
- 새로운 오픈 범위 패턴을 추가했습니다.
- 캡처된 식별자를 형식 문자열로 추가했습니다.

자세한 내용은 Rust 1.55 Rust 1.56 Rust 1.57 Rust1.58 을 참조하십시오.

RHEL 9에서는 rust-toolset 을 RPM 패키지로 쉽게 설치할 수 있습니다.

(BZ#2002885)

RHEL 9는 pcp 패키지 버전 5.3.5를 제공합니다.

RHEL 9는 Performance Co-2011(pcp) 패키지 버전 5.3.5와 함께 배포됩니다. 버전 5.3.1 이후로 BPF CO-RE (libbpf 및 BTF)를 사용하여 eBPF 프로그램의 성능 데이터를 제공하는 새로운 pcp-pmda-bpf 하위 패키지가 추가되었습니다.

(BZ#1991764)

PCP에서 SQL Server 메트릭에 액세스하기 위한 Active Directory 인증

이번 업데이트를 통해 시스템 관리자는 AD(Active Directory) 인증을 사용하여 SQL Server 메트릭에 안전하게 연결하도록 pmdamssql(1) 을 구성할 수 있습니다.

(BZ#1847808)

새로운 pcp-ss PCP 유틸리티를 사용할 수 있습니다.

pcp-ss PCP 유틸리티는 pmdasockets(1) PMDA에 의해 수집된 소켓 통계를 보고합니다. 명령은 많은 ss 명령줄 옵션 및 보고 형식과 호환됩니다. 또한 실시간 모드에서 로컬 또는 원격 모니터링의 이점과 이전에 기록된 PCP 아카이브의 기록 재생 기능도 제공합니다.

(BZ#1981223)

RHEL 9에서는 grafana 7.5.11을 제공합니다.

RHEL 9는 grafana 패키지 버전 7.5.11과 함께 배포됩니다. 7.5.9 버전 관련 주요 변경 사항은 다음과 같습니다.

- 새 데이터 프레임 형식을 지원하지 않는 패널의 이전 버전과 호환성을 위해 새로운 준비 시계열 변환을 추가했습니다.
- SHA-1 대신 HMAC-SHA-256을 사용하여 암호 재설정 토큰을 생성하도록 업데이트된 암호 복구 기능

(BZ#1993215)

RHEL 9에서는 grafana-pcp 3.2.0을 제공합니다.

RHEL 9는 grafana-pcp 패키지 버전 3.2.0과 함께 배포됩니다. 주요 버그 수정 및 버전 3.1.0에 대한 개선 사항은 다음과 같습니다.

- PCP Redis에 대한 새 MS SQL 서버 대시보드를 추가했습니다.
- PCP 벡터 eBPF/BCC 개요 대시보드에서 빈 히스토그램 버킷의 가시성이 추가되었습니다.

- PCP Redis의 metric() 기능이 모든 메트릭 이름을 반환하지 않은 버그가 수정되었습니다.

(BZ#1993156)

grafana-pcp의 벡터 데이터 소스에 대한 중앙 pmproxy 를 통해 원격 호스트에 액세스

일부 환경에서는 네트워크 정책에서 대시보드 뷰어의 브라우저에서 모니터링된 호스트로의 연결을 직접 허용하지 않습니다. 이번 업데이트를 통해 중앙 pmproxy 에 연결하도록 hostspec 을 사용자 지정하여 개별 호스트에 요청을 전달할 수 있습니다.

(BZ#1845592)

새로운 패키지: ansible-pcp

ansible-pcp 패키지에는 메트릭 RHEL 시스템 역할을 구현하는 데 사용되는 PCP(Performance Co-dpdk) 및 Redis 및 Grafana와 같은 관련 소프트웨어가 포함되어 있습니다.

(BZ#1957566)

RHEL 9에서는 python-jsonpointer 2.0을 제공합니다.

RHEL 9는 python-jsonpointer 패키지 버전 2.0과 함께 배포됩니다.

버전 1.9에 대한 주요 변경 사항은 다음과 같습니다.

- Python 버전 2.6 및 3.3은 더 이상 사용되지 않습니다.
- python-jsonpointer 모듈은 이제 잘못된 이스케이프 시퀀스에 대한 포인터를 자동으로 확인합니다.
- 이제 명령줄에서 포인터를 인수로 작성할 수 있습니다.
- 포인터는 더 이상 URL 인코딩 형식으로 제출할 수 없습니다.

(BZ#1980256)

.NET 6.0 사용 가능

RHEL 9는 .NET 버전 6.0과 함께 배포됩니다. 주요 개선사항은 다음과 같습니다.

- 64비트 ARM(aarch64) 지원
- IBM Z 및 LinuxONE 지원 (s390x)

자세한 내용은 .NET 6.0 RPM 패키지 릴리스 정보 및 .NET 6.0 컨테이너 릴리스 정보를 참조하십시오.

.NET 6.0 은 이 Application Stream의 초기 버전으로 RPM 패키지로 쉽게 설치할 수 있습니다. .NET 6.0 은 RHEL 9보다 라이프사이클이 짧습니다. 자세한 내용은 Red Hat Enterprise Linux Application Streams 라이프 사이클 문서를 참조하십시오.

(BZ#1986211)

RHEL 9의 Java 구현

RHEL 9 AppStream 리포지토리에는 다음이 포함됩니다.

- OpenJDK 17 Java 런타임 환경 및 OpenJDK 17 Java 소프트웨어 개발 키트를 제공하는 java-17-openjdk 패키지
- OpenJDK 11 Java 런타임 환경 및 OpenJDK 11 Java 소프트웨어 개발 키트를 제공하는 java-11-openjdk 패키지
- OpenJDK 8 Java 런타임 환경 및 OpenJDK 8 Java 소프트웨어 개발 키트를 제공하는 java-1.8.0-openjdk 패키지

자세한 내용은 OpenJDK 설명서 를 참조하십시오.

(BZ#2021262)

RHEL 9의 Java 툴

RHEL 9 AppStream 리포지토리에는 다음과 같은 Java 툴이 포함되어 있습니다.

- Maven 3.6. 3: 소프트웨어 프로젝트 관리 및 이해 툴.
- ant 1.10.9 - Java 애플리케이션을 컴파일, 어셈블링, 테스트 및 실행을 위한 Java 라이브러리 및 명령줄 툴입니다.

Maven 3.6 및 Ant 1.10 은 이러한 애플리케이션 스트림의 초기 버전입니다.이 Application Streams는 비모듈 RPM 패키지로 쉽게 설치할 수 있습니다.

(BZ#1951482)

CRB 리포지토리에서 SWIG 4.0 사용 가능

SWIG(Simplified Wrapper and Interface Generator) 버전 4.0은 CodeReady Linux Builder(CRB) 리포지토리에서 사용할 수 있습니다. 이 릴리스에는 PHP 8 에 대한 지원이 추가되었습니다.

RHEL 9에서는 SWIG 를 RPM 패키지로 쉽게 설치할 수 있습니다.

CodeReady Linux Builder 리포지토리에 포함된 패키지는 지원되지 않습니다.

(BZ#1943580)

4.15. IDM (IDENTITY MANAGEMENT)

Directory Server에서 더 이상 글로벌 변경 로그를 사용하지 않음

이번 개선된 기능을 통해 Directory Server 변경 로그가 기본 데이터베이스에 통합되었습니다. 이전에는 Directory Server가 글로벌 변경 로그를 사용했습니다. 그러나 디렉터리가 여러 데이터베이스를 사용하는 경우 문제가 발생할 수 있습니다. 결과적으로 각 접미사는 이제 일반 데이터베이스 파일과 동일한 디렉터리에 자체 변경 로그를 가졌습니다.

(BZ#1805717)

모든 종속 항목이 있는 AppStream 리포지토리에서 Ansible -freeipa 를 사용할 수 있습니다.

이전에는 ansible-freeipa 패키지를 설치하기 전에 Ansible 리포지토리를 활성화하고 ansible 패키지를 설치해야 했습니다. RHEL 8.6 및 RHEL 9에서는 예비 단계 없이 ansible-freeipa 를 설치할 수 있습니다. ansible -freeipa 를 설치하면 보다 기본적인 Ansible 버전인 ansible-core 패키지가 종속성으로 자동으로 설치됩니다. ansible-freeipa 및 ansible-core 는 모두 rhel-9-for-x86_64-appstream-rpms 리포지토리에서 사용할 수 있습니다.

RHEL 8.6 및 RHEL 9의 Ansible -freeipa 에는 RHEL 8에 포함된 모든 모듈이 포함되어 있습니다.

(JIRA:RHELPLAN-100359)

IdM에서 자동 마운트 위치 지원,자동 마운트 맵, 자동 마운트 키 Ansible 모듈 지원

이번 업데이트를 통해 ansible-freeipa 패키지에는 ipaautomountlocation,ipaautomountmap, ipaautomountkey 모듈이 포함되어 있습니다. IdM 위치에 있는 IdM 클라이언트에 로그인한 IdM 사용자를 위해 자동으로 마운트되도록 디렉터리를 구성하는 데 이러한 모듈을 사용할 수 있습니다. 현재는 직접 맵만 지원됩니다.

(JIRA:RHELPLAN-79161)

하위 ID 범위를 관리하는 지원은 shadow-utils에서 사용할 수 있습니다.

이전에는 shadow-utils 에서 /etc/subuid 및 /etc/subgid 파일에서 자동으로 subID 범위를 구성했습니다. 이번 업데이트를 통해 subid 필드에 값을 설정하여 /etc/nsswitch.conf 파일에서 subID 범위의 구성을 사용할 수 있습니다. 자세한 내용은 man subuid 및 man subgid 를 참조하십시오. 또한 이번 업데이트를 통해 IPA 서버의 하위 ID 범위를 제공하는 shadow-utils 플러그인의 SSSD 구현을 사용할 수 있습니다. 이 기능을 사용하려면 subid: sss 값을 /etc/nsswitch.conf 파일에 추가합니다. 이 솔루션은 rootless 컨테이너를 용이하게 하기 위해 컨테이너화된 환경에서 유용할 수 있습니다.

/etc/nsswitch.conf 파일이 authselect 툴에 의해 구성된 경우 authselect 설명서에 설명된 절차를 따라야 합니다. 그렇지 않은 경우 /etc/nsswitch.conf 파일을 수동으로 수정할 수 있습니다.

(BZ#1859252)

하위 ID 범위 관리 지원은 IdM에서 사용할 수 있습니다.

이번 업데이트를 통해 ID 관리에서 사용자의 ID 하위 범위를 관리할 수 있습니다. ipa CLI 툴 또는 IdM WebUI 인터페이스를 사용하여 자동으로 구성된 하위 ID 범위를 사용자에게 할당할 수 있습니다. 이 범위는 컨테이너화된 환경에서 유용할 수 있습니다.

(BZ#1952028)

ID 관리 설치 패키지가 데모되었습니다.

이전에는 RHEL 8에서 IdM 패키지가 모듈로 배포되었으므로 스트림을 활성화하고 원하는 설치에 해당하는 프로필을 설치해야 합니다. IdM 설치 패키지는 RHEL 9에서 시연되었으므로 다음 dnf 명령을 사용하여 IdM 서버 패키지를 설치할 수 있습니다.

통합된 DNS 서비스가 없는 서버의 경우:

```
# dnf install ipa-server
```

통합된 DNS 서비스가 있는 서버의 경우:

```
# dnf install ipa-server ipa-server-dns
```

(BZ#2080875)

기존 RHEL ansible-freeipa 리포지터리의 대안: Ansible Automation Hub

이번 업데이트를 통해 표준 RHEL 리포지토리에서 다운로드하지 않고 AAH(Ansible Automation Hub)에서 ansible-freeipa 모듈을 다운로드할 수 있습니다. AAH를 사용하면 이 리포지토리에서 사용할 수 있는 ansible-freeipa 모듈의 빠른 업데이트의 이점을 누릴 수 있습니다.

AAH에서 ansible-freeipa 역할 및 모듈은 컬렉션 형식으로 배포됩니다. AAH 포털의 콘텐츠에 액세스하려면 Ansible Automation Platform (AAP) 서브스크립션이 필요합니다. ansible 버전 2.9 이상도 필요합니다.

redhat.rhel_idm 컬렉션에는 기존 ansible-freeipa 패키지와 동일한 콘텐츠가 있습니다. 그러나 컬렉션 형식은 네임스페이스와 컬렉션 이름으로 구성된 정규화된 FQCN(정규화된 컬렉션 이름)을 사용합니다. 예를 들어 redhat.rhel_idm.ipadnsconfig 모듈은 RHEL 리포지토리에서 제공하는 ansible-freeipa 의 ipadnsconfig 모듈에 해당합니다. 네임스페이스와 컬렉션 이름을 조합하면 개체가 고유하며 충돌 없이 공유할 수 있습니다.

(JIRA:RHELPLAN-103147)

이제 Ansible-freeipa 모듈이 IdM 클라이언트에서 원격으로 실행될 수 있습니다.

이전에는 ansible-freeipa 모듈을 IdM 서버에서만 실행할 수 있었습니다. 이를 위해서는 Ansible 관리자가 IdM 서버에 대한 SSH 액세스 권한을 보유하여 잠재적인 보안 위협을 초래해야 했습니다. 이번 업데이트를 통해 IdM 클라이언트인 시스템에서 ansible-freeipa 모듈을 원격으로 실행할 수 있습니다. 따라서 IdM 구성 및 엔터티를 보다 안전한 방식으로 관리할 수 있습니다.

IdM 클라이언트에서 ansible-freeipa 모듈을 실행하려면 다음 옵션 중 하나를 선택합니다.

- 플레이북의 hosts 변수를 IdM 클라이언트 호스트로 설정합니다.
- ansible-free ipa 모듈을 사용하는 플레이북 작업에 ipa_context: 클라이언트 행을 추가합니다.

ipa_context 변수를 IdM 서버의 클라이언트로 설정할 수도 있습니다. 그러나 서버 컨텍스트는 일반적으로 더 나은 성능을 제공합니다. ipa_context 가 설정되지 않은 경우 ansible-freeipa 는 서버 또는 클라이언트에서 실행 중인지 확인하고 그에 따라 컨텍스트를 설정합니다. IdM 클라이언트 호스트에서 컨텍스트 가 server 로 설정된 ansible-freeipa 모듈을 실행하면 라이브러리가 누락된 오류가 발생합니다.

(JIRA:RHELPLAN-103146)

ipadnsconfig 모듈에는 global forwarder를 제외하려면 action: member 가 필요합니다.

이번 업데이트를 통해 ansible-freeipa ipadnsconfig 모듈을 사용하여 IdM(Identity Management)에서 전역 전달자를 제외하려면 state: absent 옵션 외에 action: member 옵션을 사용해야 합니다. action: member 를 사용하지 않고 플레이북에서 state: absent 만 사용하는 경우 플레이북이 실패합니다. 결과적으로 모든 글로벌 전달자를 제거하려면 플레이북에서 모두 개별적으로 지정해야 합니다. 반대로 state: present 옵션에는 action: member 가 필요하지 않습니다.

(BZ#2046325)

AD 사용자의 자동 개인 그룹은 중앙 집중식 구성을 지원합니다.

이제 IdM 클라이언트에서 인증된 SSSD 버전을 신뢰할 수 있는 Active Directory 도메인에서 사용자의 개인 그룹을 관리하는 방법을 중앙에서 정의할 수 있습니다. 이번 개선된 기능을 통해 AD 사용자를 처리하는 ID 범위에 대해 SSSD의 auto_private_groups 옵션 값을 명시적으로 설정할 수 있습니다.

auto_private_groups 옵션이 명시적으로 설정되지 않은 경우 기본값을 사용합니다.

- ipa-ad-trust-posix ID 범위의 경우 기본값은 false 입니다. SSSD는 항상 AD 항목의 uidNumber 및 gidNumber 를 사용합니다. gidNumber 가 있는 그룹은 AD에 있어야 합니다.

- ipa-ad-trust ID 범위의 기본값은 true 입니다. SSSD에서 uidNumber 를 항목 InstallPlan에서 매핑하고 gidNumber 는 항상 동일한 값으로 설정되고 개인 그룹은 항상 매핑됩니다.

auto_private_groups 을 세 번째 설정인 hybrid 에도 설정할 수 있습니다. 이 설정을 사용하면 사용자 항목에 UID와 동일한 GID가 있지만 이 GID가 있는 그룹이 없는 경우 SSSD는 개인 그룹을 매핑합니다. UID와 GID가 다르면 이 GID 번호가 있는 그룹이 있어야 합니다.

이 기능은 사용자 개인 그룹에 대해 별도의 그룹 오브젝트를 유지 관리하는 것을 중지하고 기존 사용자 개인 그룹도 유지하려는 관리자에게 유용합니다.

(BZ#1957736)

BIND에 대한 사용자 정의 로깅 설정

이번 개선된 기능을 통해 /etc/named/ipa-logging-ext.conf 구성 파일에서 ID 관리 서버의 BIND DNS 서버 구성 요소에 대한 로깅 설정을 구성할 수 있습니다.

(BZ#1966101)

IdM 키 탭을 검색할 때 IdM 서버 자동 검색

이번 개선된 기능을 통해 ipa-getkeytab 명령을 사용하여 Kerberos 키 탭을 검색할 때 IdM 서버 호스트 이름을 더 이상 지정할 필요가 없습니다. 서버 호스트 이름을 지정하지 않으면 DNS 검색이 IdM 서버를 찾는 데 사용됩니다. 서버를 찾을 수 없는 경우 이 명령은 /etc/ipa/default.conf 구성 파일에 지정된 호스트 값으로 대체합니다.

(BZ#1988383)

익명의 PKINIT CMS 메시지는 이제 SHA-2로 서명됩니다.

SHA-1 다이제스트 알고리즘은 RHEL 9에서 더 이상 사용되지 않으며 익명의 PKINIT에 대한 CMS 메시지는 더 강력한 SHA-2 알고리즘으로 서명됩니다.

RHEL 7.8 및 RHEL 8.6 이하에서는 여전히 SHA-1 다이제스트 알고리즘을 사용하여 CMS 메시지에 서명하기 위해 RHEL 7.9 및 RHEL 8.7 이전의 KDC(Kerberos Key Distribution Center)에서 SHA-2를 사용합니다. 이러한 이전 requirements와의 상호 운용성이 필요한 경우 다음 명령을 사용하여 RHEL 9 서버에서 SHA-1 알고리즘을 지원할 수 있습니다.

```
# update-crypto-policies --set DEFAULT:SHA1
```

RHEL 9 Kerberos 클라이언트가 Heimdal EgressIP에 대해 PKINIT를 사용하여 사용자를 인증하지 못했습니다. 도 참조하십시오.

(BZ#2060798)

RHEL 9에서 Samba 4.15.5 제공

RHEL 9는 버전 4.14에 대한 버그 수정 및 개선 사항을 제공하는 Samba 4.15.5와 함께 배포됩니다.

- 일관된 사용자 환경을 위해 Samba 유틸리티의 옵션의 이름이 변경 및 제거되었습니다.
- 서버 다중 채널 지원이 기본적으로 활성화되어 있습니다.
- Windows 기술 프리뷰에서만 사용되었던 SMB2_22,SMB2_24, SMB3_10 방언이 제거되었습니다.

Samba를 시작하기 전에 데이터베이스 파일을 백업하십시오. redhatd ,nmbd, 또는 winbind 서비스가 시작되면 Samba는 tdb 데이터베이스 파일을 자동으로 업데이트합니다. Red Hat은 downgrading tdb 데이터베이스 파일을 지원하지 않습니다.

Samba를 업데이트한 후 testparm 유틸리티를 사용하여 /etc/hiera/hiera.conf 파일을 확인합니다.

주요 변경 사항에 대한 자세한 내용은 업데이트 전에 업스트림 릴리스 노트를 참조하십시오.

(BZ#2013578)

로그 분석기 툴을 사용하여 클라이언트 요청 추적

SSSD(System Security Services Daemon)에는 여러 SSSD 구성 요소의 로그 파일에서 처음부터 끝까지 요청을 추적하는 로그 구문 분석 도구가 포함되어 있습니다.

로그 분석기 툴을 사용하면 SSSD 디버그 로그를 보다 쉽게 검토할 수 있으므로 SSSD의 문제를 보다 쉽게 해결할 수 있습니다. 예를 들어 SSSD 프로세스의 특정 클라이언트 요청에만 관련된 SSSD 로그를 추출하고 출력할 수 있습니다. Analyzer 툴을 실행하려면 sssctl analyze 명령을 사용합니다.

(JIRA:RHELPLAN-97899)

SSSD는 기본적으로 역추적을 기록합니다.

이 향상된 기능을 통해 SSSD는 메모리 내 버퍼에 자세한 디버그 로그를 저장하고 오류가 발생할 때 로그 파일에 추가합니다. 기본적으로 다음 오류 수준에서는 역추적이 트리거됩니다.

- 수준 0: 치명적인 오류
- 레벨 1 : 중요한 실패
- 레벨 2 : 심각한 실패

sssd.conf 설정 파일의 해당 섹션에 debug_level 옵션을 설정하여 각 SSSD 프로세스에 대해 이 동작을 수정할 수 있습니다.

- 디버깅 수준을 0으로 설정하면 수준 0 이벤트만 역추적을 트리거합니다.
- 디버깅 수준을 1로 설정하고 수준 0 및 1로 설정하면 역추적이 트리거됩니다.
- 디버깅 수준을 2 이상으로 설정하면 수준 0에서 2까지의 이벤트가 역추적을 트리거합니다.

sssd.conf 의 해당 섹션에서 debug_backtrace_enabled 옵션을 false 로 설정하여 SSSD 프로세스별로 이 기능을 비활성화할 수 있습니다.

-

```
[sssd]
debug_backtrace_enabled = true
debug_level=0
...

[nss]
debug_backtrace_enabled = false
...

[domain/idm.example.com]
debug_backtrace_enabled = true
debug_level=2
...

...
```

(BZ#1949149)

SSSD 기본 SSH 해싱 값이 OpenSSH 설정과 일치합니다.

ssh_hash_known_hosts 의 기본값이 false로 변경되었습니다. 이제 기본적으로 해시 호스트 이름이 아닌 OpenSSH 설정과 일치합니다.

그러나 호스트 이름을 계속 해시해야 하는 경우 /etc/sssd/sssd.conf 구성 파일의 [ssh] 섹션에 ssh_hash_known_hosts = True 를 추가합니다.

(BZ#2014249)

Directory Server 12.0은 업스트림 버전 2.0.14를 기반으로 합니다.

Directory Server 12.0은 이전 버전에 비해 여러 버그 수정 및 개선 사항을 제공하는 업스트림 버전 2.0.14를 기반으로 합니다. 주요 변경 사항 전체 목록은 업데이트하기 전에 업스트림 릴리스 노트를 참조하십시오.

- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-14.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-13.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-12.html

- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-11.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-10.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-9.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-8.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-7.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-6.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-5.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-4.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-3.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-2.html
- https://directory.fedoraproject.org/docs/389ds/releases/release-2-0-1.html

(BZ#2024693)

Directory Server는 이제 **tmpfs** 파일 시스템에 데이터베이스의 메모리 매핑된 파일을 저장합니다.

Directory Server에서 **nsslapd-db-home-directory** 매개 변수는 데이터베이스의 메모리 매핑된 파일의 위치를 정의합니다. 이번 개선된 기능을 통해 매개변수의 기본값을 **/var/lib/dirsrv/slapd-*instance_name*/db/** 에서 **/dev/shm/** 로 변경합니다. 결과적으로 **tmpfs** 파일 시스템에 저장된 내부 데이터베이스를 사용하면 **Directory Server** 성능이 향상됩니다.

(BZ#2088414)

4.16. 데스크탑

GNOME이 버전 40으로 업데이트

GNOME 환경이 GNOME 3.28에서 GNOME 40으로 업데이트되고 많은 새로운 기능이 추가되었습니다.

GNOME 40에는 새로운 활동 개요 설계가 포함되어 있습니다. 이렇게 하면 개요가 더 일관된 보기를 제공하며, 시스템 탐색 및 애플리케이션 시작을 위한 향상된 환경을 제공합니다. 이제 작업 공간이 수평으로 정렬되고 창 개요 및 애플리케이션 그리드가 수직으로 액세스됩니다.

GNOME의 기타 개선 사항은 다음과 같습니다.

- GNOME의 성능과 리소스 사용량이 크게 향상되었습니다.
- 사용자 인터페이스, 아이콘 및 데스크탑을 포함한 시각적 스타일이 새로 고쳐졌습니다.
- GNOME 애플리케이션은 상단 패널에서 사용할 수 있는 애플리케이션 메뉴를 더 이상 사용하지 않습니다. 이제 기능이 애플리케이션 창의 주 메뉴에 있습니다.
- Settings(설정) 애플리케이션이 재설계되었습니다.
- 화면 공유 및 원격 데스크탑 세션이 개선되었습니다.
- 독점 NVIDIA 드라이버를 사용하는 경우 이제 개별 GPU를 사용하여 애플리케이션을 실행할 수 있습니다.

  a. 개요를 엽니다.

  b. 대시에서 애플리케이션 아이콘을 마우스 오른쪽 버튼으로 클릭합니다.

c. 메뉴에서 **Launch on Discrete GPU** 항목을 선택합니다.

- 이제 **Power Off / Log Out** (전원 끄기/로그아웃) 메뉴에 **Suspend** (일시 중지) 옵션과 새 **Restart** 옵션이 포함되어 **Alt** 를 사용할 때 부트 로더 메뉴로 시스템을 재부팅할 수 있습니다.

- **Flatpak** 애플리케이션이 자동으로 업데이트됩니다.

- 이제 드래그 앤 드롭을 사용하여 개요에서 애플리케이션 아이콘을 폴더로 그룹화할 수 있습니다.

- 터미널 애플리케이션에서는 이제 오른쪽에서 왼쪽으로 또는 양방향 텍스트를 지원합니다.

- 포인터 위치 접근성 기능은 **Wayland** 세션에서 작동합니다. 기능을 활성화하면 **Ctrl** 을 누르면 화면의 포인터 위치가 강조 표시됩니다.

- **GNOME** 쉘 확장은 이제 **Software**가 아닌 **Extensions** 응용 프로그램에서 관리합니다. 확장 프로그램 애플리케이션은 업데이트 확장 기능, 확장 기본 설정 구성, 확장 제거 또는 비활성화를 처리합니다.

- 알림 팝업에 **Do Not Disturb** 버튼이 포함됩니다. 버튼을 활성화하면 화면에 알림이 표시되지 않습니다.

- 이제 암호가 필요한 시스템 대화 상자에 눈 (**A**) 아이콘을 클릭하여 암호 텍스트를 표시하는 옵션이 있습니다.

- 소프트웨어 애플리케이션은 이제 모바일 데이터 네트워크와 같은 측정된 네트워크를 자동으로 감지합니다. 현재 네트워크가 측정되면 소프트웨어는 데이터 사용량을 줄이기 위해 업데이트를 일시 중지합니다.

- 연결된 각 디스플레이는 **Wayland** 세션에서 다른 새로 고침 속도를 사용할 수 있습니다.

- 부분 표시 스케일링은 실험적 옵션으로 사용할 수 있습니다. 여기에는 사전 구성된 몇 가지 분수 비율이 포함됩니다.

실험적 분별 확장을 활성화하려면 활성화된 실험적 기능 목록에 **scale-monitor-framebuffer** 값을 추가합니다.

```
$ dconf write \
    /org/gnome/mutter/experimental-features \
    "['scale-monitor-framebuffer']"
```

결과적으로 부분 확장 옵션은 **Settings** (설정)의 **Display** (표시) 패널에서 액세스할 수 있습니다.

GNOME의 변경 사항에 대한 자세한 내용은 릴리스 노트 의 버전 **3.30~40.0**을 참조하십시오.

(JIRA:RHELPLAN-101240)

pipeWire가 이제 기본 오디오 서비스입니다.

이제 **Pipe wire** 서비스에서 모든 오디오 출력 및 입력을 관리합니다. **pipeWire** 는 일반적으로 사용 사례에서 **PulseAudio** 서비스를 대체하고 전문적인 사용 사례의 **JACK** 서비스를 대체합니다. 이제 시스템에서 **PulseAudio,JACK** 또는 **ALSA** 프레임워크를 사용하는 애플리케이션에서 오디오를 **Pipe wire**로 리디렉션합니다.

이전 솔루션보다 파이프 와 연결된의 이점은 다음과 같습니다.

- 소비자 및 전문 사용자를 위한 통합 솔루션
- 유연한 모듈식 아키텍처
- **JACK** 서비스와 유사한 고성능 및 짧은 대기 시간
- 향상된 보안을 위해 오디오 클라이언트 간의 격리

더 이상 이를 사용하는 애플리케이션에 대해 **JACK** 서비스를 구성할 필요가 없습니다. 이제 모든 **JACK** 애플리케이션이 기본 **RHEL** 구성에서 작동합니다.

PulseAudio 는 RHEL에서 계속 사용할 수 있으며 PipeWire 대신 활성화할 수 있습니다. 자세한 내용은 PipeWire에서 PulseAudio로 전환을 참조하십시오.

(JIRA:RHELPLAN-101241)

GNOME에서 power 프로필을 사용할 수 있습니다.

이제 GNOME 환경의 설정 전원 패널에서 여러 전원 프로필 간에 전환할 수 있습니다. power 프로필은 선택한 목표에 맞게 다양한 시스템 설정을 최적화합니다.

다음과 같은 전원 프로필을 사용할 수 있습니다.

성능

높은 시스템 성능을 위해 최적화되고 배터리 수명이 감소합니다. 이 프로필은 선택한 특정 시스템 구성에서만 사용할 수 있습니다.

균형

표준 시스템 성능 및 전력 소비를 제공합니다. 기본 프로필입니다.

절전 관리

배터리 수명 증가 및 시스템 성능 감소. 이 프로필은 낮은 배터리에서 자동으로 활성화됩니다.

전원 프로파일 구성은 시스템 재부팅 후에도 유지됩니다.

power 프로필 기능은 기본적으로 설치된 power-profiles-daemon 패키지에서 사용할 수 있습니다.

