



사용자 가이드

Amazon Linux 2023



Amazon Linux 2023: 사용자 가이드

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 브랜드 디자인은 Amazon 외 제품 또는 서비스와 더불어 Amazon 브랜드 이미지를 실추시키거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 해당 상표 소유자의 자산이며 해당 상표의 소유자가 Amazon의 계열사이거나, Amazon과 제휴 관계에 있거나, Amazon의 후원을 받는 업체일 수 있습니다.

Table of Contents

Amazon Linux 2023이란 무엇입니까?	1
릴리스 케이던스	1
메이저 및 마이너 릴리스	3
새 릴리스 사용	4
장기 지원 정책	4
이름 지정 및 버전 관리	4
성능 및 운영 최적화	5
Fedora와의 관계	7
사용자 지정 cloud-init	7
보안 업데이트 및 기능	8
관리 업데이트	9
클라우드 보안	9
SELinux 모드	9
규정 준수 프로그램	9
SSH 서버 기본값	9
OpenSSL 3의 주요 기능	9
네트워킹 서비스	10
코어 튜체인 패키지 glibc, gcc 및 binutils	11
패키지 관리 도구	11
SSH 서버 기본 구성	12
더 이상 사용되지 않는 기능	14
compat- 패키지	14
더 이상 사용되지 않는 기능이 AL1에서 중단되고 AL2에서 제거됨	14
32비트 x86 (i686) AMI	15
aws-apitools-*로 대체됨 AWS CLI	15
systemdupstartAL2에서 대체합니다.	16
AL2에서 더 이상 사용되지 않고 AL2023 버전에서 제거된 기능	16
32비트 x86 (i686) 패키지	17
aws-apitools-*로 대체됨 AWS CLI	17
bzip2개정 관리 시스템	18
cgroup v1	18
log4j핫패치 () log4j-cve-2021-44228-hotpatch	18
lsb_release 및 system-lsb-core 패키지	18
mccrypt	19

OpenJDK (7) java-1.7.0-openjdk	19
Python 2.7	19
rsyslog-openssl대체 rsyslog-gnutls	20
네트워크 정보 서비스 (NIS)/yp	20
AL2023 버전에서는 더 이상 사용되지 않음	20
32비트 x86 (i686) 런타임 지원	21
버클리 DB () libdb	21
cron	21
IMDSv1	21
pcre 버전 1	22
System V init (sysvinit)	22
AL2와 AL2023 비교	23
패키지 추가, 업그레이드 및 삭제	24
각 릴리스에 대한 지원	24
이름 지정 및 버전 변경	24
최적화	24
Python 2.7은 Python 3으로 대체되었습니다.	25
보안 업데이트	25
SELinux	25
OpenSSL 3	25
IMDSv2	26
log4j 핫패치 제거 (log4j-cve-2021-44228-hotpatch)	26
안정성을 위한 결정적 업그레이드	27
다수 업스트림 제공	27
AMI 루트 파일 시스템 및 기본 Amazon EBS 볼륨 유형	27
네트워킹 시스템 서비스	28
통합 제어 그룹 계층 구조 (cgroup v2)	28
작업 예약	28
glibc, gcc 및 binutils 패키지	28
패키지 관리자	29
로깅 시스템	29
curl 및 libcurl 패키지 변경	29
GNU 프라이버시 가드(GNUPG)	30
기본 JVM으로 Amazon Corretto 설정	30
AWS CLI v2	30
UEFI 기본	30

SSH 서버 기본 구성 변경	30
Extra Packages for Enterprise Linux (EPEL)	31
cloud-init 사용하기	31
데스크톱 그래픽 지원	31
컴파일러 트리플렛	32
32비트 x86 (i686) 패키지	32
lsb_release 및 system-lsb-core 패키지	32
AL2에서 AL2023 커널이 변경되었습니다.	32
보안 중심으로 커널 구성 변경	33
기타 커널 구성 변경	37
커널 파일 시스템 지원	38
Amazon Linux 2 및 AL2023 AMI 비교	43
Amazon Linux 2 및 AL2023 미니멀 AMI 비교	76
Amazon Linux 2 및 AL2023 컨테이너 비교	97
AL1과 AL2023 비교	105
각 릴리스에 대한 지원	105
systemd는 upstart를 init 시스템으로 대체합니다.	105
Python 2.6 및 2.7은 Python 3으로 대체되었습니다.	106
OpenJDK 8은 가장 오래된 JDK	106
AL1에서 AL2023 커널이 변경되었습니다.	106
커널 라이브 패치	106
커널 파일 시스템 지원	106
보안에 중점을 둔 커널 구성 변경	108
기타 커널 구성 변경	110
AL1과 AL2023 AMI 비교	111
AL1과 AL2023 미니멀 AMI 비교	145
AL1과 AL2023 컨테이너 비교	165
시스템 사양	174
AL2023 실행을 위한 CPU 요구 사항	174
AL2023 ARM CPU 사양	174
AL2023 x86-64 CPU 사양	175
AL2023 실행을 위한 메모리 (RAM) 요구 사항	175
AL2023 켜기 사용 AWS	177
시작하기 AWS	177
등록해 보세요 AWS 계정	177
관리자 액세스 권한이 있는 사용자 생성	178

프로그래밍 방식 액세스 권한 부여	179
Amazon EC2 AL2023	180
아마존 EC2 콘솔을 사용하여 AL2023 실행	181
SSM 파라미터를 사용하여 AL2023 실행 및 AWS CLI	182
를 사용하여 최신 AL2023 AMI 실행 AWS CloudFormation	183
특정 AMI ID를 사용하여 AL2023 실행	185
AL2023 AMI 지원 중단 및 수명 주기	185
AL2023 인스턴스에 연결	186
AL2023 표준(기본) AMI와 미니멀 AMI 비교	186
AL2023 컨테이너	214
AL2023 기본 컨테이너 이미지	214
AL2023 최소 컨테이너 이미지	216
기본적인 AL2023 컨테이너 이미지 구축	218
AL2023 컨테이너 이미지 패키지 목록 비교	222
AL2023 미니멀 AMI와 컨테이너 이미지 비교	228
AL2023 Elastic Beanstalk	244
AL2023 CloudShell	245
AL2023 Amazon ECS 컨테이너 호스트	245
AL2 이후 Amazon ECS 관련 변경 사항	246
Amazon ECS 최적화 AMI 사용자 지정	246
AL2023 기반 Amazon EFS	247
amazon-efs-utils	247
Amazon EFS 파일 시스템 마운트	247
AL2023 Amazon EMR	248
AL2023 기반 아마존 EMR 릴리스	248
EKS 기반 Amazon EMR에 AL2023 지원	248
AL2023 커짐 AWS Lambda	248
provided.al2023람다 런타임	248
AL2023 기반 런타임	249
튜토리얼	250
AL2023 시스템에 램프를 설치하세요	250
1단계: LAMP 서버 준비	251
2단계: LAMP 서버 테스트	255
3단계: 데이터베이스 서버 보안 설정	257
4단계: (선택 사항) 설치 phpMyAdmin	258
문제 해결	261

관련 주제	261
AL2023 기반 SSL/TLS 구성	262
필수 조건	263
1단계: 서버에서 TLS 활성화	264
2단계: CA가 서명한 인증서 가져오기	267
3단계: 보안 구성 테스트 및 강화	274
문제 해결	278
AL2023 WordPress 블로그를 호스팅하세요	278
필수 조건	279
설치 WordPress	280
다음 단계	290
도움말! 내 퍼블릭 DNS 이름이 변경되어 블로그를 사용할 수 없습니다.	291
Amazon EC2 외부 AL2023	293
AL2023 VM 이미지 다운로드	293
지원되는 구성	293
KVM 조건	294
VMware 요구 사항	296
하이퍼-V 요구 사항	298
AL2023 VM 구성	300
NoCloud seed.iso 기반 구성	301
VMware 게스트 정보 기반 구성	304
표준 AMI 및 KVM 이미지에 대한 AL2023 패키지 목록 비교	306
표준 AMI와 VMware OVA 이미지에 대한 AL2023 패키지 목록 비교	330
표준 AMI와 하이퍼-V 이미지에 대한 AL2023 패키지 목록 비교	356
AL2023 업데이트	381
새 업데이트에 대한 알림 받기	381
업데이트 관리	382
사용 가능한 패키지 업데이트 확인	382
DNF 및 리포지토리 버전에서 보안 업데이트	384
(보안) 업데이트 후 자동 서비스 재시작	386
최신 리포지토리 버전이 활성화된 상태로 인스턴스 시작	388
패키지 지원 정보 가져오기	388
최신 리포지토리 버전 확인	389
새 리포지토리 추가, 사용 또는 사용 중지	392
cloud-init를 사용하여 리포지토리 추가	394
AL2023 버전 지정 저장소를 통한 결정적 업그레이드 사용	395

메이저 및 마이너 릴리스에 포함된 업데이트를 관리하세요	395
메이저 버전과 마이너 버전 업그레이드의 차이점은 다음과 같습니다.	396
AL2023 리포지토리에서 사용할 수 있는 패키지 업데이트를 제어하십시오.	396
버전이 지정된 리포지토리를 통한 결정적 업그레이드	397
커널 라이브 패치	403
제한 사항	403
지원되는 구성 및 필수 조건	404
커널 라이브 패치 작업	404
프로그래밍 언어 및 런타임	410
C/C++ 및 포트란	410
Go	411
AL2023 람다 함수: Go	412
Java	412
Perl	412
Perl 모듈	412
PHP	413
PHP 최신 버전으로 마이그레이션	413
PHP 7.x에서 마이그레이션	413
PHP 모듈	414
Python	414
Python 모듈	414
Rust	414
AL2023 람다 함수: Rust	415
보안 및 규정 준수	416
보안 권고	416
ALAS 공지	417
ALAS FAQ	417
AL2023 SELinux 모드 설정	417
AL2023 기본 SELinux 상태 및 모드	418
enforcing 모드로 변경	419
SELinux 비활성화 옵션	420
AL2023 버전에서 FIPS 모드 활성화	421
커널 강화	423
커널 강화 옵션 (독립 아키텍처)	423
x86-64 전용 커널 강화 옵션	435
aarch64 전용 커널 강화 옵션	438

AL2023 기반 UEFI 보안 부팅	439
AL2023 기반 UEFI 보안 부팅 활성화	440
기존 인스턴스 등록	440
스냅샷 이미지 등록	441
폐기 업데이트	442
UEFI 보안 부팅이 AL2023 환경에서 작동하는 방식	442
자체 키 등록	442
.....	cdxliv

Amazon Linux 2023이란 무엇입니까?

아마존 리눅스 2023 (AL2023) 는 아마존 웹 서비스 (AWS) 의 차세대 아마존 리눅스입니다. AL2023 를 사용하면 안전하고 안정적인 고성능 런타임 환경에서 클라우드 및 엔터프라이즈 애플리케이션을 개발하고 실행할 수 있습니다. 또한 Linux의 최신 혁신 기술에 액세스하여 장기 지원을 제공하는 애플리케이션 환경을 이용할 수 있습니다. AL2023은 무료입니다.

AL2023 은 아마존 리눅스 2 (AL2) 의 후속 버전입니다. AL2023 및 AL2 간의 차이점에 대한 자세한 내용은 [AL2023 패키지 변경 사항을](#) 참조하십시오 [AL2와 AL2023 비교](#).

주제

- [릴리스 케이던스](#)
- [이름 지정 및 버전 관리](#)
- [성능 및 운영 최적화](#)
- [Fedora와의 관계](#)
- [사용자 지정 cloud-init](#)
- [보안 업데이트 및 기능](#)
- [네트워킹 서비스](#)
- [코어 틀체인 패키지 glibc, gcc 및 binutils](#)
- [패키지 관리 도구](#)
- [SSH 서버 기본 구성](#)

릴리스 케이던스

새 Amazon Linux 메이저 버전은 2년마다 출시되며 5년간의 지원입니다. 각 릴리스는 2단계로 지원됩니다. 기본 지원은 처음 2년간 적용됩니다. 그 다음, 유지 관리는 3년간 지원됩니다.

기본 지원은 분기별로 릴리스 마이너 버전 업데이트를 제공합니다. 유지 관리 기간에 릴리스에 보안 업데이트와 중요 버그 수정만 포함되며 이러한 업데이트가 있으면 즉시 제공합니다.

연도	Amazon Linux 2023	Amazon Linux 2025	Amazon Linux 2027	Amazon Linux 2028
2023년	 기본 지원			
2024년	 기본 지원			
2025년	유지 관리	 기본 지원		
2026년	유지 관리	 기본 지원		
2027년	유지 관리	유지 관리	 기본 지원	
2028년	 EOL	유지 관리	 기본 지원	

연도	Amazon Linux 2023	Amazon Linux 2025	Amazon Linux 2027	Amazon Linux 2028
2029년	 EOL	유지 관리	유지 관리	 기본 지원
2030년	 EOL	 EOL	유지 관리	 기본 지원
2031년	 EOL	 EOL	유지 관리	유지 관리

메이저 및 마이너 릴리스

Amazon Linux(메이저 버전, 마이너 버전 또는 보안 릴리스)를 릴리스 할 때마다 새로운 Linux Amazon Machine Image(AMI)를 릴리스합니다.

- 메이저 버전 출시 — 스택 전반의 보안 및 성능에 대한 새로운 기능과 향상된 기능이 포함되어 있습니다. 개선 사항에는 커널, 툴체인 Glib C. OpenSSL. 기타 시스템 라이브러리 및 유틸리티의 주요 변경 사항이 포함될 수 있습니다. Amazon Linux 메이저 릴리스는 부분적으로 업스트림 Fedora Linux 배포판을 기반으로 합니다. AWS는 Fedora 아닌 특정 업스트림 패키지를 추가하거나 교체할 수 있습니다.
- 마이너 버전 릴리스 - 보안 업데이트, 버그 수정, 새로운 기능 및 패키지가 포함된 분기별 업데이트입니다. 각 마이너 버전은 새로운 기능 및 패키지 외에도 보안 및 버그 수정이 누적된 업데이트 목록입니다. 이러한 릴리스에는 PHP 같은 최신 언어 런타임이 포함될 수 있습니다. Ansible 및 Docker와 같은 유명 소프트웨어 패키지도 포함될 수 있습니다.

새 릴리스 사용

업데이트는 새로운 Amazon Machine Image(AMI) 릴리스 및 이에 해당하는 새로운 리포지토리 조합으로 제공됩니다. 기본적으로 새 AMI와 해당 리포지토리가 한 쌍이 됩니다. 하지만 시간이 지나면 최신 리포지토리 버전에 Amazon EC2 인스턴스를 실행하여 업데이트할 수 있습니다. 최신 AMI에 새 인스턴스를 실행하여 업데이트할 수도 있습니다.

장기 지원 정책

Amazon Linux는 모든 패키지에 대한 업데이트를 제공하고 Amazon Linux를 기반으로 구축된 애플리케이션 메이저 버전은 호환성을 유지합니다. glibc 라이브러리, OpenSSL, OpenSSH 및 DNF 패키지 관리자와 같은 핵심 패키지는 주요 AL2023 릴리스를 사용하는 동안 지원합니다. 코어 패키지에 포함되지 않은 패키지는 특정 업스트림 소스 기반에서 지원합니다. 다음 명령을 실행하여 개별 패키지의 구체적인 지원 상태 및 날짜를 확인할 수 있습니다.

```
$ sudo dnf supportinfo --pkg packagename
```

다음 명령을 실행하여 현재 설치한 패키지 정보를 알 수 있습니다.

```
$ sudo dnf supportinfo --show installed
```

핵심 패키지 전체 목록은 미리 보기 중에 확정됩니다. 핵심 패키지에 포함된 패키지를 더 보고 싶다면 문의해 주세요. 피드백을 수집하여 평가합니다. AL2023 관련 피드백은 지정 AWS 담당자가 제공하거나 또는 GitHub의 [amazon-linux-2023 리포지토리](#)에 제출된 이슈로 받아볼 수 있습니다.

이름 지정 및 버전 관리

AL2023은 2년간의 표준 지원 기간 동안 3개월마다 마이너 릴리즈를 제공합니다. 각 릴리스는 0에서 N까지 증가하여 식별됩니다. 0은 해당 반복의 주요 릴리스를 나타냅니다. 모든 릴리스의 이름은 Amazon Linux 2023입니다. Amazon Linux 2025가 릴리스되면 AL2023 지원이 확대되고 보안 업데이트 및 중요한 버그 수정에 대한 업데이트를 받을 수 있습니다.

예를 들어, AL2023 마이너 릴리스의 형식은 다음과 같습니다.

- 2023.0.20230301
- 2023.1.20230601
- 2023.2.20230901

해당 AL2023 AMI는 다음 형식을 취합니다.

- al2023-ami-2023.0.20230301.0-kernel-6.1-x86_64
- al2023-ami-2023.1.20230601.0-kernel-6.1-x86_64
- al2023-ami-2023.2.20230901.0-kernel-6.1-x86_64

특정 마이너 버전의 정규 AMI 릴리스는 AMI 릴리스 날짜에 발생합니다.

- al2023-ami-2023.0.**20230301**.0-kernel-6.1-x86_64
- al2023-ami-2023.0.**20230410**.0-kernel-6.1-x86_64
- al2023-ami-2023.0.**20230520**.0-kernel-6.1-x86_64

AL2 또는 AL2023 인스턴스를 식별하는 권장 방법은 에서 공통 플랫폼 열거 (CPE) 문자열을 읽는 것부터 시작합니다. `/etc/system-release-cpe` 그런 다음 문자열을 해당 필드로 분할합니다. 마지막으로 플랫폼 및 버전 값을 읽어보세요.

또한 AL2023에 플랫폼 식별을 위한 새 파일이 도입되었습니다.

- `/etc/system-release`에 대한 심볼릭 링크 `/etc/amazon-linux-release`
- `/etc/system-release-cpe`에 대한 심볼릭 링크 `/etc/amazon-linux-release-cpe`

이 두 파일은 인스턴스가 Amazon Linux임을 나타냅니다. 특정 플랫폼 및 버전 값을 알고 싶지 않다면 파일을 읽거나 문자열을 필드로 분할하지 않아도 됩니다.

성능 및 운영 최적화

Amazon Linux 6.1 커널

- AL2023 은 엘라스틱 네트워크 어댑터 (ENA) 및 EFA (엘라스틱 패브릭 어댑터) 디바이스용 최신 드라이버를 사용합니다. AL2023 은 Amazon EC2 인프라의 하드웨어를 위한 성능 및 기능 백포트에 중점을 둡니다.
- x86_64 및 aarch64 인스턴스 유형에 커널 라이브 패치를 사용할 수 있습니다. 이렇게 하면 자주 재부팅하지 않아도 됩니다.
- 모든 커널 빌드 및 런타임 구성에는 AL2와 동일한 성능 및 운영 최적화가 많이 포함됩니다.