(JIRA:RHELPLAN-101242)

언어 지원은 이제 langpacks에서 제공합니다.

이제 langpack 패키지에서 다양한 언어에 대한 지원을 사용할 수 있습니다. 다음 패키지 이름을 사용하여 설치할 언어 지원 수준을 사용자 지정할 수 있습니다. 여기서 *code* 는 스페인어 es 와 같이 언어의 짧은 ISO 코드입니다.

langpacks-core-*code*

다음을 포함하여 기본 언어 지원을 제공합니다.

- glibc locale
- 기본 글꼴
- 언어에 필요한 기본 입력 방법

langpacks-core-font-*code*

언어의 기본 글꼴만 제공합니다.

langpacks-*code*

기본 언어 지원 외에도 다음과 같은 전체 언어 지원을 제공합니다.

- 번역
- 맞춤 검사기 사전
- 추가 글꼴

(JIRA:RHELPLAN-101247)

경량 단일 애플리케이션 환경

단일 애플리케이션만 제공하는 그래픽 사용 사례의 경우 이제 **UI(Lightweight User Interface)**를 사용할 수 있습니다.

키오스크 모드라고도 하는 단일 애플리케이션 세션에서 **GNOME**을 시작할 수 있습니다. 이 세션에서 **GNOME**은 구성한 애플리케이션의 전체 화면 창만 표시합니다.

단일 애플리케이션 세션은 표준 GNOME 세션보다 훨씬 적은 리소스 집약적입니다.

자세한 내용은 세션 제한을 단일 애플리케이션으로 제한을 참조하십시오.

(JIRA:RHELPLAN-102552)

로그인 및 데스크탑 세션의 보안 분류 배너

이제 시스템의 전체 보안 분류 수준을 표시하도록 분류 배너를 구성할 수 있습니다. 이는 사용자가 로그인한 시스템의 보안 분류 수준을 알고 있어야 하는 배포에 유용합니다.

구성에 따라 분류 배너는 다음 컨텍스트에서 표시될 수 있습니다.

- 실행 중인 세션 내에서
- 잠금 화면에서
- 로그인 화면에서

분류 배너는 준수할 수 있는 알림 형식 또는 영구 배너를 사용할 수 있습니다.

자세한 내용은 시스템 보안 분류 표시를 참조하십시오.

(BZ#2031186)

기본 월페이퍼는 Red Hat 로고를 추가합니다.

기본 RHEL 배경 화면이 이제 Red Hat 로고가 표시됩니다. 로고는 화면의 왼쪽 상단에 있습니다.

로고를 비활성화하려면 Background Logo GNOME Shell 확장을 비활성화합니다.

(BZ#2057150)

Firefox에서 PKCS#12 파일에서 더 강력한 암호화 사용

Firefox 웹 브라우저에서는 PKCS#12 파일을 사용하여 클라이언트 인증 인증서를 설정합니다. 이전에는 Firefox에서 기존 알고리즘을 사용하여 이러한 파일을 암호화했습니다.

- PBE-SHA1-RC2-40 - PKCS#12 파일의 인증서를 암호화
- PKCS#12 파일의 키를 암호화하기 위한 PBE-SHA1-3DES

이번 릴리스에서는 Firefox는 기본적으로 더 강력한 알고리즘을 사용하여 파일을 암호화합니다.

- PKCS#12 파일의 인증서를 암호화하기 위해 PBKDF2가 포함된 AES-256-CBC
- PKCS#12 파일의 키를 암호화하기 위한 PBKDF2가 있는 AES-128-CBC

이러한 변경으로 PKCS#12 파일은 이제 연방 정보 처리 표준(FIPS)과 호환됩니다.

레거시 암호화 알고리즘은 Firefox에서 기본이 아닌 옵션으로 계속 지원됩니다.

(BZ#1764205)

4.17. 그래픽 인프라

Wayland 세션은 이제 NVIDIA 드라이버에서 기본으로 설정되어 있습니다.

이제 NVIDIA 드라이버를 사용하는 경우 드라이버 구성에서 Wayland를 지원하는 경우 데스크탑 세션은 기본적으로 Wayland 디스플레이 프로토콜을 선택합니다. 이전 RHEL 릴리스에서 NVIDIA 드라이버는 항상 Wayland를 비활성화했습니다.

시스템에서 NVIDIA 드라이버를 사용하여 Wayland를 활성화하려면 커널 명령줄에 다음 옵션을 추가합니다.

- nvidia-drm.modeset=1
- NVreg_PreserveVideoMemoryAllocations=1

Wayland는 RHEL 8.0 이후 다른 그래픽 드라이버와 함께 기본 디스플레이 프로토콜이었습니다.

현재 NVIDIA 드라이버와 함께 Wayland 세션은 아직 불완전하며 특정 알려진 문제가 있습니다. Red Hat은 GPU 스택의 이러한 격차와 문제를 해결하기 위해 NVIDIA와 적극적으로 협력하고 있습니다.

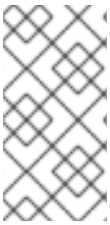
NVIDIA 드라이버와 함께 Wayland의 일부 제한 사항은 *알려진 문제* 섹션을 참조하십시오.

(JIRA:RHELPLAN-119000)

4.18. 웹 콘솔

웹 콘솔에서 sudo 및 SSH에 대한 스마트 카드 인증

이전에는 스마트 카드 인증을 사용하여 sudo 권한을 획득하거나 웹 콘솔에서 SSH를 사용할 수 없었습니다. 이번 업데이트를 통해 ID 관리 사용자는 스마트 카드를 사용하여 sudo 권한을 얻거나 SSH를 사용하여 다른 호스트에 연결할 수 있습니다.



참고

하나의 스마트 카드를 사용하여 sudo 권한을 인증하고 취득할 수 있습니다. sudo에 별도의 스마트 카드 사용은 지원되지 않습니다.

(JIRA:RHELPLAN-95126)

웹 콘솔에서 재부팅하지 않고 커널 보안 패치

이 웹 콘솔 업데이트를 사용하면 kpatch 프레임워크를 사용하여 재부팅하지 않고도 커널 보안 패치를 적용할 수 있습니다. 관리자는 또한 향후 커널을 라이브 패치 스트림에 자동으로 구독할 수 있습니다.

(JIRA:RHELPLAN-95056)

RHEL 웹 콘솔에서는 기본적으로 Insights 등록을 제공합니다.

이번 업데이트를 통해 Red Hat Enterprise Linux 웹 콘솔을 사용하여 RHEL 시스템을 등록할 때 이 시스템을 Red Hat Insights에 연결 확인란을 선택합니다. Insights 서비스에 연결하지 않으려면 상자를 선택 취소합니다.

(BZ#2049441)

Cockpit에서 기존 TLS 인증서 사용을 지원

이번 개선된 기능을 통해 인증서에 더 엄격한 파일 권한 요구 사항(예: root:cockpit-ws 0640)이 없으므로 다른 서비스와 공유할 수 있습니다.

(JIRA:RHELPLAN-103855)

4.19. RED HAT ENTERPRISE LINUX 시스템 역할

네트워킹 시스템 역할이 SAE 지원

Wi-Fi 보호 액세스 버전 3(WPA3) 네트워크에서 동일한 인증(SAE)의 동시 인증으로 암호화 키가 전송되지 않습니다. 이번 개선된 기능을 통해 네트워킹 RHEL 시스템 역할은 SAE를 지원합니다. 결과적으로 관리자는 이제 Networking 시스템 역할을 사용하여 WPA-SAE를 사용하는 Wi-Fi 네트워크에 대한 연결을 구성할 수 있습니다.

(BZ#1993304)

네트워킹 시스템 역할은 이제 owe를 지원

네트워킹 RHEL 시스템 역할은 이제 Opportunistic wireless Encryption (owe)을 지원합니다. owe는 Wi-Fi 클라이언트와 액세스 포인트 사이의 암호화를 사용하고 Wi-Fi 클라이언트를 스니핑 공격으로부터 보호하는 무선 인증 키 관리 유형입니다. owe를 사용하려면 무선 인증 키 관리 유형인key_backend 필드를 owe 로 설정합니다.

(BZ#1993377)

방화벽 시스템 역할에서 방화벽 기본 영역 설정을 지원

영역은 들어오는 트래픽을 더 투명하게 관리하는 개념을 나타냅니다. 영역은 네트워킹 인터페이스에 연결되거나 다양한 소스 주소가 할당됩니다. 각 영역의 방화벽 규칙은 관리자가 복잡한 방화벽 설정을 정의하고 트래픽에 적용할 수 있도록 독립적으로 관리됩니다. 이 기능을 사용하면 기본 영역으로 사용되는 기본 영역을 설정하여 **firewall-cmd --set-default-zone zone-name** 과 동일한 인터페이스를 할당할 수 있습니다.

(BZ#2022461)

스토리지 RHEL 시스템 역할에서 LVM VDO 볼륨 지원

이번 개선된 기능을 통해 Storage System Role을 사용하여 LVM(Logical Manager Volumes) VDO(Virtual Data Optimizer) 볼륨을 관리할 수 있습니다. LVM 파일 시스템은 VDO 볼륨을 관리하고 이 기능을 사용하면 LVM 볼륨에서 압축하고 중복될 수 있습니다. 따라서 VDO는 스토리지 볼륨의 사용을 최적화하는 데 도움이 됩니다.

(BZ#1978488)

스토리지 시스템 역할에서 백분율로 표시되는 볼륨 크기에 대한 지원을 사용할 수 있습니다.

이번 개선된 기능으로 LVM 볼륨 크기를 풀 전체의 백분율로 표현하기 위해 Storage RHEL 시스템 역할에 지원이 추가되었습니다. LVM 볼륨의 크기를 풀/VG 크기의 백분율로 지정할 수 있습니다. 예를 들면 다음과 같습니다. 파일 시스템의 사람이 읽을 수 있는 크기(예: 10g, 50GiB) 외에도 50%가 필요합니다.

(BZ#1984583)

캐시된 볼륨에 대한 지원은 스토리지 시스템 역할에서 사용할 수 있습니다.

이번 개선된 기능에는 캐시된 LVM 논리 볼륨을 생성 및 관리할 수 있는 스토리지 RHEL 시스템 역할이 추가되었습니다. LVM 캐시는 LV 데이터의 하위 집합을 임시로 더 빠르고 더 빠른 장치(예: SSD)에 저장하여 느린 논리 볼륨의 성능을 개선하는 데 사용할 수 있습니다.

(BZ#2016517)

Firewall 역할에 소스를 추가하거나 제거하는 기능

이번 업데이트를 통해 소스 매개변수를 사용하여 방화벽 설정 구성에서 소스를 추가하거나 제거할 수 있습니다.

(BZ#2021667)

Microsoft SQL Server 관리를 위한 새로운 Ansible 역할

새로운 **microsoft.sql.server** 역할은 **IT** 및 데이터베이스 관리자가 **Red Hat Enterprise Linux**에서 **SQL Server**의 설정, 구성 및 성능 튜닝과 관련된 프로세스를 자동화할 수 있도록 설계되었습니다.

(BZ#2013853)

Microsoft SQL System Role은 연결이 끊긴 또는 **Satellite** 서브스크립션에 대한 사용자 지정 리포지토리 지원

이전에는 **Satellite** 또는 **Capsule**을 가리켜야 하는 사용자 지정 서버 또는 **Satellite** 사용자로부터 패키지를 가져오는 데 필요한 연결이 끊긴 환경의 사용자는 **microsoft.sql.server** 역할을 지원하지 않았습니다. 이번 업데이트에서는 **mssql_rpm_key,mssql_server_repository** 및 **mssql_client_repository** 변수를 제공하여 패키지를 다운로드할 리포지토리를 사용자 지정하는 데 사용할 수 있습니다. **URL**을 제공하지 않으면 **mssql** 역할은 공식 **Microsoft** 서버를 사용하여 **RPM**을 다운로드합니다.

(BZ#2064648)

MSSQL 역할은 관리되는 구성 파일에서 **"Ansible_managed"** 주석을 일관되게 사용합니다.

MSSQL 역할은 **/var/opt/mssql/mssql.conf** 구성 파일을 생성합니다. 이번 업데이트를 통해 **MSSQL** 역할은 **Ansible** 표준 **ansible_managed** 변수를 사용하여 구성 파일에 **"Ansible** 관리**"** 주석을 삽입합니다. 주석은 **MSSQL** 역할이 파일을 덮어쓸 수 있으므로 구성 파일을 직접 편집하지 않아야 함을 나타냅니다. 결과적으로 구성 파일에는 구성 파일이 **Ansible**에서 관리됨을 알리는 선언이 포함되어 있습니다.

(BZ#2064690)

RHEL 시스템 역할에 대한 **Ansible Core** 지원

RHEL 9 GA 릴리스에서 **Ansible Core**는 **RHEL** 지원 자동화 사용 사례를 지원하기 위해 제한된 지원 범위로 제공됩니다. **Ansible Core**는 **RHEL**의 이전 버전에서 제공되는 **Ansible Engine**을 별도의 리포지토리에서 대체합니다. **Ansible Core**는 **RHEL**용 **AppStream** 리포지토리에서 사용할 수 있습니다. 지원되는 사용 사례에 대한 자세한 내용은 **RHEL 9 AppStream**에 포함된 **Ansible Core** 패키지에 대한 지원 범위를 참조하십시오.

Ansible Engine 지원이 필요하거나 **RHEL**이 아닌 사용 사례에 대한 지원이 필요한 경우 **Red Hat** 지원 케이스에서 케이스 를 생성합니다.

(JIRA:RHELPLAN-103540)

하나의 elasticsearch 출력 사전에서 여러 elasticsearch 호스트 구성 지원

이전에는 server_host 매개변수가 단일 호스트에 문자열 값을 가져오는 데 사용되었습니다. 이번 개선된 기능을 통해 기본 rsyslog omelasticsearch의 사양을 조정하므로 여러 호스트를 지원하는 데 문자열 목록도 걸립니다. 결과적으로 기본 rsyslog omelasticsearch의 사양에 따라 호스트에 조정됩니다. 결과적으로 사용자는 하나의 elasticsearch 출력 사전에 여러 elasticsearch 호스트를 구성할 수 있습니다.

(BZ#1986460)

RHEL 시스템 역할에서 VPN 관리 지원

이전에는 Linux에서 IPsec 터널링 및 VPN(Virtual Private Networking) 솔루션을 올바르게 구성하고 올바르게 구성하기 어려웠습니다. 이번 개선된 기능을 통해 VPN RHEL 시스템 역할을 사용하여 호스트 간 및 메시 연결에 대한 VPN 터널을 설정하고 구성할 수 있습니다. 결과적으로 RHEL System Roles 프로젝트 내에서 VPN 및 IPsec 터널링 구성을 위한 일관되고 안정적인 구성 인터페이스가 있습니다.

(BZ#2019341)

SSHD RHEL 시스템 역할은 이제 비독점 구성 스니펫 지원

이 기능을 사용하면 네임스페이스를 사용하여 이전 구성을 다시 작성하지 않고도 다양한 역할 및 플레이북을 통해 SSHD를 구성할 수 있습니다. 네임스페이스는 드롭인 디렉터리와 유사하며 SSHD에 대한 비독점 구성 스니펫을 정의합니다. 따라서 전체 구성 파일이 아닌 구성의 일부만 구성해야 하는 경우 다른 역할의 SSHD RHEL 시스템 역할을 사용할 수 있습니다.

(BZ#1978752)

Time Security (NTS) 옵션이 timesync RHEL System Role에 추가되었습니다.

클라이언트 서버에서 NTS 를 사용하도록 사용하도록 Timesync RHEL 시스템 역할에 NTS 옵션이 추가되었습니다. NTS는 NTP(Network Time Protocol)에 대해 지정된 새로운 보안 메커니즘입니다. NTS는 클라이언트별 구성 없이 NTP 클라이언트의 동기화를 보호할 수 있으며 다수의 클라이언트로 확장할 수 있습니다. NTS 옵션은 버전 4.0 이상의 chrony NTP 공급자에서만 지원됩니다.

(BZ#1978753)

HA Cluster RHEL 시스템 역할 지원

HA Cluster(고가용성 클러스터) 역할이 이제 완전히 지원됩니다. 다음과 같은 주요 구성을 사용할 수 있습니다.

- 메타 속성 및 리소스 작업을 포함하여 펜스 장치, 리소스, 리소스 그룹 및 리소스 복제 구성
- 리소스 위치 제약 조건, 리소스 공동 배치 제한 조건, 리소스 순서 제약 조건 및 리소스 티켓 제약 조건 구성
- 클러스터 속성 구성
- 클러스터 노드, 사용자 정의 클러스터 이름 및 노드 이름 구성
- 다중 링크 클러스터 구성
- 부팅 시 클러스터가 자동으로 시작되는지 여부 구성

역할을 실행하면 역할을 실행할 때 지정하지 않거나 지정되지 않은 구성이 제거됩니다.

HA 클러스터 시스템 역할은 현재 SBD를 지원하지 않습니다.

(BZ#2054401)

Elasticsearch에 대한 Rsyslog 사용자 이름 및 암호 인증 지원

이번 업데이트에서는 로깅 시스템 역할에 Elasticsearch 사용자 이름 및 암호 매개변수가 추가되었습니다. 결과적으로 사용자 이름 및 암호를 사용하여 Elasticsearch에 인증하도록 Rsyslog를 활성화할 수 있습니다.

(BZ#1990490)

Clevis 클라이언트 시스템 역할은 고정 IP 주소를 지원합니다.

이전 버전의 RHEL에서는 고정 IP 주소로 시스템을 다시 시작하고 NBDE(Network Bound Disk

Encryption) 클라이언트 시스템 역할은 시스템의 IP 주소를 변경합니다. 이 변경으로 인해 고정 IP 주소가 있는 시스템은 Clevis 클라이언트 시스템 역할에서 지원되며 IP 주소는 재부팅 후 변경되지 않습니다.

기본적으로 Clevis 역할은 부팅 시 DHCP를 사용하고 시스템이 부팅될 때 구성된 고정 IP로 전환합니다.

(BZ#2031555)

LVM에 raid_level 을 지정하는 지원이 추가되었습니다.

RHEL 9.0에서는 lvmraid 기능을 사용하여 LVM(Logical Volume Management) 볼륨을 RAID로 그룹화할 수 있습니다.

(BZ#2016518)

인증서 역할은 후크 스크립트에서 "Ansible_managed" 주석을 일관되게 사용합니다.

이번 개선된 기능을 통해 인증서 역할은 Ansible 표준 "ansible_managed" 변수를 사용하여 "Ansible 관리" 주석을 삽입하는 공급자를 지원하기 위해 사전 스크립트 및 post-scripts를 생성합니다.

- /etc/certmonger/pre-scripts/script_name.sh
- /etc/certmonger/post-scripts/script_name.sh

주석은 인증서 역할에서 파일을 덮어쓸 수 있으므로 스크립트 파일을 직접 편집하지 않아야 함을 나타냅니다. 결과적으로 구성 파일에는 구성 파일이 Ansible에서 관리됨을 알리는 선언이 포함되어 있습니다.

(BZ#2054364)

새 옵션 auto_gateway 는 기본 경로 동작을 제어합니다.

이전에는 DEFROUTE 매개변수는 구성 파일을 사용하여 구성할 수 없지만 모든 경로의 이름을 지정하여 수동으로 구성할 수 있었습니다. 이번 업데이트에서는 연결에 대한 ip 구성 섹션에 새 auto_gateway 옵션이 추가되어 기본 경로 동작을 제어할 수 있습니다. 다음과 같은 방법으로 auto_gateway 를 구성할 수 있습니다.

- **true** 로 설정하면 기본 게이트웨이 설정이 기본 경로에 적용됩니다.
- **false** 로 설정하면 기본 경로가 제거됩니다.
- 지정되지 않은 경우 네트워크 역할은 선택한 **network_provider** 의 기본 동작을 사용합니다.

(BZ#1978773)

네트워크 시스템 역할에 추가된 모든 본딩 옵션에 대한 지원

이번 업데이트에서는 네트워크 **RHEL** 시스템 역할에 대한 모든 본딩 옵션을 지원합니다. 결과적으로 결합된 인터페이스를 통해 네트워크 전송을 유연하게 제어할 수 있습니다. 결과적으로 해당 인터페이스에 여러 옵션을 지정하여 결합된 인터페이스를 통한 네트워크 전송을 제어할 수 있습니다.

(BZ#2054435)

NetworkManager는 PCI 주소를 사용하여 네트워크 카드 지정 지원

이전 버전에서는 연결 프로필을 설정하는 동안 **NetworkManager**는 이름 또는 **MAC** 주소를 사용하여 네트워크 카드만 지정할 수 있었습니다. 이 경우 장치 이름은 안정적이지 않으며 **MAC** 주소에 사용된 **MAC** 주소 레코드가 유지 관리해야 하는 인벤토리가 필요합니다. 이제 연결 프로필에서 **PCI** 주소를 기반으로 네트워크 카드를 지정할 수 있습니다.

(BZ#1999162)

이제 Network System 역할이 Ansible의 구성 파일을 직접 관리합니다.

이 향상된 기능을 통해 네트워크 역할은 **/etc/sysconfig/network-scripts** 에 **ifcfg** 파일을 생성합니다. 그런 다음 표준 **ansible_managed** 변수를 사용하여 **"Ansible managed"** 주석을 삽입합니다. 이 주석은 네트워크 역할이 덮어쓸 수 있으므로 **ifcfg** 파일을 직접 편집할 수 없음을 나타냅니다. **"Ansible** 관리**"** 주석을 추가하기 위해 **ifcfg** 파일을 처리하는 데 중요한 차이점은 **NetworkManager**에서 **nm** 패키지를 사용하는 동안 네트워크 역할에서 **initscripts** 패키지를 사용한다는 것입니다.

(BZ#2057657)

RHEL 시스템 역할에 대한 Ansible Core 지원

RHEL 9.0에서는 RHEL에서 자동화 사용 사례를 지원하기 위해 제한된 지원 범위로 Ansible Core가 제공됩니다. Ansible Core는 이전에 별도의 리포지토리에 제공된 Ansible Engine을 대체합니다. Ansible Core는 RHEL용 AppStream 리포지토리에서 사용할 수 있습니다. 지원되는 사용 사례에 대한 자세한 내용은 RHEL 9 및 RHEL 8.6 이상 AppStream 리포지토리에 포함된 Ansible Core 패키지에 대한 지원 범위를 참조하십시오. 사용자는 Ansible Engine에서 Ansible Core로 시스템을 수동으로 마이그레이션해야 합니다.

(BZ#2012298)

Cockpit 시스템 역할이 지원됨

이번 개선된 기능을 통해 시스템에 웹 콘솔을 설치하고 구성할 수 있습니다. 따라서 자동화된 방식으로 웹 콘솔을 관리할 수 있습니다.

(BZ#2021028)

터미널 세션 기록 시스템 역할은 관리 구성 파일에서 "Ansible 관리" 주석을 사용합니다.

터미널 세션 레코딩 역할은 2개의 구성 파일을 생성합니다.

- /etc/sssd/conf.d/sssd-session-recording.conf
- /etc/tlog/tlog-rec-session.conf

이번 업데이트를 통해 터미널 세션 레코딩 역할은 표준 Ansible 변수 ansible_managed 를 사용하여 구성 파일에 "Ansible 관리" 주석을 삽입합니다. 주석은 터미널 세션 레코딩 역할이 파일을 덮어쓸 수 있으므로 구성 파일을 직접 편집하지 않아야 함을 나타냅니다. 결과적으로 구성 파일에는 구성 파일이 Ansible에서 관리됨을 알리는 선언이 포함되어 있습니다.

(BZ#2054367)

VPN 역할은 관리 구성 파일에서 "Ansible_managed" 주석을 일관되게 사용합니다.

VPN 역할은 다음 구성 파일을 생성합니다.

- /etc/ipsec.d/mesh.conf

- /etc/ipsec.d/policies/clear
- /etc/ipsec.d/policies/private
- /etc/ipsec.d/policies/private-or-clear

이번 업데이트를 통해 VPN 역할은 Ansible 표준 ansible_managed 변수를 사용하여 구성 파일에 "Ansible 관리" 주석을 삽입합니다. 코멘트는 VPN 역할이 파일을 덮어쓸 수 있기 때문에 구성 파일을 직접 편집하지 않아야 함을 나타냅니다. 결과적으로 구성 파일에는 구성 파일이 Ansible에서 관리됨을 알리는 선언이 포함되어 있습니다.

(BZ#2054369)

Postfix 역할은 관리 구성 파일에서 "Ansible_managed" 주석을 일관되게 사용합니다.

Postfix 역할은 /etc/postfix/main.cf 구성 파일을 생성합니다. 이번 업데이트를 통해 Postfix 역할은 Ansible 표준 ansible_managed 변수를 사용하여 구성 파일에 "Ansible 관리" 주석을 삽입합니다. 주석은 Postfixrole에서 파일을 덮어쓸 수 있으므로 구성 파일을 직접 편집하지 않아야 함을 나타냅니다. 결과적으로 구성 파일에는 구성 파일이 Ansible에서 관리됨을 알리는 선언이 포함되어 있습니다.

(BZ#2057662)

방화벽 RHEL 시스템 역할이 RHEL 9에 추가되었습니다.

이번 개선된 기능을 통해 rhel-system-roles.firewall RHEL 시스템 역할이 rhel-system-roles 패키지에 추가되었습니다. 결과적으로 관리자는 관리형 노드의 방화벽 설정을 자동화할 수 있습니다.

(BZ#2021665)

SSH 클라이언트 RHEL 시스템 역할은 OpenSSH 8.7에서 새 구성 옵션을 지원합니다.

이번 개선된 기능을 통해 OpenSSH가 최신 버전으로 업데이트되어 새 호스트 구성을 위한 SSH 클라이언트 역할에서 사용 가능한 새로운 구성 옵션을 제공합니다.

(BZ#2029427)

4.20. 가상화

RHEL 웹 콘솔의 새로운 가상화 기능

이번 업데이트를 통해 RHEL 웹 콘솔에는 Virtual Machines 페이지에 새로운 기능이 포함되어 있습니다. 지금 할 수 있습니다:

- VM 이름 변경
- 클라우드 이미지 인증을 사용하여 VM 생성
- VM에 USB 및 PCI 장치 추가 및 제거
- 네트워크 인터페이스 모델 지정
- 호스트와 해당 VM 간에 파일 공유 및 해제

(JIRA:RHELPLAN-102009)

QEMU 사용 Clang

이제 QEMU 에뮬레이터가 Clang 컴파일러를 사용하여 빌드되었습니다. 이를 통해 RHEL 9 KVM 하이퍼바이저는 여러 고급 보안 및 디버깅 기능을 사용할 수 있으며 향후 기능 개발을 보다 효율적으로 수행할 수 있습니다.

(BZ#1940132)

가상 머신용 SafeStack

AMD64 및 Intel 64 하드웨어(x86_64)의 RHEL 9에서는 QEMU 에뮬레이터에서 향상된 컴파일러 기반 스택 보호 기능인 SafeStack을 사용할 수 있습니다. SafeStack은 스택 기반 버퍼 오버플로를 악용하여 스택의 반환 포인터를 변경하고 잘못된 프로그래밍(ROP) 공격을 생성할 수 있는 기능을 줄입니다. 결과적으로 RHEL 9에서 호스팅되는 가상 머신은 ROP 기반 취약점에 대해 훨씬 더 안전합니다.

(BZ#1939509)

virtiofs 전체 지원 Intel 64, AMD64 및 IBM Z

virtio 파일 시스템(virtiofs)은 이제 Intel 64, AMD64 및 IBM Z 아키텍처에서 완전하게 지원됩니다. virtiofs 를 사용하면 호스트 시스템과 가상 시스템 간에 파일을 효율적으로 공유할 수 있습니다.

(JIRA:RHELPLAN-64576)

KVM 게스트에서 지원되는 AMD EPYC7003 시리즈 프로세서

AMD EPYC7003 시리즈 프로세서(Blug Milan이라고도 함)에 대한 지원이 이제 KVM 하이퍼바이저 및 커널 코드와 libvirt API에 추가되었습니다. 이를 통해 KVM 가상 머신은 AMD EPYC7003 시리즈 프로세서를 사용할 수 있습니다.

(JIRA:RHELPLAN-65223)

QEMU-kvm 에서 추가 머신 유형 지원

RHEL 9에서는 qemu-kvm 가상화 시스템에서 다음 머신 유형을 지원합니다.

- Intel 64 및 AMD64(x86_64) 아키텍처에서 다음을 수행합니다.
  - PC-i440fx-rhel7.6.0 RHEL 7.6.0 PC (i440databind +databindX, [4]) (기본값)
  - PC RHEL 7.6.0 PC (i440databind +ReplicasX, [4]) (VOP-i440fx-rhel7.6.0)
  - q35 RHEL-8.5.0 PC (Q35 + ICH9, 2009) (PC-q35-rhel8.5.0의 별칭)
  - pc-q35-rhel8.5.0 RHEL-8.5.0 PC (Q35 + ICH9, 2009)
  - pc-q35-rhel8.4.0 RHEL-8.4.0 PC (Q35 + ICH9, 2009)
  - pc-q35-rhel8.3.0 RHEL-8.3.0 PC (Q35 + ICH9, 2009)

- pc-q35-rhel8.2.0 RHEL-8.2.0 PC (Q35 + ICH9, 2009)
- pc-q35-rhel8.1.0 RHEL-8.1.0 PC (Q35 + ICH9, 2009)
- pc-q35-rhel8.0.0 RHEL-8.0.0 PC (Q35 + ICH9, 2009)
- pc-q35-rhel7.6.0 RHEL-7.6.0 PC (Q35 + ICH9, 2009)

- IBM Z(s390x) 아키텍처의 경우:
  - s390-ccw-virtio-rhel7.6.0 VirtIO-ccw 기반 S390 머신 rhel7.6.0
  - s390-ccw-virtio-rhel8.2.0 VirtIO-ccw based S390 machine rhel8.2.0
  - s390-ccw-virtio-rhel8.4.0 VirtIO-ccw 기반 S390 시스템 rhel8.4.0
  - s390-ccw-virtio-rhel8.5.0 VirtIO-ccw 기반 S390 시스템 rhel8.5.0 (기본값)
  - s390-ccw-virtio VirtIO-ccw 기반 S390 머신 rhel8.5.0 (스390-ccw-virtio-rhel8.5.0)

추가로 RHEL 7.5.0 또는 이전 버전을 기반으로 하는 모든 머신 유형이 지원되지 않습니다.