기본 튜체인 선택 및 기본 빌드 플래그

- AL2023 패키지는 컴파일러 최적화 () 가 기본적으로 활성화된 상태로 빌드됩니다. -O2
- AL2023 패키지를 빌드하려면 x86-64 시스템(-march=x86-64-v2)에 x86-64v2, aarch64 (-march=armv8.2-a+crypto -mtune=neoverse-n1)에 Graviton 2 이상 버전이 필요합니다.
- AL2023 패키지는 자동 벡터화가 활성화된 상태로 빌드됩니다(-ftree-vectorize).
- AL2023 패키지는 링크 시간 최적화(LTO)가 활성화된 상태로 빌드됩니다.
- AL2023은 업데이트된 Rust, Clang/LLVM, 및 Go 버전을 사용합니다.

패키지 선택 및 버전

- 주요 시스템 구성 요소에 대한 일부 백포트에는 Amazon EC2 인프라, 특히 Graviton 인스턴스에서 실행하기 위한 몇 가지 성능 개선 사항이 포함되어 있습니다.
- AL2023 는 여러 가지 AWS 서비스 및 기능과 통합되어 있습니다. 여기에는 AWS CLI, SSM 에이전트, Amazon Kinesis 에이전트 등이 포함됩니다. CloudFormation
- AL2023에서 Amazon Corretto를 Java 개발 키트(JDK)로 사용할 수 있습니다.
- AL2023은 업스트림 프로젝트에서 최신 버전이 출시될 때 데이터베이스 엔진 및 프로그래밍 언어 런타임 업데이트를 제공합니다. 출시되면 최신 버전이 포함된 프로그래밍 언어 런타임이 추가됩니다.

클라우드 환경 배포

- 기본 AL2023 AMI 및 컨테이너 이미지는 패치 인스턴스 교체를 위해 자주 업데이트됩니다.
- 커널 업데이트는 AL2023 AMI 업데이트에 포함되어 있습니다. 즉, 커널을 업데이트하기 위해 yum update 및 reboot 같은 명령을 사용하지 않아도 됩니다.
- 표준 AL2023 AMI 외에도 미니멀 AMI 및 컨테이너 이미지도 사용할 수 있습니다. 서비스를 실행하는데 필요한 최소한의 패키지로 환경을 실행하려면 미니멀 AMI를 선택하세요.
- 기본적으로 AL2023 AMI 및 컨테이너는 패키지 리포지토리의 특정 버전에서만 실행됩니다. 실행 시 자동 업데이트가 되지 않습니다. 즉, 패키지 업데이트 시기를 관리할 수 있습니다. 프로덕션으로 출시하기 전에 언제든지 베타/감마 환경에서 테스트할 수 있습니다. 문제가 있는 경우 사전 검증된 롤백 경로를 이용할 수 있습니다.

Fedora와의 관계

AL2023은 Fedora와 독립적으로 자체 릴리스 및 지원 수명 주기를 유지합니다. AL2023은 오픈 소스 소프트웨어 업데이트, 더욱 다양한 패키지, 신속한 릴리스를 제공합니다. 이렇게 하면 익숙한 RPM 기반 운영 체제를 유지할 수 있습니다.

정식 출시된 AL2023 (GA) 버전은 특정 Fedora 릴리스와 직접 비교할 수 없습니다. AL2023 GA 버전에는 Fedora 34, 35, 36의 구성 요소가 포함되어 있습니다. 일부 구성 요소는 Fedora 구성 요소와 동일하며 일부는 수정되었습니다. 다른 구성 요소는 CentOS 9 Streams의 구성 요소와 비슷하거나 독립적으로 개발되었습니다. Amazon Linux 커널은 Fedora와는 별개로 선택된 kernel.org에 있는 장기 지원 옵션에서 가져왔습니다.

사용자 지정 cloud-init

cloud-init 패키지는 클라우드 컴퓨팅 환경에서 Linux 이미지 부트스트랩을 수행하는 오픈 소스 애플리케이션입니다. [자세한 내용은 cloud-init 설명서를 참조하십시오.](#)

AL2023에 cloud-init 사용자 지정 버전이 포함되어 있습니다. cloud-init를 사용하면 부팅 시 인스턴스 작업을 지정할 수 있습니다.

인스턴스를 시작할 때 사용자 데이터 필드를 사용하여 작업을 전달할 수 있습니다. cloud-init 즉, 여러 사용 사례에 공통 Amazon Machine Images (AMIs)를 사용하고 인스턴스 실행 시 이러한 AMI를 동적으로 구성할 수 있습니다. AL2023에서도 cloud-init을 사용하여 ec2-user 계정을 구성할 수 있습니다.

AL2023에서 /etc/cloud/cloud.cfg.d와 /etc/cloud/cloud.cfg의 cloud-init을 사용할 수 있습니다. /etc/cloud/cloud.cfg.d 디렉터리에 자체 cloud-init 파일을 만들 수 있습니다. Cloud-init는 이 디렉터리의 모든 파일을 사전순으로 읽습니다. 나중 파일이 이전 파일 값을 덮어씁니다. cloud-init로 인스턴스를 실행하면 cloud-init 패키지는 다음과 같은 구성 작업을 수행합니다.

- 기본 로케일 설정
- 호스트 이름 설정
- 사용자 데이터 파싱 및 처리
- 호스트 프라이빗 SSH 키 생성
- 간편한 로그인 및 관리를 위해 사용자 공개 SSH 키를 .ssh/authorized_keys에 추가
- 패키지 관리를 위한 리포지토리 준비
- 사용자 데이터에 정의된 패키지 작업 처리

- 사용자 데이터에 있는 사용자 스크립트를 실행
- 해당하는 경우 인스턴스 스토어 볼륨 마운트
 - 기본적으로 ephemeral0 인스턴스 스토어 볼륨이 존재하고 유효한 파일 시스템이 있다면 인스턴스 스토어 볼륨은 /media/ephemeral0에 마운트됩니다. 그렇지 않다면 마운트되지 않습니다.
 - m1.small 및 c1.medium 인스턴스 유형인 경우 인스턴스와 연결된 모든 스왑 볼륨이 마운트됩니다.
 - 다음 cloud-init 명령을 사용하여 기본 인스턴스 스토어 볼륨 마운트를 재정의할 수 있습니다.

```
#cloud-config
mounts:
- [ ephemeral0 ]
```

마운트 관리에 대한 자세한 내용은 cloud-init 설명서의 [마운트\(Mounts\)](#)를 참조하세요.

- 인스턴스를 실행하면 TRIM을 지원하는 인스턴스 스토어 볼륨은 포맷되지 않습니다. 인스턴스 스토어 볼륨을 마운트하려면 먼저 인스턴스 스토어 볼륨의 파티션을 나누고 포맷해야 합니다.

자세한 내용은 Amazon EC2 사용 설명서의 [인스턴스 스토어 볼륨 TRIM 지원](#)을 참조하십시오.

- 인스턴스 실행 후 disk_setup 모듈을 사용하여 인스턴스 스토어 볼륨의 파티션을 나누고 포맷할 수 있습니다.

자세한 내용은 cloud-init 설명서의 [디스크 설정](#)을 참조하세요.

SELinux에서 cloud-init를 사용하는 방법은 [cloud-init를 사용하여 enforcing을 활성화합니다.](#)를 참조하세요.

cloud-init 사용자 데이터 형식에 대한 자세한 내용은 cloud-init 설명서의 [사용자 데이터 형식](#)을 참조하세요.

보안 업데이트 및 기능

AL2023 은 다양한 보안 업데이트 및 솔루션을 제공합니다.

주제

- [관리 업데이트](#)
- [클라우드 보안](#)
- [SELinux 모드](#)

- [규정 준수 프로그램](#)
- [SSH 서버 기본값](#)
- [OpenSSL 3의 주요 기능](#)

관리 업데이트

DNF 및 저장소 버전을 사용하여 보안 업데이트를 적용합니다. 자세한 정보는 [AL2023 패키지 및 운영 체제 업데이트 관리](#)를 참조하세요.

클라우드 보안

보안은 사용자와 사용자 AWS 간의 공동 책임입니다. [공동 책임 모델](#)은 이를 클라우드의 보안과 클라우드에서의 보안으로 설명합니다. 자세한 정보는 [아마존 리눅스 2의 보안 및 규정 준수를 참조](#)하세요.

SELinux 모드

SELinux는 AL2023에서 활성화되고 허용 모드로 기본 설정되어 있습니다. 허용 모드에서는 권한 거부 기록되지만 강제하지는 않습니다.

SELinux 정책은 사용자, 프로세스, 프로그램, 파일 및 장치에 대한 권한을 정의합니다. SELinux에서는 두 정책 중 하나를 선택할 수 있습니다. 정책은 타겟팅 또는 다중 보안 (MLS)입니다.

SELinux 모드 및 정책에 대한 자세한 내용은 [AL2023 SELinux 모드 설정](#)과 [SELinux Project Wiki](#)를 참조하세요.

규정 준수 프로그램

독립 감사관은 여러 규정 AWS 준수 프로그램과 함께 AL2023 보안 및 규정 준수를 평가합니다.

SSH 서버 기본값

AL2023에 OpenSSH 8.7이 포함되어 있습니다. OpenSSH 8.7은 기본적으로 ssh-rsa 키 교환 알고리즘이 비활성화되어 있습니다. 자세한 정보는 [SSH 서버 기본 구성](#)을 참조하세요.

OpenSSL 3의 주요 기능

- 인증서 관리 프로토콜 (CMP, RFC 4210)에는 CRMF (RFC 4211) 와 HTTP 전송 (RFC 6712) 모두 포함되어 있습니다.

- libcrypto의 HTTP 또는 HTTPS 클라이언트는 GET, POST 작업, 리디렉션, 일반 및 ASN.1으로 인코딩된 콘텐츠, 프록시 및 타임아웃을 지원합니다.
- EVP_KDF는 키 파생 함수와 함께 작동합니다.
- EVP_MAC API는 MACs와 함께 작동합니다.
- 리눅스 커널 TLS 지원.

자세한 내용은 [OpenSSL 마이그레이션 가이드](#)를 참조하세요.

네트워킹 서비스

오픈 소스 프로젝트 systemd-networkd는 최신 Linux 배포판에서 널리 사용할 수 있습니다. 이 프로젝트는 나머지 systemd 프레임워크와 유사한 선언적 구성 언어를 사용합니다. 기본 구성 파일 유형은 .network 및 .link 파일입니다.

amazon-ec2-net-utils 패키지는 /run/systemd/network 디렉터리에 인터페이스별로 구성됩니다. 이렇게 구성하면 인스턴스에 연결된 인터페이스에서 IPv4 네트워킹과 IPv6 네트워킹 모두 사용할 수 있습니다. 또한 로컬에서 소싱된 트래픽이 해당 인스턴스의 네트워크 인터페이스를 통해 네트워크로 라우팅 시키는 정책 라우팅 규칙을 설치합니다. 이러한 규칙은 관련 주소 또는 접두사로부터 올바른 트래픽이 ENI (Elastic Network Interface) 를 통해 라우팅되도록 합니다. ENI 사용에 대한 자세한 내용은 Amazon EC2 [사용 설명서의 ENI 사용을](#) 참조하십시오.

사용자 지정 구성 파일을 /etc/systemd/network 디렉터리에 배치하고 /run/systemd/network에 포함된 기본 구성 설정을 재정의하여 이 네트워킹 동작을 사용자 지정할 수 있습니다.

[systemd.network](#) 설명서에 systemd-networkd 서비스가 특정 인터페이스에 적용되는 구성을 결정하는 방법이 나와 있습니다. 또한 다양한 리소스의 속성을 반영하기 위해 ENI 지원 인터페이스에 대한 대체 이름 (소위) 을 생성합니다. altnames AWS 이러한 ENI 지원 인터페이스 속성은 ENI 첨부 파일의 ENI ID 및 DeviceIndex 필드입니다. ip 명령과 같은 다양한 도구를 사용할 때 해당 속성으로 이러한 인터페이스를 참조할 수 있습니다.

AL2023 인스턴스 인터페이스 이름은 systemd 슬롯 이름 지정 체계를 사용하여 생성됩니다. 자세한 내용은 [systemd.net 명명 방식](#)을 참조하세요.

또한 AL2023은 기본적으로 fq_code1 활성 대기열 관리 네트워크 전송 스케줄링 알고리즘을 사용합니다. 자세한 내용은 [CoDel개요](#)를 참조하십시오.

코어 툴체인 패키지 glibc, gcc 및 binutils

Amazon Linux의 일부 패키지는 핵심 툴체인 패키지로 지정됩니다. AL2023 주요 구성 요소인 코어 패키지는 5년간 지원을 받습니다. 패키지 버전은 변경될 수 있지만 Amazon Linux 릴리스에 포함된 패키지는 장기 지원을 받습니다.

이 세 가지 핵심 패키지는 Amazon Linux 배포판에서 대부분의 소프트웨어를 빌드하는 데 사용되는 시스템 도구 체인을 제공합니다.

패키지	정의	용도
glibc 2.34	시스템 C 라이브러리	표준 함수를 제공하는 대부분의 바이너리 프로그램과 프로그램과 커널 간의 인터페이스에 사용할 수 있습니다.
gcc 11.2	gcc 컴파일러 제품군	컴파일 C, C++. Fortran
binutils 2.35	어셈블러, 링커 및 기타 바이너리 도구	바이너리 프로그램을 조작하거나 검사합니다.

glibc 라이브러리 업데이트 후 항상 재부팅하는 것이 좋습니다. 서비스를 제어하는 패키지를 업데이트 할 경우 서비스를 다시 시작하여 업데이트를 선택하세요. 시스템을 재부팅하면 이전의 모든 패키지 및 라이브러리가 업데이트됩니다.

패키지 관리 도구

AL2023 기본 소프트웨어 패키지 관리 도구는 rpm입니다. DNF DNFAL2의 패키지 관리 YUM 도구인 후속 도구입니다.

DNF는 YUM의 사용법과 비슷합니다. 많은 DNF 명령 및 명령 옵션이 명령과 동일합니다YUM. 명령줄 인터페이스 (CLI) 명령에서 대부분의 경우 dnf로 yum을 대체할 수 있습니다.

예를 들어, 다음 AL2 yum 명령의 경우

```
$ sudo yum install packagename
$ sudo yum search packagename
$ sudo yum remove packagename
```

AL2023에서는 이러한 명령이 다음과 같은 명령이 됩니다.

```
$ sudo dnf install packagename
$ sudo dnf search packagename
$ sudo dnf remove packagename
```

AL2023에서 yum 명령을 여전히 사용할 수 있지만 dnf 명령에 대한 포인터로 사용할 수 있습니다. 따라서 셸이나 스크립트에서 yum 명령을 사용하면 모든 명령과 옵션은 DNF CLI와 동일합니다. YUM CLI와 DNF CLI의 차이점에 대한 자세한 내용은 [YUM와 다른 DNF CLI 변경 사항](#)을 참조하세요.

dnf 명령의 명령 및 옵션에 대한 전체 참조는 매뉴얼 페이지 `man dnf`를 참조하세요. 자세한 내용은 [DNF명령 참조](#)를 참조하십시오.

SSH 서버 기본 구성

몇 년 전의 SSH 클라이언트가 있는 경우 인스턴스에 연결할 때 오류가 발생할 수 있습니다. 일치하는 호스트 키 유형을 찾지 못했다는 오류 메시지가 표시되면 SSH 호스트 키를 업데이트하여 이 문제를 해결하세요.

ssh-rsa 서명 기본 비활성화

AL2023에는 기존 ssh-rsa 호스트 키 알고리즘을 비활성화하고 축소된 호스트 키 세트를 생성하는 기본 구성이 포함되어 있습니다. 클라이언트는 ssh-ed25519 또는 ecdsa-sha2-nistp256 호스트 키 알고리즘을 지원합니다.

기본 구성에서는 다음과 같은 키 교환 알고리즘을 모두 사용할 수 있습니다.

- curve25519-sha256
- curve25519-sha256@libssh.org
- ecdh-sha2-nistp256
- ecdh-sha2-nistp384
- ecdh-sha2-nistp521
- diffie-hellman-group-exchange-sha256
- diffie-hellman-group14-sha256
- diffie-hellman-group16-sha512
- diffie-hellman-group18-sha512

기본적으로 AL2023은 ed25519와 ECDSA 호스트 키를 생성합니다. 클라이언트는 ssh-ed25519 또는 ecdsa-sha2-nistp256 호스트 키 알고리즘을 지원합니다. SSH로 인스턴스에 연결할 때 호환 가능한 알고리즘(예: ssh-ed25519 또는 ecdsa-sha2-nistp256)을 지원하는 클라이언트를 사용해야 합니다. 다른 키 유형을 사용해야 하는 경우 사용자 데이터의 cloud-config 프래그먼트로 생성된 키 목록을 재정의하세요.

다음 예제에서 cloud-config가 ecdsa 및 ed25519 키를 사용하여 rsa 호스트 키를 생성합니다.

```
#cloud-config
ssh_genkeytypes:
- ed25519
- ecdsa
- rsa
```

공개 키 인증에 RSA 키 페어를 사용하는 경우 SSH 클라이언트가 rsa-sha2-256 또는 rsa-sha2-512 서명을 지원해야 합니다. 호환되지 않는 클라이언트 사용 중에 업그레이드할 수 없는 경우 인스턴스에서 ssh-rsa 지원을 다시 활성화하세요. ssh-rsa 지원을 다시 활성화하려면 다음 명령을 사용하여 LEGACY 시스템 암호화 정책을 활성화하십시오.

```
$ sudo dnf install crypto-policies-scripts
$ sudo update-crypto-policies --set LEGACY
```

호스트 키 관리에 대한 자세한 내용은 [Amazon Linux 호스트 키를](#) 참조하십시오.

AL2023 에서 더 이상 사용되지 않는 기능

AL2에서는 더 이상 사용되지 않으며 AL2023 에는 없는 기능에 대해 여기에 설명되어 있습니다. 이는 AL2에는 있지만 AL2023 에는 없고 AL2023 에는 추가되지 않는 기능 및 패키지 등의 기능입니다. AL2 에서 기능이 지원되는 기간에 대한 자세한 내용은 AL2에서 더 [이상 사용되지 않는](#) 기능을 참조하십시오.

또한 AL2023 에는 더 이상 사용되지 않는 기능이 있으며 향후 릴리스에서 제거될 예정입니다. 이 장에서는 해당 기능이 무엇인지, 언제 더 이상 지원되지 않는지, Amazon Linux에서 언제 제거되는지에 대해 설명합니다. 더 이상 사용되지 않는 기능을 이해하면 AL2023 배포는 물론 Amazon Linux의 다음 메이저 버전을 준비하는 데도 도움이 됩니다.