(JIRA:RHELPLAN-75866)

IBM Z의 가상화 CLI에서 중재 장치 지원

virt-install 또는 virt-xml 을 사용하면 이제 vfio-ap 및 vfio-ccw와 같은 중재 장치를 VM(가상 머신)에 연결할 수 있습니다. 예를 들어 IBM Z 호스트에서 DASD 스토리지 장치 및 암호화 공동 프로세서를 보다 유연하게 관리할 수 있습니다. 또한 virt-install 을 사용하여 기존 DASD 중재 장치를 기본 디스크로 사용하는 VM을 만들 수 있습니다. 이에 대한 지침은 RHEL 9에서 가상화 구성 및 관리 가이드를 참조하십시오.

(BZ#1995131)

모듈식 libvirt 데몬

RHEL 9에서 libvirt 라이브러리는 호스트의 개별 가상화 드라이버 세트를 처리하는 모듈식 데몬을 사용합니다. 예를 들어 virtqemud 데몬은 QEMU 드라이버를 처리합니다. 따라서 리소스 로드 최적화 및 모니터링과 같은 가상화 드라이버와 관련된 다양한 작업을 미세 조정할 수 있습니다.

또한 모놀리식 libvirt 데몬인 libvirtd 도 더 이상 사용되지 않습니다. 그러나 RHEL 8에서 RHEL 9로 업그레이드하는 경우 호스트는 여전히 libvirtd 를 사용하므로 RHEL 9에서 계속 사용할 수 있습니다. 하지만 Red Hat은 대신 모듈식 libvirt 데몬으로 전환하는 것을 권장합니다.

(JIRA:RHELPLAN-113994)

Windows 11 및 Windows Server 2022 게스트 지원

RHEL 9는 KVM 가상 머신의 게스트 운영 체제로 Windows 11 및 Windows Server 2022 사용을 지원합니다.

(BZ#2036856, BZ#2004161)

ksmtuned 는 qemu-kvm과 별도로 배포되었습니다.

KVM 하이퍼바이저의 공간을 줄이기 위해 ksmtuned 유틸리티는 더 이상 qemu-kvm 의 종속성이 아닙니다. 결과적으로 커널 동일 페이지 병합(KSM)을 구성해야 하는 경우 ksmtuned 패키지를 수동으로 설치해야 합니다.

(BZ#2069501, BZ#1971678, BZ#1972158)

새로운 기능: vTPM

가상 신뢰할 수 있는 플랫폼 모듈 (vTPM)은 RHEL 9에서 완전히 지원됩니다. vTPM을 사용하면 RHEL 9 KVM 하이퍼바이저에서 실행 중인 VM(가상 머신)에 TPM 가상 암호화 프로세서를 추가할 수 있습니다. 따라서 VM을 사용하여 암호화 키를 생성, 저장 및 관리할 수 있습니다.

(JIRA:RHELPLAN-98617)

Intel Atom P59 시리즈 프로세서에 대한 가상화 지원

이번 업데이트를 통해 RHEL 9에서 가상화는 이전에 Snow Ridge로 알려진 Intel Atom P59 시리즈 프로세서에 대한 지원을 추가합니다. 결과적으로 RHEL 9에서 호스팅되는 가상 머신은 이제 Snowridge CPU 모델을 사용하고 프로세서에서 제공하는 새로운 기능을 사용할 수 있습니다.

(BZ#1874187)

4.21. 클라우드 환경의 RHEL

RHEL 9에서는 WALinuxAgent 2.3.0.2를 제공합니다.

RHEL 9는 Windows Azure Linux Agent (WALinuxAgent) 패키지 버전 2.3.0.2와 함께 배포됩니다. 주요 버그 수정 및 버전 2.2.49 개선 사항은 다음과 같습니다.

- RequiredFeatures 및 GoalStateAggregateStatus API에 대한 지원이 추가되었습니다.
- 확장 매니페스트의 대체 위치가 추가되었습니다.
- exceptions를 생성할 때 str.format()에 대한 누락된 호출이 추가되었습니다.

(BZ#1972101)

Azure의 RHEL은 이제 MANA를 지원합니다.

Microsoft Azure에서 실행되는 RHEL 9 가상 머신은 이제 MANA(Microsoft Azure Network Adapter)를 사용할 수 있습니다.

(BZ#1957818)

cloud-init 에서 VMware GuestInfo 데이터 소스 지원

이번 업데이트를 통해 cloud-init 유틸리티에서 VMware guestinfo 데이터에 대한 데이터 소스를 읽을 수 있습니다. 결과적으로 cloud-init 를 사용하여 VMware vSphere에서 RHEL 9 가상 머신을 설정하는 것이 더 효율적이고 안정적입니다.

(BZ#2040090)

RHEL 9 가상 머신은 Azure의 특정 ARM64 호스트에서 지원됩니다.

RHEL 9를 게스트 운영 체제로 사용하는 가상 머신은 이제 Ampere Altra ARM 기반 프로세서에서 실행되는 Microsoft Azure 하이퍼바이저에서 지원됩니다.

(BZ#1949613)

cloud-init 는 Microsoft Azure에서 사용자 데이터를 지원합니다.

cloud-init 유틸리티에 대해 --user-data 옵션이 도입되었습니다. 이 옵션을 사용하면 Azure에서 RHEL 9 가상 머신을 설정할 때 IMDS(Azure Instance Metadata Service)에서 스크립트 및 메타데이터를 전달할 수 있습니다.

(BZ#2042351)

4.22. 지원 관련 기능

SOS 보고서는 이제 예상 모드 실행을 제공합니다.

이 보고서 업데이트에서는 RHEL 서버에서 sos 보고서를 수집하는 데 필요한 디스크 공간을 대략적으로 수행할 수 있는 -- expectations-only 옵션을 추가합니다. sos report --neutron-only 명령을 실행합니다.

- sos report의 예행 실행 실행
- 모든 플러그인을 연속하고 디스크 크기를 추정합니다.

최종 디스크 공간 추정은 매우 대략적인 점에 유의하십시오. 따라서 예상 값을 두 배로 늘리는 것이 좋습니다.

(BZ#2011537)

4.23. 컨테이너

Podman에서 보안 짧은 이름 지원

[aliases] 테이블에 있는 registries.conf 파일에서 이미지의 단축 별칭을 구성할 수 있습니다. 짧은 이

름 모드는 다음과 같습니다.

- 강제: 이미지 가져오기 중에 일치하는 별칭을 찾을 수 없는 경우 **Podman**은 사용자에게 **unqualified-search** 레지스트리 중 하나를 선택하라는 메시지를 표시합니다. 선택한 이미지를 성공적으로 가져오면 **Podman**은 **$HOME/.cache/containers/short-name-aliases.conf** 파일(**rootless** 사용자) 및 **/var/cache/containers/short-name-aliases.conf (root user)**에 자동으로 새로운 단축 별칭을 기록합니다. 사용자에게 메시지를 표시할 수 없는 경우 (예: **stdin** 또는 **stdout**는 **TTY**가 아님) **Podman**은 실패합니다. 둘 다 동일한 별칭을 지정하는 경우 **short-name-aliases.conf** 파일이 **registries.conf** 파일보다 우선합니다.

- 허용: 강제 모드와 유사하지만, 사용자에게 메시지를 표시할 수 없는 경우 **Podman**은 실패하지 않습니다. 대신 **Podman**은 지정된 순서로 모든 정규화되지 않은 레지스트리에서 검색합니다. 별칭은 기록되지 않습니다.

예제:

```
unqualified-search-registries=["registry.fedoraproject.org", "quay.io"]

[aliases]

"fedora"="registry.fedoraproject.org/fedora"
```

(JIRA:RHELPLAN-74542)

container-tools 모듈의 변경 사항

container-tools 모듈에는 **Podman, Buildah, Skopeo, runc** 툴이 포함되어 있습니다. **RHEL 8**에서 **container-tools:rhel8** 스트림으로 표시되는 롤링 스트림 이름은 **RHEL 9**에서 **container-tools:latest** 라고 합니다. **RHEL 8**과 마찬가지로 안정적인 버전의 컨테이너 툴은 번호가 지정된 스트림(예: **3.0**)에서 사용할 수 있습니다.

컨테이너 툴 애플리케이션 스트림에 대한 자세한 내용은 Container Tools AppStream - Content Availability 를 참조하십시오.

(JIRA:RHELPLAN-73678)

containers-common 패키지를 사용할 수 있습니다.

container-common 패키지가 **container-tools:latest** 모듈에 추가되었습니다. **containers-common** 패키지에는 **Podman, Buildah, Skopeo**와 같은 컨테이너 툴 에코시스템에 대한 일반적인 구성 파일과 문

서가 포함되어 있습니다.

(JIRA:RHELPLAN-77549)

새 패키지로 컨테이너 이미지 업데이트

예를 들어 최신 패키지로 **registry.access.redhat.com/rhel9** 컨테이너 이미지를 업데이트하려면 다음 명령을 사용합니다.

```
# podman run -it registry.access.redhat.com/rhel9
# dnf update -y && rm -rf /var/cache/dnf
```

특정 ***<package>*** 를 설치하려면 다음을 입력합니다.

```
# dnf install <package>
```

자세한 내용은 실행 중인 **UBI** 컨테이너에 소프트웨어 추가를 참조하십시오.

RHEL 9의 경우 이미지에서 새 패키지를 업데이트하거나 설치하려면 권한이 부여된 호스트에서 실행해야 합니다. 개인용 **Red Hat Enterprise Linux** 개발자 서브스크립션을 사용하여 무료로 제공되는 저장소에 액세스할 수 있습니다.

자세한 내용은 대부분의 **Red Hat Enterprise Linux** 개별 개발자 서브스크립션을 참조하십시오**. FAQ**.

(JIRA:RHELPLAN-84168)

container-tools meta-package가 업데이트됨

Podman, Buildah, Skopeo 및 **runc** 툴을 포함하는 **container-tools RPM meta-package**를 사용할 수 있습니다. 이번 업데이트에서는 이전 버전에 비해 버그 수정 및 개선 사항 목록을 제공합니다.

(JIRA:RHELPLAN-118914)

podman-py 패키지를 사용할 수 있습니다.

podman-py 패키지가 **container-tools:3.0** 의 안정적인 모듈 스트림과 **container-tools:latest** 모듈에 추가되었습니다. **podman-py** 패키지는 **Podman**의 **RESTful API**를 사용하기 위한 바인딩 라이브러리입

니다.

(BZ#1975462)

컨트롤 그룹 버전 2 사용 가능

이전 버전의 제어 그룹인 cgroups 버전 1(cgroups v1)으로 인해 다양한 애플리케이션에서 성능 문제가 발생했습니다. 제어 그룹의 최신 릴리스인 cgroups 버전 2(cgroups v2)를 사용하면 시스템 관리자가 성능 문제를 유발하지 않고도 모든 애플리케이션의 리소스를 제한할 수 있습니다.

이 새 버전의 제어 그룹인 cgroups v2는 RHEL 8에서 활성화될 수 있으며 RHEL 9에서는 기본적으로 활성화되어 있습니다.

(JIRA:RHELPLAN-73697)

container-tools meta-package를 사용할 수 있습니다.

container-tools RPM 메타 패키지에는 Podman, Buildah, Skopeo, CRIU, Udica 및 필요한 모든 라이브러리가 RHEL 9에서 사용할 수 있습니다. RHEL 9에서는 stable 스트림을 사용할 수 없습니다. Podman, Buildah, Skopeo 등에 대한 안정적인 액세스 권한을 받으려면 RHEL EUS 서브스크립션을 사용합니다.

container-tools meta-package를 설치하려면 다음을 입력합니다.

```
# dnf install container-tools
```

(BZ#2000871)

커널에서 기본 오버레이 파일 시스템 지원 사용 가능

오버레이 파일 시스템 지원은 이제 커널 5.11에서 사용할 수 있습니다. 루트가 아닌 사용자는 rootless를 실행할 때(사용자로) 기본 오버레이 성능을 갖습니다. 따라서 이 향상된 기능은 바인딩 마운트 없이 overlayfs를 사용하려는 루트가 아닌 사용자에게 더 나은 성능을 제공합니다.

(JIRA:RHELPLAN-99892)

NFS 스토리지를 사용할 수 있음

파일 시스템에 **xattr** 지원이 있는 경우 **NFS** 파일 시스템을 컨테이너 및 이미지의 백엔드 스토리지로 사용할 수 있습니다.

(JIRA:RHELPLAN-74543)

container-tools meta-package가 업데이트됨

container-tools meta-package에는 **Podman, Buildah, Skopeo, CRIU, Udica** 및 모든 필수 라이브러리가 포함되어 있습니다. 이번 업데이트에서는 이전 버전에 비해 버그 수정 및 개선 사항 목록을 제공합니다.

주요 변경 사항은 다음과 같습니다.

- 네트워크 스택의 변경으로 인해 **Podman v3** 및 이전 버전에서 생성된 컨테이너는 **Podman v4.0**에서 사용할 수 없습니다.
- 기본 오버레이 파일 시스템을 **rootless** 사용자로 사용할 수 있습니다.
- **NFS** 스토리지가 컨테이너 내에서 지원됨
- 제어 그룹 버전 **2(cgroup v2)**는 기본적으로 활성화되어 있습니다.
- 모든 컨테이너가 제거되고 다시 생성되지 않는 한 **Podman v4**에서 **v3**로 다운그레이드하는 것은 지원되지 않습니다.

Podman의 주요 변경 사항에 대한 자세한 내용은 업스트림 릴리스 노트 를 참조하십시오.

(JIRA:RHELPLAN-99889)

crun 컨테이너 런타임이 기본값입니다.

crun 컨테이너 런타임은 이제 기본 런타임입니다. **crun** 컨테이너 런타임은 컨테이너가 **rootless** 사용자의 추가 그룹에 액세스할 수 있는 주석을 지원합니다. **setgid**가 설정된 디렉터리 또는 사용자에게 그룹

액세스 권한만 있는 디렉터리의 볼륨 마운트에 유용합니다. **crun** 및 **runc** 런타임 모두 **cgroup v2** 를 완전히 지원합니다.

(JIRA:RHELPLAN-99890)

이제 제어 그룹 버전 2 사용 가능

이전 버전의 제어 그룹인 **cgroup** 버전 **1(cgroup v1)**으로 인해 다양한 애플리케이션의 성능 문제가 발생했습니다. **cgroup** 버전 **2(cgroup v2)**의 최신 제어 그룹 릴리스를 통해 시스템 관리자는 성능 문제가 발생하지 않고 애플리케이션의 리소스를 제한할 수 있습니다.

RHEL 9에서는 **cgroup v2**가 기본적으로 활성화되어 있습니다.

(JIRA:RHELPLAN-75322)

Docker Hub에서 Universal Base Images를 사용할 수 있습니다.

이전에는 **Universal Base Images**를 **Red Hat** 컨테이너 카탈로그에서만 사용할 수 있었습니다. 이번 개선된 기능을 통해 **Docker Hub**에서 **Verified Publisher** 이미지로 도 **Universal Base Images**를 사용할 수 있습니다.

(JIRA:RHELPLAN-100032)

openssl 컨테이너 이미지를 사용할 수 있습니다.

openssl 이미지는 **OpenSSL crypto** 라이브러리의 다양한 기능을 사용하기 위한 **openssl** 명령줄 툴을 제공합니다. **OpenSSL** 라이브러리를 사용하여 개인 키를 생성하고, 인증서 서명 요청(**CSR**)을 생성하고, 인증서 정보를 표시할 수 있습니다.

openssl 컨테이너 이미지는 다음 리포지토리에서 사용할 수 있습니다.

- **registry.redhat.io/rhel9/openssl**
- **registry.access.redhat.com/ubi9/openssl**

(JIRA:RHELPLAN-100034)

Netavark 네트워크 스택 사용 가능

Netavark 스택은 컨테이너의 네트워크 구성 툴입니다. RHEL 9에서는 Netavark 스택이 완전히 지원되며 기본적으로 활성화됩니다.

이 네트워크 스택에는 다음과 같은 기능이 있습니다.

- 브리지 및 MACVLAN 인터페이스를 포함한 네트워크 인터페이스 생성, 관리 및 제거
- NAT(네트워크 주소 변환) 및 포트 매핑 규칙과 같은 방화벽 설정 구성
- IPv4 및 IPv6
- 여러 네트워크에서 컨테이너의 기능 개선

(JIRA:RHELPLAN-101141)

Podman은 YAML 파일을 사용하여 자동 빌드 및 자동 실행 Pod 지원

podman play kube 명령은 YAML 파일을 사용하여 Pod에서 여러 컨테이너가 있는 여러 Pod를 자동으로 빌드하고 실행합니다.

(JIRA:RHELPLAN-108830)

Podman은 이제 IdM에서 subUID 및 하위GID 범위를 소싱할 수 있습니다.

이제 subUID 및 subGID 범위를 IdM에서 관리할 수 있습니다. 모든 호스트에 동일한 /etc/subuid 및 /etc/subgid 파일을 배포하는 대신 단일 중앙 스토리지에 범위를 정의할 수 있습니다. /etc/nsswitch.conf 파일을 수정하고 sss s를 서비스 맵 라인에 추가해야 합니다. services: files sss.

자세한 내용은 IdM 문서의 하위 ID 범위 관리 섹션을 참조하십시오.

(JIRA:RHELPLAN-100020)

5장. 버그 수정

이 부분에서는 Red Hat Enterprise Linux 9.0에서 수정된 버그에 대해 설명합니다.

5.1. 설치 프로그램 및 이미지 생성

--leavebootorder 가 더 이상 부팅 순서를 변경하지 않음

이전에는 부트로더 **kickstart** 명령에 **--leavebootorder** 를 사용하는 것이 **UEFI** 시스템에서 제대로 작동하지 않고 부팅 순서를 변경했습니다. 이로 인해 설치 프로그램이 **UEFI** 부팅 메뉴에 설치된 시스템 목록의 맨 위에 **RHEL**을 추가했습니다.

이번 업데이트에서는 문제가 해결되어 **--leavebootorder** 를 사용하면 부트 로더에서 더 이상 부팅 순서를 변경하지 않습니다. **--leavebootorder** 는 이제 **UEFI** 시스템의 **RHEL**에서 지원됩니다.

(BZ#2025953)

Anaconda는 **%post** 스크립트를 실행하기 전에 정적 호스트 이름을 설정합니다.

이전에는 **Anaconda**에서 설치 프로그램 환경 호스트 이름을 **kickstart** 구성(네트워크 **--hostname**)의 값으로 설정할 때 일시적인 호스트 이름을 설정하는 데 사용되었습니다. **%post** 스크립트 실행 중에 수행되는 일부 동작(예: 네트워크 장치 활성화)으로 인해 호스트 이름이 역방향 **dns** 에서 얻은 값으로 재설정되었습니다.

이번 업데이트를 통해 **Anaconda**는 이제 **kickstart %post** 스크립트를 실행하는 동안 설치 프로그램 환경의 정적 호스트 이름을 안정적으로 설정합니다.

(BZ#2009403)

이제 사용자가 에지 설치 관리자의 **RHEL**에서 사용자 계정을 지정할 수 있습니다.

이전 버전에서는 **RPM** 패키지 추가와 같이 업그레이드에 대한 에지 커밋에 정의된 사용자 계정 없이 청사진에서 업데이트를 수행하면 업그레이드가 적용된 후 사용자가 시스템에서 잠길 수 없었습니다. 이로 인해 사용자가 기존 시스템을 업그레이드할 때 사용자 계정을 재정의했습니다. 이 문제는 사용자가 **OS tree** 커밋의 일부로 사용자를 설정하지 않고 설치 시 **RHEL**에서 **RHEL**에서 사용자 계정을 지정할 수 있도록 수정되었습니다.

(BZ#2060575)

부팅 메뉴에서 기본 그래픽 모드가 제거되었습니다.

이전에는 기본 그래픽 모드가 지원되지 않는 그래픽 카드가 있는 하드웨어에 RHEL을 설치하거나 그래픽 인터페이스를 시작하지 못하도록 그래픽 드라이버의 문제를 해결하는 데 사용되었습니다. 이번 업데이트를 통해 기본 그래픽 모드에서 설치할 옵션이 설치 프로그램 부팅 메뉴에서 제거됩니다. 지원되지 않는 하드웨어에서 그래픽 설치 또는 드라이버 버그를 해결하기 위해 VNC 설치 옵션을 사용합니다.

VNC를 사용한 설치에 대한 자세한 내용은 VNC를 사용하여 원격 RHEL 설치 수행 섹션을 참조하십시오.

(BZ#1961092)

5.2. 서브스크립션 관리

Hyper-V 호스트에서 virt-who 가 올바르게 작동합니다.

이전에는 virt-who 를 사용하여 Hyper-V 하이퍼바이저에서 RHEL 9 가상 머신(VM)을 설정할 때 virt-who 가 하이퍼바이저와 올바르게 통신하지 않았으며 설정이 실패했습니다. 이는 openssl 패키지에서 더 이상 사용되지 않는 암호화 방법 때문입니다.

이번 업데이트를 통해 Hyper-V용 virt-who 인증 모드가 수정되었으며 virt-who 를 사용하여 Hyper-V에서 RHEL 9 VM을 설정하는 작업이 올바르게 작동합니다. 이 경우 하이퍼바이저는 기본 인증 모드를 사용해야 합니다. 이 모드를 활성화하려면 다음 명령을 사용하십시오.

```
winrm set winrm/config/service/auth '@{Basic="true"}'
winrm set winrm/config/service '@{AllowUnencrypted="true"}'
```

(BZ#2008215)

5.3. 소프트웨어 관리

모듈식 리포지토리에서 createrepo_c --update 를 실행하면 이제 모듈식 메타데이터가 유지됩니다.

이전 버전에서는 모듈식 메타데이터의 원래 소스가 없는 기존 모듈식 리포지토리에서 createrepo_c --update 명령을 실행할 때 기본 정책은 이 리포지토리에서 모듈식 메타데이터를 포함하여 모든 추가 메타데이터를 제거하는 것이었습니다. 메타데이터를 보존하려면 추가 --keep-all-metadata 옵션과 함께 createrepo_c --update 명령을 실행해야 했습니다.

이번 업데이트를 통해 추가 옵션 없이 createrepo_c --update 를 실행하여 모듈식 리포지터리의 모듈식 메타데이터를 유지할 수 있습니다.

추가 메타데이터를 제거하려면 new --discard-additional-metadata 옵션을 사용합니다.

(BZ#2055032)

5.4. 쉘 및 명령행 툴

RHEL 9에서 libservicelog 1.1.19제공

RHEL 9는 libservicelog 버전 1.1.19와 함께 배포됩니다. 주요 버그 수정에는 다음이 포함됩니다.

- 수정된 출력 정렬 문제.
- servicelog_open() 실패에서 고정 segfault.

(BZ#1869568)

5.5. 보안

FIPS 모드에서 libgcrypt 에서 활성화 된 하드웨어 최적화

이전에는 연방 정보 처리 표준(FIPSRuntimeConfig)에서 하드웨어 최적화를 사용할 수 없었습니다. 따라서 이전 버전의 RHEL에서는 FIPS 모드인 경우 libgcrypt 패키지에서 작업이 비활성화되었습니다. RHEL 9는 FIPS 모드에서 하드웨어 최적화를 가능하게 하고 결과적으로 모든 암호화 작업이 더 빨리 수행됩니다.

(BZ#1990059)

이제 crypto-policies 에서 ChaCha20 암호화 사용을 비활성화할 수 있습니다.

이전에는 crypto-policies 패키지에서 잘못된 키워드를 사용하여 OpenSSL에서 ChaCha20 암호를 비활성화했습니다. 결과적으로 OpenSSL에서 crypto-policies 를 통해 TLS 1.2 프로토콜의 ChaCha20 을 비활성화할 수 없었습니다. 이번 업데이트에서는 -CHACHA20-POLY1305 대신 -CHACHA20 키워드가 사용됩니다. 결과적으로 TLS 1.2 및 TLS 1.3에 대해 OpenSSL에서 ChaCha20 암호 사용을 비활성화하는 암호화 정책을 사용할 수 있습니다.

(BZ#2004207)

FIPS 모드에서 설치하는 경우 64비트 IBM Z 시스템은 더 이상 부팅할 수 없습니다.

이전에는 --no-bootcfg 옵션이 있는 fips-mode-setup 명령이 zipl 도구를 실행하지 않았습니다. fips-mode-setup 이 초기 RAM 디스크(initrd)를 다시 생성하므로 결과 시스템에서 부팅을 위해 zipl 내부 상태를 업데이트해야 하므로 FIPS 모드에 설치한 후 64비트 IBM Z 시스템을 부팅할 수 없는 상태가 됩니다. 이번 업데이트를 통해 이제 --no - bootcfg 로 호출되어 새로 설치된 시스템이 성공적으로 부팅되는 경우에도 64비트 IBM Z 시스템에서 zipl 을 실행합니다.

(BZ#2013195)

GNUTLS_NO_EXPLICIT_INIT 가 더 이상 암시적 라이브러리 초기화를 비활성화하지 않음

이전에는 GNUTLS_NO_EXPLICIT 환경 변수가 암시적 라이브러리 초기화를 비활성화했습니다. RHEL 9에서 GNUTLS_NO_IMPLICIT_INIT 변수는 대신 암시적 라이브러리 초기화를 비활성화합니다.

(BZ#1999639)

이제 OpenSSL 기반 애플리케이션이 터키어 로케일에서 올바르게 작동합니다.

OpenSSL 라이브러리에서는 대소문자를 구분하지 않는 문자열 비교 기능을 사용하므로 OpenSSL 기반 애플리케이션이 Turkish 로케일에서 올바르게 작동하지 않았으며 생략된 검사로 인해 이 로케일을 사용하는 애플리케이션이 충돌했습니다. 이번 업데이트에서는 대소문자를 구분하지 않는 문자열 비교를 위해 이식 가능한 운영 체제 인터페이스(POSIX) 로케일을 사용하는 패치를 제공합니다. 결과적으로 curl과 같은 OpenSSL 기반 애플리케이션은 터키어 로케일을 통해 올바르게 작동합니다.

(BZ#2071631)

SELinux 권한으로 인해 kdump 가 더 이상 충돌하지 않음

kdump 충돌 복구 서비스를 사용하려면 추가 SELinux 권한이 올바르게 시작되어야 합니다. 따라서 이전 버전에서 SELinux는 kdump 가 작동하지 않고 kdump 가 작동하지 않음을 보고했으며 AVC(Access Vector Cache) 거부를 감사했습니다. 이 버전에서는 필요한 권한이 selinux-policy 에 추가되어 kdump 가 올바르게 작동하고 AVC 거부를 감사하지 않습니다.

(BZ#1932752)

usbguard-selinux 패키지는 더 이상 usbguard에 의존하지 않습니다.

이전에는 **usbguard-selinux** 패키지가 **usbguard** 패키지에 종속되었습니다. 이는 이러한 패키지의 다른 종속 항목과 함께 **usbguard** 를 설치할 때 파일 충돌이 발생했습니다. 이로 인해 특정 시스템에서 **usbguard** 를 설치할 수 없었습니다. 이 버전을 사용하면 **usbguard-selinux** 가 더 이상 **usbguard** 에 의존하지 않으므로 **dnf** 가 **usbguard** 를 올바르게 설치할 수 있습니다.

(BZ#1986785)

DNF 설치 및 **dnf** 업데이트 는 이제 **SELinux**에서 **fapolicyd** 와 함께 작동합니다.

fapolicyd에 대한 **SELinux** 규칙이 포함된 **fapolicyd-selinux** 패키지에는 모든 파일 및 디렉터리를 감시할 수 있는 권한이 포함되어 있지 않습니다. 그 결과 **fapolicyd-dnf-plugin** 이 제대로 작동하지 않아 **dnf install** 및 **dnf update** 명령이 시스템에서 무기한 응답을 중지하도록 했습니다. 이 버전에서 모든 파일 유형을 조사할 권한이 **fapolicyd-selinux** 에 추가되었습니다. 결과적으로 **fapolicyd-dnf-plugin** 이 올바르게 작동하고 **dnf install** 및 **dnf** 업데이트가 작동합니다.

(BZ#1932225)

이제 루트가 아닌 사용자에게 앰비언트 기능이 올바르게 적용됩니다.

안전 조치로 **UID(User Identifier)**를 **root**에서 루트가 아닌 사용자로 변경하면 허용된 유효 기능 및 앰비언트 세트의 기능을 알립니다.

그러나 기능이 허용되고 상속 가능한 세트가 앰비언트 세트에 있어야 하므로 **pam_cap.so** 모듈은 앰비언트 기능을 설정할 수 없습니다. 또한, 허용된 세트는 **UID**를 변경한 후(예: **setuid** 유틸리티 사용) **nullified**이므로 앰비언트 기능을 설정할 수 없습니다.

이 문제를 해결하기 위해 **pam_cap.so** 모듈에서 **keepcaps** 옵션을 지원하므로 루트에서 루트가 아닌 사용자로 **UID**를 변경한 후 프로세스에서 허용된 기능을 유지할 수 있습니다. 이제 **pam_cap.so** 모듈에서는 **pam_cap.so**로 콜백에서 **pam_end()** 에 대한 콜백 내에서 앰비언트 기능을 다시 적용하는 **defer** 옵션을 지원합니다. 이 콜백은 **UID**를 변경한 후 다른 애플리케이션에서 사용할 수 있습니다.