주제

- [compat- 패키지](#)
- [더 이상 사용되지 않는 기능이 AL1에서 중단되고 AL2에서 제거됨](#)
- [AL2에서 더 이상 사용되지 않고 AL2023 버전에서 제거된 기능](#)
- [AL2023 버전에서는 더 이상 사용되지 않음](#)

compat- 패키지

접두사가 인 AL2의 모든 패키지는 최신 버전의 패키지용으로 아직 다시 빌드되지 않은 이전 바이너리와의 바이너리 호환성을 위해 compat- 제공됩니다. Amazon Linux의 각 새 메이저 버전은 이전 릴리스의 compat- 패키지를 전달하지 않습니다.

Amazon Linux 릴리스의 모든 compat- 패키지 (예: AL2) 는 더 이상 사용되지 않으며 후속 버전 (예: AL2023) 에는 없습니다. 업데이트된 버전의 라이브러리에 맞춰 소프트웨어를 다시 빌드하는 것이 좋습니다.

더 이상 사용되지 않는 기능이 AL1에서 중단되고 AL2에서 제거됨

이 섹션에서는 AL1에서 사용할 수 있고 AL2에서는 더 이상 사용할 수 없는 기능에 대해 설명합니다.

Note

AL1의 유지 관리 지원 단계의 일환으로 일부 패키지의 end-of-life (EOL) 날짜는 AL1의 EOL보다 이전이었습니다. 자세한 내용은 [AL1 패키지 지원 설명](#)을 참조하십시오.

Note

일부 AL1 기능은 이전 릴리스에서 중단되었습니다. 자세한 내용은 [AL1 릴리스 노트](#)를 참조하십시오.

주제

- [32비트 x86 \(i686\) AMI](#)
- [aws-apitools-*로 대체됨 AWS CLI](#)
- [systemdupstartAL2에서 대체합니다.](#)

32비트 x86 (i686) AMI

[2014.09 AL1 릴리스](#)의 일환으로 아마존 리눅스는 32비트 AMI를 생성하는 마지막 릴리스가 될 것이라고 발표했습니다. 따라서 [2015.03 AL1 릴리스부터](#) Amazon Linux는 더 이상 32비트 모드에서의 시스템 실행을 지원하지 않습니다. AL2는 x86-64 호스트의 32비트 바이너리에 대해 제한된 런타임 지원을 제공하며 새 32비트 바이너리를 빌드할 수 있는 개발 패키지를 제공하지 않습니다. AL2023에는 더 이상 32비트 사용자 공간 패키지가 포함되지 않습니다. 사용자는 AL2023 버전으로 마이그레이션하기 전에 64비트 코드로의 전환을 완료하는 것이 좋습니다.

AL2023 상에서 32비트 바이너리를 실행해야 하는 경우 AL2023 위에서 실행되는 AL2 컨테이너 내에서 AL2의 32비트 사용자 공간을 사용할 수 있습니다.

aws-apitools-*로 대체됨 AWS CLI

2013년 AWS CLI 9월에 릴리스되기 전에 사용자가 Amazon EC2 API를 호출할 수 있도록 하는 명령줄 유틸리티 세트를 제공하고 에서 AWS Java 구현했습니다. 이러한 도구는 2015년에 중단되었으며, 명령줄에서 Amazon EC2 API와 상호 작용하는 데 AWS CLI 선호되는 방법이 되었습니다. 명령줄 유틸리티 세트에는 다음 패키지가 포함됩니다. aws-apitools-*

- aws-apitools-as
- aws-apitools-cfn
- aws-apitools-common
- aws-apitools-ec2
- aws-apitools-elb
- aws-apitools-mon

`aws-apitools-*` 패키지에 대한 업스트림 지원은 2017년 3월에 종료되었습니다. 업스트림 지원이 부족함에도 불구하고 Amazon Linux는 사용자에게 이전 버전과의 호환성을 제공하기 위해 이러한 명령줄 유틸리티 중 일부를 계속 출시했습니다. `aws-apitools-ec2` 적극적으로 유지 관리되고 모든 AWS API를 사용할 수 있는 수단을 제공하므로 이 `aws-apitools-*` 패키지보다 더 강력하고 완전한 도구입니다. AWS CLI

이 `aws-apitools-*` 패키지는 2017년 3월에 지원이 중단되었으며 추가 업데이트를 받지 않을 예정입니다. 이러한 패키지의 모든 사용자는 가능한 한 AWS CLI 빨리 으로 마이그레이션해야 합니다. 이러한 패키지는 AL2023 에는 없습니다.

또한 AL1은 `aws-apitools-iam` 및 `aws-apitools-rds` 패키지를 제공했는데, 이 패키지는 AL1에서 더 이상 사용되지 않으며 AL2 이후 Amazon Linux에는 없습니다.

systemdupstartAL2에서 대체합니다.

AL2는 systemd init 시스템을 사용한 최초의 Amazon Linux 릴리스로서 AL1을 upstart 대체했습니다. AL1에서 최신 버전의 Amazon Linux로 마이그레이션하는 과정에서 모든 upstart 특정 구성을 변경해야 합니다. AL1에서는 사용할 systemd systemd 수 없으므로 AL2나 AL2023 같은 Amazon Linux의 최신 메이저 버전으로의 전환의 일부로만 전환할 수 있습니다. upstart

AL2에서 더 이상 사용되지 않고 AL2023 버전에서 제거된 기능

이 섹션에서는 AL2에서 사용할 수 있고 AL2023 에서는 더 이상 사용할 수 없는 기능에 대해 설명합니다.

주제

- [32비트 x86 \(i686\) 패키지](#)
- [aws-apitools-*로 대체됨 AWS CLI](#)
- [bzd개정 관리 시스템](#)
- [cgroup v1](#)
- [log4j 핫패치 \(\) log4j-cve-2021-44228-hotpatch](#)
- [lsb_release 및 system-lsb-core 패키지](#)
- [mccrypt](#)
- [OpenJDK \(7\) java-1.7.0-openjdk](#)
- [Python 2.7](#)
- [rsyslog-openssl 대체 rsyslog-gnutls](#)

- [네트워크 정보 서비스 \(NIS\)/yp](#)

32비트 x86 (i686) 패키지

[AL1의 2014.09 릴리스의](#) 일환으로 32비트 AMI를 생성하는 마지막 릴리스가 될 것이라고 발표했습니다. 따라서 [2015.03 AL1 릴리스부터](#) Amazon Linux는 더 이상 32비트 모드에서의 시스템 실행을 지원하지 않습니다. AL2는 x86-64 호스트의 32비트 바이너리에 대한 제한된 런타임 지원을 제공하며 새 32비트 바이너리를 빌드할 수 있는 개발 패키지를 제공하지 않습니다. AL2023 에는 더 이상 32비트 사용자 공간 패키지가 포함되지 않습니다. 고객이 64비트 코드로 전환을 완료하는 것이 좋습니다.

AL2023 상에서 32비트 바이너리를 실행해야 하는 경우, AL2023 위에서 실행되는 AL2 컨테이너 내에서 AL2의 32비트 사용자 공간을 사용할 수 있습니다.

aws-apitools-*로 대체됨 AWS CLI

2013년 AWS CLI 9월에 릴리스되기 전에 고객이 Amazon EC2 API를 호출할 수 있도록 하는 명령줄 유틸리티 세트를 제공하고 에서 AWS Java 구현했습니다. 이러한 도구는 2015년에 더 이상 사용되지 않으며 명령줄에서 Amazon EC2 API와 상호 작용하는 데 AWS CLI 선호되는 방법이 되었습니다. 여기에는 다음 패키지가 포함됩니다. aws-apitools-*

- aws-apitools-as
- aws-apitools-cfn
- aws-apitools-common
- aws-apitools-ec2
- aws-apitools-elb
- aws-apitools-mon

aws-apitools-* 패키지에 대한 업스트림 지원은 2017년 3월에 종료되었습니다. 업스트림 지원이 부족함에도 불구하고 Amazon Linux는 고객에게 이전 버전과의 호환성을 제공하기 위해 이러한 명령줄 유틸리티 중 일부 (예:aws-apitools-ec2) 를 계속 출시했습니다. 적극적으로 유지 관리되고 모든 AWS API를 사용할 수 있는 수단을 제공하므로 이 aws-apitools-* 패키지보다 더 강력하고 완전한 도구입니다. AWS CLI

이 aws-apitools-* 패키지는 2017년 3월에 지원이 중단되었으며 추가 업데이트를 받지 않을 예정입니다. 이러한 패키지의 모든 사용자는 가능한 한 AWS CLI 빨리 으로 마이그레이션해야 합니다. 이러한 패키지는 AL2023 에는 없습니다.

bzr개정 관리 시스템

[GNU Bazaar](#)(bzr) 개정 관리 시스템은 AL2에서 중단되었으며 AL2023 버전에서는 더 이상 존재하지 않습니다.

의 사용자는 저장소를 로 bzr 마이그레이션하는 것이 좋습니다. git

cgroup v1

AL2023 은 통합 제어 그룹 계층 구조 (cgroup v2) 로 이동하는 반면, AL2는 cgroup v1을 사용합니다. AL2는 cgroup v2를 지원하지 않으므로 AL2023 로의 전환의 일환으로 이 마이그레이션을 완료해야 합니다.

log4j핫패치 () **log4j-cve-2021-44228-hotpatch**

Note

이 log4j-cve-2021-44228-hotpatch 패키지는 AL2에서 더 이상 사용되지 않으며 AL2023 에서는 제거되었습니다.

[CVE-2021-44228](#) 대응으로 아마존 리눅스는 AL1 및 [AL2용 아파치 Log4j용 핫패치의 RPM 패키지](#) 버전을 출시했습니다. [Amazon Linux에 핫패치를 추가한다는 발표에서 “핫패치](#) 설치가 CVE-2021-44228 또는 CVE-2021-45046 완화 기능을 갖춘 log4j 버전으로 업데이트하는 것을 대체하는 것은 아닙니다.” 라고 언급했습니다.

이 핫패치는 log4j 패치 시간을 벌기 위한 조치였습니다. AL2023 최초의 일반 공급 릴리스는 [CVE-2021-44228](#) 이후 15개월 후였으므로 AL2023 는 핫패치 (활성화 여부) 와 함께 제공되지 않습니다.

Amazon Linux에서 자체 log4j 버전을 실행한다면 [CVE-2021-44228](#) 또는 [CVE-2021-45046](#)의 영향을 받지 않는 버전으로 업데이트하는 것이 좋습니다.

lsb_release 및 system-lsb-core 패키지

이전에는 일부 소프트웨어로 lsb_release 명령(AL2에 설치된 system-lsb-core 패키지)을 호출하여 실행 중인 Linux 배포판에 대한 정보를 가져올 수 있었습니다. Linux 표준 베이스(LSB)에서 이 명령을 사용할 수 있고 Linux 배포판에도 이 명령을 설치되었습니다. Linux 배포판은 이 정보를 /etc/os-release 및 기타 관련 파일에 보관하는 더 간단한 표준으로 발전했습니다.

이 `os-release` 표준은 `systemd`에서 나왔습니다. 자세한 내용은 [systemd OS 관련 설명서](#)를 참조하세요.

AL2023에 `lsb_release` 명령이 사용할 수 없으며 `system-lsb-core` 패키지도 포함되어 있지 않습니다. Amazon Linux 및 기타 주요 Linux 배포판의 호환성을 유지하려면 소프트웨어를 `os-release` 표준으로 전환해야 합니다.

mcrypt

`mcrypt` 라이브러리 및 관련 PHP 확장은 AL2에서 더 이상 사용되지 않으며 AL2023 버전에서는 더 이상 존재하지 않습니다.

[업스트림은 2016년 12월에 처음 릴리스되어 2019년 10월에 최종 출시된 mcrypt 확장 프로그램을 PHP 7.1에서 PHP 지원 중단했습니다.](#)

업스트림 `mcrypt` 라이브러리는 [2007년에 마지막으로 릴리스되었으며, 2017년에 새 커밋에 SourceForge 필요한 cvs](#) 수정 제어에서 마이그레이션하지 않았습니다. 가장 최근의 커밋 (그리고 3년 전만) 은 2011년의 커밋에서 관리자가 있는 프로젝트에 대한 언급을 삭제했습니다.

AL2023 `mcrypt` 에는 추가되지 않으므로 나머지 사용자는 코드를 로 `mcrypt` 이식하는 OpenSSL 것이 좋습니다.

OpenJDK (7) `java-1.7.0-openjdk`

Note

AL2023 은 기반 워크로드를 지원하는 여러 버전의 [Amazon Corretto](#)를 Java 제공합니다. OpenJDK 7 패키지는 AL2에서 더 이상 사용되지 않으며 AL2023 버전에서는 더 이상 존재하지 않습니다. AL2023 버전에서 사용할 수 있는 가장 오래된 JDK는 Corretto 8에서 제공합니다.

Amazon Linux에서의 Java에 대한 자세한 내용은 [을 참조하십시오AL2023에서 Java 사용.](#)

Python 2.7

Note

AL2023에서 Python 2.7이 제거되었으므로, Python을 사용하는 OS 구성 요소는 Python 3에서 작동됩니다. Amazon Linux가 제공하고 지원하는 Python 버전을 계속 사용하려면 Python 2 코드를 Python 3로 변환하세요.

Amazon Linux의 Python에 대한 자세한 내용은 [을 참조하십시오](#)[AL2023에서 Python 사용](#).

rsyslog-openssl대체 rsyslog-gnutls

이 rsyslog-gnutls 패키지는 AL2에서 더 이상 사용되지 않으며 AL2023에서는 더 이상 존재하지 않습니다. 패키지는 모든 rsyslog-openssl 패키지 사용에 대한 즉석 대체품이어야 합니다.

rsyslog-gnutls

네트워크 정보 서비스 (NIS)/yp

네트워크 정보 서비스 (NIS)는 원래 옐로우 페이지라고 불렸거나 YP AL2에서 더 이상 사용되지 않으며 AL2023 버전에서는 더 이상 존재하지 않습니다. 여기에는, 및 패키지가 포함됩니다. ypbind ypserv yp-tools와 통합되는 다른 패키지의 경우 AL2023 버전에서 이 NIS 기능이 제거되었습니다.

AL2023 버전에서는 더 이상 사용되지 않음

이 섹션에서는 Amazon Linux에 존재하며 향후 Amazon Linux 버전에서 제거될 가능성이 있는 기능에 대해 설명합니다. AL2023 각 섹션에서는 이 기능이 무엇이고 Amazon Linux에서 제거될 것으로 예상되는 시기에 대해 설명합니다.

Note

이 섹션은 Linux 에코시스템이 발전하고 Amazon Linux의 향후 주요 버전 출시에 가까워짐에 따라 시간이 지남에 따라 업데이트될 예정입니다.

주제

- [32비트 x86 \(i686\) 런타임 지원](#)
- [버클리 DB \(\) libdb](#)
- [cron](#)
- [IMDSv1](#)
- [pcre 버전 1](#)
- [System V init \(sysvinit\)](#)

32비트 x86 (i686) 런타임 지원

AL2023 은 32비트 x86 (i686) 바이너리를 실행할 수 있는 기능을 유지합니다. Amazon Linux의 다음 메이저 버전에서는 더 이상 32비트 사용자 공간 바이너리 실행을 지원하지 않을 수 있습니다.

버클리 DB () **libdb**

AL2023 버전은 버클리 DB () 라이브러리 버전 5.3.28과 함께 제공됩니다. libdb 이 버전은 라이선스가 덜 제한적인 슬리피캣 라이선스에서 GNU Affero GPLv3 (AGPL) 라이선스로 변경되기 전의 버클리 DB의 마지막 버전입니다.

AL2023 패키지에는 여전히 버클리 DB (libdb) 에 의존하는 패키지가 거의 없으며, 이 라이브러리는 Amazon Linux의 다음 주요 릴리스에서 제거될 예정입니다.

Note

AL2023 dnf 패키지 관리자는 버클리 DB (BDB) 형식 데이터베이스에 대한 읽기 전용 지원을 유지합니다. rpm 이 지원은 Amazon Linux의 다음 주요 릴리스에서 제거될 예정입니다.

cron

cronie 패키지는 AL2 AMI에 기본적으로 설치되었으며, 정기 작업을 예약하는 기존 crontab 방식을 지원합니다. AL2023 cronie 버전에서는 기본적으로 포함되지 않습니다. 따라서 에 대한 crontab 지원은 더 이상 기본적으로 제공되지 않습니다.

AL2023 에서는 선택적으로 cronie 패키지를 설치하여 클래식 cron 작업을 사용할 수 있습니다. systemd에서 추가 기능을 제공하므로 systemd 타이머로 마이그레이션하는 것이 좋습니다.

Amazon Linux의 향후 버전 (아마도 다음 메이저 버전) 에서 클래식 cron 작업에 대한 지원이 더 이상 포함되지 않고 systemd 타이머로의 전환이 완료될 수 있습니다. 사용하지 cron 않는 곳으로 마이그레이션하는 것이 좋습니다.

IMDSv1

기본적으로 AL2023 AMI는 IMDSv2 전용 모드에서 시작되도록 구성되어 있으며, 이를 사용할 수 없습니다. IMDSv1 이 활성화된 상태에서도 AL2023 사용 옵션이 여전히 있습니다. Amazon Linux의 향후 버전에서는 이 IMDSv2 버전만 적용할 가능성이 높습니다.

AMI용 IMDS 구성에 대한 자세한 내용은 Amazon EC2 사용 [설명서의 AMI 구성](#)을 참조하십시오.

pcre 버전 1

기존 pcre 패키지는 더 이상 사용되지 않으며 Amazon Linux의 다음 주요 릴리스에서 제거될 예정입니다. pcre2 패키지의 후속 버전입니다. AL2023 최초 버전에는 대상 패키지 수가 제한되어 있지만 이러한 패키지는 AL2023 내에서 pcre2 마이그레이션됩니다. pcre 지원 중단된 pcre 라이브러리는 AL2023 버전에서도 계속 사용할 수 있습니다.

Note

지원 중단된 버전에는 AL2023 전체 pcre 수명 기간 동안 보안 업데이트가 제공되지 않습니다. pcre지원 수명 주기 및 패키지가 보안 업데이트를 받는 시간에 대한 자세한 내용은 패키지의 [패키지 지원 설명](#)을 참조하십시오. pcre

System V init (**sysvinit**)

System VService (init) 스크립트와의 [이전 버전과의 호환성은 유지되지만 업스트림 systemd 프로젝트는 v254 릴리스의 일환으로 System V 서비스 스크립트에 대한 지원 중단을 발표하고 향후 버전에서 지원이 제거될 것이라고 밝혔습니다.](#) systemd AL2023 자세한 정보는 [systemd](#)을 참조하세요.

AL2023 버전에서는 System V service (init) 스크립트와의 이전 버전과의 호환성은 유지되지만, 다음 메이저 릴리스에서 System V 서비스 (init) 스크립트에 대한 지원이 Amazon Linux에서 제거될 경우에 대비하기 위해 사용자는 네이티브 systemd 단위 파일을 사용하도록 마이그레이션하는 것이 좋습니다.