따라서 **su** 및 **login** 유틸리티가 업데이트되고 **PAM**을 준수하는 경우, **keepcaps** 와 함께 **pam_cap.so** 를 사용하고 루트가 아닌 사용자에 대해 앰비언트 기능을 설정하는 옵션 지연 을 사용할 수 있습니다.

(BZ#2037215)

usbguard-notifier 는 더 이상 저널에 너무 많은 오류 메시지를 기록하지 않습니다.

이전에는 **usbguard-daemon IPC** 인터페이스에 연결하기 위한 **usbguard-notifier** 서비스에 프로세스

간 통신(IPC) 권한이 없었습니다. 그 결과 usbguard-notifier 가 인터페이스에 연결하지 못하고 해당 오류 메시지를 저널에 작성했습니다. usbguard-notifier 는 --wait 옵션으로 시작했기 때문에 usbguard-notifier 가 연결 실패 후 각각 IPC 인터페이스에 연결을 시도했기 때문에 기본적으로 로그에 이러한 메시지의 과도한 양이 곧 포함되어 있었습니다.

이번 업데이트를 통해 usbguard-notifier 는 기본적으로 --wait 로 시작하지 않습니다. 서비스는 1초 간격으로 3번 데몬에 연결을 시도합니다. 결과적으로 로그에 최대 세 개의 오류 메시지가 포함됩니다.

(BZ#2009226)

5.6. 네트워킹

802.1x 이더넷 연결 프로필이 올바르게 연결되었습니다.

이전에는 많은 유선 및 802.1x 이더넷 연결 프로필을 연결할 수 없었습니다. 이 버그가 해결되었습니다. 이제 모든 프로필이 올바르게 연결됩니다. 레거시 암호화 알고리즘을 사용하는 프로필은 여전히 작동하지만 OpenSSL 레거시 공급자를 수동으로 활성화해야 합니다. 예를 들어, DES with MS-CHAPv2 및 RC4를 TKIP와 함께 사용할 때 필요합니다.

(BZ#1975718)

afterburn은 더 이상 /etc/hostname에 긴 호스트 이름을 설정하지 않습니다.

RHEL 호스트 이름의 최대 길이는 64자입니다. 그러나 특정 클라우드 공급자는 FQDN(정규화된 도메인 이름)을 호스트 이름으로 사용하며 최대 255자일 수 있습니다. 이전 버전에서는 afterburn-hostname 서비스가 지나치게 긴 호스트 이름을 /etc/hostname 파일에 직접 작성했습니다. systemd 서비스는 호스트 이름을 64자로 잘렸고 NetworkManager는 잘린 값에서 잘못된 DNS 검색 도메인을 파생했습니다. 이번 수정으로 afterburn-hostname 이 첫 번째 점 또는 64자 중 먼저 적용되는 호스트 이름을 잘라냅니다. 그 결과 NetworkManager는 더 이상 /etc/resolv.conf 에 잘못된 DNS 검색 도메인을 설정하지 않습니다.

(BZ#2008521)

5.7. 커널

modprobe 가 out-of-tree 커널 모듈을 예상대로 로드

/etc/depmod.d/dist.conf 구성 파일은 depmod 유틸리티에 대한 검색 순서를 제공합니다. 검색 순서에 따라 depmod 는 modules.dep.bin 파일을 생성합니다. 이 파일은 모듈 종속성을 나열합니다. modprobe 유틸리티는 커널 모듈을 로드 및 언로드하고 모듈 종속성을 동시에 확인하는 데 사용합니다. 이전에는 /etc/depmod.d/dist.conf 가 누락되었습니다. 그 결과 modprobe 가 일부 트리 외부 커널 모듈

을 로드할 수 없었습니다. 이번 업데이트에는 검색 순서를 수정하는 /etc/depmod.d/dist.conf 설정 파일이 포함되어 있습니다. 그 결과, modprobe 가 예상대로 out-of-tree 커널 모듈을 로드합니다.

(BZ#1985100)

시스템이 동적 LPAR 작업을 성공적으로 실행할 수 있습니다.

이전에는 다음 조건 중 하나가 충족된 경우 HMC(Hardware Management Console)에서 동적 논리 파티션(DLPAR) 작업을 실행할 수 없었습니다.

- 보안 부팅 기능이 활성화되어 무결성 모드에서 커널 잠금 메커니즘을 암시적으로 활성화합니다.
- 커널 잠금 메커니즘은 무결성 또는 기밀성 모드로 수동으로 활성화되었습니다.

RHEL 9에서는 커널 잠금 이 완전히 /dev/mem 문자 장치 파일을 통해 액세스할 수 있는 시스템 메모리에 대한 Run Time Abstraction Services(RTAS) 액세스를 차단했습니다. 여러 RTAS 호출에서 /dev/mem 에 대한 쓰기 액세스가 올바르게 작동합니다. 결과적으로 RTAS 호출이 올바르게 실행되지 않았으며 사용자에게 다음과 같은 오류 메시지가 표시됩니다.

> HSCL2957 Either there is currently no RMC connection between the management console and the partition <LPAR name> or the partition does not support dynamic partitioning operations. Verify the network setup on the management console and the partition and ensure that any firewall authentication between the management console and the partition has occurred. Run the management console diagrmc command to identify problems that might be causing no RMC connection.

이번 업데이트를 통해 잠금 에 매우 좁은 PowerPC 관련 예외를 제공하여 문제가 해결되었습니다. 예외를 통해 RTAS는 필요한 /dev/mem 영역에 액세스할 수 있습니다. 그 결과 설명된 시나리오에서 문제가 더 이상 나타나지 않습니다.

(BZ#2046472)

이제 ALSA-lib 가 UCM을 사용하는 오디오 장치를 올바르게 처리

alsa-lib 패키지의 버그로 인해 내부 UCM(사용 사례 관리자) 식별자 구문 분석이 잘못되었습니다. 결과적으로 UCM 구성을 사용한 일부 오디오 장치가 감지되지 않았거나 제대로 작동하지 않았습니다. 시스템이 pipewire 사운드 서비스를 사용하는 경우 문제가 더 자주 발생했습니다. RHEL 9 새 릴리스에서는 alsa-lib 라이브러리를 업데이트하여 문제가 해결되었습니다.

(BZ#2015863)

5.8. 파일 시스템 및 스토리지

보호 **uevents**가 더 이상 다중 경로 장치의 다시 로드 실패를 유발하지 않음

이전에는 읽기 전용 경로 장치를 다시 스캔할 때 커널은 두 개의 쓰기 보호 **uevents**를 전송했습니다. 하나는 읽기/쓰기 로 설정된 장치와 함께 다음과 같이 읽기 전용으로 설정된 장치입니다. 결과적으로 경로 장치에서 읽기/쓰기 **uevent**를 탐지하면 **multipathd** 가 다중 경로 장치를 다시 로드하려고 시도하여 다시 로드 오류 메시지가 발생했습니다. 이번 업데이트를 통해 이제 장치 읽기/쓰기를 다시 로드하기 전에 **multipathd** 가 모든 경로가 읽기/쓰기 로 설정되어 있는지 확인합니다. 결과적으로 **multipathd** 는 읽기 전용 장치를 다시 검사할 때마다 읽기/쓰기 를 다시 로드하지 않습니다.

(BZ#2017979)

device-mapper-multipath 버전이 **0.8.7**로 재기반

device-mapper-multipath 패키지가 여러 버그 수정 및 향상된 기능을 제공하는 **0.8.7** 버전으로 업그레이드되었습니다. 주요 변경 사항은 다음과 같습니다.

- 다중 경로 및 **kpartx** 명령에서 수정된 메모리 누수입니다.
- **multipathd.socket** 유닛 파일에서 반복된 트리거 오류를 수정했습니다.
- **DELL SC** 시리즈 어레이, **EMC Invista** 및 **Symmetrix** 어레이와 같은 더 많은 장치의 자동 구성이 개선되었습니다.

(BZ#2017592)

5.9. 고가용성 및 클러스터

Pacemaker 특성 관리자에서 원격 노드 속성을 올바르게 결정하여 루프에 영향을 미치지 않도록 합니다.

이전에는 특성 관리자가 이미 활성 원격 노드가 원격임을 알기 전에 노드의 **Pacemaker** 컨트롤러에서 **DC(Designated Controller)**를 선택할 수 있었습니다. 이 문제가 발생하면 노드의 스케줄러에서 원격 노드의 노드 속성이 표시되지 않습니다. 클러스터가 예기치 않은 오류를 사용한 경우 이로 인해 예기치 않은 루프가 발생할 수 있습니다. 수정으로 특성 관리자는 이제 시작 시 초기 속성 동기화를 포함하여 추가 이

벤트를 통해 원격 노드를 확인할 수 있습니다. 그 결과 어느 노드가 DC로 선택되었는지에 관계없이 예기치 않은 루프가 발생하지 않습니다.

(BZ#1975388)

5.10. 컴파일러 및 개발 도구

-wsequence-point 경고 동작이 수정되었습니다.

이전에는 GCC를 사용하여 C++ 프로그램을 컴파일할 때 -Wsequence-point warning 옵션은 매우 긴 표현식에 대해 경고하려고 시도했기 때문에 쿼드레틱 동작이 발생하여 훨씬 더 긴 컴파일 시간이 길어졌습니다. 이번 업데이트를 통해 -Wsequence-point 는 매우 큰 표현식에 대해 경고하려고 하지 않으며 결과적으로 컴파일 시간을 늘리지 않습니다.

(BZ#1481850)

5.11. IDM (IDENTITY MANAGEMENT)

OpenSSL 레거시 공급자를 사용한 MS-CHAP 인증

이전 버전에서는 MS-CHAP를 사용한 FreeRADIUS 인증 메커니즘은 MD4 해시 함수에 의존하여 RHEL 9에서 더 이상 사용되지 않기 때문에 실패했습니다. 이번 업데이트를 통해 OpenSSL 레거시 공급자를 활성화하는 경우 MS-CHAP 또는 MS-CHAPv2를 사용하여 FreeRADIUS 사용자를 인증할 수 있습니다.

기본 OpenSSL 공급자를 사용하는 경우 MS-CHAP 및 MS-CHAPv2 인증이 실패하고 수정 사항을 나타내는 다음 오류 메시지가 표시됩니다.

```
Couldn't init MD4 algorithm. Enable OpenSSL legacy provider.
```

(BZ#1978216)

sudo 명령을 실행해도 더 이상 KRB5CCNAME 환경 변수를 내보내지 않습니다.

이전 버전에서는 sudo 명령을 실행한 후 환경 변수 KRB5CCNAME 은 대상 사용자가 액세스할 수 없는 원래 사용자의 Kerberos 인증 정보 캐시를 가리켰습니다. 따라서 이 캐시에 액세스할 수 없으므로 Kerberos 관련 작업이 실패할 수 있습니다. 이번 업데이트를 통해 sudo 명령을 실행하면 더 이상 KRB5CCNAME 환경 변수가 설정되지 않으며 대상 사용자는 기본 Kerberos 인증 정보 캐시를 사용할 수 있습니다.

(BZ#1879869)

SSSD는 /etc/krb5.conf에서 Kerberos keytab 이름의 기본 설정을 올바르게 평가합니다.

이전에는 KnativeServing 5.keytab 파일에 비표준 위치를 정의한 경우 SSSD에서 이 위치를 사용하지 않고 기본 /etc/krb5.keytab 위치를 대신 사용했습니다. 결과적으로 시스템에 로그인하려고 하면 /etc/krb5.keytab 으로 로그인에 항목이 포함되지 않았습니다.

이번 업데이트를 통해 SSSD는 이제 /etc/krb5.conf 의 default_keytab_name 변수를 평가하고 이 변수에서 지정한 위치를 사용합니다. SSSD는 default_keytab_name 변수가 설정되지 않은 경우에만 기본 /etc/krb5.keytab 위치를 사용합니다.

(BZ#1737489)

PBKDF2 알고리즘으로 해시된 암호를 사용하여 FIPS 모드에서 Directory Server에 인증하면 예상대로 작동합니다.

Directory Server가 연방 정보 처리 표준(FIPS) 모드에서 실행되는 경우 PK11_ExtractKeyValue() 함수를 사용할 수 없습니다. 결과적으로 이 업데이트 이전에는 암호 기반 키 저하 기능 2(PBKDF2) 알고리즘으로 암호 해시가 지정된 사용자가 FIPS 모드가 활성화되면 서버에 인증할 수 없었습니다. 이번 업데이트를 통해 Directory Server는 이제 PK11_Decrypt() 함수를 사용하여 암호 해시 데이터를 가져옵니다. 그 결과 PBKDF2 알고리즘으로 해시된 암호를 사용한 인증이 예상대로 작동합니다.

(BZ#1779685)

5.12. RED HAT ENTERPRISE LINUX 시스템 역할

IPv6가 비활성화된 경우 네트워킹 시스템 역할은 더 이상 DNS 검색 도메인을 설정하지 못합니다.

이전에는 IPv6 프로토콜이 비활성화된 경우 libnm 라이브러리의 nm_connection_verify() 함수가 DNS 검색 도메인을 무시하지 않았습니다. 결과적으로 Networking RHEL 시스템 역할을 사용하고 ipv6_disabled: true 와 함께 dns_search 를 설정하면 시스템 역할이 다음 오류와 함께 실패했습니다.

```
nm-connection-error-quark: ipv6.dns-search: this property is not allowed for 'method=ignore' (7)
```

이번 업데이트를 통해 nm_connection_verify() 함수는 IPv6가 비활성화된 경우 DNS 검색 도메인을 무시합니다. 따라서 IPv6가 비활성화된 경우에도 dns_search 를 예상대로 사용할 수 있습니다.

(BZ#2004899)

redfish 역할 **README**는 더 이상 일반 역할 이름을 사용하지 않습니다.

이전에는 **/usr/share/ansible/roles/rhel-system-roles.journal/README.md** 에 제공된 예제에서 **rhel-system-roles.Replicas**를 사용하는 대신 역할 이름의 일반 버전을 사용했습니다. 결과적으로 사용자는 문서를 참조하고 **FQRN(Full Qualified Role Name)** 대신 일반 역할 이름을 잘못 사용합니다. 이번 업데이트에서는 이 문제가 해결되고 문서에는 **FQRN, rhel-system-roles. #189**가 포함된 예제가 포함되어 사용자가 플레이북을 올바르게 작성할 수 있습니다.

(BZ#1958964)

Postfix RHEL 시스템 역할 **README.md**에서 더 이상 "역할 변수" 섹션에서 더 이상 변수가 누락되지 않음

이전에는 **postfix_check,postfix_backup,postfix_backup_multiple** 과 같은 **Postfix RHEL** 시스템 역할 변수를 **"Role Variables"** 섹션에서 사용할 수 없었습니다. 그 결과 **users was not able to consult the advice role documentation.** 이번 업데이트에서는 **SASL README** 섹션에 역할 변수 문서가 추가되었습니다. 역할 변수는 **doc/usr/share/doc/ rhel-system-roles /#189/README.md** 설명서의 사용자에 대해 설명 및 사용할 수 있습니다.

(BZ#1978734)

동일한 출력을 실행할 때 역할 작업이 더 이상 변경되지 않음

이전에는 변경 사항이 없는 경우에도 여러 역할 작업에서 동일한 입력을 다시 실행할 때 **changesD** 로 보고되었습니다. 그 결과 이 역할은 멱등이 작동하지 않았습니다. 이 문제를 해결하려면 다음 작업을 수행합니다.

- 설정 변수를 적용하기 전에 변경되는지 확인합니다. 이 확인에는 **--check** 옵션을 사용할 수 있습니다.
- 마지막 수정: **$date** 헤더를 구성 파일에 추가하지 마십시오.

결과적으로 역할 작업은 멱등입니다.

(BZ#1978760)

logging_purge_confs 옵션은 불필요한 구성 파일을 올바르게 삭제합니다.

logging_purge_confs 옵션이 **true** 로 설정된 경우 불필요한 로깅 구성 파일을 삭제해야 합니다. 이전 버전에서는 **logging_purge_confs** 가 **true** 로 설정된 경우에도 불필요한 구성 파일이 구성 디렉터리에서 삭제되지 않았습니다. 이 문제는 해결되었으며 옵션은 다음과 같이 다시 정의되었으며 **logging_purge_confs** 가 **true** 로 설정된 경우 **Rsyslog**는 **rpm** 패키지에 속하지 않는 **rsyslog.d** 디렉토리에서 파일을 제거합니다. 여기에는 이전 로깅 역할 실행에서 생성된 구성 파일이 포함됩니다. **logging_purge_confs** 의 기본값은 **false** 입니다.

(BZ#2039106)

Grafana 관리자 암호가 변경된 경우에도 **Metrics** 역할을 사용하는 플레이북이 여러 실행에서 성공적으로 완료됩니다.

이전 버전에서는 **metrics_graph_service: yes** 부울과 함께 **Metrics** 역할을 실행한 후 **Grafana** 관리자 암호를 변경한 후 **Metrics** 역할의 후속 실행 시 실패했습니다. 이로 인해 **Metrics** 역할을 사용하는 플레이북이 실패했으며 영향을 받는 시스템은 성능 분석을 위해 부분적으로 설정되었습니다. 이제 **Metrics** 역할은 **Grafana** 배포 **API**를 사용할 수 있고 더 이상 사용자 이름 또는 암호에 대한 지식이 필요하지 않으므로 필요한 구성 작업을 수행할 수 있습니다. 결과적으로 **Metrics** 역할을 사용하는 플레이북이 관리자가 **Grafana** 관리자 암호를 변경하더라도 여러 실행에서 성공적으로 완료됩니다.

(BZ#2041632)

Metrics 역할의 구성은 이제 심볼릭 링크를 올바르게 따릅니다.

mssql pcp 패키지가 설치되면 **mssql.conf** 파일은 **/etc/pcp/mssql/** 에 있으며 심볼릭 링크 **/var/lib/pcp/pmdas/mssql/mssql.conf** 에 의해 대상입니다. 이전 버전에서는 **Metrics** 역할은 이를 따르고 **mssql.conf** 를 구성하는 대신 심볼릭 링크를 덮어 쓰기했습니다. 그 결과 **Metrics** 역할을 실행하면 심볼릭 링크가 일반 파일로 변경되어 구성에서 **/var/lib/pcp/pmdas/mssql/mssql.conf** 파일에만 영향을 미쳤습니다. 이로 인해 심볼릭 링크가 실패하고 기본 구성 파일 **/etc/pcp/mssql/mssql.conf** 가 구성의 영향을 받지 않았습니다. 이제 문제가 해결되었으며 심볼릭 링크를 따르는 **follow: yes** 옵션이 **Metrics** 역할에 추가되었습니다. 결과적으로 **Metrics** 역할은 심볼릭 링크를 유지하고 기본 구성 파일을 올바르게 구성합니다.

(BZ#2058777)

timesync 역할이 더 이상 요청된 서비스 **ptp4l**을 찾지 못했습니다.

이전에는 일부 **RHEL** 버전에서 **Ansible service_facts** 모듈이 서비스 사실을 잘못 보고했습니다. 그 결과 **timesync** 역할에서 **ptp4l** 서비스를 중지하려고 시도하는 오류가 보고되었습니다. 이번 수정을 통해 **Ansible service_facts** 모듈에서 **timesync** 서비스를 중지하기 위한 작업의 반환 값을 확인합니다. 반환된 값이 실패 하지만 오류 메시지가 **Could not find the requested service NAME:, module assumes success.** 결과적으로 **timesync** 역할은 **Could not find the requested service ptp4l** 과 같은 오류 없이 실행됩니다.

(BZ#2058645)

kernel_settings configobj 는 관리 호스트에서 사용할 수 있습니다.

이전에는 **kernel_settings** 역할이 관리 호스트에 **python3-configobj** 패키지를 설치하지 않았습니다. 결과적으로 역할에서 **configobj Python** 모듈을 찾을 수 없다는 오류가 반환되었습니다. 이번 수정으로 이 역할을 통해 관리 호스트에 **python3-configobj** 패키지가 있고 **kernel_settings** 역할이 예상대로 작동하는지 확인합니다.

(BZ#2058756)

SSSD에 의해 이제 터미널 세션 기록 역할 **tlog-rec-session** 이 올바르게 오버라이드됩니다.

이전에는 **RHEL** 시스템 역할을 기록하는 터미널 세션 기록은 **SSSD(System Security Services Daemon)** 파일 공급자를 사용하고 **authselect** 옵션을 **with-files-domain** 으로 활성화하여 **nsswitch.conf** 파일에 올바른 **passwd** 항목을 설정했습니다. **RHEL 9.0**에서는 기본적으로 파일 공급자를 암시적으로 활성화하지 않았기 때문에 **SSSD**에 의한 **tlog-rec-session** 쉘 오버레이가 작동하지 않았습니다. 이번 수정을 통해 이제 터미널 세션 기록 역할이 **nsswitch.conf** 를 업데이트하여 **SSSD**에 의해 **tlog-rec-session** 이 올바르게 오버라이드되도록 합니다.

(BZ#2071804)

SSHD 시스템 역할은 **FIPS** 모드에서 시스템을 관리할 수 있습니다.

이전에는 **SSHD** 시스템 역할은 호출 시 허용되는 **HostKey** 유형을 생성할 수 없었습니다. 그 결과 **SSHD** 시스템 역할은 연방 정보 처리 표준(**FIPS**) 모드에서 **RHEL 8** 및 이전 시스템을 관리할 수 없었습니다. 이번 업데이트를 통해 **SSHD** 시스템 역할은 **FIPS** 모드를 감지하고 기본 **HostKey** 목록을 올바르게 조정합니다. 결과적으로 시스템 역할은 기본 **HostKey** 구성을 사용하여 **FIPS** 모드에서 **RHEL** 시스템을 관리할 수 있습니다.

(BZ#2029634)

SSHD 시스템 역할은 올바른 템플릿 파일을 사용합니다.

이전에는 **SSHD** 시스템 역할에서 잘못된 템플릿 파일을 사용했습니다. 그 결과 생성된 **sshd_config** 파일에 **ansible_managed** 주석이 포함되어 있지 않았습니다. 이번 업데이트를 통해 시스템 역할은 올바른 템플릿 파일을 사용하고 **sshd_config** 에는 올바른 **ansible_managed** 주석이 포함되어 있습니다.

(BZ#2044408)

Kdump RHEL 시스템 역할은 재부팅할 수 있거나 재부팅이 필요함을 나타냅니다.

이전에는 **Kdump RHEL** 시스템 역할은 크래시 커널을 위해 예약된 메모리 없이 관리 노드를 무시했습니다. 결과적으로 시스템을 올바르게 구성하지 않은 경우에도 "성공" 상태의 역할이 완료되었습니다. 이 **RHEL 9** 업데이트에서는 문제가 해결되었습니다. 관리 노드에 크래시 커널용으로 예약된 메모리가 없는 경우 **Kdump RHEL** 시스템 역할이 실패하고 사용자가 관리 노드에서 **kdump** 서비스를 올바르게 구성하도록 **kdump_reboot_ok** 변수를 **true** 로 설정하도록 제안합니다.

(BZ#2029602)

네트워킹 시스템 역할의 **nm** 공급자에서 브릿지를 올바르게 관리

이전 버전에서는 **initscripts** 공급자를 사용한 경우 **Networking** 시스템 역할은 브릿지 인터페이스를 관리되지 않음으로 표시하도록 **NetworkManager**를 구성한 **ifcfg** 파일을 생성했습니다. 또한 **NetworkManager**는 **initscript** 작업을 감지하지 못했습니다. 예를 들어 **initscript** 공급자의 **down** 및 **absent** 작업은 **down** 및 **absent** 작업 후에 연결을 다시 로드하지 않는 경우 이 인터페이스의 관리되지 않는 상태에 대한 **NetworkManager**의 이해를 변경하지 않습니다. 이번 수정으로 **Networking** 시스템 역할은 **NM.Client.reload_connections_async()** 함수를 사용하여 **NetworkManager 1.18**을 사용하여 관리 호스트에서 **NetworkManager**를 다시 로드합니다. 결과적으로 **NetworkManager**는 공급자를 **initscript**에서 **nm** 으로 전환할 때 브리지 인터페이스를 관리합니다.

(BZ#2038957)

올바른 본딩 모드에 대해 **active-backup** 을 지원하도록 오타를 수정

이전 버전에서는 **active-backup** 본딩 모드를 지정하는 동안 **InfiniBand** 포트를 지원하는 오타**active_backup** 이 있었습니다. 이 오타로 인해 연결에서 **InfiniBand** 본딩 포트에 대해 올바른 본딩 모드를 지원하지 못했습니다. 이번 업데이트에서는 본딩 모드를 **active-backup** 으로 변경하여 오타가 수정되었습니다. 이제 연결에서 **InfiniBand** 본딩 포트를 성공적으로 지원합니다.

(BZ#2064391)

로깅 시스템 역할은 더 이상 작업을 여러 번 호출하지 않습니다.

이전에는 **Logging** 역할은 한 번만 호출해야 하는 작업을 여러 번 호출했습니다. 결과적으로 추가 작업 호출은 역할의 실행 속도가 느려졌습니다. 이번 수정을 통해 로깅 역할이 한 번만 작업을 호출하도록 변경되어 로깅 역할 성능이 향상되었습니다.

(BZ#2004303)

RHEL System Roles에서 생성된 파일에서 여러 줄 **ansible_managed** 주석을 처리

이전에는 RHEL 시스템 역할 중 일부는 # {{ ansible_managed }} 을 사용하여 파일 일부를 생성했습니다. 그 결과 고객이 여러 줄의 ansible_managed 설정이 있는 경우 파일이 잘못 생성되었습니다. 이번 수정을 통해 ansible_managed 문자열이 여러 줄 ansible_managed ansible_managed 값을 포함하여 항상 주석 처리되도록 파일을 생성할 때 {{ ansible_managed | comment }} 와 동등한 시스템 역할이 사용됩니다. 그 결과 생성된 파일에 올바른 다중 줄 ansible_managed 값이 있습니다.

(BZ#2006230)

이제 대상 변경 시 방화벽 시스템 역할이 방화벽을 즉시 다시 로드

이전에는 target 매개 변수가 변경될 때 방화벽이 다시 로드되지 않았습니다. 이번 수정을 통해 방화벽 역할은 대상이 변경될 때 방화벽을 다시 로드하므로 대상 변경 사항이 즉시 작동하고 후속 작업에 사용할 수 있습니다.

(BZ#2057164)

인증서 시스템 역할의 그룹 옵션이 더 이상 그룹에 액세스할 수 없는 인증서를 유지하지 않습니다.

이전에는 인증서에 대한 그룹을 설정할 때 그룹 읽기 권한을 허용하도록 모드가 설정되지 않았습니다. 결과적으로 그룹 멤버는 인증서 역할에서 발급한 인증서를 읽을 수 없었습니다. 이번 수정으로 그룹 설정에 이제 파일 모드에 그룹 읽기 권한이 포함되어 있는지 확인합니다. 결과적으로 그룹에 대해 Certificate 역할에서 발급한 인증서는 그룹 멤버가 액세스할 수 있습니다.

(BZ#2021025)

Logging 역할은 더 이상 immark 모듈 간격 값에 대한 따옴표를 표시하지 않습니다.

이전에는 immark 모듈이 제대로 구성되지 않았기 때문에 immark 모듈의 interval 필드 값이 제대로 인용되지 않았습니다. 이번 수정을 통해 간격 값이 올바르게 인용됩니다. 이제 immark 모듈이 예상대로 작동합니다.

(BZ#2021676)

/etc/tuned/kernel_settings/tuned.conf 파일에 적절한 ansible_managed 헤더가 있습니다.

이전에는 kernel_settings RHEL 시스템 역할에 /etc/tuned/kernel_settings/tuned.conf 파일에 ansible_managed 헤더에 대한 하드 코딩된 값이 있었습니다. 결과적으로 사용자 정의 ansible_managed 헤더를 제공할 수 없었습니다. 이번 업데이트에서는 kernel_settings 가 사용자의 ansible_managed 설정을 사용하여 /etc/tuned/kernel_settings/tuned.conf 의 헤더를 업데이트하도록

문제가 해결되었습니다. 그 결과 /etc/tuned/kernel_settings/tuned.conf 에 적절한 ansible_managed 헤더가 있습니다.

(BZ#2047506)

VPN 시스템 역할 필터 플러그인 vpn_ipaddr 은 이제 FQCN (Fully Qualified Collection Name)으로 변환됩니다.

이전에는 레거시 역할 형식에서 컬렉션 형식으로의 변환이 필터 플러그인 vpn_ipaddr 을 FQCN (Fully Qualified Collection Name) redhat.rhel_system_roles.vpn_ipaddr 로 변환하지 않았습니다. 결과적으로 VPN 역할은 짧은 이름으로 플러그인을 찾을 수 없어 오류를 보고했습니다. 이번 수정으로 필터가 컬렉션의 FQCN 형식으로 변환되도록 변환 스크립트가 변경되었습니다. 이제 VPN 역할은 오류가 발생하지 않고 실행됩니다.

(BZ#2050341)

kdump.service 에 대한 작업이 더 이상 실패하지 않음

이전에는 커널 크래시 크기를 구성하기 위한 Kdump 역할 코드가 RHEL9에 대해 업데이트되지 않았습니다. 여기에는 kdumpctl reset-crashkernel 을 사용해야 했습니다. 결과적으로 kdump.service 를 시작하고 오류를 발행할 수 없었습니다. 이번 업데이트를 통해 kdump.service 역할은 kdumpctl reset-crashkernel 을 사용하여 크래시 커널 크기를 구성합니다. 이제 kdump.service 역할이 kdump 서비스를 성공적으로 시작하고 커널 충돌 크기가 올바르게 구성됩니다.

(BZ#2050419)

5.13. 가상화

마운트된 가상 디스크를 핫 플러그 해제하면 더 이상 게스트 커널이 IBM Z에서 충돌하지 않습니다.

이전 버전에서는 IBM Z 하드웨어의 실행 중인 VM(가상 머신)에서 마운트된 디스크를 분리할 때 다음과 같은 조건에서 VM 커널이 충돌했습니다.

- 디스크는 대상 버스 유형 scsi 와 연결되어 게스트 내부에 마운트되었습니다.
- 디스크 장치를 핫 플러그 해제한 후 해당 SCSI 컨트롤러도 핫플러그되었습니다.

이번 업데이트를 통해 기본 코드가 수정되어 설명된 충돌이 더 이상 발생하지 않습니다.

(BZ#1997541)

5.14. 컨테이너

UBI 9-Beta 컨테이너는 RHEL 7 및 8 호스트에서 실행할 수 있습니다.

이전에는 UBI 9-Beta 컨테이너 이미지에 container -common 패키지에 잘못된 seccomp 프로필이 설정되어 있었습니다. 이로 인해 컨테이너가 특정 시스템 호출을 처리할 수 없어 오류가 발생했습니다. 이번 업데이트를 통해 문제가 해결되었습니다.

(BZ#2019901)

6장. 기술 프리뷰

이 부분에서는 **Red Hat Enterprise Linux 9**에서 제공되는 모든 기술 프리뷰 목록을 제공합니다.

기술 프리뷰 기능에 대한 **Red Hat** 지원 범위 내용은 기술 프리뷰 기능 지원 범위를 참조하십시오.

6.1. EDGE용 RHEL

FDO 프로세스를 기술 프리뷰로 사용 가능

자동 프로비저닝 및 에지 이미지 온보딩 **RHEL**을 위한 **FDO** 프로세스는 기술 프리뷰로 사용할 수 있습니다. 이를 통해 **Edge Simplified** 설치 프로그램 이미지용 **RHEL**을 빌드하고 **Edge** 이미지용 **RHEL**에 프로비저닝하고 **FDO(FIDO** 장치 온보딩) 프로세스를 사용하여 에지 장치를 자동으로 프로비저닝하고 온보딩하여 네트워크에 연결된 다른 장치 및 시스템과 데이터를 교환할 수 있습니다. 결과적으로 **FIDO** 장치 온보딩 프로토콜은 제조 단계에서 장치 초기화를 수행한 다음, 장치를 실제로 사용하기 위해 늦은 바인딩을 수행합니다.

(BZ#1989930)

6.2. 쉘 및 명령행 툴

64비트 **IBM Z** 아키텍처에서 기술 프리뷰로 다시 사용 가능

Basic Relax and Recover (ReaR) 기능은 이제 **64**비트 **IBM Z** 아키텍처에서 기술 프리뷰로 제공됩니다. **IBM Z**에서 **z/VM** 환경에서만 복구 이미지를 생성할 수 있습니다. **LPAR(Logical partitions)** 백업 및 복구는 테스트되지 않았습니다.

현재 사용 가능한 출력 방법은 초기 프로그램 로드**(IPL)**입니다. **IPL**은 **zIPL** 부트로더와 함께 사용할 수 있는 커널 및 초기 램디스크**(initrd)**를 생성합니다.

주의

현재 복구 프로세스는 시스템에 연결된 모든 **DASD**(직접 연결된 스토리지 장치)를 다시 포맷합니다. 시스템 스토리지 장치에 중요한 데이터가 있는 경우 시스템 복구를 시도하지 마십시오. 여기에는 복구 환경으로 부팅하는 데 사용된 **zIPL** 부트로더, **ReaR** 커널, **initrd**로 준비된 장치도 포함됩니다. 사본을 보관하십시오.

자세한 내용은 64비트 IBM Z 아키텍처에서 ReaR rescue 이미지 사용을 참조하십시오.

(BZ#2046653)

GIMP는 RHEL 9에서 기술 프리뷰로 사용 가능

GNU Image Manipulation Program(GIMP) 2.99.8은 이제 RHEL 9에서 기술 프리뷰로 사용할 수 있습니다. gimp 패키지 버전 2.99.8은 향상된 기능 집합이 포함된 시험판 버전이지만 제한된 기능 세트이며 안정성에 대한 보장은 없습니다. 공식 GIMP 3가 출시되는 즉시 이 시험판 버전의 업데이트로 RHEL 9에 도입됩니다.

RHEL 9에서는 gimp 를 RPM 패키지로 쉽게 설치할 수 있습니다.

(BZ#2047161)

6.3. 네트워킹

하이퍼레어 VPN은 기술 프리뷰로 사용할 수 있습니다.

Red Hat이 지원되지 않는 기술 프리뷰로 제공하는 Wireoctets는 Linux 커널에서 실행되는 고성능 VPN 솔루션입니다. 최신 암호화 기능을 사용하며 다른 VPN 솔루션보다 쉽게 구성할 수 있습니다. 또한 무선의 작은 코드-basis는 공격에 대한 면적을 줄이며, 따라서 보안이 향상됩니다.

자세한 내용은 Wireguard VPN 설정을 참조하십시오.

(BZ#1613522)

KTLS를 기술 프리뷰로 사용 가능

RHEL은 KVM(커널 전송 계층 보안)을 기술 프리뷰로 제공합니다. KTLS는 AES-GCM 암호화에 대해 커널에서 대칭 암호화 또는 암호 해독 알고리즘을 사용하여 TLS 레코드를 처리합니다. KTLS에는 이 기능을 제공하는 NIC(Network Interface Controller)로 TLS 레코드 암호화를 오프로드하는 인터페이스도 포함되어 있습니다.

(BZ#1570255)

systemd-resolved 서비스는 기술 프리뷰로 사용할 수 있습니다.

systemd-resolved 서비스는 로컬 애플리케이션에 대한 이름 확인을 제공합니다. 이 서비스는 캐싱 및 검증 DNS 스텁 확인자, Link-Local Multicast Name Resolution(LLMNR) 및 Multicast DNS 확인자 및 응답자를 구현합니다.

systemd 해결은 지원되지 않는 기술 프리뷰입니다.

(BZ#2020529)

6.4. 커널

커널용 Intel 데이터 스트리밍 가속기 드라이버는 기술 프리뷰로 사용할 수 있습니다.

커널에 대한 Intel 데이터 스트리밍 액셀러레이터 드라이버(IDXD)는 현재 기술 프리뷰로 사용할 수 있습니다. Intel CPU 통합 가속기이며 프로세스 주소 공간 ID(pasid) 제출 및 공유 가상 메모리(SVM)와 공유 작업 대기열을 포함합니다.

(BZ#2030412)

SGX를 기술 프리뷰로 이용 가능

SGX(Software guard Extensions)는 공개 및 수정으로부터 소프트웨어 코드 및 데이터를 보호하기 위한 Intel® 기술입니다. RHEL 커널은 SGX v1 및 v1.5 기능을 부분적으로 제공합니다. 버전 1을 사용하면 flexible Launch Control 메커니즘을 사용하는 플랫폼이 SGX 기술을 사용할 수 있습니다.

(BZ#1874182)

soft-iWARP 드라이버는 기술 프리뷰로 사용할 수 있습니다.

soft-iWARP(siw)는 Linux용 커널 드라이버인 소프트웨어, Internet Wide-area RDMA 프로토콜(iWARP)입니다. soft-iWARP는 TCP/IP 네트워크 스택에서 iWARP 프로토콜 제품군을 구현합니다. 이 프로토콜 제품군은 소프트웨어에서 완전히 구현되며 특정 RDMA(Remote Direct Memory Access) 하드웨어가 필요하지 않습니다. soft-iWARP를 사용하면 표준 이더넷 어댑터가 있는 시스템이 iWARP 어댑터 또는 이미 soft-iWARP가 설치된 다른 시스템에 연결할 수 있습니다.

(BZ#2023416)

6.5. 파일 시스템 및 스토리지

ext4 및 XFS에서 기술 프리뷰로 DAX를 사용할 수 있습니다.

RHEL 9에서는 DAX 파일 시스템을 기술 프리뷰로 사용할 수 있습니다. DAX는 애플리케이션이 영구 메모리를 주소 공간에 직접 매핑할 수 있음을 의미합니다.DAX provides means for an application to directly map persistent memory into its address space. DAX를 사용하려면 일반적으로 하나 이상의 비접근 다각형 메모리 모듈(NVDIMM)의 형태로 시스템에 사용 가능한 영구 메모리 형식이 있어야 하며 DAX 호환 파일 시스템을 NVDIMM(s)에서 생성해야 합니다. 또한 dax 마운트 옵션을 사용하여 파일 시스템을 마운트해야 합니다. 그런 다음 dax 마운트 파일 시스템에서 파일의 mmap 을 사용하면 스토리지를 애플리케이션의 주소 공간에 직접 매핑합니다.

(BZ#1995338)

Stratis는 기술 프리뷰로 사용 가능

Stratis는 로컬 스토리지 관리자입니다. 사용자에게 추가 기능을 제공하는 스토리지 풀 상단에서 관리되는 파일 시스템을 제공합니다.

- 스냅샷 및 씬 프로비저닝 관리
- 필요에 따라 파일 시스템 크기 자동 확장
- 파일 시스템 관리

Stratis 스토리지를 관리하려면 stratisd 백그라운드 서비스와 통신하는 stratis 유틸리티를 사용합니다.

Stratis는 기술 프리뷰로 제공됩니다.

자세한 내용은 Stratis 설명서를 참조하십시오. Stratis 파일 시스템 설정.

(BZ#2041558)

NVMe-oF Discovery Service 기능은 기술 프리뷰로 사용 가능

NVMexpress.org TP(Technical Proposals) 8013 및 8014에 정의된 NVMe-oF 검색 서비스 기능은 기술 프리뷰로 사용할 수 있습니다. 이러한 기능을 미리 보려면 nvme-cli 2.0 패키지를 사용하여 호스트를 TP-8013 또는 TP-8014를 구현하는 NVMe-oF 대상 장치에 연결합니다. TP-8013 및 TP-8014에 대한 자세

한 내용은 https://nvmexpress.org/developers/nvme-specification/ 웹 사이트의 NVM Express 2.0 Ratified TPs를 참조하십시오.

(BZ#2021672)

6.6. 컴파일러 및 개발 도구

JMC-core 및 owasp-java-encoder 는 기술 프리뷰로 사용 가능

RHEL 9는 jmc-core 및 owasp-java-encoder 패키지와 함께 기술 프리뷰 기능으로 배포됩니다.

JMC -core 는 JDK Flight Recording 파일을 구문 분석하고 작성하기 위한 라이브러리와 JDK(Java Discovery Protocol)를 통한 JVM(Java Virtual Machine) 검색 라이브러리를 포함하여 JDK(Java Development Kit) 미션 제어를 위한 핵심 API를 제공하는 라이브러리입니다.

owasp-java-encoder 패키지는 Java에 대한 고성능 저 헤드 컨텍스트 인코더 컬렉션을 제공합니다.

(BZ#1980981)

6.7. IDM (IDENTITY MANAGEMENT)

IdM에서 기술 프리뷰로 DNSSEC 사용 가능

통합된 DNS가 있는 IdM(Identity Management) 서버는 DNS 프로토콜의 보안을 향상시키는 DNSSEC(DNS Security Extensions) 세트를 구현합니다. IdM 서버에서 호스팅되는 DNS 영역은 DNSSEC를 사용하여 자동으로 서명할 수 있습니다. 암호화 키는 자동으로 생성되고 순환됩니다.

DNSSEC로 DNS 영역을 보호하려는 사용자는 다음 문서를 읽고 따르는 것이 좋습니다.

- DNSSEC 운영 사례, 버전 2
- 안전한 DNS(Domain Name System) 배포 가이드
- DNSSEC 키 롤링 고려 사항

통합된 DNS가 있는 IdM 서버는 DNSSEC를 사용하여 다른 DNS 서버에서 얻은 DNS 응답을 검증합니다. 이는 권장되는 이름 지정 방법에 따라 구성되지 않은 DNS 영역의 가용성에 영향을 줄 수 있습니다.

(BZ#2084180)

Identity Management JSON-RPC API를 기술 프리뷰로 사용 가능

IdM(Identity Management)에 API를 사용할 수 있습니다. API를 보기 위해 IdM은 API 브라우저도 기술 프리뷰로 제공합니다.

이전에는 여러 버전의 API 명령을 사용하도록 IdM API가 향상되었습니다. 이러한 개선 사항은 호환되지 않는 방식으로 명령의 동작을 변경할 수 있습니다. 이제 사용자는 IdM API가 변경되어도 기존 툴과 스크립트를 계속 사용할 수 있습니다. 이를 통해 다음을 수행할 수 있습니다.

- 관리자는 관리 클라이언트보다 서버에서 이전 또는 이후 버전의 IdM을 사용할 수 있습니다.
- 개발자는 IdM 버전이 서버에서 변경하더라도 특정 버전의 IdM 호출을 사용할 수 있습니다.

모든 경우에서 서버와의 통신은 한 쪽에서 사용하는 경우(예: 기능에 대한 새로운 옵션을 도입하는 최신 버전)에 관계없이 가능합니다.

API 사용에 대한 자세한 내용은 IdM 서버(TECHNOLOGY PREresolv)와 함께 ID 관리 API 사용을 참조하십시오.

(BZ#2084166)

RHCS에서 ACME 구현이 기술 프리뷰로 사용 가능

AMQP(Automated Certificate Management Environment)를 통한 서버 인증서는 RHCS(Red Hat Certificate System)에서 사용할 수 있습니다. ACME 응답자는 ACME v2 프로토콜(RFC 8555)을 지원합니다.

이전에는 사용자가 CA(인증 기관)의 CSR(인증서 서명 요청) 제출 루틴을 사용해야 했습니다. 루틴은 요청을 수동으로 검토하고 인증서를 발행하기 위해 CA(인증 기관) 에이전트가 필요한 경우가 있습니다.

이제 RHCS ACME 응답자는 CA 에이전트 없이 자동 서버 인증서 발급 및 라이프 사이클 관리를 위한 표준 메커니즘을 제공합니다. 이 기능을 사용하면 RHCS CA에서 기존 인증서 issuance 인프라와 통합하여 배포를 위해 공용 CA 및 개발을 위한 내부 CA를 대상으로 할 수 있습니다.

이 기술 프리뷰에는 ACME 서버 구현만 포함됩니다. ACME 클라이언트는 이 릴리스의 일부로 제공되지 않습니다. 또한 이 ACME 프리뷰는 issuance 데이터를 유지하거나 사용자 등록을 처리하지 않습니다.

향후 RHEL 업데이트는 잠재적으로 ACME 설치를 중단할 수 있습니다.

자세한 내용은 ACME의 IETF 정의 및 GitHub에 ACME Responder 설치를 참조하십시오.

(BZ#2084181)

6.8. 웹 콘솔

Stratis는 RHEL 웹 콘솔에서 기술 프리뷰로 사용 가능

이번 업데이트를 통해 Red Hat Enterprise Linux 웹 콘솔에서는 Stratis 스토리지를 기술 프리뷰로 관리할 수 있습니다.

Stratis에 대한 자세한 내용은 Stratis를 참조하십시오.
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/managing_file_systems/setting-up-stratis-file-systems_managing-file-systems#the-purpose-and-features-of-stratis_setting-up-stratis-file-systems

(JIRA:RHELPLAN-122345)

6.9. 가상화

KVM 가상 머신용 AMD SEV 및 SEV-ES

RHEL 9는 기술 프리뷰로 KVM 하이퍼바이저를 사용하는 AMD EPYC 호스트 머신의 SEV(Secure Encrypted Virtualization) 기능을 제공합니다. VM(가상 머신)에서 활성화된 경우 SEV는 VM의 메모리를 암호화하여 VM이 호스트의 액세스 권한을 방지합니다. 이로 인해 VM의 보안이 향상됩니다.

또한 SEV(SEV-ES)의 향상된 암호화 상태 버전도 기술 프리뷰로 제공됩니다. SEV-ES는 VM의 실행을

중지하면 모든 CPU 레지스터 콘텐츠를 암호화합니다. 이렇게 하면 호스트에서 VM의 CPU 레지스터를 수정하거나 해당 호스트의 정보를 읽을 수 없습니다.

SEV 및 SEV-ES는 2세대 AMD EPYC CPU (코드명 STA) 이상에서만 작동합니다. 또한 RHEL 9에는 SEV 및 SEV-ES 암호화가 포함되지만 SEV 및 SEV-ES 보안 인증에는 포함되지 않습니다.

(JIRA:RHELPLAN-65217)

이제 ARM 64에서 가상화를 사용할 수 있습니다.

이제 기술 프리뷰로 ARM 64 CPU를 사용하여 시스템에 KVM 가상 머신을 생성할 수 있습니다.

(JIRA:RHELPLAN-103993)

virtio-mem 은 이제 AMD64 및 Intel 64에서 사용 가능

RHEL 9는 기술 프리뷰로 AMD64 및 Intel 64 시스템에 virtio-mem 기능을 도입했습니다. virtio-mem 을 사용하면 VM(가상 머신)에서 호스트 메모리를 동적으로 추가하거나 제거할 수 있습니다.

virtio-mem 을 사용하려면 VM의 XML 구성에 virtio-mem 메모리 장치를 정의하고 virsh update-memory-device 명령을 사용하여 VM이 실행되는 동안 메모리 장치 크기 변경을 요청합니다. 이러한 메모리 장치에서 실행 중인 VM에 의해 노출되는 현재 메모리 크기를 보려면 VM의 XML 구성을 확인합니다.

(BZ#2014487)

7장. 사용되지 않는 기능

이 부분에서는 **Red Hat Enterprise Linux 9**에서 *더 이상 사용되지* 않는 기능에 대한 개요를 제공합니다.

사용되지 않는 기능은 이 제품의 향후 주요 릴리스에서 지원되지 않을 가능성이 높으며 새로운 배포에 구현하는 것은 권장되지 않습니다. 특정 주요 릴리스 내에서 더 이상 사용되지 않는 기능의 최신 목록은 최신 릴리스 노트를 참조하십시오.

더 이상 사용되지 않는 기능의 지원 상태는 **Red Hat Enterprise Linux 9**에서는 변경되지 않습니다. 지원 기간에 대한 자세한 내용은 **Red Hat Enterprise Linux** 라이프 사이클 및 **Red Hat Enterprise Linux Application Streams** 라이프 사이클을 참조하십시오.

사용되지 않는 하드웨어 구성 요소는 현재 또는 향후 주요 릴리스의 새로운 배포에 구현하는 것을 권장하지 않습니다. 하드웨어 드라이버 업데이트는 보안 및 중요 수정 사항으로만 제한됩니다. **Red Hat**은 최대한 빠른 시일 내에 이 하드웨어를 교체할 것을 권장합니다.

패키지가 더 이상 사용되지 않으며 향후 사용이 권장되지 않는 경우가 있습니다. 경우에 따라 패키지가 제품에서 삭제될 수 있습니다. 제품 설명서에 더 이상 사용되지 않는 기능과 유사 또는 동일하거나 보다 고급 기능을 제공하는 최근 패키지가 지정된 권장 사항이 기재됩니다.

RHEL 8에 존재하지만 **RHEL 9**에서 *제거된* 기능에 대한 자세한 내용은 **RHEL 9** 채택 시 고려 사항을 참조하십시오.

7.1. 설치 프로그램 및 이미지 생성

더 이상 사용되지 않는 Kickstart 명령

다음 **Kickstart** 명령은 더 이상 사용되지 않습니다.

- **timezone --ntpservers**
- **timezone --nontp**
- **logging --level**

- %packages --excludeWeakdeps
- %packages --instLangs
- %Anaconda
- pwpolicy

특정 옵션만 나열되는 경우 기본 명령 및 기타 옵션은 계속 사용할 수 있으며 더 이상 사용되지 않습니다. Kickstart 파일에서 더 이상 사용되지 않는 명령을 사용하면 로그에 경고가 출력됩니다. inst.ksstrict 부팅 옵션을 사용하여 더 이상 사용되지 않는 명령 경고를 오류로 전환할 수 있습니다.

(BZ#1899167)

7.2. 보안

SHA-1은 암호화 목적으로 사용되지 않습니다.

암호화 목적으로 SHA-1 메시지 다이제스트를 사용하는 것은 RHEL 9에서 더 이상 사용되지 않습니다. SHA-1에서 생성된 다이제스트는 해시 충돌을 찾는 데 따른 문서화된 여러 공격으로 인해 안전하지 않은 것으로 간주되지 않습니다. RHEL 핵심 암호화 구성 요소는 기본적으로 SHA-1을 사용하여 더 이상 서명을 생성하지 않습니다. RHEL 9의 애플리케이션은 보안 관련 사용 사례에서 SHA-1을 사용하지 않도록 업데이트되었습니다.

예외적으로 HMAC-SHA1 메시지 인증 코드와 UUID(Universal Unique Identifier) 값은 SHA-1을 사용하여 계속 생성할 수 있습니다. 이러한 사용 사례는 현재 보안 위험을 초래하지 않기 때문입니다. SHA-1은 Kerberos 및 WPA-2와 같은 중요한 상호 운용성 및 호환성 문제와 관련된 제한된 경우에도 사용할 수 있습니다. 자세한 내용은 RHEL 9 보안 강화 문서의 FIPS 140-3 섹션을 지원하지 않는 암호화를 사용하는 RHEL 애플리케이션 목록을 참조하십시오.

시나리오에 SHA-1을 사용하여 기존 또는 타사 암호화 서명을 확인해야 하는 경우 다음 명령을 입력하여 활성화할 수 있습니다.

```
# update-crypto-policies --set DEFAULT:SHA1
```

또는 시스템 전체 암호화 정책을 LEGACY 정책으로 전환할 수 있습니다. LEGACY 는 또한 안전하지 않은 다른 많은 알고리즘을 사용할 수 있습니다.

(JIRA:RHELPLAN-110763)

RHEL 9에서 SCP가 더 이상 사용되지 않음

보안 복사 프로토콜(SCP)은 알려진 보안 취약점이 있기 때문에 더 이상 사용되지 않습니다. SCP API는 RHEL 9 라이프사이클에서 계속 사용할 수 있지만 이를 사용하면 시스템 보안이 저하됩니다.

- scp 유틸리티에서 SCP는 기본적으로 SFTP(SSH 파일 전송 프로토콜)로 교체됩니다.
- OpenSSH 제품군은 RHEL 9에서 SCP를 사용하지 않습니다.
- libssh 라이브러리에서 SCP가 더 이상 사용되지 않습니다.

(JIRA:RHELPLAN-99136)

SASL의 digest-MD5는 더 이상 사용되지 않음

SASL(Simple Authentication Security Layer) 프레임워크의 다이제스트-MD5 인증 메커니즘은 더 이상 사용되지 않으며 향후 주요 릴리스에서 cyrus-sasl 패키지에서 제거될 수 있습니다.

(BZ#1995600)

OpenSSL은 MD2, MD4, MDC2, Whirlpool, RIPEMD160, Blowfish, CAST, DES, IDEA, RC4, RC4, RC5, SEED 및 PBKDF1을 더 이상 사용하지 않습니다.

OpenSSL 프로젝트는 안전하지 않거나 일반적으로 사용되거나 둘 다 사용되기 때문에 암호화 알고리즘 세트를 더 이상 사용하지 않습니다. 또한 Red Hat은 이러한 알고리즘 사용을 권장하지 않으며 RHEL 9는 암호화된 데이터를 마이그레이션하여 새로운 알고리즘을 사용하기 위해 제공합니다. 사용자는 시스템의 보안을 위해 이러한 알고리즘에 의존하지 않아야 합니다.

다음 알고리즘의 구현은 OpenSSL의 기존 공급자로 이동되었습니다. MD2, MD4, MDC2, Whirlpool, RIPEMD160, Blowfish, CAST,DES, IDEA, RC4, RC4, RC5, SEED, PBKDF1.

레거시 공급자를 로드하고 더 이상 사용되지 않는 알고리즘에 대한 지원을 활성화하는 방법에 대한 지침은 /etc/pki/tls/openssl.cnf 구성 파일을 참조하십시오.

(BZ#1975836)

/etc/system-fips 가 더 이상 사용되지 않음

/etc/system-fips 파일을 통한 FIPS 모드를 나타내는 지원은 제거되었으며 향후 RHEL 버전에서는 파일이 포함되지 않습니다. FIPS 모드에서 RHEL을 설치하려면 시스템 설치 중에 fips=1 매개변수를 커널 명령줄에 추가합니다. fips-mode-setup --check 명령을 사용하여 RHEL이 FIPS 모드에서 작동하는지 확인할 수 있습니다.

(JIRA:RHELPLAN-103232)

libcrypt.so.1 이 더 이상 사용되지 않음

libcrypt.so.1 라이브러리는 더 이상 사용되지 않으며 향후 RHEL 버전에서 제거될 수 있습니다.

(BZ#2034569)

fapolicyd.rules 가 더 이상 사용되지 않음

허용 및 거부 규칙이 포함된 파일의 /etc/fapolicyd/rules.d/ 디렉터리는 /etc/fapolicyd/fapolicyd.rules 파일을 대체합니다. 이제 fagenrules 스크립트는 이 디렉터리의 모든 구성 요소 규칙 파일을 /etc/fapolicyd/ compiled.rules 파일에 병합합니다. /etc/fapolicyd/fapolicyd.trust 의 규칙은 여전히 fapolicyd 프레임워크에서 처리되지만 이전 버전과의 호환성을 위해서만 계속 처리합니다.

(BZ#2054740)

7.3. 네트워킹

ipset 및 iptables-nft 가 더 이상 사용되지 않음

RHEL에서 ipset 및 iptables-nft 패키지가 더 이상 사용되지 않습니다. iptables-nft 패키지에는 iptables,ip6tables,ebtables 및 arptables 와 같은 다양한 도구가 포함되어 있습니다. 이러한 툴은 더 이상 새 기능을 받지 않으며 새 배포에 사용하는 것은 권장되지 않습니다. 대신 nftables 패키지에서 제공하는 nft 명령줄 툴을 사용하는 것을 선호합니다. 가능한 경우 기존 설정을 nft 로 마이그레이션해야 합니다.

iptables,ip6tables,ebtables,arptables,nft_compat 또는 ipset 모듈을 로드할 때 모듈은 /var/log/message 파일에 다음 경고를 기록합니다.

> Warning: <module_name> - this driver is not recommended for new deployments. It continues to be supported in this RHEL release, but it is likely to be removed in the next major release. Driver updates and fixes will be limited to critical issues. Please contact Red Hat Support for additional information.

nftables로 마이그레이션하는 방법에 대한 자세한 내용은 **iptables**에서 **nftables**로 마이그레이션, **iptables-translate(8)** 및 **ip6tables-translate(8)** 도움말 페이지를 참조하십시오.

(BZ#1945151)

RHEL 9에서 네트워크 팀이 더 이상 사용되지 않음

teamd 서비스와 **libteam** 라이브러리는 **Red Hat Enterprise Linux 9**에서 더 이상 사용되지 않으며 다음 주요 릴리스에서 제거될 예정입니다. 교체로 네트워크 팀 대신 본딩을 구성합니다.

Red Hat은 비슷한 기능을 갖춘 두 가지 기능, 본딩 및 팀을 유지 관리하기 위해 커널 기반 본딩에 중점을 두고 있습니다. 본딩 코드에는 높은 고객 채택이 있고 견고하며 활발한 커뮤니티 개발이 있습니다. 결과적으로 본딩 코드는 개선 사항 및 업데이트를 수신합니다.

팀을 본딩으로 마이그레이션하는 방법에 대한 자세한 내용은 네트워크 팀 구성 마이그레이션을 참조하십시오.

(BZ#1935544)

NetworkManager는 새 네트워크 구성을 **/etc/NetworkManager/system-connections/** 에 키 파일 형식으로 저장합니다.

이전에는 **NetworkManager**에서 새 네트워크 구성을 **/etc/sysconfig/network-scripts/** 에 **ifcfg** 형식으로 저장했습니다. **RHEL 9.0**부터 **RHEL**은 새 네트워크 구성을 **/etc/NetworkManager/system-connections/** 에 키 파일 형식으로 저장합니다. **/etc/sysconfig/network-scripts/** 에 구성이 저장되는 연결은 여전히 중단되지 않습니다. 기존 프로필의 수정 사항은 이전 파일을 계속 업데이트합니다.

(BZ#1894877)

firewalld 에서 **iptables** 백엔드가 더 이상 사용되지 않음

RHEL 9에서는 **iptables** 프레임워크가 더 이상 사용되지 않습니다. 결과적으로 **firewalld** 의 **iptables** 백엔드 및 직접 인터페이스 도 더 이상 사용되지 않습니다. 직접 인터페이스 대신 **firewalld** 에서 네이티브

기능을 사용하여 필요한 규칙을 구성할 수 있습니다.

(BZ#2089200)

7.4. 커널

RHEL 9에서 2016, 캡슐화가 더 이상 사용되지 않음

A TM(Asynchronous Transfer Mode) 캡슐화는 AAL-5(AAL-5)에 Layer-2(Point-to-Point Protocol, 이더넷) 또는 Layer-3(IP) 연결을 활성화합니다. Red Hat은 RHEL 7 이후의 ATM NIC 드라이버를 지원하지 않습니다. ATM 구현에 대한 지원은 RHEL 9에서 중단되고 있습니다. 이러한 프로토콜은 현재 ADSL 기술을 지원하고 제조업체가 단계적으로 제공하는 칩셋에서만 사용됩니다. 따라서 Red Hat Enterprise Linux 9에서는 ATM 캡슐화가 더 이상 사용되지 않습니다.

자세한 내용은 CHAP Over AAL5,ATM Adaptation Layer 5를 통한 Multiprotocol Encapsulation, and classical IP 및 ARP 를 참조하십시오.

(BZ#2058153)

7.5. 파일 시스템 및 스토리지

lvm2-activation-generator 및 생성된 서비스 RHEL 9.0에서 제거됨

lvm2-activation-generator 프로그램 및 생성된 서비스 lvm2-activation -early, lvm2-activation-net 은 RHEL 9.0에서 제거됩니다. 서비스를 활성화하는 데 사용되는 lvm.conf event_activation 설정이 더 이상 작동하지 않습니다. 볼륨 그룹을 자동 활성화하는 유일한 방법은 이벤트 기반 활성화입니다.