AL2와 AL2023 비교

다음 항목에서는 AL2와 AL2023 간의 주요 차이점을 설명합니다.

주제

- [패키지 추가, 업그레이드 및 삭제](#)
- [각 릴리스에 대한 지원](#)
- [이름 지정 및 버전 변경](#)
- [최적화](#)
- [Python 2.7은 Python 3으로 대체되었습니다.](#)
- [보안 업데이트](#)
- [안정성을 위한 결정적 업그레이드](#)
- [다수 업스트림 제공](#)
- [AMI 루트 파일 시스템 및 기본 Amazon EBS 볼륨 유형](#)
- [네트워킹 시스템 서비스](#)
- [통합 제어 그룹 계층 구조 \(cgroup v2\)](#)
- [작업 예약](#)
- [glibc, gcc 및 binutils 패키지](#)
- [패키지 관리자](#)
- [로깅 시스템](#)
- [curl 및 libcurl 패키지 변경](#)
- [GNU 프라이버시 가드\(GNUPG\)](#)
- [기본 JVM으로 Amazon Corretto 설정](#)
- [AWS CLI v2](#)
- [UEFI 기본](#)
- [SSH 서버 기본 구성 변경](#)
- [Extra Packages for Enterprise Linux \(EPEL\)](#)
- [cloud-init 사용하기](#)
- [데스크톱 그래픽 지원](#)
- [컴파일러 트리플렛](#)

- [32비트 x86 \(i686\) 패키지](#)
- [lsb_release 및 system-lsb-core 패키지](#)
- [AL2에서 AL2023 커널이 변경되었습니다.](#)
- [Amazon Linux 2와 Amazon Linux 2023 AMI에 설치된 패키지 비교](#)
- [Amazon Linux 2 및 Amazon Linux 2023 미니멀 AMI에 설치된 패키지 비교](#)
- [Amazon Linux 2 및 Amazon Linux 2023 기본 컨테이너 이미지에 설치된 패키지 비교](#)

패키지 추가, 업그레이드 및 삭제

AL2023에는 사용할 수 있는 수천 개의 소프트웨어 패키지가 포함되어 있습니다. Amazon Linux 구 버전과 비교해서 AL2023 버전에 추가, 업그레이드 또는 삭제된 모든 패키지의 전체 목록은 [AL2023 패키지 변경 사항](#)을 참조하세요.

AL2023 버전에 패키지를 추가하거나 변경하도록 요청하려면 [Amazon-linux-2023](#) 리포지토리에 문제를 제출하십시오. GitHub

각 릴리스에 대한 지원

AL2023은 5년간 지원합니다.

자세한 정보는 [릴리스 케이던스](#)을 참조하세요.

이름 지정 및 버전 변경

AL2023은 플랫폼 식별을 위해 AL2가 지원하는 것과 동일한 메커니즘을 제공합니다. 또한 AL2023에 플랫폼 식별을 위한 새 파일을 제공합니다.

자세한 정보는 [이름 지정 및 버전 관리](#)을 참조하세요.

최적화

AL2023 부팅 시간을 최적화하여 인스턴스 실행부터 고객 워크로드 실행까지의 시간이 단축됩니다. Amazon EC2 인스턴스 커널 구성, cloud-init 구성 및 kmod와 systemd 같은 OS 패키지에 내장된 기능을 최적화합니다.

최적화에 대한 자세한 내용은 [성능 및 운영 최적화](#)를 참조하세요.

Python 2.7은 Python 3으로 대체되었습니다.

AL2 코어 패키지에 대한 장기 지원(LTS) 약속의 일환으로 2025년 6월까지 Python 2.7 지원 및 보안 패치를 제공합니다. 이 지원은 2020년 1월 Python end-of-life 2.7의 업스트림 Python 커뮤니티 선언을 넘어섭니다.

AL2는 Python 2.7에 크게 의존하는 yum 패키지 관리자를 사용합니다. AL2023 버전의 dnf 패키지 관리자가 Python 3으로 마이그레이션되어 더 이상 Python 2.7이 필요하지 않습니다. AL2023은 파이썬 3으로 완전히 전환되었습니다.

Note

AL2023에서 Python 2.7이 제거되었으므로, Python을 사용하는 OS 구성 요소는 Python 3에서 작동됩니다. Amazon Linux가 제공하고 지원하는 Python 버전을 계속 사용하려면 Python 2 코드를 Python 3로 변환하세요.

Amazon Linux Python에 대한 자세한 내용은 [AL2023에서 Python 사용](#)을 참조하세요.

보안 업데이트

SELinux

기본적으로 AL2023 Security Enhanced Linux (SELinux)는 enabled되어 있으며 permissive 모드로 설정되어 있습니다. permissive 모드에서 권한 거부는 기록되지만 강제하지는 않습니다.

SELinux는 AL2 disabled에 있던 Amazon Linux 커널의 보안 기능입니다. SELinux는 커널의 주요 하위 시스템에 필수 액세스 제어(MAC) 아키텍처를 제공하는 커널 기능 및 유틸리티입니다.

자세한 정보는 [AL2023 SELinux 모드 설정](#)을 참조하세요.

SELinux 리포지토리, 도구 및 정책에 대한 자세한 내용은 [SELinux 노트북](#), [SELinux 정책 유형](#) 및 [SELinux 프로젝트](#)를 참조하세요.

OpenSSL 3

AL2023에는 Open Secure Sockets Layer version 3 (OpenSSL 3) 암호화 툴킷이 있습니다. AL2023 지원 TLS 1.3 및 네트워크 프로토콜. TLS 1.2

기본적으로 AL2는 OpenSSL 1.0.2과 함께 제공됩니다. OpenSSL 1.1.1에 애플리케이션을 빌드할 수 있습니다.

OpenSSL에 대한 자세한 내용은 [OpenSSL 마이그레이션 가이드](#)를 참조하세요.

보안에 대한 자세한 내용은 [보안 업데이트 및 기능](#)을 참조하세요.

IMDSv2

기본적으로 AL2023 AMI로 시작하는 모든 인스턴스는 단독으로 필요하며 IMDSv2 컨테이너식 워크로드 지원이 가능하도록 기본 홉 제한이 2로 설정됩니다. `imds-support` 파라미터를 `v2.0`로 설정하면 됩니다. 자세한 내용은 Amazon EC2 사용 설명서의 [AMI 구성](#)을 참조하십시오.

Note

세션 토큰의 유효 시간은 1초에서 6시간 사이입니다. IMDSv2 쿼리에 API 요청을 전달하는 주소는 다음과 같습니다.

- IPv 4:169.254.169.254
- IPv6: fd00:ec2::254

인스턴스 메타데이터 옵션 시작 속성을 IMDSv1 사용하여 이러한 설정을 수동으로 재정의하고 활성화할 수 있습니다. 또한 IAM 컨트롤을 사용하여 다양한 설정을 적용할 수 있습니다. IMDS 인스턴스 메타데이터 서비스 설정 및 사용에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [사용 IMDSv2, 새 인스턴스에 대한 인스턴스 메타데이터 옵션 구성 및 기존 인스턴스의 인스턴스 메타데이터 옵션 수정](#)을 참조하십시오.

log4j 핫패치 제거 (**log4j-cve-2021-44228-hotpatch**)

Note

AL2023에는 `log4j-cve-2021-44228-hotpatch` 패키지가 없습니다.

[CVE-2021-44228](#) 대응으로 아마존 리눅스는 AL1 및 [AL2용 아파치 Log4j용 핫패치의 RPM 패키지](#) 버전을 출시했습니다. [Amazon Linux에 핫패치를 추가한다고 발표](#)하면서 “핫패치 설치가 CVE-2021-44228 또는 CVE-2021-45046 완화 기능을 갖춘 log4j 버전 업데이트를 대체하는 것이 아니라고 언급했습니다.

이 핫패치는 log4j 패치 시간을 벌기 위한 조치였습니다. [CVE-2021-44228](#)에 대응한 지 15개월 후에 첫 AL2023 정식 릴리스(GA)가 이루어졌기 때문에 AL2023에 핫패치 (활성화 여부와 관계없이)를 포함 되어 있지 않습니다.

Amazon Linux에서 자체 log4j 버전을 실행하는 사용자는 [CVE-2021-44228](#) 또는 [CVE-2021-45046](#) 영향을 받지 않는 버전으로 업데이트했는지 확인해야 합니다.

AL2023은 보안 패치를 최신 상태로 유지할 수 있도록 [AL2023 업데이트](#) 지침을 제공합니다. 보안 권고는 [Amazon Linux 보안 센터](#)에 나와 있습니다.

안정성을 위한 결정적 업그레이드

버전이 지정된 리포지토리를 통한 결정적 업그레이드 기능을 사용하면 모든 AL2023 AMI는 기본적으로 특정 리포지토리 버전에 고정됩니다. 결정적 업그레이드를 통해 패키지 버전과 업데이트 간의 일관성을 높일 수 있습니다. 메이저 또는 마이너 릴리스에는 특정 리포지토리 버전이 포함되어 있습니다.

최신 AL2023 버전에는 결정적 업그레이드가 기본적으로 활성화되어 있습니다. 이는 AL2 및 기타 구 버전에서 사용되었던 수동 증분 잠금 방법에서 개선된 방식입니다.

자세한 정보는 [AL2023 버전 지정 저장소를 통한 결정적 업그레이드 사용](#)을 참조하세요.

다수 업스트림 제공

AL2023 버전은 RPM 기반이며 여러 버전의 Fedora 및 기타 배포판 (예: CentOS 9 Stream) 에서 가져온 구성 요소를 포함합니다. Amazon Linux 커널은 kernel.org에서 직접 제공하는 장기 지원 (LTS) 릴리스에서 가져온 것으로, 다른 배포판과는 별개로 선택된 것입니다.

자세한 정보는 [Fedora와의 관계](#)을 참조하세요.

AMI 루트 파일 시스템 및 기본 Amazon EBS 볼륨 유형

AL2023 AMI와 AL2 모두 루트 파일 시스템의 XFS 파일 시스템을 사용합니다. 루트 디바이스 파일 시스템 mkfs 옵션은 AL2023 Amazon EC2에 맞게 최적화되었습니다. 또한 AL2023은 특정 요구 사항에 따라 다른 볼륨에서 사용할 수 있는 여러 다른 파일 시스템을 지원합니다.

AL2023 AMI는 기본적으로 Amazon EBS gp3 볼륨을 사용하는 반면, AL2 AMI는 기본적으로 Amazon EBS gp2 볼륨을 사용합니다. 인스턴스 시작 시 볼륨을 변경할 수 있습니다.

Amazon EBS 볼륨 유형에 대한 자세한 내용은 [Amazon EBS 범용 볼륨](#)(Amazon EBS General Purpose Volumes)을 참조하세요.

Amazon EC2 인스턴스 시작에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [인스턴스 시작](#)을 참조하십시오.

네트워킹 시스템 서비스

systemd-networkd 시스템 서비스는 AL2023 네트워크 인터페이스를 관리합니다. 이는 ISC dhclient 또는 dhclient를 사용하는 AL2와 다른 점입니다.

자세한 정보는 [네트워킹 서비스](#)을 참조하세요.

통합 제어 그룹 계층 구조 (cgroup v2)

제어 그룹(cgroup)은 프로세스를 계층적으로 구성하고 프로세스 간에 시스템 리소스를 배포하는 Linux 커널 기능입니다. 제어 그룹은 systemd로 컨테이너 런타임을 광범위하게 구현할 수 있습니다.

AL2는 cgroupv1 지원하며 AL2023 지원도 지원합니다. cgroupv2 이는 컨테이너 워크로드를 실행할 때 특히 유용합니다(예: [AL2023 기반 Amazon ECS AMI를 사용하여 컨테이너식 워크로드를 호스팅합니다.](#)).

Amazon Linux에는 여전히 을 사용하여 cgroupv1 시스템을 실행할 수 있는 코드가 포함되어 있지만 이 구성은 권장되거나 지원되는 구성이 아니며 향후 Amazon Linux의 주요 릴리스에서 완전히 제거될 예정입니다. AL2023

[저사양 Linux 커널 인터페이스](#)에 관한 광범위한 설명서와 [systemd cgroup 위임 설명서](#)가 있습니다.

컨테이너 이외의 일반적인 사용 사례는 사용할 수 있는 시스템 리소스에 제한이 있는 systemd 유닛을 만드는 것입니다. 자세한 내용은 [systemd.resource-control](#)을 참조하십시오.

작업 예약

cronie 패키지는 AL2 AMI에 기본적으로 설치되었으며, 정기 작업을 예약하는 기존 crontab 방식을 지원합니다. AL2023에서는 기본적으로 포함되지 않습니다. 따라서 에 대한 crontab 지원은 더 이상 기본적으로 제공되지 않습니다.

cronie 패키지를 옵션으로 설치하여 기존 cron 작업을 할 수 있습니다. systemd에서 추가 기능을 제공하므로 systemd 타이머로 마이그레이션하는 것이 좋습니다.

glibc, gcc 및 binutils 패키지

AL2023에는 AL2와 동일한 코어 패키지가 다수 포함되어 있습니다.

AL2023에 다음 세 가지 핵심 툴체인 패키지를 업데이트했습니다.

패키지 이름	AL2	AL2023
glibc	2.26	2.34
gcc	7.3	11.3
binutils	2.29	2.39

자세한 정보는 [코어 툴체인 패키지 glibc, gcc 및 binutils](#)을 참조하세요.

패키지 관리자

AL2023 기본 소프트웨어 패키지 관리 도구는 DNF입니다. DNF는 AL2 패키지 관리 도구인 YUM의 후속 도구입니다.

자세한 정보는 [패키지 관리 도구](#)을 참조하세요.

로깅 시스템

AL2023 로깅 시스템 패키지는 AL2에서 업그레이드되었습니다. AL2023에 rsyslog가 설치되어 있지 않으므로 AL2에 있던 /var/log/messages와 같은 텍스트 기반 로그 파일은 사용할 수 없습니다. AL2023 기본 구성은 systemd-journal이며 journalctl으로 검사할 수 있습니다. rsyslog는 AL2023 옵션 패키지이지만 새로운 systemd 기반 journalctl 인터페이스와 관련 패키지를 사용하는 것이 좋습니다. 자세한 내용은 [journalctl](#) 매뉴얼 페이지를 참조하세요.

curl 및 libcurl 패키지 변경

AL2023에서 curl 및 libcurl 패키지의 공통 프로토콜과 기능을 curl-minimal 및 libcurl-minimal로 구분할 수 있습니다. 이는 AL2023 AMI 및 컨테이너 기본 패키지이며 대부분 사용자의 디스크, 메모리 및 종속성 풋프린트 용량을 줄입니다.

gopher://를 지원하기 위해 curl의 전체 기능이 필요한 경우 다음 명령을 실행하여 curl-full 및 libcurl-full 패키지를 설치하세요.

```
$ dnf swap libcurl-minimal libcurl-full
```

```
$ dnf swap curl-minimal curl-full
```

GNU 프라이버시 가드(GNUPG)

AL2023은 gnupg2 패키지의 최소 기능과 전체 기능을 gnupg2-minimal과 gnupg2-full 패키지로 구분합니다. gnupg2-minimal 패키지가 기본적으로 설치되어 있습니다. 이는 rpm 패키지의 디지털 서명을 확인하는 데 필요한 최소한의 기능을 제공합니다.

키 서버에서 키를 다운로드하기 등 더 많은 기능을 사용하려면 gnupg2-full 패키지가 설치되어 있어야 합니다. gnupg2 gnupg2-minimal을 gnupg2-full로 변경하려면 다음 명령을 실행합니다.

```
$ dnf swap gnupg2-minimal gnupg2-full
```

기본 JVM으로 Amazon Corretto 설정

AL2023에는 기본 (그리고 유일한) Java 개발 키트(JDK)로 [Amazon Corretto](#)를 사용할 수 있습니다. AL2023 내 모든 Java 기반 패키지는 모두 다음과 같이 빌드됩니다. Amazon Corretto 17

AL2에서 마이그레이션하는 경우 AL2의 해당 OpenJDK 버전에서 다음으로 원활하게 전환할 수 있습니다. Amazon Corretto

AWS CLI v2

AL2023 은 AWS CLI 버전 2와 함께 제공되는 반면 AL2는 버전 1과 함께 제공됩니다. AWS CLI

UEFI 기본

기본적으로 UEFI 펌웨어를 지원하는 인스턴스 유형에서 AL2023 AMI로 실행된 인스턴스는 UEFI 모드에서 시작하려면 부팅 모드 AMI 파라미터를 uefi-preferred로 설정하면 됩니다. 자세한 내용은 Amazon EC2 사용 설명서의 [부팅 모드를](#) 참조하십시오.

SSH 서버 기본 구성 변경

AL2023 AMI에서 릴리스에서 생성하는 sshd 호스트 키 유형을 변경했습니다. 또한 실행 시 키가 생성되지 않도록 일부 레거시 키 유형을 삭제했습니다. 클라이언트는 rsa-sha2-256 및 rsa-sha2-512 프로토콜 또는 ed25519 키를 사용하는 ssh-ed25519를 지원해야 합니다. 기본적으로 ssh-rsa 서명은 비활성화되어 있습니다.

또한 기본 `sshd_config` 파일의 AL2023 구성 설정에는 `UseDNS=no`이 포함되어 있습니다. 이렇게 설정하면 DNS 장애로 인해 인스턴스와 `ssh` 세션을 설정하는 기능이 차단될 가능성이 줄어듭니다. 단점은 `authorized_keys` 파일의 `from=hostname.domain,hostname.domain` 줄 항목이 확인되지 않는다는 것입니다. `sshd`은 더 이상 DNS 이름을 확인하지 않으므로 쉼표로 구분된 각 `hostname.domain` 값을 해당 IP address 값으로 변환해야 합니다.

자세한 정보는 [SSH 서버 기본 구성](#)을 참조하세요.

Extra Packages for Enterprise Linux (EPEL)

Extra Packages for Enterprise Linux (EPEL)는 엔터프라이즈급 Linux 운영 체제를 위한 다양한 패키지를 만드는 것을 목표로 하는 Fedora 커뮤니티의 프로젝트입니다. 이 프로젝트를 통해 주로 RHEL 및 CentOS 패키지를 제작했습니다. AL2는 높은 수준의 CentOS 7 호환성을 제공합니다. 결과적으로 많은 EPEL7 패키지가 AL2에서 작동합니다. 하지만, EPEL 또는 EPEL 같은 리포지토리는 AL2023에서 지원되지 않습니다.

cloud-init 사용하기

cloud-init는 AL2023 패키지 리포지토리를 관리합니다. Amazon Linux 구 버전 cloud-init에 기본적으로 보안 업데이트가 설치되어 있습니다. 이는 AL2023 기본값이 아닙니다. AL2023에서 실행할 때 `releasever` 업데이트하는 새로운 결정적 업그레이드 기능은 실행할 때 패키지 업데이트를 활성화할 수 있습니다. 자세한 내용은 [AL2023 패키지 및 운영 체제 업데이트 관리](#) 및 [안정성을 위한 결정적 업그레이드](#) 섹션을 참조하세요.