(BZ#2038183)

7.6. 동적 프로그래밍 언어, 웹 서버 및 데이터베이스 서버

libdb 가 더 이상 사용되지 않음

RHEL 8 및 RHEL 9는 현재 LGPLv2 라이센스에 따라 배포되는libdb(Lerkeley DB) 버전 5.3.28을 제공합니다. 업스트림 Berkeley DB 버전 6은 AGPLv3 라이센스에 따라 사용할 수 있으며, 이는 보다 제한적입니다.

libdb 패키지는 RHEL 9부터 더 이상 사용되지 않으며 향후 주요 RHEL 릴리스에서는 사용할 수 없을 수 있습니다.

또한 RHEL 9의 libdb 에서 암호화 알고리즘이 제거되고 RHEL 9에서는 여러 libdb 종속성이 제거되었습니다.

libdb 사용자는 다른 키-값 데이터베이스로 마이그레이션하는 것이 좋습니다. 자세한 내용은 RHEL에서 더 이상 사용되지 않는 Berkeley DB(libdb)에 대한 지식 베이스 문서 Available replacements for the deprecated Berkeley DB(libdb) 를 참조하십시오.

(BZ#1927780, BZ#1974657, JIRA:RHELPLAN-80695)

7.7. IDM (IDENTITY MANAGEMENT)

OpenDNSSec의 SHA-1 이 더 이상 사용되지 않음

OpenDNSSec은 SHA-1 알고리즘을 사용하여 디지털 서명 및 인증 레코드 내보내기를 지원합니다. SHA-1 알고리즘 사용은 더 이상 지원되지 않습니다. RHEL 9 릴리스에서는 OpenDNSSec의 SHA-1 이 더 이상 사용되지 않으며 향후 마이너 릴리스에서 제거될 수 있습니다. 또한 OpenDNSSec 지원은 Red Hat Identity Management와의 통합으로 제한됩니다. OpenDNSSec은 독립형으로 지원되지 않습니다.

(BZ#1979521)

SSSD 암시적 파일 공급자 도메인은 기본적으로 비활성화되어 있습니다.

/etc/shadow 및 /etc/ groups 에서 그룹 정보와 같은 로컬 파일에서 사용자 정보를 검색하는 SSSD 암시적 파일 프로바이더 도메인이 이제 기본적으로 비활성화됩니다.

SSSD를 사용하여 로컬 파일에서 사용자 및 그룹 정보를 검색하려면 다음을 수행합니다.

1. SSSD 구성. 다음 옵션 중 하나를 선택합니다.

   a. sssd.conf 구성 파일에서 id_provider=files 옵션을 사용하여 로컬 도메인을 명시적으로 구성합니다.

```
[domain/local]
id_provider=files
...
```

b. **sssd.conf** 구성 파일에서 **enable_files_domain=true** 를 설정하여 파일 공급자를 활성화합니다.

```
[sssd]
enable_files_domain = true
```

2. 이름 서비스 스위치를 구성합니다.

```
# authselect enable-feature with-files-provider
```

(JIRA:RHELPLAN-100639)

7.8. 그래픽 인프라

X.org Server가 더 이상 사용되지 않음

X.org 디스플레이 서버는 더 이상 사용되지 않으며 향후 주요 **RHEL** 릴리스에서 제거될 예정입니다. 대부분의 경우 기본 데스크탑 세션은 이제 **Wayland** 세션입니다.

X11 프로토콜은 **XWayland** 백엔드를 사용하여 완전히 지원됩니다. 결과적으로 **X11** 이 필요한 애플리케이션은 **Wayland** 세션에서 실행할 수 있습니다.

Red Hat은 **Wayland** 세션의 나머지 문제와 격차를 해결하기 위해 노력하고 있습니다. **Wayland** 에서 미해결 문제(**Known issues**) 섹션을 참조하십시오.

사용자 세션을 **X.org** 백엔드로 다시 전환할 수 있습니다. 자세한 내용은 **GNOME** 환경 선택 및 표시 프로토콜 을 참조하십시오.

(JIRA:RHELPLAN-121048)

motif가 더 이상 사용되지 않음

업스트림 **Motif** 커뮤니티의 개발이 비활성 상태이기 때문에 **RHEL**에서 **Motif** 위젯 툴킷이 더 이상 사용되지 않습니다.

개발 및 디버깅 변형을 포함하여 다음 **Motif** 패키지가 더 이상 사용되지 않습니다.

- **motif**
- **openmotif**
- **openmotif21**
- **openmotif22**

또한 **motif-static** 패키지가 제거되었습니다.

GTK 툴킷을 교체용으로 사용하는 것이 좋습니다. **GTK**는 보다 유지 관리할 수 있으며 **Motif**에 비해 새로운 기능을 제공합니다.

(JIRA:RHELPLAN-98983)

7.9. RED HAT ENTERPRISE LINUX 시스템 역할

RHEL 9 노드에서 팀을 구성할 때 네트워킹 시스템 역할에 사용 중단 경고가 표시됩니다.

RHEL 9에서는 네트워크 티밍 기능이 더 이상 사용되지 않습니다. 그 결과 **RHEL 8** 컨트롤러에서 네트워킹 **RHEL** 시스템 역할을 사용하여 **RHEL 9** 노드에서 네트워크 팀을 구성하여 사용 중단에 대한 경고가 표시됩니다.

(BZ#1999770)

7.10. 가상화

SHA1- 기반 서명을 사용한 **SecureBoot** 이미지 확인은 더 이상 사용되지 않음

UEFI (PE/COFF) 실행 파일에서 **SHA1-** 기반 서명을 사용하여 **SecureBoot** 이미지 확인을 수행하면 더 이상 사용되지 않습니다. 대신 **SHA2** 알고리즘을 기반으로 한 서명을 사용하는 것이 좋습니다.

(BZ#1935497)

가상 머신 스냅샷에 대한 제한된 지원

VM(가상 머신)의 스냅샷을 생성하는 것은 현재 UEFI 펌웨어를 사용하지 않는 VM에서만 지원됩니다. 또한 스냅샷 작업 중에 QEMU 모니터가 차단되어 특정 워크로드의 하이퍼바이저 성능에 부정적인 영향을 미칠 수 있습니다.

또한 VM 스냅샷 생성 메커니즘은 더 이상 사용되지 않으며 Red Hat은 프로덕션 환경에서 VM 스냅샷을 사용하는 것을 권장하지 않습니다. 그러나 새로운 VM 스냅샷 메커니즘이 개발 중이며 향후 RHEL 9의 마이너 릴리스에서 완전히 구현될 예정입니다.

(JIRA:RHELPLAN-15509, BZ#1621944)

virt-manager가 더 이상 사용되지 않음

virt-manager라고도 하는 Virtual Machine Manager 애플리케이션은 더 이상 사용되지 않습니다. Cockpit 라고도 하는 RHEL 웹 콘솔은 후속 릴리스에서 대체될 예정입니다. 따라서 GUI에서 가상화를 관리하기 위해 웹 콘솔을 사용하는 것이 좋습니다. 그러나 virt-manager 에서 사용 가능한 일부 기능은 RHEL 웹 콘솔에서 아직 제공되지 않을 수 있습니다.

(JIRA:RHELPLAN-10304)

libvirtd 가 더 이상 사용되지 않음

monolithic libvirt 데몬인 libvirtd 는 RHEL 9에서 더 이상 사용되지 않으며 RHEL의 향후 주요 릴리스에서 제거될 예정입니다. 여전히 libvirtd 를 사용하여 하이퍼바이저의 가상화를 관리할 수 있지만, 새로 도입된 모듈식 libvirt 데몬으로 전환하는 것이 좋습니다. 자세한 내용은 RHEL 9 구성 및 가상화 관리 문서를 참조하십시오.

(JIRA:RHELPLAN-113995)

가상 플로피 드라이버가 더 이상 사용되지 않습니다.

가상 플로피 디스크 장치를 제어하는 isa-fdc 드라이버는 더 이상 사용되지 않으며 향후 RHEL 릴리스에서 지원되지 않습니다. 따라서 마이그레이션된 가상 머신(VM)과의 호환성을 보장하기 위해 RHEL 9에서 호스팅되는 VM에서 플로피 디스크 장치를 사용하는 것을 권장하지 않습니다.

(BZ#1965079)

qcow2-v2 이미지 형식이 더 이상 사용되지 않음

RHEL 9에서는 가상 디스크 이미지의 **qcow2-v2** 형식이 더 이상 사용되지 않으며 향후 **RHEL** 주요 릴리스에서 지원되지 않습니다. 또한 **RHEL 9** 이미지 빌더는 **qcow2-v2** 형식으로 디스크 이미지를 생성할 수 없습니다.

qcow2-v2 대신 **qcow2-v3** 사용을 강력히 권장합니다. **qcow2-v2** 이미지를 최신 형식 버전으로 변환하려면 **qemu-img amend** 명령을 사용합니다.

(BZ#1951814)

7.11. 컨테이너

RHEL 7 호스트에서 **RHEL 9** 컨테이너 실행은 지원되지 않습니다.

RHEL 7 호스트에서 **RHEL 9** 컨테이너 실행은 지원되지 않습니다. 효과가 있을 수 있지만 보장되지는 않습니다.

자세한 내용은 Red Hat Enterprise Linux 컨테이너 호환성 매트릭스 를 참조하십시오.

(JIRA:RHELPLAN-100087)

Podman 내에서 **SHA1** 해시 알고리즘이 더 이상 사용되지 않음

Podman에서 **rootless** 네트워크 네임스페이스의 파일 이름을 생성하는 데 사용되는 **SHA1** 알고리즘은 더 이상 지원되지 않습니다. 따라서 **rootless** 컨테이너는 RHBA-2022:5951 에서 **Podman 4.1.1**로 업데이트하기 전에 시작되어 업그레이드 후 시작되는 컨테이너를 사용하기 위해 네트워크에 가입되어 네트워크에 가입되어 있지 않은 경우 재시작해야 합니다.

(BZ#2069279)

rhel9/pause 가 더 이상 사용되지 않음

rhel9/pause 컨테이너 이미지가 더 이상 사용되지 않습니다.

(BZ#2106816)

7.12. 더 이상 사용되지 않는 패키지

이 섹션에는 더 이상 사용되지 않으며 향후 Red Hat Enterprise Linux 주요 릴리스에 포함되지 않는 패키지가 나와 있습니다.

RHEL 8과 RHEL 9 간의 패키지 변경 사항은 *RHEL 9 문서 채택 시 고려 사항의* 패키지 변경 사항을 참조하십시오.

다음 패키지는 더 이상 사용되지 않으며 RHEL 9의 라이프 사이클이 종료될 때까지 계속 지원됩니다.

- iptables-devel
- iptables-libs
- iptables-nft
- iptables-nft-services
- iptables-utils
- libdb
- mcpp
- python3-pytz

8장. 확인된 문제

이 부분에서는 Red Hat Enterprise Linux 9.0의 알려진 문제에 대해 설명합니다.

8.1. 설치 프로그램 및 이미지 생성

reboot --kexec 및 inst.kexec 명령은 예측 가능한 시스템 상태를 제공하지 않습니다.

reboot --kexec Kickstart 명령 또는 inst.kexec 커널 부팅 매개 변수를 사용하여 RHEL 설치를 수행해도 전체 재부팅과 동일한 예측 가능한 시스템 상태가 제공되지 않습니다. 결과적으로 재부팅하지 않고 설치된 시스템으로 전환하면 예기치 않은 결과가 발생할 수 있습니다.

kexec 기능은 더 이상 사용되지 않으며 향후 Red Hat Enterprise Linux 릴리스에서 제거될 예정입니다.

(BZ#1697896)

타사 도구를 사용하여 생성된 USB에서 설치를 부팅할 때 로컬 미디어 설치 소스가 탐지되지 않습니다.

타사 툴을 사용하여 생성된 USB에서 RHEL 설치를 부팅할 때 설치 프로그램이 로컬 미디어 설치 소스를 감지하지 못합니다(*Red Hat CDN* 만 감지됨).

이 문제는 기본 부팅 옵션 int.stage2= 에서 iso9660 이미지 형식을 검색하려고 시도하기 때문에 발생합니다. 그러나 타사 툴은 다른 형식의 ISO 이미지를 생성할 수 있습니다.

해결 방법으로 다음 해결 방법 중 하나를 사용합니다.

- 설치를 부팅할 때 Tab 키를 클릭하여 커널 명령줄을 편집하고 부팅 옵션 inst.stage2= 를 inst. repo=로 변경합니다.
- Windows에서 부팅 가능한 USB 장치를 만들려면 Fedora Media Writer를 사용하십시오.
- rufus와 같은 타사 도구를 사용하여 부팅 가능한 USB 장치를 만드는 경우 먼저 Linux 시스템에서 RHEL ISO 이미지를 다시 생성한 다음 타사 도구를 사용하여 부팅 가능한 USB 장치를 생성합니다.

지정된 해결방법을 수행하는 단계에 대한 자세한 내용은 RHEL 8.3 설치 중에 설치 미디어가 자동으로 탐지되지 않습니다.

(BZ#1877697)

auth 및 authconfig Kickstart 명령에는 AppStream 리포지토리가 필요

authselect-compat 패키지는 설치하는 동안 auth 및 authconfig Kickstart 명령이 필요합니다. 이 패키지가 없으면 auth 또는 authconfig가 사용되는 경우 설치에 실패합니다. 설계에 따라 authselect-compat 패키지는 AppStream 리포지토리에서만 사용할 수 있습니다.

이 문제를 해결하려면 설치 프로그램에 BaseOS 및 AppStream 리포지토리를 사용할 수 있는지 확인하거나 설치 중에 authselect Kickstart 명령을 사용합니다.

(BZ#1640697)

Anaconda가 애플리케이션으로 실행되는 시스템에서 예기치 않은 SELinux 정책

Anaconda가 이미 설치된 시스템에서 애플리케이션으로 실행되는 경우(예: -image anaconda 옵션을 사용하여 이미지 파일에 다른 설치를 수행하는 등) 설치 중에 시스템이 SELinux 유형 및 속성을 수정할 수 없습니다. 결과적으로 Anaconda가 실행되는 시스템에서 SELinux 정책의 특정 요소가 변경될 수 있습니다. 이 문제를 해결하려면 production 시스템에서 Anaconda를 실행하지 말고 임시 가상 머신에서 실행합니다. 따라서 프로덕션 시스템의 SELinux 정책은 수정되지 않습니다. boot.iso 또는 dvd.iso 로 설치하는 것과 같이 시스템 설치 프로세스의 일부로 anaconda를 실행하면 이 문제의 영향을 받지 않습니다.

(BZ#2050140)

USB CD-ROM 드라이브는 Anaconda에서 설치 소스로 사용할 수 없습니다.

USB CD-ROM 드라이브가 해당 드라이브의 소스이고 Kickstart ignoredisk --only-use= 명령이 지정되면 설치에 실패합니다. 이 경우 Anaconda에서 이 소스 디스크를 찾아서 사용할 수 없습니다.

이 문제를 해결하려면 hard drive --partition=sdX --dir=/ 명령을 사용하여 USB CD-ROM 드라이브에서 설치합니다. 이로 인해 설치에 실패하지 않습니다.

(BZ#1914955)

최소 RHEL 설치에는 s390utils-base 패키지가 더 이상 포함되지 않습니다.

RHEL 8.4 이상에서 s390utils-base 패키지는 s390utils-core 패키지와 보조 s390utils-base 패키지로 나뉩니다. 결과적으로 RHEL 설치를 minimal-environment 로 설정하면 보조 s390utils-base 패키지가 아닌 필요한 s390utils-core 패키지만 설치됩니다. 이 문제를 해결하려면 RHEL 설치를 완료한 후 s390utils-base 패키지를 수동으로 설치하거나 kickstart 파일을 사용하여 s390utils-base 를 명시적으로 설치합니다.

(BZ#1932480)

iso9660 파일 시스템을 사용하여 하드 드라이브 분할 설치에 실패합니다.

하드 드라이브가 iso9660 파일 시스템으로 분할된 시스템에는 RHEL을 설치할 수 없습니다. 이는 iso9660 파일 시스템 파티션이 포함된 하드 디스크를 무시하도록 설정된 업데이트된 설치 코드로 인해 발생합니다. 이는 DVD를 사용하지 않고 RHEL을 설치하는 경우에도 발생합니다.

이 문제를 해결하려면 설치를 시작하기 전에 Kickstart 파일에 다음 스크립트를 추가하여 디스크를 포맷합니다.

참고: 해결 방법을 수행하기 전에 디스크에서 사용 가능한 데이터를 백업하십시오. over fs 명령은 디스크의 모든 기존 데이터를 포맷합니다.

```
%pre
wipefs -a /dev/sda
%end
```

결과적으로 설치 작업이 오류 없이 예상대로 작동합니다.

(BZ#1929105)

Anaconda에서 관리자 사용자 계정의 존재를 확인하지 못했습니다.

그래픽 사용자 인터페이스를 사용하여 RHEL을 설치하는 동안 Anaconda는 관리자 계정이 생성되었는지 확인하지 않습니다. 결과적으로 사용자는 관리자 사용자 계정없이 시스템을 설치할 수 있습니다.

이 문제를 해결하려면 관리자 사용자 계정 또는 root 암호가 설정되고 root 계정을 잠금 해제해야 합니다. 따라서 사용자는 설치된 시스템에서 관리 작업을 수행할 수 있습니다.

(BZ#2047713)

Anaconda에서 CHAP 인증 시도에 실패한 후 인증 방법을 사용하지 않고 iSCSI 서버에 로그인하지 못했습니다.

CHAP 인증을 사용하여 iSCSI 디스크를 추가하고 잘못된 자격 증명으로 인해 로그인 시도가 실패하면 인증 방법이 실패하지 않은 디스크를 다시 로그인합니다. 이 문제를 해결하려면 현재 세션을 닫고 인증 방법을 사용하지 않고 로그인합니다.

(BZ#1983602)

새로운 XFS 기능은 버전 5.10보다 펌웨어가 있는 PowerNV IBM POWER 시스템의 부팅을 방지합니다.

PowerNV IBM POWER 시스템은 펌웨어에 Linux 커널을 사용하고, GRUB 대신 Petitboot를 사용합니다. 그러면 펌웨어 커널이 /boot 및 Petitboot를 통해 GRUB 설정을 읽고 RHEL을 부팅합니다.

RHEL 9 커널은 bigtime=1 및 inobtcount=1 기능을 XFS 파일 시스템에 도입합니다. 이 기능은 버전 5.10 미만의 펌웨어가 있는 커널은 이해하지 못합니다.

이 문제를 해결하려면 /boot 에 다른 파일 시스템을 사용할 수 있습니다(예: ext4).

(BZ#1997832)

PReP 크기가 4 또는 8MiB가 아닌 경우 RHEL을 설치할 수 없습니다.

PowerPC 참조 플랫폼(PReP) 파티션이 4 kiB 섹터를 사용하는 디스크의 크기가 4MiB 또는 8MiB인 경우 RHEL 설치 프로그램에서 부트 로더를 설치할 수 없습니다. 따라서 디스크에 RHEL을 설치할 수 없습니다.

이 문제를 해결하려면 PReP 파티션 크기가 정확히 4MiB 또는 8MiB이고 크기가 다른 값으로 반올림되지 않았는지 확인하십시오. 결과적으로 설치 프로그램이 이제 디스크에 RHEL을 설치할 수 있습니다.

(BZ#2026579)

새로운 XFS 기능은 버전 5.10보다 오래된 펌웨어 커널을 사용하여 PowerNV IBM POWER 시스템의 부팅을 방지합니다.

PowerNV IBM POWER 시스템은 펌웨어에 Linux 커널을 사용하고, GRUB 대신 Petitboot를 사용합니다. 그러면 펌웨어 커널이 /boot 및 Petitboot를 통해 GRUB 설정을 읽고 RHEL을 부팅합니다.

RHEL 9 커널은 bigtime=1 및 inobtcount=1 기능을 XFS 파일 시스템에 도입했습니다. 이 기능은 버전 5.10보다 오래된 커널이 있는 펌웨어를 이해하지 못합니다. 결과적으로 Anaconda는 다음 오류 메시지가 있는 설치를 방지합니다.

펌웨어는 /boot 파일 시스템에서 XFS 파일 시스템 기능을 지원하지 않습니다. 이 시스템은 부팅할 수 없습니다. 펌웨어를 업그레이드하거나 파일 시스템 유형을 변경하십시오.

해결 방법으로 /boot 에 대한 다른 파일 시스템을 사용하십시오(예: ext4).

(BZ#2008792)

8.2. 서브스크립션 관리

FIPS 모드에서는 virt-who 가 ESX 서버에 연결할 수 없습니다.

FIPS 모드에서 RHEL 9 시스템에서 virt-who 유틸리티를 사용하는 경우 virt-who 는 ESX 서버에 연결할 수 없습니다. 결과적으로 virt-who 는 ESX 서버를 보고하지 않으며, 구성된 경우에도 다음 오류 메시지를 기록합니다.

```
ValueError: [digital envelope routines] unsupported
```

이 문제를 해결하려면 다음 중 하나를 수행합니다.

- virt-who 를 실행하는 데 사용하는 RHEL 9 시스템을 FIPS 모드로 설정하지 마십시오.
- virt-who 를 버전 9.0으로 실행하는 데 사용하는 RHEL 시스템을 업그레이드하지 마십시오.

(BZ#2054504)

8.3. 소프트웨어 관리

설치 프로세스가 응답하지 않는 경우가 있음

RHEL을 설치할 때 설치 프로세스가 응답하지 않는 경우가 있습니다. /tmp/packaging.log 파일은 끝에 다음 메시지를 표시합니다.

```
10:20:56,416 DDEBUG dnf: RPM transaction over.
```

이 문제를 해결하려면 설치 프로세스를 다시 시작하십시오.

(BZ#2073510)

8.4. 쉘 및 명령행 툴

ifcfg 파일을 사용하여 네트워크 인터페이스 이름 변경이 실패합니다.

RHEL 9에서는 initscripts 패키지가 기본적으로 설치되지 않습니다. 결과적으로 ifcfg 파일을 사용하여 네트워크 인터페이스 이름 변경이 실패합니다. 이 문제를 해결하기 위해 Red Hat은 udev 규칙을 사용하거나 파일 링크를 사용하여 인터페이스 이름을 변경할 것을 권장합니다. 자세한 내용은 Consistent network interface device naming and the systemd.link(5) 매뉴얼 페이지를 참조하십시오.

권장 솔루션 중 하나를 사용할 수 없는 경우 initscripts 패키지를 설치합니다.

(BZ#2018112)

chkconfig 패키지는 RHEL 9에 기본적으로 설치되지 않습니다.

시스템 서비스에 대한 실행 수준 정보를 업데이트하고 쿼리하는 chkconfig 패키지는 RHEL 9에 기본적으로 설치되지 않습니다.

서비스를 관리하려면 systemctl 명령을 사용하거나 chkconfig 패키지를 수동으로 설치합니다.

systemd 에 대한 자세한 내용은 systemd 소개를 참조하십시오. systemctl 유틸리티를 사용하는 방법에 대한 지침은 systemctl을 사용하여 시스템 서비스 관리를 참조하십시오.

(BZ#2053598)

8.5. 인프라 서비스

모두 **SHA-1** 기반 서명의 바인딩 및 바인딩 되지 않은 유효성 검사를 비활성화합니다.

바인딩 및 바인딩 되지 않은 구성 요소는 모든 **RSA/SHA1**(알고리즘 번호 5) 및 **RSASHA1-NSEC3-SHA1** 서명의 유효성 검사 지원을 비활성화하고, **SHA-1** 사용은 **DEFAULT** 시스템 차원의 암호화 정책에 제한되어 있습니다.

결과적으로 **SHA-1, RSA/SHA1** 및 **RSASHA1-NSEC3-SHA1** 다이제스트 알고리즘으로 서명된 특정 **DNSSEC** 레코드가 **Red Hat Enterprise Linux 9**에서 검증되지 않고 영향을 받는 도메인 이름이 취약해집니다.

이 문제를 해결하려면 **RSA/SHA-256** 또는 **elliptic** 곡선 키와 같은 다른 서명 알고리즘으로 업그레이드하십시오.

영향을 받고 취약한 최상위 도메인 목록은 **RSASHA1**로 서명된 **DNSSEC** 레코드를 참조하여 솔루션을 확인하지 못했습니다.

(**BZ#2070495**)

여러 영역에서 동일한 쓰기 가능 영역 파일을 사용하는 경우 **named** 가 시작되지 않습니다.

BIND에서는 여러 영역에서 동일한 쓰기 가능 영역 파일을 허용하지 않습니다. 따라서 구성에 명명된 서비스에서 수정할 수 있는 파일의 경로를 공유하는 여러 영역이 포함된 경우 **named** 가 시작되지 않습니다. 이 문제를 해결하려면 **in-view** 절을 사용하여 여러 뷰 간에 하나의 영역을 공유하고 다양한 영역에 대해 다른 경로를 사용해야 합니다. 예를 들어 경로에 뷰 이름을 포함합니다.

일반적으로 쓰기 가능 영역 파일은 **DNSSEC**에서 유지 관리하는 동적 업데이트, 슬레이브 영역 또는 영역이 허용된 영역에서 사용됩니다.

(**BZ#1984982**)

8.6. 보안

PKCS #11 토큰이 원시 **RSA** 또는 **RSA-PSS** 서명 생성을 지원하는 경우 **OpenSSL** 은 이를 감지하지 않습니다.

TLS 1.3 프로토콜은 **RSA-PSS** 서명을 지원해야 합니다. **PKCS #11** 토큰이 원시 **RSA** 또는 **RSA-PSS** 서명을 지원하지 않는 경우 **OpenSSL** 라이브러리를 사용하는 서버 애플리케이션은 **PKCS #11** 토큰에서 키를 보유하는 경우 **RSA** 키로 작동하지 않습니다. 그 결과 설명된 시나리오에서 **TLS** 통신이 실패합니다.

이 문제를 해결하려면 TLS 버전 1.2를 사용 가능한 최고 TLS 프로토콜 버전으로 사용하도록 서버와 클라이언트를 구성합니다.

(BZ#1681178)

OpenSSL 이 raw RSA 또는 RSA-PSS 서명을 지원하지 않는 PKCS #11 토큰을 잘못 처리

OpenSSL 라이브러리는 PKCS #11 토큰의 키 관련 기능을 감지하지 않습니다. 결과적으로 원시 RSA 또는 RSA-PSS 서명을 지원하지 않는 토큰으로 서명이 생성될 때 TLS 연결 설정이 실패합니다.

이 문제를 해결하려면 /etc/pki/tls/openssl.cnf 파일의 crypto_policy 섹션 끝에 .include 행 뒤에 다음 행을 추가합니다.

```
SignatureAlgorithms =
RSA+SHA256:RSA+SHA512:RSA+SHA384:ECDSA+SHA256:ECDSA+SHA512:ECDSA+SHA384
MaxProtocol = TLSv1.2
```

결과적으로 설명된 시나리오에서 TLS 연결을 설정할 수 있습니다.

(BZ#1685470)

FIPS가 승인되지 않은 암호화는 FIPS 모드에서 OpenSSL에서 작동하지 않음

FIPS가 승인되지 않은 암호화는 시스템 설정에 관계없이 OpenSSL 툴킷에서 작동합니다. 따라서 시스템이 FIPS 모드에서 실행될 때 비활성화해야 하는 암호화 알고리즘 및 암호를 사용할 수 있습니다. 예를 들면 다음과 같습니다.

- RSA 키 교환 작업을 사용하는 TLS 암호화 제품군.
- PKCS #1 및 SSLv23 패딩을 사용하거나 2048비트보다 짧은 키를 사용하더라도 공개 키 암호화 및 암호 해독을 위한 RSA 기반 알고리즘이 작동합니다.

(BZ#2053289)

OpenSSL은 FIPS 모드에서 엔진을 사용할 수 없습니다.

engine API는 OpenSSL 3.0에서 더 이상 사용되지 않으며 OpenSSL Federal Information Processing Standards (FIPS) 구현 및 기타 FIPS 호환 구현과 호환되지 않습니다. 따라서 OpenSSL은 FIPS 모드에서 엔진을 실행할 수 없습니다. 이 문제에 대한 해결방법은 없습니다.

(BZ#2087253)

PSK ciphersuites는 FUTURE 암호화 정책에서 작동하지 않습니다.

PSK(Pre-shared key) 암호화suites는 PFS(Smart forward secrecy) 키 교환 방법을 수행하는 것으로 인식되지 않습니다. 결과적으로 ECDHE-PSK 및 DHE-PSK 암호suites는 FUTURE 암호화 정책 (예: FUTURE crypto policy)과 함께 구성된 OpenSSL에서 작동하지 않습니다. 이 문제를 해결하려면 덜 제한적인 암호화 정책을 설정하거나 PSK 암호suites를 사용하는 애플리케이션에 대해 낮은 보안 수준 (ECLEVEL)을 설정할 수 있습니다.

(BZ#2060044)

GnuPG를 잘못 사용하면 암호화에서 허용되지 않는 경우에도 SHA-1 서명을 사용할 수 있습니다.

GnuPG(GNUPG) 암호화 소프트웨어는 시스템 전체의 암호화 정책에 정의된 설정과 관계없이 SHA-1 알고리즘을 사용하는 서명을 생성하고 확인할 수 있습니다. 따라서 DEFAULT 암호화 정책의 암호화 목적에 SHA-1을 사용할 수 있으며 이는 서명에 대해 이 안전하지 않은 알고리즘의 시스템 전체 사용 중단과 일치하지 않습니다.

이 문제를 해결하려면 SHA-1과 관련된 GnuPG 옵션을 사용하지 마십시오. 결과적으로 보안되지 않은 SHA-1 서명을 사용하여 GnuPG가 기본 시스템 보안을 낮추지 않도록 합니다.

(BZ#2070722)

일부 OpenSSH 작업은 FIPS 승인 인터페이스를 사용하지 않습니다.

OpenSSH에서 사용하는 OpenSSL 암호화 라이브러리는 레거시 및 최신의 두 가지 인터페이스를 제공합니다. OpenSSL 내부가 변경되어 최신 인터페이스만 암호화 알고리즘의 FIPS 인증 구현을 사용합니다. OpenSSH는 일부 작업에 기존 인터페이스를 사용하므로 FIPS 요구 사항을 준수하지 않습니다.

(BZ#2087121)

GPG-agent 가 FIPS 모드에서 SSH 에이전트로 작동하지 않음

gpg-agent 툴은 FIPS 모드에서 MD5 다이제스트를 비활성화하더라도 ssh-agent 프로그램에 키를 추가할 때 MD5 지문을 생성합니다. 결과적으로 ssh-add 유틸리티에서 키를 인증 에이전트에 추가하지 못

했습니다.

이 문제를 해결하려면 **gpg-agent --daemon --enable-ssh-support** 명령을 사용하지 않고 **~/.gnupg/sshcontrol** 파일을 만듭니다. 예를 들어 **< FINGERPRINT> 0** 형식으로 **gpg --list-keys** 명령의 출력을 **~/.gnupg/sshcontrol** 에 붙여넣을 수 있습니다. 결과적으로 **gpg-agent** 는 **SSH** 인증 에이전트로 작동합니다.

(BZ#2073567)

SELinux staff_u 사용자가 **unconfined_r**로 잘못 전환할 수 있습니다.

secure_mode 부울이 활성화되면 **staff_u** 사용자가 **unconfined_r** 역할로 잘못 전환할 수 있습니다. 결과적으로 **staff_u** 사용자는 시스템의 보안에 영향을 미치는 권한 있는 작업을 수행할 수 있습니다.

(BZ#2021529)

기본 **SELinux** 정책을 통해 제한되지 않은 실행 파일로 스택을 실행할 수 있습니다.

SELinux 정책에 있는 **selinuxuser_execstack** 부울의 기본 상태는 에 있습니다. 즉, **Bitbucket** 실행 파일이 스택을 실행 가능하게 만들 수 있습니다. 실행 파일은 이 옵션을 사용하지 않아야 하며 잘못 코딩된 실행 파일이나 가능한 공격을 나타낼 수 있습니다. 그러나 **Red Hat**은 다른 툴, 패키지 및 타사 제품과의 호환성으로 인해 기본 정책에서 부울의 값을 변경할 수 없습니다. 시나리오가 이러한 호환성 측면에 의존하지 않는 경우 명령 **setsebool -P selinuxuser_execstack**을 입력하여 로컬 정책에서 부울을 켤 수 있습니다.

(BZ#2064274)

ImageStreamTag 프로필의 **SSH** 시간 초과 규칙 구성 잘못된 옵션

OpenSSH 업데이트는 다음과 같은 **Defense** 정보 시스템국(**DISAonnectionFactory**) 프로필의 규칙에 영향을 미쳤습니다.

- **DISA STIG for RHEL 9 (xccdf_org.ssgproject.content_profile_stig)**
- **RHEL 9**용 **GUI(xccdf_org.ssgproject.content_profile_stig_gui)**

이러한 각 프로필에서는 다음 두 가지 규칙이 영향을 받습니다.

```
Title: Set SSH Client Alive Count Max to zero
CCE Identifier: CCE-90271-8
Rule ID: xccdf_org.ssgproject.content_rule_sshd_set_keepalive_0

Title: Set SSH Idle Timeout Interval
CCE Identifier: CCE-90811-1
Rule ID: xccdf_org.ssgproject.content_rule_sshd_set_idle_timeout
```

SSH 서버에 적용할 때 이러한 각 규칙은 더 이상 이전처럼 작동하지 않는 옵션(**ClientAliveCountMax** 및 **ClientAliveInterval**)을 구성합니다. 결과적으로 이러한 규칙에서 구성한 타임아웃에 도달하면 **OpenSSH**는 더 이상 유휴 **SSH** 사용자의 연결을 끊지 않습니다. 해결 방법으로 이러한 규칙은 솔루션이 개발될 때까지 **RHEL 9**용 **GUI**를 사용하여 **DISA organization**에서 임시로 제거되었으며, **RHEL 9**용 **GUI**를 사용하여 이러한 규칙이 일시적으로 제거되었습니다.

(**BZ#2038978**)

fagenrules --load 가 제대로 작동하지 않음

fapolicyd 서비스는 **SIGHUP(SIGHUP)** 신호를 올바르게 처리하지 않습니다. 결과적으로 **SIGHUP** 신호를 수신한 후 **fapolicyd** 가 종료됩니다. 따라서 **fagenrules --load** 명령이 제대로 작동하지 않으며 규칙 업데이트에서는 **fapolicyd** 를 수동으로 다시 시작해야 합니다. 이 문제를 해결하려면 규칙이 변경된 후 **fapolicyd** 서비스를 다시 시작하고 결과적으로 **fagenrules --load** 가 올바르게 작동합니다.

(**BZ#2070655**)

Ansible 수정에는 추가 컬렉션이 필요합니다.

ansible-core 패키지에서 **Ansible Engine**을 교체하면 **RHEL** 서브스크립션과 함께 제공되는 **Ansible** 모듈 목록이 줄어듭니다. 결과적으로 **scap-security-guide** 패키지에 포함된 **Ansible** 콘텐츠를 사용하는 수정을 실행하려면 **rhc-worker-playbook** 패키지의 컬렉션이 필요합니다.

Ansible 수정을 위해 다음 단계를 수행합니다.

1. 필수 패키지를 설치합니다.

```
# dnf install -y ansible-core scap-security-guide rhc-worker-playbook
```

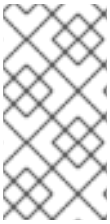
2. **/usr/share/scap-security-guide/ansible** 디렉토리로 이동합니다. **# cd /usr/share/scap-security-guide/ansible**

3. 추가 **Ansible** 컬렉션 경로를 정의하는 환경 변수를 사용하여 관련 **Ansible** 플레이북을 실행합니다.

```
# ANSIBLE_COLLECTIONS_PATH=/usr/share/rhc-worker-playbook/ansible/collections/ansible_collections/ ansible-playbook -c local -i localhost, rhel9-playbook-cis_server_l1.yml
```

cis_server_l1 을 시스템을 수정하려는 프로필의 **ID**로 바꿉니다.

결과적으로 **Ansible** 콘텐츠가 올바르게 처리됩니다.



참고

rhc-worker-playbook 에 제공된 컬렉션 지원은 **scap-security-guide** 에서 제공되는 **Ansible** 콘텐츠를 사용하도록 제한됩니다.

(BZ#2105162)

8.7. 네트워킹

커널 명령줄에 빈 **rd.znet** 옵션을 사용하면 네트워크 구성이 실패합니다.

커널의 네트워크 유형 또는 하위 채널과 같은 인수 없이 **rd.znet** 옵션이 없습니다. 이 문제를 해결하려면 명령줄에서 **rd.znet** 옵션을 완전히 제거하거나 관련 **net** 유형, 하위 채널 및 기타 관련 옵션을 지정합니다. 이러한 옵션에 대한 자세한 내용은 **dra cut.cmdline(7)** 도움말 페이지를 참조하십시오.

(BZ#1931284)

세션 키를 업데이트하지 않으면 연결이 끊어집니다.

kTLS(커널 전송 계층 보안) 프로토콜은 대칭 암호에서 사용하는 세션 키 업데이트를 지원하지 않습니다. 결과적으로 사용자가 키를 업데이트할 수 없으므로 연결이 끊어집니다. 이 문제를 해결하려면 **kTLS**를 비활성화합니다. 그 결과 세션 키를 성공적으로 업데이트할 수 있습니다.

(BZ#2013650)

initscripts 패키지는 기본적으로 설치되지 않습니다.

기본적으로 **initscripts** 패키지는 설치되지 않습니다. 결과적으로 **ifup** 및 **ifdown** 유틸리티를 사용할 수 없습니다. 또는 **nmcli connection up** 및 **nmcli connection down** 명령을 사용하여 연결을 활성화 및 비활성화합니다. 제안된 대체 방법이 작동하지 않는 경우 문제를 보고하고 **NetworkManager-initscripts-updown** 패키지를 설치합니다. 이 패키지는 **ifup** 및 **ifdown** 유틸리티에 대한 **NetworkManager** 솔루션을 제공합니다.

(BZ#2082303)

Alibaba Cloud에서 **nm-cloud-setup** 서비스를 시작한 후 인스턴스의 기본 **IP** 주소가 변경됩니다.

Alibaba Cloud에서 인스턴스를 시작한 후 **nm-cloud-setup** 서비스는 기본 **IP** 주소를 인스턴스에 할당합니다. 그러나 인스턴스에 여러 개의 보조 **IP** 주소를 할당하고 **nm-cloud-setup** 서비스를 시작하면 이전의 기본 **IP** 주소가 이미 할당된 보조 **IP** 주소 중 하나로 교체됩니다. 반환된 메타데이터 목록은 동일한지 확인합니다. 문제를 해결하려면 기본 **IP** 주소가 변경되지 않도록 보조 **IP** 주소를 수동으로 구성합니다. 결과적으로 인스턴스는 **IP** 주소를 모두 유지하며 기본 **IP** 주소는 변경되지 않습니다.

(BZ#2079849)

8.8. 커널

RHEL 9 커널에서 **kdump** 가 시작되지 않음

RHEL 9 커널에는 **crashkernel=auto** 매개 변수가 기본값으로 구성되어 있지 않습니다. 결과적으로 **kdump** 서비스가 기본적으로 시작되지 않습니다.

이 문제를 해결하려면 **crashkernel=** 옵션을 필수 값으로 구성합니다.

예를 들어 **grubby** 유틸리티를 사용하여 **256MB**의 메모리를 예약하려면 다음 명령을 입력합니다.

```
# grubby --args crashkernel=256M --update-kernel ALL
```

결과적으로 **RHEL 9** 커널은 **kdump** 를 시작하고 구성된 메모리 크기 값을 사용하여 **vmcore** 파일을 덤프합니다.

(BZ#1894783)

LUKS 암호화 대상에서 **vmcore** 를 캡처하지 못했습니다.

Linux 통합 키 설정(LUKS) 암호화 파티션을 사용하여 시스템에서 kdump 를 실행하는 경우 시스템에는 사용 가능한 특정 양의 메모리가 필요합니다. 사용 가능한 메모리가 필요한 메모리 양보다 작으면 systemd-cryptsetup 서비스가 파티션을 마운트하지 못합니다. 결과적으로 두 번째 커널은 LUKS 암호화 대상에서 크래시 덤프 파일(vmcore)을 캡처하지 못했습니다.

kdumpctl estimate 명령을 사용하면 kdump 에 필요한 권장 메모리 크기인 Recommended crashkernel 값을 쿼리할 수 있습니다.

이 문제를 해결하려면 다음 단계를 사용하여 LUKS 암호화된 대상에서 kdump 에 필요한 메모리를 구성합니다.

1. 크래시커널 추정치를 출력합니다.

```
# kdumpctl estimate
```

2. crashkernel 값을 늘려 필요한 메모리 양을 구성합니다.

```
# grubby --args=crashkernel=652M --update-kernel=ALL
```

3. 변경 사항을 적용하려면 시스템을 재부팅합니다.

```
# reboot
```

결과적으로 kdump 는 LUKS 암호화 파티션이 있는 시스템에서 올바르게 작동합니다.

(BZ#2017401)

부팅 시 크래시 커널 메모리 할당 실패

특정 Ampere Altra 시스템에서 사용 가능한 메모리가 1GB 미만인 경우 부팅 중에 kdump 사용을 위해 크래시 커널 메모리를 할당할 수 없습니다. 결과적으로 필요한 메모리가 사용 가능한 메모리보다 많을 때 kdump ctl 명령이 kdump 서비스를 시작하지 못합니다.

해결 방법으로 크기 요구 사항에 맞게 최소240MB로 crashkernel 매개변수의 값을 줄입니다(예: crashkernel=240M). 결과적으로 kdump 에 대한 크래시 커널 메모리 할당이 Ampere Altra 시스템에서 실패하지 않습니다.

(BZ#2065013)

kTLS는 TLS 1.3을 NIC로 오프로드하는 것을 지원하지 않습니다.

kTLS(커널 전송 계층 보안)는 TLS 1.3을 NIC로 오프로드하는 것을 지원하지 않습니다. 결과적으로 NIC가 TLS 오프로드를 지원하는 경우에도 소프트웨어 암호화가 TLS 1.3과 함께 사용됩니다. 이 문제를 해결하려면 오프로드가 필요한 경우 TLS 1.3을 비활성화합니다. 따라서 TLS 1.2만 오프로드할 수 있습니다. TLS 1.3을 사용하면 TLS 1.3을 오프로드할 수 없기 때문에 성능이 저하됩니다.

(BZ#2000616)

8.9. 부트 로더

새 커널의 이전 명령줄 옵션 손실

GRUB 부트 로더는 이전에 구성한 커널 명령줄 옵션을 새 커널에 적용하지 않습니다. 결과적으로 커널 패키지를 업그레이드할 때 누락된 옵션으로 인해 재부팅 후 시스템 동작이 변경될 수 있습니다.

문제를 해결하려면 각 커널 업그레이드 후 모든 사용자 지정 커널 명령줄 옵션을 수동으로 추가합니다. 결과적으로 커널은 다음 커널 업그레이드까지 사용자 지정 옵션을 예상대로 적용합니다.

(BZ#1969362)

8.10. 파일 시스템 및 스토리지

NVMe/TCP에서는 장치 매퍼 Multipath가 지원되지 않습니다.

nvme-tcp 드라이버와 함께 장치 매퍼 Multipath를 사용하면 호출 추적 경고 및 시스템 불안정성이 발생할 수 있습니다. 이 문제를 해결하려면 NVMe/TCP 사용자가 기본 NVMe 다중 경로를 활성화하고 NVMe와 함께 device-mapper-multipath 툴을 사용하지 않아야 합니다.

기본적으로 Native NVMe 다중 경로는 RHEL 9에서 활성화됩니다. 자세한 내용은 NVMe 장치에서 다중 경로 활성화를 참조하십시오.

(BZ#2033080)

blk-availability systemd 서비스는 복잡한 장치 스택을 비활성화합니다.

systemd 에서 기본 블록 비활성화 코드가 항상 복잡한 가상 블록 장치의 스택을 올바르게 처리하지는 않습니다. 일부 구성에서는 종료 중에 가상 장치가 제거되지 않을 수 있으므로 오류 메시지가 기록됩니다. 이 문제를 해결하려면 다음 명령을 실행하여 복잡한 블록 장치 스택을 비활성화합니다.

```
# systemctl enable --now blk-availability.service
```

결과적으로 종료 중에 복잡한 가상 장치 스택이 올바르게 비활성화되고 오류 메시지가 생성되지 않습니다.

(BZ#2011699)

supported_speeds에 대한 잘못된 **sysfs** 값

qla2xxx 드라이버는 예상 64Gb/s 대신 sysfs supported_speeds 속성에서 지원되는 포트 속도 중 하나로 20Gb/s를 보고합니다.

```
$ cat /sys/class/fc_host/host12/supported_speeds
16 Gbit, 32 Gbit, 20 Gbit
```

결과적으로 HBA에서 64Gb/s 링크 속도를 지원하는 경우 sysfs supported_speeds 값이 올바르지 않습니다. 이는 sysfs 의 supported_speeds 값에만 영향을 미치고 포트는 예상 협상 링크 속도에서 작동합니다.

(BZ#2069758)

AMD EPYC 시스템의 **Broadcom** 이니시에이터에서 **NVMe** 네임스페이스에 연결할 수 없음

기본적으로 RHEL 커널은 AMD 기반 플랫폼에서 IOMMU를 활성화합니다. 결과적으로 AMD 프로세서가 있는 서버에서 IOMMU 지원 플랫폼을 사용할 때 전송 길이 불일치로 인한 I/O와 같은 NVMe I/O 문제가 발생할 수 있습니다.

이 문제를 해결하려면 커널 명령줄 옵션인 iommu=pt 를 사용하여 passthrough 모드에 IOMMU를 추가합니다. 결과적으로 AMD EPYC 시스템의 Broadcom 이니시에이터에서 NVMe 네임스페이스에 연결할 수 있습니다.

(BZ#2073541)

8.11. 동적 프로그래밍 언어, 웹 서버 및 데이터베이스 서버

MySQL 및 MariaDB 의 --ssl-fips-mode 옵션은 FIPS 모드를 변경하지 않습니다.

MySQL 의 --ssl-fips-mode 옵션과 RHEL의 MariaDB 는 업스트림에서와 다르게 작동합니다.

RHEL 9에서는 --ssl-fips-mode 를 mysqld 또는 mariadbd 데몬의 인수로 사용하거나 MySQL 또는 MariaDB 서버 구성 파일에서 ssl-fips-mode 를 사용하는 경우 --ssl-fips-mode 는 이러한 데이터베이스 서버에 대해 FIPS 모드를 변경하지 않습니다.

대신 다음을 수행합니다.

- **--ssl-fips-mode 를 ON 으로 설정하면 mysqld 또는 mariadbd 서버 데몬이 시작되지 않습니다.**
- **FIPS 활성화 시스템에서 --ssl-fips-mode 를 OFF 로 설정하면 mysqld 또는 mariadbd 서버 데몬은 계속 FIPS 모드에서 실행됩니다.**

이는 특정 구성 요소에 대해 **FIPS** 모드를 활성화 또는 비활성화해야 하며 전체 **RHEL** 시스템에 대해 비활성화되어 있어야 하기 때문입니다.

따라서 **RHEL**의 **MySQL** 또는 **MariaDB** 에서 **--ssl-fips-mode** 옵션을 사용하지 마십시오. 대신 전체 **RHEL** 시스템에서 **FIPS** 모드가 활성화되어 있는지 확인하십시오.

- **FIPS** 모드가 활성화된 **RHEL**을 설치하는 것이 좋습니다. 설치 중에 **FIPS** 모드를 활성화하면 시스템이 **FIPS** 승인 알고리즘 및 지속적인 모니터링 테스트로 모든 키를 생성합니다. **FIPS** 모드에서 **RHEL**을 설치하는 방법에 대한 자세한 내용은 **FIPS** 모드에서 시스템 설치를 참조하십시오.
- 또는 시스템을 **FIPS** 모드로 전환하는 절차에 따라 전체 **RHEL** 시스템의 **FIPS** 모드를 전환할 수 있습니다.

(BZ#1991500)

8.12. 컴파일러 및 개발 도구

특정 기호 기반 프로브는 64비트 ARM 아키텍처의 SystemTap 에서 작동하지 않음

커널 구성은 SystemTap 에 필요한 특정 기능을 비활성화합니다. 따라서 일부 기호 기반 프로브는 64 비트 ARM 아키텍처에서 작동하지 않습니다. 결과적으로 영향을 받는 SystemTap 스크립트가 실행되지 않거나 원하는 프로브 포인트에서 히트를 수집하지 않을 수 있습니다.

이 버그는 RHBA-2022:5259 권고가 릴리스되어 나머지 아키텍처에서 수정되었습니다.

(BZ#2083727)

8.13. IDM (IDENTITY MANAGEMENT)

RHEL 9 Kerberos 클라이언트가 Heimdal EgressIP에 대해 PKINIT를 사용하여 사용자를 인증하지 못했습니다.

RHEL 9 Kerberos 클라이언트에서 IdM 사용자의 PKINIT 인증 동안 RHEL 9 또는 이전 버전의 Heimdal Kerberos Distribution Center(KDC)는 지원되는CMSTypes 필드를 지원하지 않기 때문에 SHA-1 백업 서명 알고리즘을 사용합니다. 그러나 SHA-1 알고리즘은 RHEL 9에서 더 이상 사용되지 않으므로 사용자 인증이 실패합니다.

이 문제를 해결하려면 다음 명령을 사용하여 RHEL 9 클라이언트에서 SHA-1 알고리즘 지원을 활성화하십시오.

```
# update-crypto-polices --set DEFAULT:SHA1
```

결과적으로 PKINIT 인증은 Kerberos 클라이언트와 Heimdal EgressIP 간에 작동합니다.

지원되는 백업 서명 알고리즘에 대한 자세한 내용은 Kerberos Encryption Types Defined for CMS Algorithm Identifiers 를 참조하십시오.

RHEL 9 Kerberos 에이전트가 RHEL 9 Kerberos 에이전트와 통신하면 사용자의 PKINIT 인증이 실패합니다. 에서 참조하십시오.

(BZ#2068935)

RHEL 9 Kerberos 에이전트가 RHEL 9 Kerberos 에이전트와 통신하면 사용자의 PKINIT 인증이 실패합니다.

RHEL 9 Kerberos 에이전트가 사용자 환경의 RHEL 9 Kerberos 에이전트와 상호 작용하는 경우 사용자의 초기 인증(PKINIT) 인증을 위한 Public Key Cryptography가 실패합니다. 이 문제를 해결하려면 다

음 작업 중 하나를 수행하십시오.

- **SHA-1** 서명을 확인할 수 있도록 **RHEL 9** 에이전트의 **crypto-policy**를 **DEFAULT:SHA1** 로 설정합니다.

```
# update-crypto-polices --set DEFAULT:SHA1
```

- **SHA-1** 알고리즘을 사용하여 **CMS** 데이터에 서명하지 않도록 **RHEL 9** 에이전트를 업데이트합니다. 이를 위해 **SHA-1** 대신 **SHA-256**을 사용하는 버전으로 **Kerberos** 패키지를 업데이트하십시오.
  - **CentOS 9 Stream: jenkinsfile5-1.19.1-15**
  - **RHEL 8.7: krb5-1.18.2-17**
  - **RHEL 7.9: krb5-1.15.1-53**
  - **Fedora Rawhide/36:tekton5-1.19.2-7**
  - **Fedora 35/34:qcow5-1.19.2-3**

패치되지 않은 에이전트가 **Kerberos** 클라이언트인지 **Kerberos** 배포 센터(**KDC**)인지와 관계없이 이러한 동작 중 하나를 수행해야 합니다.

그 결과 사용자의 **PKINIT** 인증이 제대로 작동합니다.

다른 운영 체제의 경우, **agent**가 **SHA-1** 대신 **SHA-256**으로 **CMS** 데이터에 서명할 수 있도록 하는 **jenkinsfile5-1.20** 릴리스입니다.

익명의 **PKINIT CMS** 메시지는 이제 **SHA-2**로 서명됩니다. 에서 참조하십시오.

(**BZ#2077450**)

AD trust에 대한 FIPS 지원에는 AD-SUPPORT crypto sub-policy가 필요합니다.

AD(Active Directory)는 기본적으로 RHEL 9에서 FIPS 모드에서 허용되지 않는 AES SHA-1 HMAC 암호화 유형을 사용합니다. AD 신뢰와 함께 RHEL 9 IdM 호스트를 사용하려면 IdM 소프트웨어를 설치하기 전에 AES SHA-1 HMAC 암호화 유형에 대한 지원을 활성화하십시오.

FIPS 컴플라이언스 규정 준수는 기술 및 조직적 계약과 관련된 프로세스이므로 AD-SUPPORT 하위 정책을 활성화하여 기술 측정에서 AES SHA-1 HMAC 암호화 유형을 지원하는 다음 RHEL IdM을 설치하기 전에 FIPS 감사자를 확인하십시오.

```
# update-crypto-policies --set FIPS:AD-SUPPORT
```

(BZ#2057471)

Directory Server는 추천 모드에서 시작될 때 예기치 않게 종료됩니다.

버그로 인해 전역 추천 모드가 Directory Server에서 작동하지 않습니다. refer 옵션을 dirsrv 사용자로 사용하여 ns-slapd 프로세스를 시작하는 경우 Directory Server는 포트 설정을 무시하고 예기치 않게 종료됩니다. 루트 사용자가 SELinux 레이블을 변경하고 나중에 일반 모드에서 서비스가 시작되지 않도록 프로세스를 실행하려고 합니다. 사용 가능한 해결방법이 없습니다.

(BZ#2053204)

Directory Server에서 접미사에 대한 참조를 구성하는 데 실패합니다.

Directory Server에서 백엔드 참조를 설정하는 경우 dsconf <instance_name> 백엔드 접미사 세트를 사용하여 백엔드 상태를 설정하는 작업이 다음 오류와 함께 실패합니다.

```
Error: 103 - 9 - 53 - Server is unwilling to perform - [] - need to set nsslapd-referral before moving to referral state
```

결과적으로 접미사에 대한 참조를 구성하는 데 실패합니다. 문제를 해결하기 위해:

1. **nsslapd-referral 매개변수를 수동으로 설정합니다.**

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com

dn: cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config
```

```
changetype: modify
add: nsslapd-referral
nsslapd-referral: ldap://remote_server:389/dc=example,dc=com
```

2. 백엔드 상태를 설정합니다.

```
# dsconf <instance_name> backend suffix set --state referral
```

이 문제를 해결하려면 접미사에 대한 참조를 구성할 수 있습니다.

(BZ#2063140)

dsconf 유틸리티에는 entryUUID 플러그인에 대한 수정 작업 생성 옵션이 없습니다.

dsconf 유틸리티는 **entryUUID** 플러그인에 대한 수정 작업 생성 옵션을 제공하지 않습니다. 따라서 관리자는 **dsconf** 를 사용하여 기존 항목에 항목**UUID** 특성을 자동으로 추가하는 작업을 생성할 수 없습니다. 이 문제를 해결하려면 작업을 수동으로 생성합니다.

```
# ldapadd -D "cn=Directory Manager" -W -H ldap://server.example.com -x

dn: cn=entryuuid_fixup___<time_stamp__,cn=entryuuid task,cn=tasks,cn=config
objectClass: top
objectClass: extensibleObject
basedn: __<fixup base tree>__
cn: entryuuid_fixup___<time_stamp>__
filter: __<filtered_entry>__
```

작업을 만들고 나면 **Directory Server**에서 누락되거나 잘못된 항목 **UUID** 속성으로 항목을 수정합니다.

(BZ#2047175)

8.14. 데스크탑

RHEL 9로 업그레이드한 후 **Firefox** 애드온은 비활성화되어 있습니다.

RHEL 8에서 **RHEL 9**로 업그레이드하는 경우 이전에 **Firefox**에서 활성화한 모든 애드온이 비활성화됩니다.

문제를 해결하려면 애드온을 수동으로 다시 설치하거나 업데이트합니다. 그 결과 애드온이 예상대로 활성화됩니다.

(BZ#2013247)

RHEL 9로 업그레이드한 후 **VNC**가 실행되고 있지 않습니다.

RHEL 8에서 **RHEL 9**로 업그레이드한 후 이전에 활성화된 경우에도 **VNC** 서버가 시작되지 않습니다.

이 문제를 해결하려면 시스템을 업그레이드한 후 **vncserver** 서비스를 수동으로 활성화합니다.