AL2023에서 cloud-init와 SELinux를 함께 사용할 수 있습니다. 자세한 정보는 [cloud-init를 사용하여 enforcing을 활성화합니다](#)을 참조하세요.

Cloud-init은 HTTP(S)를 사용하여 원격 위치에서 cloud-init으로 구성 내용을 로드합니다. Amazon Linux 구 버전에서는 원격 리소스를 사용할 수 없을 때 알림을 보내지 않았습니다. AL2023 환경에서 원격 리소스를 사용할 수 없으면 치명적인 오류가 발생하여 cloud-init 실행을 할 수 없습니다. AL2과 다른 이 동작으로 더 안전하게 “실패 종료”를 할 수 있습니다.

자세한 내용은 [cloud-init 설명서](#)의 [사용자 지정 cloud-init](#) 섹션을 참조하세요.

데스크톱 그래픽 지원

Amazon EC2에 최적화된 클라우드 중심의 AL2023은 현재 그래픽 또는 데스크톱 환경을 지원하지 않습니다. 에 대한 GitHub 피드백을 제공하려면 <https://github.com/> 을 참조하십시오.

컴파일러 트리플렛

AL2023에서 컴파일러 트리플렛을 GCC와 LLVM로 설정하면 amazon이 공급업체임을 나타냅니다.

따라서 AL2 aarch64-redhat-linux-gcc는 AL2023에서 aarch64-amazon-linux-gcc가 됩니다.

이는 대부분의 사용자에게 완전히 투명해야 하며 AL2023 기반 컴파일러를 빌드하는 사용자에게만 영향을 미칠 수 있습니다.

32비트 x86 (i686) 패키지

[AL1 2014.09 릴리스의 일환으로 AL1이](#) 32비트 AMI를 생성하는 마지막 릴리스가 될 것이라고 발표되었습니다. 따라서 [AL1 2015.03 릴리스](#)부터 Amazon Linux는 더 이상 32비트 모드에서 실행되지 않습니다. AL2는 x86-64 호스트의 32비트 바이너리에 대해 제한된 런타임을 지원했으나 새 32비트 바이너리를 빌드할 수 있는 개발 패키지를 제공하지 않았습니다. AL2023에는 더 이상 32비트 사용자 공간 패키지가 없습니다. 64비트 코드로의 전환을 완료하는 것이 좋습니다.

AL2023에서 32비트 바이너리를 실행하는 경우, AL2023에서 실행되는 AL2 컨테이너 내에서 AL2의 32비트 사용자 공간을 사용할 수 있습니다.

lsb_release 및 system-lsb-core 패키지

이전에는 일부 소프트웨어로 lsb_release 명령(AL2에 설치된 system-lsb-core 패키지)을 호출하여 실행 중인 Linux 배포판에 대한 정보를 가져올 수 있었습니다. Linux 표준 베이스(LSB)에서 이 명령을 사용할 수 있고 Linux 배포판에도 이 명령을 설치되었습니다. Linux 배포판은 이 정보를 /etc/os-release 및 기타 관련 파일에 보관하는 더 간단한 표준으로 발전했습니다.

이 os-release 표준은 systemd에서 나왔습니다. 자세한 내용은 [systemd OS 관련 설명서](#)를 참조하세요.

AL2023에 lsb_release 명령이 사용할 수 없으며 system-lsb-core 패키지도 포함되어 있지 않습니다. Amazon Linux 및 기타 주요 Linux 배포판의 호환성을 유지하려면 소프트웨어를 os-release 표준으로 전환해야 합니다.

AL2에서 AL2023 커널이 변경되었습니다.

AL2023 버전에서는 Amazon Linux를 클라우드용으로 더욱 최적화하기 위해 6.1 커널과 많은 구성 변경 사항을 제공합니다. 대부분의 사용자에게 이러한 변경 사항은 완전히 투명해야 합니다.

보안 중심으로 커널 구성 변경

CONFIG 옵션	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_BUG_ON_DATA_CORRUPTION	n	y	n	y	y	y
CONFIG_DEBUG_FAULT_MMAP_MIN_ADDR	4096	4096	4096	4096	65536	65536
CONFIG_DEBUG_VM	n	y	n	y	n	n
CONFIG_DEBUG_VPORT	n	y	n	y	n	n
CONFIG_FORTIFY_SOURCE	n	y	n	y	y	y
CONFIG_HARDENED_USERCOPY_FALLBACK	N/A	해당 사항 없음	y	y	해당 사항 없음	해당 사항 없음
CONFIG_INIT_ON_ALLOC_DEFAULT_ON	해당 사항 없음	해당 사항 없음	n	n	n	n
CONFIG_INIT_ON_FREE	해당 사항 없음	해당 사항 없음	n	n	n	n

CONFIG 옵션	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
<u>E_DEFAULT_ON</u>						
<u>CONFIG_IOMMU_DEFAULT_DMA_STRICT</u>	해당 사항 없음	해당 사항 없음	해당 사항 없음	해당 사항 없음	n	n
<u>CONFIG_LD_ISC_AUTOLOAD</u>	y	y	y	y	n	n
<u>CONFIG_SCHED_CORE</u>	해당 사항 없음	해당 사항 없음	해당 사항 없음	해당 사항 없음	해당 사항 없음	y
<u>CONFIG_SCHED_STACK_END_CHECK</u>	n	y	n	y	y	y
<u>CONFIG_SECURITY_DMESG_RESTRICT</u>	n	n	n	n	y	y
<u>CONFIG_SECURITY_SELINUX_DISABLE</u>	y	y	y	y	n	n
<u>CONFIG_SHUFFLE_PAGE_ALLOCATOR</u>	해당 사항 없음	N/A	y	y	y	y

CONFIG 옵션	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_SLAB_FREELIST_HARDENED	n	y	y	y	y	y
CONFIG_SLAB_FREELIST_RANDOM	n	n	y	y	y	y

x86-64 특정 보안 중심 커널 구성 변경

CONFIG 옵션	AL2/4.14/x86_64	AL2/5.10/x86_64	AL2023/6.1/x86_64
CONFIG_AMD_IOMMU	y	y	y
CONFIG_AMD_IOMMU_V2	m	m	y
CONFIG_RANDOMIZE_MEMORY	N/A	y	y

aarch64 (ARM/Graviton) 전용 보안 중심 커널 구성 변경

CONFIG 옵션	AL2/4.14/aarch64	AL2/5.10/aarch64	AL2023/6.1/aarch64
CONFIG_ARM64_PTR_AUTH	N/A	y	y
CONFIG_ARM64_PTR_AUTH_KERNEL	해당 사항 없음	N/A	y

CONFIG 옵션	AL2/4.14/aarch64	AL2/5.10/aarch64	AL2023/6.1/aarch64
CONFIG_AR M64_SW_TT BR0_PAN	y	y	y

/dev/mem, /dev/kmem 및 /dev/port

Amazon Linux 2023은 AL2에 이미 적용되어 있는 제한을 /dev/mem CONFIG_DEVMEM 비활성화하고 CONFIG_DEVPORT 이를 완전히 기반으로 합니다. /dev/port

이 /dev/kmem 코드는 Linux 5.13 커널에서 완전히 제거되었으며, AL2에서는 비활성화되었지만 이제는 AL2023 에는 적용할 수 없습니다.

이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

FORTIFY_SOURCE

AL2023 는 지원되는 모든 CONFIG_FORTIFY_SOURCE 아키텍처에서 활성화됩니다. 이 기능은 보안 강화 기능입니다. 컴파일러가 버퍼 크기를 확인하고 검증할 수 있는 경우 이 기능은 공통 문자열 및 메모리 함수의 버퍼 오버플로를 탐지할 수 있습니다.

이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

라인 규율 자동 로드 () CONFIG_LDISC_AUTOLOAD

권한이 있는 프로세스에서 요청이 오지 않는 한 AL2023 커널은 라인 규율 (예: 를 사용하는 소프트웨어) 을 사용하여 라인 규율을 자동으로 로드하지 않습니다. TIOCSETD ioctl CAP_SYS_MODULE

이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

dmesg 권한이 없는 사용자를 위한 액세스 () CONFIG_SECURITY_DMESG_RESTRICT

기본적으로 AL2023 에서는 권한이 없는 사용자의 액세스를 허용하지 않습니다. dmesg

이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

SELinux 비활성화 **selinuxfs**

AL2023 은 더 이상 사용되지 않는 CONFIG_SECURITY_SELINUX_DISABLE 커널 옵션을 비활성화합니다. 이 옵션을 사용하면 정책을 로드하기 전에 SELinux를 비활성화하는 런타임 메서드를 사용할 수 있습니다.

이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

기타 커널 구성 변경

CONFIG 옵션	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_HZ	100	250	100	250	100	100
CONFIG_NR_CPUS	4096	8192	4096	8192	512	512
CONFIG_PANIC_ON_OOPS	y	n	y	n	y	y
CONFIG_PANIC_ON_OOPS_VALUE	1	0	1	0	1	1
CONFIG_PREEMPT	m	m	m	m	n	n
CONFIG_SLIP	m	m	m	m	n	n
CONFIG_XEN_PV	N/A	y	해당 사항 없음	n	N/A	n

CONFIG_HZ

AL2023 설정은 플랫폼 및 플랫폼 모두에서 CONFIG_HZ 100으로 설정됩니다. x86-64 aarch64

CONFIG_NR_CPUS

AL2023 값은 Amazon EC2에서 사용할 수 있는 최대 CPU 코어 수에 가까운 수로 설정합니다. CONFIG_NR_CPUS.

Panic on OOPS

실패하면 AL2023 커널이 패닉 상태가 됩니다. 이 기능은 커널 명령줄에서 `oops=panic`로 부팅하는 것과 같습니다.

커널 오프스는 커널이 시스템 신뢰성에 영향을 미칠 수 있는 내부 오류를 감지합니다.

PPP 및 SLIP 지원

AL2023 프로토콜은 PPP 또는 SLIP 프로토콜을 지원하지 않습니다.

Xen PV 게스트 지원

AL2023 버전은 Xen PV 게스트로 실행하는 것을 지원하지 않습니다.

커널 파일 시스템 지원

AL2의 커널이 마운트를 지원하는 파일 시스템이 몇 가지 변경되었으며 커널이 파싱할 파티션 구성표도 변경되었습니다.

CONFIG 옵션	AL2/4.14/ aarch64	AL2/4.14/ x86_64	AL2/5.10/ aarch64	AL2/5.10/ x86_64	AL2023/6. 1/aarch64	AL2023/6. 1/x86_64
CONFIG_AFS_FS	n	m	n	m	n	n
CONFIG_AFS_RXRPC	n	m	n	m	n	n
CONFIG_BSD_DISKLABEL	y	y	y	y	n	n
CONFIG_CRAMFS	m	m	m	m	n	n

CONFIG 옵션	AL2/4.14/ aarch64	AL2/4.14/ x86_64	AL2/5.10/ aarch64	AL2/5.10/ x86_64	AL2023/6. 1/aarch64	AL2023/6. 1/x86_64
<u>CONFIG_CRAMFS_BLOCKDEV</u>	N/A	해당 사항 없음	y	n	해당 사항 없음	해당 사항 없음
<u>CONFIG_DM_CLONE</u>	해당 사항 없음	해당 사항 없음	n	n	n	n
<u>CONFIG_DM_ERA</u>	m	n	m	n	n	n
<u>CONFIG_DM_INTEGRITY</u>	n	m	n	m	m	m
<u>CONFIG_DM_LOG_WRITES</u>	n	n	m	m	m	m
<u>CONFIG_DM_SWITCH</u>	m	n	m	n	n	n
<u>CONFIG_DM_VERITY</u>	m	n	m	n	n	n
<u>CONFIG_ECRYPT_FS</u>	n	m	n	m	n	n
<u>CONFIG_EXFAT_FS</u>	해당 사항 없음	N/A	m	m	m	m
<u>CONFIG_EXFAT2_FS</u>	n	m	n	m	n	n
<u>CONFIG_EXFAT3_FS</u>	n	m	n	m	n	n

CONFIG 옵션	AL2/4.14/ aarch64	AL2/4.14/ x86_64	AL2/5.10/ aarch64	AL2/5.10/ x86_64	AL2023/6. 1/aarch64	AL2023/6. 1/x86_64
<u>CONFIG_GFS2_FS</u>	m	m	m	m	n	n
<u>CONFIG_HFSPLUS_FS</u>	n	m	n	m	n	n
<u>CONFIG_HFS_FS</u>	n	m	n	m	n	n
<u>CONFIG_JFS_FS</u>	n	m	n	m	n	n
<u>CONFIG_LDM_PARTITION</u>	n	y	n	y	n	n
<u>CONFIG_MSC_PARTITION</u>	n	y	n	y	n	n
<u>CONFIG_NFS_V2</u>	n	m	n	m	n	n
<u>CONFIG_NTFS_FS</u>	n	m	n	n	n	n
<u>CONFIG_ROMFS_FS</u>	n	m	n	m	n	n
<u>CONFIG_SOLARIS_X86_PARTITION</u>	n	y	n	y	n	n

CONFIG 옵션	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_SQUASHFS_ZSTD	n	y	n	y	y	y
CONFIG_SU_N_PARTITIONS	n	y	n	y	n	n

앤드류 파일 시스템 지원(AFS)

커널은 더 이상 afs 파일 시스템을 지원하도록 빌드되지 않습니다. AL2는 사용자 공간 지원과 함께 제공되지 않았습니다. afs

cramfs 지원

커널은 더 이상 cramfs 파일 시스템을 지원하도록 빌드되지 않습니다. AL2023 후속 버전은 파일 시스템입니다. squashfs

BSD 디스크 레이블 지원

커널은 더 이상 BSD 디스크 레이블을 지원하도록 빌드되지 않습니다. BSD 디스크 레이블이 있는 볼륨을 읽어야 하는 경우 여러 BSD를 실행할 수 있습니다.

디바이스 매퍼 변경

AL2023 커널에 구성된 디바이스 매퍼 타겟이 몇 가지 변경되었습니다.

eCryptFs 지원

ecryptfs 파일 시스템은 Amazon Linux에서 더 이상 사용할 수 없습니다. 의 사용자 공간 구성 요소는 AL1에 존재하고 AL2에서 ecryptfs 제거되었으며 AL2023 커널은 더 이상 지원을 통해 커널을 빌드하지 않습니다. ecryptfs

exFAT

exFAT 파일 시스템에 대한 지원이 AL2의 5.10 커널에 추가되었습니다. AL2 출시 당시에는 4.14 커널과 함께 제공되지 않았습니다. AL2023 은 exFAT 파일 시스템을 계속 지원합니다.

ext2, ext3 및 ext4 파일 시스템

AL2023 CONFIG_EXT4_USE_FOR_EXT2 옵션과 함께 제공됩니다. 즉, ext4 파일 시스템 코드를 사용하여 기존 ext2 파일 시스템을 읽을 수 있습니다.

CONFIG_GFS2_FS

커널은 더 이상 CONFIG_GFS2_FS로 빌드되지 않습니다.

Apple Extended HFS 파일 시스템 지원 (HFS+)

AL2에서는 x86-64 커널만 hfsplus 파일 시스템 지원을 통해 빌드되었습니다. AL2 5.15 커널에는 아키텍처에 대한 hfsplus 지원이 포함되어 있지 않습니다. AL2023 버전에서는 Amazon Linux에서의 hfsplus 지원 중단이 완료되었습니다.

HFS 파일 시스템 지원

AL2에서는 x86-64 커널만 파일 시스템 지원을 통해 구축되었습니다. hfs AL2 5.15 커널에는 아키텍처에 대한 hfs 지원이 포함되어 있지 않습니다. AL2023 버전에서는 Amazon Linux에서의 hfs 지원 중단이 완료되었습니다.

JFS 파일 시스템 지원

AL2에서는 x86-64 커널만 파일 시스템 지원을 통해 구축되었습니다. jfs AL2 5.15 커널에는 아키텍처에 대한 jfs 지원이 포함되어 있지 않습니다. AL1과 AL2 모두 JFS 사용자 공간과 함께 제공되지 않습니다. AL2023 버전에서는 Amazon Linux에서의 jfs 지원 중단이 완료되었습니다.

업스트림 Linux 커널은 [제거를 고려하고 있습니다. JFS](#) 따라서 파일 시스템에 데이터가 있는 경우 다른 JFS 파일 시스템으로 마이그레이션해야 합니다.

Windows 논리적 디스크 관리자 (동적 디스크) 지원 (CONFIG_LDM_PARTITION)

AL2023 은 더 이상 또는 MS-DOS 스타일 파티션이 있는 Windows Vista 동적 디스크를 지원하지 않습니다. Windows XP 에 도입된 최신 GPT 기반 동적 디스크는 이 코드가 전혀 지원하지 않았습니다. Windows Vista

Macintosh 파티션 맵 지원

AL2023 버전은 더 이상 클래식 매킨토시 파티션 맵을 지원하지 않습니다. 최신 macOS 버전은 구 버전에 최신 GPT 파티션 테이블을 생성합니다.

NFSv2 지원

AL2023 버전은 더 이상 NFSv2를 지원하지 않지만 NFSv3, NFSv4, NFSv4.1 및 NFSv4.2는 계속 지원합니다. NFSv3 이상으로 마이그레이션하는 것이 좋습니다.

NTFS(**CONFIG_NTFS_FS**)

AL2의 5.10 커널부터 Amazon Linux에서 NTFS 파일 시스템에 액세스하기 위한 `ntfs` 코드가 `ntfs3` 코드가 대체되었습니다. AL2023 에는 더 이상 `ntfs` 코드가 포함되지 않으며, NTFS 파일 시스템에 액세스하기 위한 `ntfs3` 코드에만 의존합니다.

romfs 파일 시스템

`squashfs` 파일 시스템은 Amazon Linux `romfs` 파일 시스템의 후속 버전이며, AL2023 커널은 더 이상 `romfs`를 지원하지 않습니다.

Solaris x86 하드 디스크 파티션 형식

AL2023 버전은 더 이상 Solaris x86 하드 디스크 파티션 형식을 지원하지 않습니다.

squashfszstd 압축

AL2023 기능은 지원되는 모든 아키텍처의 `zstd` 압축 `squashfs` 파일 시스템에 대한 지원을 추가합니다.

Sun 파티션 테이블 지원

AL2023 는 더 이상 Sun 파티션 테이블 형식 (`CONFIG_SUN_PARTITION`) 을 지원하지 않습니다.

Amazon Linux 2와 Amazon Linux 2023 AMI에 설치된 패키지 비교

아마존 리눅스 2와 AL2023 표준 AMI에 있는 RPM을 비교한 것입니다.