```
# systemctl enable --now vncserver@:port-number
```

결과적으로 **VNC**가 활성화되고 모든 시스템이 예상대로 부팅된 후 시작됩니다.

(BZ#2060308)

8.15. 그래픽 인프라

기존 **BIOS**의 경우 **VGA**에 출력이 표시되지 않음

다음 시스템 구성을 사용하는 경우 디스플레이에 그래픽 출력이 표시되지 않을 수 있습니다.

- **VGA** 컨트롤러에 연결된 디스플레이
- **UEFI**가 레거시 모드로 전환

따라서 이 구성에 **RHEL**을 사용하거나 설치할 수 없습니다.

문제를 해결하려면 다음 절차를 따르십시오.

1. 시스템을 부트 로더 메뉴로 부팅합니다.

2. **kernel** 명령줄에 **rfca =791** 옵션을 추가합니다.

결과적으로 **RHEL**은 부팅되고 그래픽 출력을 예상대로 표시하지만 최대 해상도는 **16**비트 색상 깊이에서 **1024x768**로 제한됩니다.

(BZ#1960467)

X.org 설정 유틸리티가 **Wayland**에서 작동하지 않음

화면을 조작하는 **X.org** 유틸리티가 **Wayland** 세션에서는 작동하지 않습니다. 특히 **xrandr** 유틸리티는 해상도, 회전 및 레이아웃 처리 방법이 다르기 때문에 **Wayland**에서 작동하지 않습니다.

(JIRA:RHELPLAN-121049)

NVIDIA 드라이버는 **X.org**로 되돌릴 수 있습니다.

특정 조건에서 독점 **NVIDIA** 드라이버는 **Wayland** 디스플레이 프로토콜을 비활성화하고 **X.org** 디스플레이 서버로 되돌립니다.

- **NVIDIA** 드라이버 버전이 **470**보다 낮은 경우
- 시스템이 하이브리드 그래픽을 사용하는 랩탑인 경우
- 필수 **NVIDIA** 드라이버 옵션을 활성화하지 않은 경우

또한 **Wayland**가 활성화되어 있지만, **NVIDIA** 드라이버 버전이 **510** 미만이면 데스크탑 세션은 기본적으로 **X.org**를 사용합니다.

(JIRA:RHELPLAN-119001)

Night Light는 **NVIDIA**를 사용하는 **Wayland**에서 사용할 수 없습니다.

시스템에서 독점적 **NVIDIA** 드라이버가 활성화되면 **GNOME**의 **Night Light** 기능은 **Wayland** 세션에서 사용할 수 없습니다. **NVIDIA** 드라이버는 현재 **Night Light** 을 지원하지 않습니다.

(JIRA:RHELPLAN-119852)

8.16. 웹 콘솔

웹 콘솔을 사용하여 USB 호스트 장치를 제거해도 예상대로 작동하지 않습니다.

USB 장치를 VM(가상 머신)에 연결하면 USB 장치의 장치 번호와 버스 번호가 VM에 전달된 후 변경될 수 있습니다. 결과적으로 웹 콘솔을 사용하여 장치와 버스 번호의 잘못된 상관 관계로 인해 이러한 장치를 제거할 수 없습니다. 이 문제를 해결하려면 VM의 XML 구성에서 USB 장치의 < hostdev > 부분을 제거합니다.

(JIRA:RHELPLAN-109067)

웹 콘솔을 사용하여 여러 호스트 장치를 연결할 수 없습니다.

웹 콘솔을 사용하여 VM(가상 머신)에 연결할 여러 장치를 선택하면 단일 장치만 연결되어 나머지는 무시됩니다. 이 문제를 해결하려면 한 번에 하나의 장치만 첨부합니다.

(JIRA:RHELPLAN-115603)

8.17. 가상화

https를 통한 가상 머신 설치가 실패하는 경우

현재 https 연결을 통해 ISO 소스에서 게스트 운영 체제를 설치하려고 할 때 virt-install 유틸리티가 실패합니다(예: virt-install --cdrom https://example/path/to/image.iso). VM(가상 머신)을 생성하는 대신, 모니터 메시지에 연결하는 동안 설명된 작업은 내부 오류가 발생하여 예기치 않게 종료됩니다.

이 문제를 해결하려면 호스트에 qemu-kvm-block-curl 을 설치하여 https 프로토콜 지원을 활성화하십시오. 또는 다른 연결 프로토콜 또는 다른 설치 소스를 사용합니다.

(BZ#2014229)

가상 머신에서 NVIDIA 드라이버를 사용하면 Wayland가 비활성화됩니다.

현재 NVIDIA 드라이버는 Wayland 그래픽 세션과 호환되지 않습니다. 결과적으로 NVIDIA 드라이버를 사용하는 RHEL 게스트 운영 체제는 Wayland를 자동으로 비활성화하고 대신 Xorg 세션을 로드합니다. 이러한 문제는 주로 다음과 같은 시나리오에서 발생합니다.

- NVIDIA GPU 장치를 RHEL 가상 머신(VM)에 통과하면
- RHEL VM에 NVIDIA vGPU 중재 장치를 할당하는 경우

(JIRA:RHELPLAN-117234)

Milan VM CPU 유형은 AMD Milan 시스템에서 사용할 수 없는 경우가 있습니다.

특정 AMD Milan 시스템에서는 향상된 REP신SB (erms) 및 Fast Short REP RSSB (fsrm) 기능 플래그가 BIOS에서 기본적으로 비활성화되어 있습니다. 결과적으로 이러한 시스템에서 'Milan' CPU 유형을 사용할 수 없습니다. 또한 기능 플래그 설정이 다른 Milan 호스트 간에 VM 실시간 마이그레이션이 실패할 수 있습니다. 이러한 문제를 해결하려면 호스트의 BIOS에서 erms 및 fsrm 을 수동으로 설정합니다.

(BZ#2077767)

가상 머신의 네트워크 트래픽 성능이 저하될 수 있습니다.

경우에 따라 RHEL 9.0 게스트 가상 머신(VM)은 높은 수준의 네트워크 트래픽을 처리할 때 성능이 다소 감소했습니다.

(BZ#1945040)

AVX를 비활성화하면 VM이 재부팅할 수 없게 됩니다.

AMQP(Advanced Vector Extensions) 지원이 있는 CPU를 사용하는 호스트 시스템에서 현재 사용 안 함으로 AVX를 명시적으로 비활성화한 VM을 부팅하려고 하면 VM에서 커널 패닉이 발생합니다.

(BZ#2005173)

장애 조치 virtio NIC에는 Windows 가상 머신의 IP 주소가 할당되지 않습니다.

현재는 장애 조치 virtio NIC로만 VM(Windows 가상 머신)을 시작할 때 VM이 NIC에 IP 주소를 할당하지 못합니다. 결과적으로 NIC는 네트워크 연결을 설정할 수 없습니다. 현재는 해결방법이 없습니다.

(BZ#1969724)

장애 조치 설정이 있는 hostdev 인터페이스는 핫 플러그 연결 해제 후 핫플러그할 수 없습니다.

실행 중인 VM(가상 머신)에서 장애 조치(failover) 구성이 있는 hostdev 네트워크 인터페이스를 제거한 후 현재 동일한 실행 중인 VM에 인터페이스를 다시 연결할 수 없습니다.

(BZ#2052424)

장애 조치 VF가 있는 VM의 실시간 복사 마이그레이션 실패

현재 VM에서 VF(가상 기능) 장애 조치 기능이 활성화된 장치를 사용하는 경우 실행 중인 VM(가상 머신)을 마이그레이션 후 마이그레이션하지 않습니다. 문제를 해결하려면 복사 후 마이그레이션 대신 표준 마이그레이션 유형을 사용합니다.

(BZ#1817965, BZ#1789206)

8.18. 클라우드 환경의 RHEL

SR-IOV는 Azure의 ARM 64 RHEL 9 가상 머신에서 하위 최적화를 수행합니다.

현재 SR-IOV 네트워킹 장치는 Microsoft Azure 플랫폼에서 실행되는 ARM 64 RHEL 9 가상 머신 VM에서 예상보다 대기 시간이 훨씬 더 높습니다.

(BZ#2068432)

콘솔 프록시를 사용하여 XenServer 7의 RHEL 9 VM에서 마우스를 사용할 수 없습니다.

콘솔 프록시가 있는 XenServer 7 플랫폼에서 RHEL 9 VM(가상 머신)을 실행할 때 VM의 GUI에서 마우스를 사용할 수 없습니다. 이 문제를 해결하려면 다음과 같이 VM에서 Wayland 컴포저 프로토콜을 비활성화합니다.

1. /etc/gdm/custom.conf 파일을 엽니다.
2. WaylandEnable=false 행의 주석 처리를 해제합니다.
3. 파일을 저장합니다.

또한 Red Hat은 RHEL VM을 실행하기 위한 플랫폼으로 XenServer를 지원하지 않으며 프로덕션 환경에서 RHEL과 함께 XenServer를 사용하지 않는 것에 유의하십시오.

(BZ#2019593)

Nutanix AHV에서 LVM을 사용하는 RHEL 9 가상 머신 복제 또는 복원으로 인해 루트가 아닌 파티션이 사라집니다.

Nutanix AHV 하이퍼바이저에서 호스팅되는 가상 머신(VM)에서 RHEL 9 게스트 운영 체제를 실행하는 경우 스냅샷에서 VM을 복원하거나 VM 복제를 복제하면 게스트에서 논리 볼륨 관리(LVM)를 사용하는 경우 현재 VM의 루트가 아닌 파티션이 사라집니다. 결과적으로 다음과 같은 문제가 발생합니다.

- 스냅샷에서 VM을 복원하면 VM을 부팅할 수 없으며 대신 긴급 모드로 전환됩니다.
- 복제를 통해 생성된 VM은 부팅할 수 없으며 대신 긴급 모드로 전환됩니다.

이러한 문제를 해결하려면 VM의 긴급 모드에서 다음을 수행합니다.

1. LVM 시스템 장치 파일 제거: rm /etc/lvm/devices/system.devices
2. LVM 장치 설정 재생성: firstimportdevices -a
3. VM 재부팅

이렇게 하면 복제된 VM 또는 복원된 VM이 올바르게 부팅될 수 있습니다.

(BZ#2059545)

Hyper-V 가상 머신에 연결된 네트워크 어댑터의 SR-IOV 기능이 작동하지 않을 수 있습니다.

현재 일부 경우 SR-IOV(Single-root I/O Virtualization)가 활성화된 네트워크 어댑터를 Microsoft Hyper-V 하이퍼바이저에서 실행되는 RHEL 9 가상 머신(VM)에 연결할 때 일부 경우 SR-IOV 기능이 올바르게 작동하지 않습니다.

이 문제를 해결하려면 VM 구성에서 SR-IOV를 비활성화한 다음 다시 활성화합니다.

1. Hyper-V 관리자 창에서 VM을 마우스 오른쪽 버튼으로 클릭합니다.
2. 상황에 맞는 메뉴에서 설정/네트워크 어댑터/하이프롬으로 이동합니다.
3. SR-IOV 활성화를 선택 해제합니다.
4. 적용을 클릭합니다.
5. 1단계와 2단계를 반복하여 Enable SR-IOV 옵션으로 다시 이동합니다.
6. SR-IOV 활성화를 확인합니다.
7. 적용을 클릭합니다.

(BZ#2030922)

ESXi에서 RHEL 9 게스트를 사용자 정의하면 네트워킹 문제가 발생하는 경우가 있습니다.

현재 VMware ESXi 하이퍼바이저에서 RHEL 9 게스트 운영 체제를 사용자 정의해도 NetworkManager 키 파일에서 올바르게 작동하지 않습니다. 결과적으로 게스트가 이러한 키 파일을 사용하는 경우 IP 주소 또는 게이트웨이와 같은 잘못된 네트워크 설정이 있습니다.

자세한 내용 및 해결 방법은 VMware 지식 베이스 를 참조하십시오.

(BZ#2037657)

8.19. 지원 관련 기능

IBM Power Systems에서 보고서를 실행할 때 시간 초과, Little Endian

IBM Power Systems, 수천 개의 **CPU**가 있는 **Little Endian**에서 **sos report** 명령을 실행하는 경우, 프로세서 플러그인은 **/sys/devices/system/cpu** 디렉토리의 거대한 콘텐츠를 수집할 때 **300**초의 기본 타임아웃에 도달합니다. 해결 방법으로 다음과 같이 플러그인의 시간 제한을 늘립니다.

- 일회성 설정의 경우 다음을 실행합니다.

```
# sos report -k processor.timeout=1800
```

- 영구 변경의 경우 **/etc/sos/sos.conf** 파일의 **[plugin_options]** 섹션을 편집합니다.

```
[plugin_options]
# Specify any plugin options and their values here. These options take the form
# plugin_name.option_name = value
#rpm.rpmva = off
processor.timeout = 1800
```

예제 값은 **1800**으로 설정됩니다. 특정 시간 제한 값은 특정 시스템에 따라 크게 달라집니다. 플러그인의 시간 제한을 적절하게 설정하려면 먼저 다음 명령을 실행하여 시간 초과 없이 하나의 플러그인을 수집하는 데 필요한 시간을 추정할 수 있습니다.

```
# time sos report -o processor -k processor.timeout=0 --batch --build
```

(BZ#1869561)

8.20. 컨테이너

베타 GPG 키로 서명된 컨테이너 이미지를 가져올 수 없습니다.

현재 **RHEL 9** 베타 컨테이너 이미지를 가져오려고 하면 **podman** 이 오류 메시지로 종료됩니다. **error:** 소스 이미지가 거부됨 서명은 허용되지 않았습니다. 현재 빌드가 기본적으로 **RHEL Beta GPG** 키를 신뢰하지 않도록 구성되어 있으므로 이미지를 가져오지 못했습니다.

이 문제를 해결하려면 **Red Hat Beta GPG** 키가 로컬 시스템에 저장되었는지 확인하고 해당 베타 네임스페이스에 대해 **podman image trust set** 명령으로 기존 신뢰 범위를 업데이트합니다.

베타 **GPG** 키가 로컬에 저장되어 있지 않은 경우 다음 명령을 실행하여 가져올 수 있습니다.

```
sudo wget -O /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-beta
https://www.redhat.com/security/data/f21541eb.txt
```

베타 GPG 키를 네임스페이스에 신뢰할 수 있는 것으로 추가하려면 다음 명령 중 하나를 사용합니다.

```
$ sudo podman image trust set -f /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-beta
registry.access.redhat.com/namespace
```

및

```
$ sudo podman image trust set -f /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-beta
registry.redhat.io/namespace
```

namespace 를 *ubi9-beta* 또는 *rhel9-beta* 로 바꿉니다.

(BZ#2020026)

Podman은 알 수 없는 기관에서 서명한 컨테이너 "X509: 인증서"를 가져오지 못했습니다.

자체 CA 인증서로 서명된 자체 내부 레지스트리가 있는 경우 해당 인증서를 호스트 머신으로 가져와야 합니다. 그렇지 않으면 오류가 발생합니다.

```
x509: certificate signed by unknown authority
```

호스트에서 CA 인증서를 가져옵니다.

```
# cd /etc/pki/ca-trust/source/anchors/
[anchors]# curl -O <your_certificate>.crt

[anchors]# update-ca-trust
```

그런 다음 내부 레지스트리에서 컨테이너 이미지를 가져올 수 있습니다.

(BZ#2027576)

이전 컨테이너 이미지에서 systemd를 실행해도 작동하지 않음

이전 컨테이너 이미지(예: centos: 7)에서 systemd를 실행하면 작동하지 않습니다.

```
$ podman run --rm -ti centos:7 /usr/lib/systemd/systemd
```

```
Storing signatures
Failed to mount cgroup at /sys/fs/cgroup/systemd: Operation not permitted
[!!!!!!] Failed to mount API filesystems, freezing.
```

이 문제를 해결하려면 다음 명령을 사용하십시오.

```
# mkdir /sys/fs/cgroup/systemd
# mount none -t cgroup -o none,name=systemd /sys/fs/cgroup/systemd
# podman run --runtime /usr/bin/crun --annotation=run.oci.systemd.force_cgroup_v1=/sys/fs/cgroup -
-rm -ti centos:7 /usr/lib/systemd/systemd
```

(JIRA:RHELPLAN-96940)

podman 시스템 연결 추가 및 podman image scp 실패

Podman은 **RSA** 키 교환에 **SHA-1** 해시를 사용합니다. **RSA** 키를 사용하는 시스템 간의 일반적인 **SSH** 연결은 작동하지만 **podman system connection add** 및 **podman image scp** 명령은 **RHEL 9**에서 키 교환에 허용되지 않기 때문에 동일한 **RSA** 키를 사용하여 작동하지 않습니다.

```
$ podman system connection add --identity ~/.ssh/id_rsa test_connection
$REMOTE_SSH_MACHINE
Error: failed to connect: ssh: handshake failed: ssh: unable to authenticate, attempted methods [none
publickey], no supported methods remain
```

이 문제를 해결하려면 **ED25519** 키를 사용하십시오.

1. 원격 시스템에 연결합니다.

```
$ ssh -i ~/.ssh/id_ed25519 $REMOTE_SSH_MACHINE
```

2. **Podman** 서비스의 **ssh** 대상을 기록합니다.

```
$ podman system connection add --identity ~/.ssh/id_ed25519 test_connection
$REMOTE_SSH_MACHINE
```

3. **ssh** 대상이 기록되었는지 확인합니다.

```
$ podman system connection list
```

RHBA-2022:5951 권고가 릴리스되면서 문제가 해결되었습니다.

(JIRA:RHELPLAN-121180)

부록 A. 구성 요소별 티켓 목록

Bugzilla 및 JIRA ID는 참조용으로 이 문서에 나열됩니다. 공개적으로 액세스할 수 있는 Bugzilla 버그에는 티켓에 대한 링크가 포함됩니다.

구성 요소	티켓
389-ds-base	BZ#2024693, BZ#1805717, BZ#1779685, BZ#2053204, BZ#2063140, BZ#2047175
ModemManager	BZ#1996716
NetworkManager	BZ#1980387, BZ#1949127, BZ#2060013, BZ#1931284, BZ#1894877, BZ#2079849
RHCOS	BZ#2008521
WALinuxAgent	BZ#1972101
alsa-lib	BZ#2015863
anaconda	BZ#1951709, BZ#1978264, BZ#2025953, BZ#2009403, BZ#2050140, BZ#1877697, BZ#1914955, BZ#1929105, BZ#19836032, BZ#1997832, BZ#2008792
ansible-collection-microsoft-sql	BZ#2064648, BZ#2064690
ansible-collection-redhat-rhel_mgmt	BZ#2023381
ansible-pcp	BZ#1957566
bash	BZ#2079078
bind	BZ#1984982
binutils	BZ#2030554
boost	BZ#1957950
Chrony	BZ#1961131
clevis	BZ#1956760
cloud-init	BZ#2040090, BZ#2042351

구성 요소	티켓
cmake	BZ#1957948
container-tools	BZ#2000871
containers-common	BZ#2019901
crash	BZ#1896647
createrepo_c	BZ#2055032
crypto-policies	BZ#2004207, BZ#2013195
cyrus-sasl	BZ#1947971, BZ#1995600
device-mapper-multipath	BZ#2017979, BZ#2017592, BZ#2011699
distribution	BZ#1878583
dnf	BZ#2005305, BZ#2073510
dotnet6.0	BZ#1986211
edk2	BZ#1935497
eigen3	BZ#2032423
fapolicyd	BZ#2032408, BZ#1932225, BZ#2054740, BZ#2070655
fence-agents	BZ#1977588
fetchmail	BZ#1999276
fido-device-onboard	BZ#1989930
firefox	BZ#1764205, BZ#2013247
firewalld	BZ#2029211
freeradius	BZ#1978216
gcc	BZ#1986836, BZ#1481850
gdb	BZ#1870029, BZ#1870031
gfs2-utils	BZ#1616432

구성 요소	티켓
gimp	BZ#2047161
Git	BZ#1956345
glibc	BZ#2023422, BZ#2024347
gnome-shell-extension-background-logo	BZ#2057150
gnome-shell-extensions	BZ#2031186
gnupg2	BZ#2070722, BZ#2073567
gnutls	BZ#2033220, BZ#1999639
golang	BZ#2014087, BZ#1984110
grafana-pcp	BZ#1993156, BZ#1845592
grafana	BZ#1993215
grub2	BZ#2026579
grubby	BZ#1969362
hostapd	BZ#2019830
ipa	BZ#1952028, BZ#1957736, BZ#1966101,BZ#1988383,BZ#2084180,BZ#2084166,BZ#2057471
iptables	BZ#1945151
javapackages-tools	BZ#1951482
jigawatts	BZ#1972029
jmc-core	BZ#1980981
kdump-anaconda-addon	BZ#1894783, BZ#2017401
kernel-rt	BZ#2002474

구성 요소	티켓
kernel	BZ#1844416, BZ#1851933, BZ#1780258, BZ#1874195, BZ#1953515,BZ#1960556, BZ#1948340, BZ#1952863, BZ#1978382, BZ#1957818,BZ#2002499, BZ#2050415, BZ#1951951, BZ#1949613, BZ#2036856, BZ#2034490, BZ#1943423, BZ #2046472, BZ#2068432, BZ#1997541, BZ#1613522, BZ#1874182, BZ#1995338, BZ#1995338 BZ#1570255, BZ#2023416, BZ#2021672, BZ#2019593, BZ#2000616, BZ#2013650, BZ#2033080, BZ#2069758, BZ#2059545, BZ#2030922, BZ#1940 , BZ# 2073541, BZ#2073541, BZ#1960467, BZ#2005173
kexec-tools	BZ#1988894, BZ#1895232, BZ#1958452, BZ#2065013
kmod	BZ#1985100
krb5	BZ#2060798, BZ#2068935, BZ#2077450
libburn	BZ#2015861
libcap	BZ#2037215
libgcrypt	BZ#1990059
libmodulemd	BZ#1984403
libreswan	BZ#2017355, BZ#2039877
libseccomp	BZ#2019887
libservicelog	BZ#1869568
libvirt	BZ#2014487
libxcrypt	BZ#2034569
llvm-toolset	BZ#2001107
lorax-templates-rhel	BZ#1961092
lsvpd	BZ#1869564
lvm2	BZ#1899214, BZ#1749513, BZ#2038183
mariadb	BZ#1971248
mod_security_crs	BZ#1947962

구성 요소	티켓
nettle	BZ#1986712
nfs-utils	BZ#2059245
nginx	BZ#1953639
nmstate	BZ#1969941
nodejs	BZ#1953491
nss	BZ#2008320, BZ#2099438
numatop	BZ#1874125
nvml	BZ#1874208
opal-prd	BZ#1869560
open-vm-tools	BZ#2037657
opencryptoki	BZ#1869533
openscap	BZ#2041782
openssh	BZ#1952957, BZ#2002734, BZ#1821501, BZ#2087121
openssl	BZ#1990814, BZ#1871147, BZ#1970388, BZ#1975836, BZ#1681178, BZ#1685470, BZ#2053289, BZ#2087253, BZ#2060044, BZ#2071631
osbuild-composer	BZ#2060575
oscap-anaconda-addon	BZ#1893753
ostree	BZ#1961254
p11-kit	BZ#1966680
pacemaker	BZ#1850145, BZ#1443666, BZ#1470834, BZ#1082146, BZ#1376538, BZ#1975388
pcp	BZ#1991764, BZ#1847808, BZ#1981223
pcs	BZ#1290830, BZ#1909901, BZ#1872378, BZ#2018969, BZ#1996067
perl-Module-Signature	BZ#2039361

구성 요소	티켓
php	BZ#1949319
pki-core	BZ#2084181
podman	JIRA:RHELPLAN-77549, JIRA:RHELPLAN-75322, JIRA:RHELPLAN-108830, BZ#2027576
powerpc-utils	BZ#1873868
ppc64-diag	BZ#1869567
python-jsonpointer	BZ#1980256
python-podman	BZ#1975462
qemu-kvm	BZ#1940132, BZ#1939509, JIRA:RHELPLAN-75866, BZ#1874187, BZ#1965079,BZ#1951814,BZ#2014229,BZ#2052424,BZ#1817965
Redis	BZ#1959756
rhel-system-roles	BZ#1993304,BZ#1993377,BZ#2022461,BZ#1978488,BZ#1984583,BZ#2016517,BZ#2021667,BZ#1986460,BZ#1978752, BZ#1978753,BZ#1990490, BZ#2031555, BZ#2016518,BZ#2054364,BZ#1978773,BZ#2054435,BZ#1999162,BZ#2057657, BZ#2012298,BZ#2021028,BZ#2054367,BZ#2054369,BZ#2054369 , BZ#2021665, BZ#2029427,BZ#2004899,BZ#1958964, BZ#1978734,BZ#1978760,BZ#2039106,BZ#2041632,BZ#2058777, BZ#2058645, BZ#2058756,BZ#2071804,BZ#2029634, , BZ#2044408,BZ#2029602,BZ#2038957,BZ#2064391,BZ#2004303, BZ#2006230,BZ#2057164,BZ#2021025, . BZ#2021676,BZ#2047506, BZ#2050341, BZ#2050419, BZ#1999770
rpm-ostree	BZ#1961324
rpm	BZ#1942549, BZ#1962234
rsyslog	BZ#2027971, BZ#1992155
rust-toolset	BZ#2002885
s390utils	BZ#1932480
samba	BZ#2013578
scap-security-guide	BZ#2028435, BZ#2014561, BZ#2045341, BZ#2038978

구성 요소	티켓
selinux-policy	BZ#2055822, BZ#1932752, BZ#2021529, BZ#2064274
shadow-utils	BZ#1859252
sos	BZ#2011537, BZ#1869561
squid	BZ#1990517
sssd	BZ#1949149, BZ#2014249, BZ#1879869, BZ#1737489
strace	BZ#2038965
stratisd	BZ#2041558
stunnel	BZ#2039299
subscription-manager	BZ#1898563, BZ#2049441
sudo	BZ#1981278
swig	BZ#1943580
systemd	BZ#2018112
systemtap	BZ#2083727
tigervnc	BZ#2060308
trace-cmd	BZ#1933980
tuned	BZ#2003838
unbound	BZ#2070495
usbguard	BZ#1986785, BZ#2009226
varnish	BZ#1984185
virt-manager	BZ#1995131
virt-who	BZ#2008215, BZ#2054504
virtio-win	BZ#1969724

구성 요소	티켓
wpa_supplicaant	BZ#1975718
기타	BZ#2077836, BZ#2019806, BZ#1937651,BZ#2010291, BZ#1941810, BZ#1941810, BZ#1941595, JIRA:RHELPLAN-80758, JIRA:RHELPLAN-80759, JIRA:RHELPLAN-82578, JIRA:RHELPLAN-68364, JIRA:RHELPLAN-68364 JIRA:RHELPLAN-78673, JIRA:RHELPLAN-78675, BZ#1940863, BZ#2079313, JIRA:RHELPLAN-100497, BZ#2068532, BZ#2089193, JIRA:RHELPLAN-102009, BZ#2065646, BZ#2088414, JIRA:RHELPLAN-80734, BZ#2013853, JIRA:RHELPLAN-103540, BZ#2019341,BZ#2008558,BZ#2008575,BZ#2009455, JIRA:RHELPLAN-74542, JIRA:RHELPLAN-73678, JIRA:RHELPLAN-84168, JIRA:RHELPLAN-73697, JIRA:RHELPLAN-95126, BZ# 20875 , JIRA:RHELPLAN-97899, JIRA:RHELPLAN-100359, JIRA:RHELPLAN-103147, JIRA:RHELPLAN-103147, JIRA:RHELPLAN-103147, JIRA:RHELPLAN-103147, JIRA:RHELPLAN-103146 JIRA:RHELPLAN-79161, BZ#2046325, BZ#2046325 , BZ#2021262, JIRA:RHELPLAN-64576, JIRA:RHELPLAN-65223, BZ#2083036,BZ#2011448,BZ2019318, JIRA:RHELPLAN-101240, JIRA:RHELPLAN-101241, JIRA:RHELPLAN-101242, JIRA:RHELPLAN-101246, JIRA:RHELPLAN-101247, JIRA:RHELPLAN-102552, JIRA:RHELPLAN-99892, BZ#2027596, JIRA:RHELPLAN-119000, BZ#1940653, JIRA:RHELPLAN-101246, JIRA:RHELPLAN-99892, JIRA:RHELPLAN-119000, JIRA:RHELPLAN-101246, JIRA BZ#2054401 JIRA:RHELPLAN-113994, BZ#2059183, JIRA:RHELPLAN-74543, JIRA:RHELPLAN-99889, JIRA:RHELPLAN-99890, JIRA:RHELPLAN-100032, JIRA:RHELPLAN-100034, JIRA:RHELPLAN-100034, JIRA:RHELPLAN-101111, JIRA:RHELPLAN-1011, JIRA:RHELPLAN-100034, JIRA JIRA:RHELPLAN-100020, BZ#2069501, BZ#2070506, JIRA:RHELPLAN-117903, JIRA:RHELPLAN-98617, JIRA:RHELPLAN-103855, BZ#2091653,BZ#2082306, JIRA:RHELPLAN-65217, JIRA:RHELPLAN-65217, JIRA BZ#2020529,BZ#2030412, BZ#2046653, JIRA:RHELPLAN-103993, JIRA:RHELPLAN-122345, BZ#1927780, BZ#1927780, BZ#1935763, BZ#1935544, BZ#2089200, JIRA:RHELPLAN-15509, JIRA:RHELPLAN-15509, JIRA:RHELPLAN-99136, JIRA:RHELPLAN-103232, BZ#1899167, BZ#1979521, JIRA:RHELPLANPLAN-100087, JIRA:RHELPLAN-100639, JIRA:RHELPLAN-10304, BZ#2058153, JIRA:RHELPLAN-113995, JIRA:RHELPLAN-113995, JIRA:RHELPLAN-121048, JIRA:RHELPLAN-98983, BZ#1640697, BZ#1697896, BZ#207896, BZ# 2047713, JIRA:RHELPLAN-109067, JIRA:RHELPLAN-115603, JIRA:RHELPLAN-96940, JIRA:RHELPLAN-96940, JIRA:RHELPLAN-96940, JIRA JIRA:RHELPLAN-11PPC, JIRA:RHELPLAN-119852, BZ#2077767, BZ#2053598, JIRA:RHELPLAN-121180, BZ#2082303, JIRA:RHELPLAN-121049

부록 B. 감사 인사

RHEL 9 준비 챌린지의 일부로 피드백을 제공하는 아래의 Red Hat Associates에게 감사드립니다.

- Buland Singh
- Pradeep Jagtap
- Omkar Andhekar
- Ju keys
- Suresh Jagtap
- Prijesh Patel
- Nikhil euwanshi
- Amit Yadav
- Pranav Lawate
- John Pittman

부록 C. 리버전 내역

0.1-0

2017년 8월 22일mailto:lspackova@redhat.com

- 더 이상 사용되지 않는 기능 **BZ#2069279** 및 **BZ#2106816** **(Containers)**이 추가되었습니다.
- **z-stream** 수정(컨테이너)에 대한 정보가 포함된 **JIRA-RHELPLAN-121180** 업데이트

0.0-9

2022년 8월 10일mailto:lspackova@redhat.com

- **BZ#1991500** **(Dynamic** 프로그래밍 언어, 웹 및 데이터베이스 서버)의 알려진 문제가 추가되었습니다.

0.0-8

Thu Aug 4, 2022, Gabriela Fialov] (gfialova@redhat.com)

- **JIRA-RHELPLAN-118914** 기능 개선 사항.
- **BZ#2105162** (보안)에 알려진 문제가 추가되었습니다.
- 알려진 문제 **BZ#1960467** **(sst_gpu_rhel_9, Marek Suchanek, msuchane@redhat.com)**이 추가되었습니다.

0.0-7

2017년 7월 28일mailto:lspackova@redhat.com

- 개선 사항 **BZ#2099438** (보안)이 추가되었습니다.

- BZ#2087253 (Security)에 알려진 문제가 추가되었습니다.
- 배포의 Application Streams에 대한 확장된 정보입니다.

0.0-6

2022년 7월 11일, 2017년 3월11일

- 알려진 문제 BZ#2077450.
- 개선 사항 BZ#2091653.
- 버그 수정 BZ#2006230 을 추가했습니다.

0.0-5

2022년 6월 29일, 2017년 3월15일

- 알려진 문제 BZ#2087121,BZ#2073567,BZ#2083727, 및 BZ#2005173.
- BZ#2091643 개선 사항이 추가되었습니다.

0.0-4

2022년 6월 1일 Gabriela Fialov Korea (mailto:gfialova@redhat.com)

- BZ#2027576 알려진 문제가 추가되었습니다.

0.0-3

Tue 2022년 5월 24일 Gabriela FialovSample (gfialova@redhat.com)

- Reefadias의 이메일에 따라 10 개 주요 고객 포털 랩 목록을 업데이트

- 추가 및 게시되지 않는 기능 **BZ2089200**

0.0-2

Wed 2022년 5월 18일 Gabriela Fialovmake (gfialova@redhat.com)

- **Red Hat Enterprise Linux 9.0 릴리스 노트.**

0.0-1

Wed November 03, 2021년 11월 03일, Lenka hierakovoctets(참조하십시오.)

- **Red Hat Enterprise Linux 9.0 베타 릴리스 노트.**