패키지	AL2 AMI	2023년 AMI
<code>acl</code>	2.2.51	2.3.1
<code>acpid</code>	2.0.19	2.0.32
<code>alternatives</code>		1.15

패키지	AL2 AMI	2023년 AMI
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.4.1
amazon-linux-extras	2.0.3	
amazon-linux-extras-yum-plugin	2.0.3	
amazon-linux-repo-s3		2023.4.20240513
amazon-linux-sb-keys		2023.1
amazon-rpm-config		228
amazon-ssm-agent	3.3.131.0	3.3.380.0
at	3.1.13	3.1.23
attr	2.4.46	2.5.1
audit	2.8.1	3.0.6
audit-libs	2.8.1	3.0.6
authconfig	6.2.8	
aws-cfn-bootstrap	2.0	2.0
awscli	1.18.147	
awscli-2		2.15.30
basesystem	10.0	11
bash	4.2.46	5.2.15
bash-completion	2.1	2.11
bc	1.06.95	1.07.1

패키지	AL2 AMI	2023년 AMI
bind-export-libs	9.11.4	
bind-libs	9.11.4	9.16.48
bind-libs-lite	9.11.4	
bind-license	9.11.4	9.16.48
bind-utils	9.11.4	9.16.48
binutils	2.29.1	2.39
blktrace	1.0.5	
boost-date-time	1.53.0 (x86_64)	
boost-filesystem		1.75.0
boost-system	1.53.0 (x86_64)	1.75.0
boost-thread	1.53.0 (x86_64)	1.75.0
bridge-utils	1.5	
bzip2	1.0.6	1.0.8
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023.2.64	2023.2.64
c-ares		1.19.0
checkpolicy		3.4
chkconfig	1.7.4	1.15
chrony	4.2	4.3
cloud-init	19.3	22.2.2

패키지	AL2 AMI	2023년 AMI
cloud-init-cfg-ec2		22.2.2
cloud-utils-growpart	0.31	0.31
coreutils	8.22	8.32
coreutils-common		8.32
cpio	2.12	2.13
cracklib	2.9.0	2.9.6
cracklib-dicts	2.9.0	2.9.6
cronie	1.4.11	
cronie-anacron	1.4.11	
crontabs	1.11	1.11
crypto-policies		20220428
crypto-policies-scripts		20220428
cryptsetup	1.7.4	2.6.1
cryptsetup-libs	1.7.4	2.6.1
curl	8.3.0	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.26	2.1.27
cyrus-sasl-plain	2.1.26	2.1.27
dbus	1.10.24	1.12.28
dbus-broker		32

패키지	AL2 AMI	2023년 AMI
dbus-common		1.12.28
dbus-libs	1.10.24	1.12.28
device-mapper	1.02.170	1.02.185
device-mapper-event	1.02.170	
device-mapper-event-libs	1.02.170	
device-mapper-libs	1.02.170	1.02.185
device-mapper-persistent-data	0.7.3	
dhclient	4.2.5	
dhcp-common	4.2.5	
dhcp-libs	4.2.5	
diffutils	3.3	3.8
dmidecode	3.2	
dmraid	1.0.0.rc16	
dmraid-events	1.0.0.rc16	
dnf		4.14.0
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0

패키지	AL2 AMI	2023년 AMI
dnf-plugin-support-info		1.2
dnf-utils		4.3.0
dosfstools	3.0.20	4.2
dracut	033	055
dracut-config-ec2	2.0	3.0
dracut-config-generic	033	055
dwz		0.14
dyninst	9.3.1 (x86_64)	10.2.1
e2fsprogs	1.42.9	1.46.5
e2fsprogs-libs	1.42.9	1.46.5
ec2-hibinit-agent	1.0.8	1.0.8
ec2-instance-connect	1.1	1.1
ec2-instance-connect-selinux	1.1	1.1
ec2-net-utils	1.7.3	
ec2-utils	1.2	2.2.0
ed	1.9	1.14.2
efibootmgr	15 (aarch64)	
efi-filesystem		5

패키지	AL2 AMI	2023년 AMI
efi-srpm-macros		5
efivar		38
efivar-libs	31 (aarch64)	38
elfutils-debuginfod-client		0.188
elfutils-default-yama-scope	0.176	0.188
elfutils-libelf	0.176	0.188
elfutils-libs	0.176	0.188
ethtool	4.8	5.15
expat	2.1.0	2.5.0
file	5.11	5.39
file-libs	5.11	5.39
filesystem	3.2	3.14
findutils	4.5.11	4.8.0
fipscheck	1.4.1	
fipscheck-lib	1.4.1	
fonts-srpm-macros		2.0.5
freetype	2.8	
fstrm		0.6.1
fuse-libs	2.9.2	2.9.9

패키지	AL2 AMI	2023년 AMI
gawk	4.0.2	5.1.0
gdbm	1.13	
gdbm-libs		1.19
gdisk	0.8.10	1.0.8
generic-logos	18.0.0	
GeoIP	1.5.0	
gettext	0.19.8.1	0.21
gettext-libs	0.19.8.1	0.21
ghc-srpm-macros		1.5.0
glib2	2.56.1	2.74.7
glibc	2.26	2.34
glibc-all-langpacks	2.26	2.34
glibc-common	2.26	2.34
glibc-gconv-extra		2.34
glibc-locale-source	2.26	2.34
glibc-minimal-lang pack	2.26	
gmp	6.0.0	6.2.1
gnupg2	2.0.22	
gnupg2-minimal		2.3.7
gnutls		3.8.0

패키지	AL2 AMI	2023년 AMI
go-srpm-macros		3.2.0
gpgme	1.3.2	1.15.1
gpm-libs	1.20.7	1.20.7
grep	2.20	3.8
groff-base	1.22.2	1.22.4
grub2	2.06	
grub2-common	2.06	2.06
grub2-efi-aa64	2.06 (aarch64)	
grub2-efi-aa64-ec2	2.06 (아키텍처64)	2.06 (아키텍처64)
grub2-efi-aa64-modules	2.06 (노아크)	
grub2-efi-x64-ec2	2.06 (x86_64)	2.06 (x86_64)
grub2-pc	2.06 (x86_64)	
grub2-pc-modules	2.06 (노아크)	2.06
grub2-tools	2.06	2.06
grub2-tools-minimal	2.06	2.06
grubby	8.28	8.40
gssproxy	0.7.0	0.8.4
gzip	1.5	1.12
hardlink	1.3	
hibagent	1.1.0	

패키지	AL2 AMI	2023년 AMI
hostname	3.13	3.23
hunspell	1.3.2	1.7.0
hunspell-en	0.20121024	0.20140811.1
hunspell-en-GB	0.20121024	0.20140811.1
hunspell-en-US	0.20121024	0.20140811.1
hunspell-filesystem		1.7.0
hwdata	0.252	0.353
info	5.1	6.7
inih		49
initscripts	9.49.47	10.09
iproute	5.10.0	5.10.0
iptables	1.8.4	
iptables-libs	1.8.4	
iputils	20180629	20210202
irqbalance	1.7.0	1.9.0
jansson	2.10	2.14
jbigkit-libs	2.0	
jitterentropy		3.4.1
jq		1.7.1
json-c	0.11	0.14

패키지	AL2 AMI	2023년 AMI
kbd	1.15.5	2.4.0
kbd-legacy	1.15.5	
kbd-misc	1.15.5	2.4.0
kernel	5.10.215	6.1.90
kernel-livepatch-r epo-s3		2023.4.20240513
kernel-srpm-macros		1.0
kernel-tools	5.10.215	6.1.90
keyutils	1.5.8	1.6.3
keyutils-libs	1.5.8	1.6.3
kmod	25	29
kmod-libs	25	29
kpartx	0.4.9	
kpatch-runtime	0.9.4	0.9.7
krb5-libs	1.15.1	1.21
langtable	0.0.31	
langtable-data	0.0.31	
langtable-python	0.0.31	
less	458	608
libacl	2.2.51	2.3.1
libaio	0.3.109	0.3.111

패키지	AL2 AMI	2023년 AMI
libarchive		3.5.3
libargon2		20171227
libassuan	2.1.0	2.5.5
libattr	2.4.46	2.5.1
libbasicobjects	0.1.1	0.1.1
libblkid	2.30.2	2.37.4
libcap	2.54	2.48
libcap-ng	0.7.5	0.8.2
libcbor		0.7.0
libcollection	0.7.0	0.7.0
libcom_err	1.42.9	1.46.5
libcomps		0.1.20
libconfig	1.4.9	1.7.2
libcroco	0.6.12	
libcrypt	2.26	
libcurl	8.3.0	
libcurl-minimal		8.5.0
libdaemon	0.14	
libdb	5.3.21	5.3.28
libdb-utils	5.3.21	

패키지	AL2 AMI	2023년 AMI
libdhash		0.5.0
libdnf		0.69.0
libdrm	2.4.97	
libdwarf	20130207 (x86_64)	
libeconf		0.4.0
libedit	3.0	3.1
libestr	0.1.9	
libev		4.33
libevent	2.0.21	2.1.12
libfastjson	0.99.4	
libfdisk	2.30.2	2.37.4
libffi	3.0.13	3.4.4
libfido2		1.10.0
libgcc	7.3.1	11.4.1
libgcrypt	1.5.3	1.10.2
libgomp	7.3.1	11.4.1
libgpg-error	1.12	1.42
libibverbs		48.0
libicu	50.2	
libidn	1.28	

패키지	AL2 AMI	2023년 AMI
libidn2	2.3.0	2.3.2
libini_config	1.3.1	1.3.1
libjpeg-turbo	2.0.90	
libkcap1		1.4.0
libkcap1-hmacalc		1.4.0
libldb		2.6.2
libmaxminddb		1.5.2
libmetalink	0.1.3	0.1.3
libmnl	1.0.3	1.0.4
libmodulemd		2.13.0
libmount	2.30.2	2.37.4
libnetfilter_conntrack	1.0.6	
libnfnetlink	1.0.1	
libnfsidmap	0.25	2.5.4
libnghttp2	1.41.0	1.59.0
libnl3	3.2.28	3.5.0
libnl3-cli	3.2.28	
libpath_utils	0.2.1	0.2.1
libpcap	1.5.3	1.10.1
libpciaccess	0.14 (x86_64)	

패키지	AL2 AMI	2023년 AMI
libpipeline	1.2.3	1.5.3
libpkgconf		1.8.0
libpng	1.5.13	
libpsl	0.21.5	0.21.1
libpwquality	1.2.3	1.4.4
libref_array	0.1.5	0.1.5
librepo		1.14.5
libreport-filessystem		2.15.2
libseccomp	2.5.2	2.5.3
libselinux	2.5	3.4
libselinux-utils	2.5	3.4
libsemanage	2.5	3.4
libsepol	2.5	3.4
libsigsegv		2.13
libsmartcols	2.30.2	2.37.4
libsolv		0.7.22
libss	1.42.9	1.46.5
libssh2	1.4.3	
libsss_certmap		2.9.4
libsss_idmap	1.16.5	2.9.4

패키지	AL2 AMI	2023년 AMI
libsss_nss_idmap	1.16.5	2.9.4
libsss_sudo		2.9.4
libstdc++	7.3.1	11.4.1
libstoragegmt	1.6.1	1.9.4
libstoragegmt-python	1.6.1	
libstoragegmt-python-clibs	1.6.1	
libsysfs	2.1.0	
libtalloc		2.3.4
libtasn1	4.10	4.19.0
libtdb		1.4.7
libteam	1.27	
libtevent		0.13.0
libtextstyle		0.21
libtiff	4.0.3	
libtirpc	0.2.4	1.3.3
libunistring	0.9.3	0.9.10
libuser	0.60	0.63
libutempter	1.1.6	1.2.1
libuuid	2.30.2	2.37.4

패키지	AL2 AMI	2023년 AMI
libuv		1.47.0
libverto	0.2.5	0.3.2
libverto-libev		0.3.2
libverto-libevent	0.2.5	
libwebp	0.3.0	
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libxml2-python	2.9.1	
libyaml	0.1.4	0.2.5
libzstd		1.5.5
lm_sensors-libs	3.4.0	3.6.0
lmbd-libs		0.9.29
logrotate	3.8.6	3.20.1
lsof	4.87	4.94.0
lua	5.1.4	
lua-libs		5.4.4
lua-srpm-macros		1
lvm2	2.02.187	
lvm2-libs	2.02.187	
lz4	1.7.5	

패키지	AL2 AMI	2023년 AMI
lz4-libs		1.9.4
make	3.82	
man-db	2.6.3	2.9.3
man-pages	3.53	5.10
man-pages-overrides	7.5.2	
mariadb-libs	5.5.68	
mdadm	4.0	
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)
mlocate	0.26	
mpfr		4.1.0
mtr	0.92	
nano	2.9.8	5.8
ncurses	6.0	6.2
ncurses-base	6.0	6.2
ncurses-libs	6.0	6.2
nettle	2.7.1	3.8
net-tools	2.0	2.0
newt	0.52.15	0.52.21
newt-python	0.52.15	
nfs-utils	1.3.0	2.5.4

패키지	AL2 AMI	2023년 AMI
npth		1.6
nspr	4.35.0	4.35.0
nss	3.90.0	3.90.0
nss-pem	1.0.3	
nss-softokn	3.90.0	3.90.0
nss-softokn-freebl	3.90.0	3.90.0
nss-sysinit	3.90.0	3.90.0
nss-tools	3.90.0	
nss-util	3.90.0	3.90.0
ntsysv	1.7.4	1.15
numactl-libs	2.0.9	2.0.14
ocaml-srpm-macros		6
oniguruma		6.9.7.1
openblas-srpm-macros		2
openldap	2.4.44	2.4.57
openssh	7.4p1	8.7p1
openssh-clients	7.4p1	8.7p1
openssh-server	7.4p1	8.7p1
openssl	1.0.2k	3.0.8
openssl-libs	1.0.2k	3.0.8

패키지	AL2 AMI	2023년 AMI
openssl-pkcs11		0.4.12
os-prober	1.58	1.77
p11-kit	0.23.22	0.24.1
p11-kit-trust	0.23.22	0.24.1
package-notes-srpm-macros		0.4
pam	1.1.8	1.5.1
parted	3.1	3.4
passwd	0.79	0.80
pciutils	3.5.1	3.7.0
pciutils-libs	3.5.1	3.7.0
pcre	8.32	
pcre2	10.23	10.40
pcre2-syntax		10.40
perl	5.16.3	
perl-Carp	1.26	1.50
perl-Class-Struct		0.66
perl-constant	1.27	1.33
perl-DynaLoader		1.47
perl-Encode	2.51	3.15
perl-Errno		1.30

패키지	AL2 AMI	2023년 AMI
perl-Exporter	5.68	5.74
perl-Fcntl		1.13
perl-File-Basename		2.85
perl-File-Path	2.09	2.18
perl-File-stat		1.09
perl-File-Temp	0.23.01	0.231.100
perl-Filter	1.49	
perl-Getopt-Long	2.40	2.52
perl-Getopt-Std		1.12
perl-HTTP-Tiny	0.033	0.078
perl-if		0.60.800
perl-interpreter		5.32.1
perl-IO		1.43
perl-IPC-Open3		1.21
perl-libs	5.16.3	5.32.1
perl-macros	5.16.3	
perl-MIME-Base64		3.16
perl-mro		1.23
perl-overload		1.31
perl-overloading		0.02

패키지	AL2 AMI	2023년 AMI
perl-parent	0.225	0.238
perl-PathTools	3.40	3.78
perl-Pod-Escapes	1.04	1.07
perl-podlators	2.5.1	4.14
perl-Pod-Perldoc	3.20	3.28.01
perl-Pod-Simple	3.28	3.42
perl-Pod-Usage	1.63	2.01
perl-POSIX		1.94
perl-Scalar-List-Utils	1.27	1.56
perl-SelectSaver		1.02
perl-Socket	2.010	2.032
perl-srpm-macros		1
perl-Storable	2.45	3.21
perl-subst		1.03
perl-Symbol		1.08
perl-Term-ANSIColor		5.01
perl-Term-Cap		1.17
perl-Text-ParseWords	3.29	3.30
perl-Text-Tabs+Wrap		2021.0726
perl-threads	1.87	

패키지	AL2 AMI	2023년 AMI
perl-threads-shared	1.43	
perl-Time-HiRes	1.9725	
perl-Time-Local	1.2300	1.300
perl-vars		1.05
pinentry	0.8.1	
pkgconf		1.8.0
pkgconfig	0.27.1	
pkgconf-m4		1.8.0
pkgconf-pkg-config		1.8.0
plymouth	0.8.9	
plymouth-core-libs	0.8.9	
plymouth-scripts	0.8.9	
pm-utils	1.4.1	
policycoreutils	2.5	3.4
policycoreutils-python-utils		3.4
popt	1.13	1.18
postfix	2.10.1	
procps-ng	3.3.10	3.3.17
protobuf-c		1.4.1
psacct	6.6.1	6.6.4

패키지	AL2 AMI	2023년 AMI
psmisc	22.20	23.4
pth	2.0.7	
publicsuffix-list-dafsa	20240208	20240212
pygpme	0.3	
pyliblzma	0.5.3	
pystache	0.5.3	
python	2.7.18	
python2-botocore	1.18.6	
python2-colorama	0.3.9	
python2-cryptography	1.7.2	
python2-dateutil	2.6.1	
python2-futures	3.0.5	
python2-jmespath	0.9.3	
python2-jsonschema	2.5.1	
python2-oauthlib	2.0.1	
python2-pyasn1	0.1.9	
python2-rpm	4.11.3	
python2-rsa	3.4.1	
python2-s3transfer	0.3.3	
python2-setuptools	41.2.0	

패키지	AL2 AMI	2023년 AMI
python2-six	1.11.0	
python3	3.7.16	3.9.16
python3-attrs		20.3.0
python3-audit		3.0.6
python3-awscrt		0.19.19
python3-babel		2.9.1
python3-cffi		1.14.5
python3-chardet		4.0.0
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36.0.1
python3-daemon	2.2.3	2.3.0
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils	0.14	0.16
python3-gpg		1.15.1
python3-hawkey		0.69.0

패키지	AL2 AMI	2023년 AMI
python3-idna		2.10
python3-jinja2		2.11.3
python3-jmespath		0.10.0
python3-jsonpatch		1.21
python3-jsonpointer		2.0
python3-jsonschema		3.2.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0
python3-libs	3.7.16	3.9.16
python3-libselenium		3.4
python3-libsemanage		3.4
python3-libstorage mgmt		1.9.4
python3-lockfile	0.11.0	0.12.2
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip	20.2.2	
python3-pip-wheel		21.3.1
python3-ply		3.11

패키지	AL2 AMI	2023년 AMI
python3-policycore utils		3.4
python3-prettytable		0.7.2
python3-prompt-too lkit		3.0.24
python3-pycparser		2.20
python3-pyrsistent		0.17.3
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pystache	0.5.4	
python3-pytz		2022.7.1
python3-pyyaml		5.4.1
python3-requests		2.25.1
python3-rpm		4.16.1.3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml- clib		0.1.2
python3-setools		4.4.1
python3-setuptools	49.1.3	59.6.0
python3-setuptools- wheel		59.6.0
python3-simplejson	3.2.0	

패키지	AL2 AMI	2023년 AMI
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1.25.10
python3-wcwidth		0.2.5
python-babel	0.9.6	
python-backports	1.0	
python-backports-s sl_match_hostname	3.5.0.1	
python-cffi	1.6.0	
python-chardet	2.2.1	
python-chevron		0.13.1
python-configobj	4.7.2	
python-daemon	1.6	
python-devel	2.7.18	
python-docutils	0.12	
python-enum34	1.0.4	
python-idna	2.4	
python-iniparse	0.4	
python-ipaddress	1.0.16	
python-jinja2	2.7.2	
python-jsonpatch	1.2	

패키지	AL2 AMI	2023년 AMI
python-jsonpointer	1.9	
python-jwcrypto	0.4.2	
python-kitchen	1.1.1	
python-libs	2.7.18	
python-lockfile	0.9.1	
python-markupsafe	0.11	
python-pillow	2.0.0	
python-ply	3.4	
python-pycparser	2.14	
python-pycurl	7.19.0	
python-repoze-lru	0.4	
python-requests	2.6.0	
python-simplejson	3.2.0	
python-srpm-macros		3.9
python-urlgrabber	3.10	
python-urllib3	1.25.9	
pyxattr	0.5.1	
PyYAML	3.10	
qrencode-libs	3.4.1	
quota	4.01	4.06

패키지	AL2 AMI	2023년 AMI
quota-nls	4.01	4.06
rdate	1.4	
readline	6.2	8.1
rng-tools	6.8	6.14
rootfiles	8.1	8.1
rpcbind	0.2.0	1.2.6
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
rsync	3.1.2	3.2.6
rsyslog	8.24.0	
rust-srpm-macros		21
sbsigntools		0.9.4
scl-utils	20130529	
screen	4.1.0	4.8.0
sed	4.2.2	4.8
selinux-policy	3.13.1	37.22

패키지	AL2 AMI	2023년 AMI
selinux-policy-targeted	3.13.1	37.22
setserial	2.17	
setup	2.8.71	2.13.7
setuptools	1.19.11	
sgpio	1.2.0.10	
shadow-utils	4.1.5.1	4.9
shared-mime-info	1.8	
slang	2.2.4	2.3.2
sqlite	3.7.17	
sqlite-libs		3.40.0
sssd-client	1.16.5	2.9.4
sssd-common		2.9.4
sssd-kcm		2.9.4
sssd-nfs-idmap		2.9.4
strace	4.26	6.8
sudo	1.8.23	1.9.15
sysctl-defaults	1.0	1.0
sysstat	10.1.5	12.5.6
systemd	219	252.16
systemd-libs	219	252.16

패키지	AL2 AMI	2023년 AMI
systemd-networkd		252.16
systemd-pam		252.16
systemd-resolved		252.16
systemd-sysv	219	
systemd-udev		252.16
system-release	2	2023.4.20240513
systemtap-runtime	4.5	4.8
sysvinit-tools	2.88	
tar	1.26	1.34
tbb		2020.3
tcp_wrappers	7.6	
tcp_wrappers-libs	7.6	
tcpdump	4.9.2	4.99.1
tcsh	6.18.01	6.24.07
teamd	1.27	
time	1.7	1.9
traceroute	2.0.22	2.1.3
tzdata	2024a	2024a
unzip	6.0	6.0
update-motd	1.1.2	2.2

패키지	AL2 AMI	2023년 AMI
usermode	1.111	
userspace-rcu		0.12.1
ustr	1.0.4	
util-linux	2.30.2	2.37.4
util-linux-core		2.37.4
vim-common	9.0.2153	9.0.2153
vim-data	9.0.2153	9.0.2153
vim-enhanced	9.0.2153	9.0.2153
vim-filesystem	9.0.2153	9.0.2153
vim-minimal	9.0.2153	9.0.2153
virt-what	1.18	
wget	1.14	1.21.3
which	2.20	2.21
words	3.0	3.0
xfsdump	3.1.8	3.1.11
xfspgrog	5.0.0	5.18.0
xxd	9.0.2153	9.0.2153
xxhash-libs		0.8.0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5

패키지	AL2 AMI	2023년 AMI
yajl	2.0.4	
yum	3.4.3	4.14.0
yum-langpacks	0.4.2	
yum-metadata-parser	1.1.4	
yum-plugin-priorities	1.1.31	
yum-utils	1.1.31	
zip	3.0	3.0
zlib	1.2.7	1.2.11
zram-generator		1.1.2
zram-generator-defaults		1.1.2
zstd		1.5.5

Amazon Linux 2 및 Amazon Linux 2023 미니멀 AMI에 설치된 패키지 비교

아마존 리눅스 2와 AL2023 미니멀 AMI에 있는 RPM을 비교한 것입니다.

패키지	AL2 미니멀	AL2023 미니멀
acl	2.2.51	
alternatives		1.15
amazon-chrony-config		4.3

패키지	AL2 미니멀	AL2023 미니멀
amazon-ec2-net-utils		2.4.1
amazon-linux-extras	2.0.3	
amazon-linux-repo-s3		2023.4.20240513
amazon-linux-sb-keys		2023.1
audit	2.8.1	3.0.6
audit-libs	2.8.1	3.0.6
authconfig	6.2.8	
awscli-2		2.15.30
basesystem	10.0	11
bash	4.2.46	5.2.15
bind-export-libs	9.11.4	
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023.2.64	2023.2.64
checkpolicy		3.4
chkconfig	1.7.4	
chrony	4.2	4.3
cloud-init	19.3	22.2.2
cloud-init-cfg-ec2		22.2.2
cloud-utils-growpart	0.31	0.31
coreutils	8.22	8.32

패키지	AL2 미니멀	AL2023 미니멀
coreutils-common		8.32
cpio	2.12	2.13
cracklib	2.9.0	2.9.6
cracklib-dicts	2.9.0	2.9.6
cronie	1.4.11	
cronie-anacron	1.4.11	
crontabs	1.11	
crypto-policies		20220428
cryptsetup-libs	1.7.4	2.6.1
curl	8.3.0	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.26	2.1.27
dbus	1.10.24	1.12.28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.10.24	1.12.28
device-mapper	1.02.170	1.02.185
device-mapper-libs	1.02.170	1.02.185
dhclient	4.2.5	
dhcp-common	4.2.5	

패키지	AL2 미니멀	AL2023 미니멀
dhcp-libs	4.2.5	
diffutils	3.3	3.8
dnf		4.14.0
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0
dnf-plugin-support-info		1.2
dracut	033	055
dracut-config-ec2	2.0	3.0
dracut-config-generic	033	055
e2fsprogs	1.42.9	1.46.5
e2fsprogs-libs	1.42.9	1.46.5
ec2-utils	1.2	2.2.0
efibootmgr	15 (aarch64)	
efi-filesystem		5
efivar		38
efivar-libs	31 (aarch64)	38
elfutils-default-yama-scope	0.176	0.188

패키지	AL2 미니멀	AL2023 미니멀
elfutils-libelf	0.176	0.188
elfutils-libs	0.176	0.188
expat	2.1.0	2.5.0
file	5.11	5.39
file-libs	5.11	5.39
filesystem	3.2	3.14
findutils	4.5.11	4.8.0
fipscheck	1.4.1	
fipscheck-lib	1.4.1	
freetype	2.8	
fuse-libs	2.9.2	2.9.9
gawk	4.0.2	5.1.0
gdbm	1.13	
gdbm-libs		1.19
gdisk	0.8.10	1.0.8
gettext	0.19.8.1	0.21
gettext-libs	0.19.8.1	0.21
glib2	2.56.1	2.74.7
glibc	2.26	2.34
glibc-all-langpacks	2.26	2.34

패키지	AL2 미니멀	AL2023 미니멀
glibc-common	2.26	2.34
glibc-locale-source	2.26	2.34
glibc-minimal-langpack	2.26	
gmp	6.0.0	6.2.1
gnupg2	2.0.22	
gnupg2-minimal		2.3.7
gnutls		3.8.0
gpgme	1.3.2	1.15.1
grep	2.20	3.8
groff-base	1.22.2	1.22.4
grub2	2.06	
grub2-common	2.06	2.06
grub2-efi-aa64	2.06 (aarch64)	
grub2-efi-aa64-ec2	2.06 (아키텍처64)	2.06 (아키텍처64)
grub2-efi-aa64-modules	2.06 (노아크)	
grub2-efi-x64-ec2	2.06 (x86_64)	2.06 (x86_64)
grub2-pc	2.06 (x86_64)	
grub2-pc-modules	2.06 (노아크)	2.06
grub2-tools	2.06	2.06

패키지	AL2 미니멀	AL2023 미니멀
grub2-tools-minimal	2.06	2.06
grubby	8.28	8.40
gzip	1.5	1.12
hardlink	1.3	
hostname	3.13	3.23
hwdata		0.353
info	5.1	
inih		49
initscripts	9.49.47	10.09
iproute	5.10.0	5.10.0
iptables	1.8.4	
iptables-libs	1.8.4	
iputils	20180629	20210202
irqbalance	1.7.0	1.9.0
jansson		2.14
jitterentropy		3.4.1
jq		1.7.1
json-c		0.14
kbd		2.4.0
kbd-misc		2.4.0

패키지	AL2 미니멀	AL2023 미니멀
kernel	4.14.343	6.1.90
kernel-livepatch-r epo-s3		2023.4.20240513
keyutils-libs	1.5.8	1.6.3
kmod	25	29
kmod-libs	25	29
kpartx	0.4.9	
krb5-libs	1.15.1	1.21
less	458	608
libacl	2.2.51	2.3.1
libarchive		3.5.3
libargon2		20171227
libassuan	2.1.0	2.5.5
libattr	2.4.46	2.5.1
libblkid	2.30.2	2.37.4
libcap	2.54	2.48
libcap-ng	0.7.5	0.8.2
libcbor		0.7.0
libcom_err	1.42.9	1.46.5
libcomps		0.1.20
libcroco	0.6.12	

패키지	AL2 미니멀	AL2023 미니멀
libcrypt	2.26	
libcurl	8.3.0	
libcurl-minimal		8.5.0
libdb	5.3.21	5.3.28
libdb-utils	5.3.21	
libdnf		0.69.0
libeconf		0.4.0
libedit	3.0	3.1
libestr	0.1.9	
libfastjson	0.99.4	
libfdisk	2.30.2	2.37.4
libffi	3.0.13	3.4.4
libfido2		1.10.0
libgcc	7.3.1	11.4.1
libgcrypt	1.5.3	1.10.2
libgomp	7.3.1	11.4.1
libgpg-error	1.12	1.42
libicu	50.2	
libidn	1.28	
libidn2	2.3.0	2.3.2

패키지	AL2 미니멀	AL2023 미니멀
libkcapi		1.4.0
libkcapi-hmaccalc		1.4.0
libmetalink	0.1.3	
libmnl	1.0.3	1.0.4
libmodulemd		2.13.0
libmount	2.30.2	2.37.4
libnetfilter_connt rack	1.0.6	
libnfnetworking	1.0.1	
libnghttp2	1.41.0	1.59.0
libpcap	1.5.3	
libpipeline	1.2.3	1.5.3
libpng	1.5.13	
libpsl	0.21.5	0.21.1
libpwquality	1.2.3	1.4.4
librepo		1.14.5
libreport-filesystem		2.15.2
libseccomp	2.5.2	2.5.3
libselinux	2.5	3.4
libselinux-utils	2.5	3.4
libsemanage	2.5	3.4

패키지	AL2 미니멀	AL2023 미니멀
libsepol	2.5	3.4
libsigsegv		2.13
libsmartcols	2.30.2	2.37.4
libsolv		0.7.22
libss	1.42.9	1.46.5
libssh2	1.4.3	
libstdc++	7.3.1	11.4.1
libsysfs	2.1.0	
libtasn1	4.10	4.19.0
libtextstyle		0.21
libunistring	0.9.3	0.9.10
libuser	0.60	0.63
libutempter	1.1.6	1.2.1
libuuid	2.30.2	2.37.4
libverto	0.2.5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libyaml	0.1.4	0.2.5
libzstd		1.5.5
logrotate	3.8.6	3.20.1

패키지	AL2 미니멀	AL2023 미니멀
lua	5.1.4	
lua-libs		5.4.4
lz4	1.7.5	
lz4-libs		1.9.4
make	3.82	
man-db	2.6.3	2.9.3
mariadb-libs	5.5.68	
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)
mpfr		4.1.0
ncurses	6.0	6.2
ncurses-base	6.0	6.2
ncurses-libs	6.0	6.2
nettle	2.7.1	3.8
net-tools	2.0	2.0
newt	0.52.15	
newt-python	0.52.15	
npth		1.6
nspr	4.35.0	
nss	3.90.0	
nss-pem	1.0.3	

패키지	AL2 미니멀	AL2023 미니멀
nss-softokn	3.90.0	
nss-softokn-freebl	3.90.0	
nss-sysinit	3.90.0	
nss-tools	3.90.0	
nss-util	3.90.0	
numactl-libs	2.0.9	2.0.14
oniguruma		6.9.7.1
openldap	2.4.44	2.4.57
openssh	7.4p1	8.7p1
openssh-clients	7.4p1	8.7p1
openssh-server	7.4p1	8.7p1
openssl	1.0.2k	3.0.8
openssl-libs	1.0.2k	3.0.8
openssl-pkcs11		0.4.12
os-prober	1.58	1.77
p11-kit	0.23.22	0.24.1
p11-kit-trust	0.23.22	0.24.1
pam	1.1.8	1.5.1
passwd	0.79	0.80
pciutils		3.7.0

패키지	AL2 미니멀	AL2023 미니멀
pciutils-libs		3.7.0
pcre	8.32	
pcre2	10.23	10.40
pcre2-syntax		10.40
pinentry	0.8.1	
pkgconfig	0.27.1	
policycoreutils	2.5	3.4
popt	1.13	1.18
postfix	2.10.1	
procps-ng	3.3.10	3.3.17
psmisc	22.20	23.4
pth	2.0.7	
publicsuffix-list-dafsa	20240208	20240212
pygpgme	0.3	
pyliblzma	0.5.3	
python	2.7.18	
python2-cryptography	1.7.2	
python2-jsonschema	2.5.1	
python2-oauthlib	2.0.1	
python2-pyasn1	0.1.9	

패키지	AL2 미니멀	AL2023 미니멀
python2-rpm	4.11.3	
python2-setuptools	41.2.0	
python2-six	1.11.0	
python3		3.9.16
python3-attrs		20.3.0
python3-audit		3.0.6
python3-awscrt		0.19.19
python3-babel		2.9.1
python3-cffi		1.14.5
python3-chardet		4.0.0
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36.0.1
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils		0.16
python3-gpg		1.15.1

패키지	AL2 미니멀	AL2023 미니멀
python3-hawkey		0.69.0
python3-idna		2.10
python3-jinja2		2.11.3
python3-jmespath		0.10.0
python3-jsonpatch		1.21
python3-jsonpointer		2.0
python3-jsonschema		3.2.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0
python3-libs		3.9.16
python3-libseltlinux		3.4
python3-libsemanage		3.4
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip-wheel		21.3.1
python3-ply		3.11
python3-policycore utils		3.4
python3-prettytable		0.7.2

패키지	AL2 미니멀	AL2023 미니멀
python3-prompt-toolkit		3.0.24
python3-pycparser		2.20
python3-pyrsistent		0.17.3
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pytz		2022.7.1
python3-pyyaml		5.4.1
python3-requests		2.25.1
python3-rpm		4.16.1.3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools		59.6.0
python3-setuptools-wheel		59.6.0
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1.25.10
python3-wcwidth		0.2.5

패키지	AL2 미니멀	AL2023 미니멀
python-babel	0.9.6	
python-backports	1.0	
python-backports-s sl_match_hostname	3.5.0.1	
python-cffi	1.6.0	
python-chardet	2.2.1	
python-configobj	4.7.2	
python-devel	2.7.18	
python-enum34	1.0.4	
python-idna	2.4	
python-iniparse	0.4	
python-ipaddress	1.0.16	
python-jinja2	2.7.2	
python-jsonpatch	1.2	
python-jsonpointer	1.9	
python-jwcrypto	0.4.2	
python-libs	2.7.18	
python-markupsafe	0.11	
python-ply	3.4	
python-pycparser	2.14	
python-pycurl	7.19.0	

패키지	AL2 미니멀	AL2023 미니멀
python-repoze-lru	0.4	
python-requests	2.6.0	
python-urlgrabber	3.10	
python-urllib3	1.25.9	
pyxattr	0.5.1	
PyYAML	3.10	
qrencode-libs	3.4.1	
readline	6.2	8.1
rng-tools	6.8	6.14
rootfiles	8.1	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
rsyslog	8.24.0	
sbsigntools		0.9.4
sed	4.2.2	4.8
selinux-policy	3.13.1	37.22

패키지	AL2 미니멀	AL2023 미니멀
selinux-policy-targeted	3.13.1	37.22
setup	2.8.71	2.13.7
shadow-utils	4.1.5.1	4.9
shared-mime-info	1.8	
slang	2.2.4	
sqlite	3.7.17	
sqlite-libs		3.40.0
sudo	1.8.23	1.9.15
sysctl-defaults	1.0	1.0
systemd	219	252.16
systemd-libs	219	252.16
systemd-networkd		252.16
systemd-pam		252.16
systemd-resolved		252.16
systemd-sysv	219	
systemd-udev		252.16
system-release	2	2023.4.20240513
sysvinit-tools	2.88	
tar	1.26	1.34
tcp_wrappers-libs	7.6	

패키지	AL2 미니멀	AL2023 미니멀
tzdata	2024a	2024a
update-motd	1.1.2	2.2
userspace-rcu		0.12.1
ustr	1.0.4	
util-linux	2.30.2	2.37.4
util-linux-core		2.37.4
vim-data	9.0.2153	9.0.2153
vim-minimal	9.0.2153	9.0.2153
which	2.20	2.21
xfspgrog	5.0.0	5.18.0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	1.1.4	
yum-plugin-priorities	1.1.31	
zlib	1.2.7	1.2.11
zram-generator		1.1.2
zram-generator-defaults		1.1.2
zstd		1.5.5

Amazon Linux 2 및 Amazon Linux 2023 기본 컨테이너 이미지에 설치된 패키지 비교

아마존 리눅스 2와 AL2023 기본 컨테이너 이미지에 있는 RPM을 비교한 것입니다.

패키지	AL2 컨테이너	AL2023 컨테이너
alternatives		1.15
amazon-linux-extras	2.0.3	
amazon-linux-repo-cdn		2023.4.20240513
audit-libs		3.0.6
basesystem	10.0	11
bash	4.2.46	5.2.15
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023.2.64	2023.2.64
chkconfig	1.7.4	
coreutils	8.22	
coreutils-single		8.32
cpio	2.12	
crypto-policies		20220428
curl	8.3.0	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.26	

패키지	AL2 컨테이너	AL2023 컨테이너
diffutils	3.3	
dnf		4.14.0
dnf-data		4.14.0
elfutils-default-yama-scope		0.188
elfutils-libelf	0.176	0.188
elfutils-libs		0.188
expat	2.1.0	2.5.0
file-libs	5.11	5.39
filesystem	3.2	3.14
findutils	4.5.11	
gawk	4.0.2	5.1.0
gdbm	1.13	
gdbm-libs		1.19
glib2	2.56.1	2.74.7
glibc	2.26	2.34
glibc-common	2.26	2.34
glibc-langpack-en	2.26	
glibc-minimal-langpack	2.26	2.34
gmp	6.0.0	6.2.1

패키지	AL2 컨테이너	AL2023 컨테이너
gnupg2	2.0.22	
gnupg2-minimal		2.3.7
gpgme	1.3.2	1.15.1
grep	2.20	3.8
info	5.1	
json-c		0.14
keyutils-libs	1.5.8	1.6.3
krb5-libs	1.15.1	1.21
libacl	2.2.51	2.3.1
libarchive		3.5.3
libassuan	2.1.0	2.5.5
libattr	2.4.46	2.5.1
libblkid	2.30.2	2.37.4
libcap	2.54	2.48
libcap-ng		0.8.2
libcom_err	1.42.9	1.46.5
libcomps		0.1.20
libcrypt	2.26	
libcurl	8.3.0	
libcurl-minimal		8.5.0

패키지	AL2 컨테이너	AL2023 컨테이너
libdb	5.3.21	
libdb-utils	5.3.21	
libdnf		0.69.0
libffi	3.0.13	3.4.4
libgcc	7.3.1	11.4.1
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.12	1.42
libidn2	2.3.0	2.3.2
libmetalink	0.1.3	
libmodulemd		2.13.0
libmount	2.30.2	2.37.4
libnghttp2	1.41.0	1.59.0
libpsl	0.21.5	0.21.1
librepo		1.14.5
libreport-filessystem		2.15.2
libselinux	2.5	3.4
libsepol	2.5	3.4
libsigsegv		2.13
libsmartcols		2.37.4

패키지	AL2 컨테이너	AL2023 컨테이너
libsolv		0.7.22
libssh2	1.4.3	
libstdc++	7.3.1	11.4.1
libtasn1	4.10	4.19.0
libunistring	0.9.3	0.9.10
libuuid	2.30.2	2.37.4
libverto	0.2.5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libyaml		0.2.5
libzstd		1.5.5
lua	5.1.4	
lua-libs		5.4.4
lz4-libs		1.9.4
mpfr		4.1.0
ncurses	6.0	
ncurses-base	6.0	6.2
ncurses-libs	6.0	6.2
npth		1.6
nspr	4.35.0	

패키지	AL2 컨테이너	AL2023 컨테이너
nss	3.90.0	
nss-pem	1.0.3	
nss-softokn	3.90.0	
nss-softokn-freebl	3.90.0	
nss-sysinit	3.90.0	
nss-tools	3.90.0	
nss-util	3.90.0	
openldap	2.4.44	
openssl-lib	1.0.2k	3.0.8
p11-kit	0.23.22	0.24.1
p11-kit-trust	0.23.22	0.24.1
pcre	8.32	
pcre2		10.40
pcre2-syntax		10.40
pinentry	0.8.1	
popt	1.13	1.18
pth	2.0.7	
publicsuffix-list-dafsa	20240208	20240212
pygpgme	0.3	
pyliblzma	0.5.3	

패키지	AL2 컨테이너	AL2023 컨테이너
python	2.7.18	
python2-rpm	4.11.3	
python3		3.9.16
python3-dnf		4.14.0
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0
python3-libs		3.9.16
python3-pip-wheel		21.3.1
python3-rpm		4.16.1.3
python3-setuptools-wheel		59.6.0
python-iniparse	0.4	
python-libs	2.7.18	
python-pycurl	7.19.0	
python-urlgrabber	3.10	
pyattr	0.5.1	
readline	6.2	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3

패키지	AL2 컨테이너	AL2023 컨테이너
rpm-libs	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
sed	4.2.2	4.8
setup	2.8.71	2.13.7
shared-mime-info	1.8	
sqlite	3.7.17	
sqlite-libs		3.40.0
system-release	2	2023.4.20240513
tzdata	2024a	2024a
vim-data	9.0.2153	
vim-minimal	9.0.2153	
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	1.1.4	
yum-plugin-ovl	1.1.31	
yum-plugin-priorities	1.1.31	
zlib	1.2.7	1.2.11

AL1과 AL2023 비교

다음 항목에서는 [AL2와의 비교에서](#) 아직 다루지 않은 AL1과 AL2023 간의 주요 차이점에 대해 설명합니다.

Note

AL1은 2023년 12월 31일에 EOL end-of-life (EOL) 에 도달했으며 2024년 1월 1일부터 보안 업데이트나 버그 수정을 받을 수 없습니다. AL1 EOL 및 유지 관리 지원에 대한 자세한 내용은 [Amazon Linux AMI에서의 업데이트](#) 블로그 게시물을 참조하십시오. end-of-life 애플리케이션을 AL2023으로 업그레이드하는 것이 좋습니다. 이 버전은 2028년까지 장기 지원을 포함합니다.

주제

- [각 릴리스에 대한 지원](#)
- [systemd는 upstart를 init 시스템으로 대체합니다.](#)
- [Python 2.6 및 2.7은 Python 3으로 대체되었습니다.](#)
- [OpenJDK 8은 가장 오래된 JDK](#)
- [아마존 리눅스 1 \(AL1\) 에서의 AL2023 커널 변경](#)
- [Amazon Linux 1\(AL1\) 과 Amazon Linux 2023 AMI에 설치된 패키지 비교](#)
- [Amazon Linux 1\(AL1\)과 Amazon Linux 2023 미니멀 AMI에 설치된 패키지 비교](#)
- [Amazon Linux 1\(AL1\) 과 Amazon Linux 2023의 기본 컨테이너 이미지에 설치된 패키지 비교](#)

각 릴리스에 대한 지원

AL2023 관련 지원은 출시일로부터 5년간 제공됩니다. AL1은 2020년 12월 31일에 표준 지원을 종료하고 2023년 12월 31일에 유지 관리 지원을 종료했습니다.

자세한 정보는 [릴리스 케이던스](#)를 참조하세요.

systemd는 upstart를 init 시스템으로 대체합니다.

AL2에서는 systemd AS 시스템으로 upstart 대체되었습니다. init AL2023 또한 systemd init 시스템으로 사용하여 의 새로운 특징과 기능을 더욱 채택하고 있습니다. systemd

Python 2.6 및 2.7은 Python 3으로 대체되었습니다.

AL1이 2018.03 릴리스에서 Python 2.6을 EOL로 표시했지만 패키지는 여전히 리포지토리에서 설치할 수 있었습니다. AL2는 가장 먼저 지원되는 파이썬 버전으로 Python 2.7과 함께 제공되었으며, AL2023 버전은 Python 3으로의 전환을 완료했습니다. AL2023 리포지토리에는 Python 2.x 버전이 포함되어 있지 않습니다.

Amazon Linux의 Python에 대한 자세한 내용은 [AL2023에서 Python 사용](#)을 참조하십시오.

OpenJDK 8은 가장 오래된 JDK

AL2023은 기본 (그리고 유일한) Java 개발 키트(JDK)로 [Amazon Corretto](#)를 지원합니다. AL2023 내 모든 Java 기반 패키지는 다음과 같이 빌드됩니다. Amazon Corretto 17

AL1에서는 OpenJDK 1.6.0 java-1.6.0-openjdk () 이 첫 번째 2018.03 릴리스와 함께 단종되었고, OpenJDK 1.7.0 () 은 2020년 중반에 단종되었습니다. 하지만 AL1 리포지토리에서 두 버전 모두 사용할 수 있었습니다. java-1.7.0-openjdk. AL2023 버전에서 사용할 수 있는 가장 초기의 OpenJDK 버전은 에서 제공하는 OpenJDK 8입니다. Amazon Corretto 8

아마존 리눅스 1 (AL1) 에서의 AL2023 커널 변경

커널 라이브 패치

AL2023 및 AL2 모두 커널 라이브 패치 기능에 대한 지원을 추가합니다. 이를 통해 재부팅이나 다운타임 없이 Linux 커널의 중요하고 중요한 보안 취약성을 패치할 수 있습니다. 자세한 정보는 [AL2023 기반 커널 라이브 패칭](#)을 참조하세요.

커널 파일 시스템 지원

AL1의 커널이 마운트를 지원하는 파일 시스템이 몇 가지 변경되었으며 커널이 파싱할 파티션 구성표도 변경되었습니다.

CONFIG 옵션	AL1/4.14/x86_64	AL2023/6.1/AArch64	AL2023/6.1/x86_64
CONFIG_AFS_FS	m	n	n
CONFIG_AF_RXRPC	m	n	n

CONFIG 옵션	AL1/4.14/x86_64	AL2023/6.1/AArch64	AL2023/6.1/x86_64
<u>CONFIG_BS D_DISKLABEL</u>	y	n	n
<u>CONFIG_CRAMFS</u>	m	n	n
<u>CONFIG_CRAMFS_BLOCKDEV</u>	N/A	해당 사항 없음	해당 사항 없음
<u>CONFIG_DM_CLONE</u>	해당 사항 없음	n	n
<u>CONFIG_DM_ERA</u>	n	n	n
<u>CONFIG_DM_INTEGRITY</u>	m	m	m
<u>CONFIG_DM_LOG_WRITES</u>	n	m	m
<u>CONFIG_DM_SWITCH</u>	n	n	n
<u>CONFIG_DM_VERITY</u>	n	n	n
<u>CONFIG_ECRYPT_FS</u>	m	n	n
<u>CONFIG_EXFAT_FS</u>	N/A	m	m
<u>CONFIG_EXT2_FS</u>	m	n	n
<u>CONFIG_EXT3_FS</u>	m	n	n
<u>CONFIG_GFS2_FS</u>	n	n	n
<u>CONFIG_HFSPLUS_FS</u>	m	n	n

CONFIG 옵션	AL1/4.14/x86_64	AL2023/6.1/AArch64	AL2023/6.1/x86_64
CONFIG_HFS_FS	m	n	n
CONFIG_JFS_FS	m	n	n
CONFIG_LDM_PARTITION	y	n	n
CONFIG_MAC_PARTITION	y	n	n
CONFIG_NFS_V2	m	n	n
CONFIG_NTFS_FS	m	n	n
CONFIG_ROMFS_FS	m	n	n
CONFIG_SOLARIS_X86_PARTITION	y	n	n
CONFIG_SQUASHFS_ZSTD	y	y	y
CONFIG_SUN_PARTITION	y	n	n

보안에 중점을 둔 커널 구성 변경

CONFIG 옵션	AL1/4.14/x86_64	AL2023/6.1/AArch64	AL2023/6.1/x86_64
CONFIG_BUG_ON_DATA_CORRUPTION	y	y	y

CONFIG 옵션	AL1/4.14/x86_64	AL2023/6.1/AArch64	AL2023/6.1/x86_64
<u>CONFIG_DEF</u> <u>FAULT_MMA</u> <u>P_MIN_ADDR</u>	4096	65536	65536
<u>CONFIG_DEVMEM</u>	y	n	n
<u>CONFIG_DEVPOR</u>	y	n	n
<u>CONFIG_FO</u> <u>RTIFY_SOURCE</u>	y	y	y
<u>CONFIG_HA</u> <u>RDENED_US</u> <u>ERCOPY_FA</u> <u>LLBACK</u>	N/A	해당 사항 없음	해당 사항 없음
<u>CONFIG_IN</u> <u>IT_ON_ALL</u> <u>OC_DEFAULT_ON</u>	해당 사항 없음	n	n
<u>CONFIG_IN</u> <u>IT_ON_FRE</u> <u>E_DEFAULT_ON</u>	해당 사항 없음	n	n
<u>CONFIG_IO</u> <u>MMU_DEFAU</u> <u>LT_DMA_STRICT</u>	해당 사항 없음	n	n
<u>CONFIG_LD</u> <u>ISC_AUTOLOAD</u>	y	n	n
<u>CONFIG_SC</u> <u>HED_CORE</u>	해당 사항 없음	해당 사항 없음	y
<u>CONFIG_SC</u> <u>HED_STACK</u> <u>_END_CHECK</u>	y	y	y

CONFIG 옵션	AL1/4.14/x86_64	AL2023/6.1/AArch64	AL2023/6.1/x86_64
CONFIG_SECURITY_DMESG_RESTRICT	n	y	y
CONFIG_SECURITY_LINUX_DISABLE	y	n	n
CONFIG_SHUFFLE_PAGE_ALLOCATOR	N/A	y	y
CONFIG_SLAB_FREELIST_HARDENED	y	y	y
CONFIG_SLAB_FREELIST_RANDOM	n	y	y

기타 커널 구성 변경

CONFIG 옵션	AL1/4.14/x86_64	AL2023/6.1/AArch64	AL2023/6.1/x86_64
CONFIG_HZ	250	100	100
CONFIG_NR_CPUS	8192	512	512
CONFIG_PANIC_ON_OOPS	n	y	y
CONFIG_PANIC_ON_OOPS_VALUE	0	1	1

CONFIG 옵션	AL1/4.14/x86_64	AL2023/6.1/AArch64	AL2023/6.1/x86_64
CONFIG_PPP	m	n	n
CONFIG_SLIP	m	n	n
CONFIG_XEN_PV	y	N/A	n

Amazon Linux 1(AL1) 과 Amazon Linux 2023 AMI에 설치된 패키지 비교

AL1 및 AL2023 표준 AMI에 있는 RPM을 비교한 것입니다.

패키지	AL1 AMI	2023년 AMI
acl	2.2.49	2.3.1
acpid	2.0.19	2.0.32
alsa-lib	1.0.22	
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.4.1
amazon-linux-repo-s3		2023.4.20240513
amazon-linux-sb-keys		2023.1
amazon-rpm-config		228
amazon-ssm-agent	3.2.2222.0	3.3.380.0
at	3.1.10	3.1.23
attr	2.4.46	2.5.1

패키지	AL1 AMI	2023년 AMI
audit	2.6.5	3.0.6
audit-libs	2.6.5	3.0.6
authconfig	6.2.8	
aws-amitools-ec2	1.5.13	
aws-cfn-bootstrap	1.4	2.0
aws-cli	1.18.107	
awscli-2		2.15.30
basesystem	10.0	11
bash	4.2.46	5.2.15
bash-completion		2.11
bc	1.06.95	1.07.1
bind-libs	9.8.2	9.16.48
bind-license		9.16.48
bind-utils	9.8.2	9.16.48
binutils	2.27	2.39
boost-filesystem		1.75.0
boost-system		1.75.0
boost-thread		1.75.0
bzip2	1.0.6	1.0.8
bzip2-libs	1.0.6	1.0.8

패키지	AL1 AMI	2023년 AMI
ca-certificates	2023.2.62	2023.2.64
c-ares		1.19.0
checkpolicy	2.1.10	3.4
chkconfig	1.3.49.3	1.15
chrony		4.3
cloud-disk-utils	0.27	
cloud-init	0.7.6	22.2.2
cloud-init-cfg-ec2		22.2.2
cloud-utils-growpart		0.31
copy-jdk-configs	3.3	
coreutils	8.22	8.32
coreutils-common		8.32
cpio	2.10	2.13
cracklib	2.8.16	2.9.6
cracklib-dicts	2.8.16	2.9.6
cronie	1.4.4	
cronie-anacron	1.4.4	
crontabs	1.10	1.11
crypto-policies		20220428
crypto-policies-scripts		20220428

패키지	AL1 AMI	2023년 AMI
cryptsetup	1.6.7	2.6.1
cryptsetup-libs	1.6.7	2.6.1
curl	7.61.1	
curl-minimal		8.5.0
cyrus-sasl	2.1.23	
cyrus-sasl-lib	2.1.23	2.1.27
cyrus-sasl-plain	2.1.23	2.1.27
dash	0.5.5.1	
db4	4.7.25	
db4-utils	4.7.25	
dbus	1.6.12	1.12.28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.6.12	1.12.28
dejavu-fonts-common	2.33	
dejavu-sans-fonts	2.33	
dejavu-serif-fonts	2.33	
device-mapper	1.02.135	1.02.185
device-mapper-event	1.02.135	
device-mapper-event-libs	1.02.135	

패키지	AL1 AMI	2023년 AMI
device-mapper-libs	1.02.135	1.02.185
device-mapper-persistent-data	0.6.3	
dhclient	4.1.1	
dhcp-common	4.1.1	
diffutils	3.3	3.8
dmraid	1.0.0.rc16	
dmraid-events	1.0.0.rc16	
dnf		4.14.0
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0
dnf-plugin-support-info		1.2
dnf-utils		4.3.0
dosfstools		4.2
dracut	004	055
dracut-config-ec2		3.0
dracut-config-generic		055

패키지	AL1 AMI	2023년 AMI
dracut-modules-gro wroot	0.20	
dump	0.4	
dwz		0.14
dyninst		10.2.1
e2fsprogs	1.43.5	1.46.5
e2fsprogs-libs	1.43.5	1.46.5
ec2-hibinit-agent	1.0.0	1.0.8
ec2-instance-connect		1.1
ec2-instance-conne ct-selinux		1.1
ec2-net-utils	0.7	
ec2-utils	0.7	2.2.0
ed	1.1	1.14.2
efi-filesystem		5
efi-srpm-macros		5
efivar		38
efivar-libs		38
elfutils-debuginfod- client		0.188
elfutils-default-y ama-scope		0.188

패키지	AL1 AMI	2023년 AMI
elfutils-libelf	0.168	0.188
elfutils-libs		0.188
epel-release	6	
ethtool	3.15	5.15
expat	2.1.0	2.5.0
file	5.37	5.39
file-libs	5.37	5.39
filesystem	2.4.30	3.14
findutils	4.4.2	4.8.0
fipscheck	1.3.1	
fipscheck-lib	1.3.1	
fontconfig	2.8.0	
fontpackages-files system	1.41	
fonts-srpm-macros		2.0.5
freetype	2.3.11	
fstrm		0.6.1
fuse-libs	2.9.4	2.9.9
gawk	3.1.7	5.1.0
gdbm	1.8.0	
gdbm-libs		1.19

패키지	AL1 AMI	2023년 AMI
gdisk	0.8.10	1.0.8
generic-logos	17.0.0	
get_reference_source	1.2	
gettext		0.21
gettext-libs		0.21
ghc-srpm-macros		1.5.0
giflib	4.1.6	
glib2	2.36.3	2.74.7
glibc	2.17	2.34
glibc-all-langpacks		2.34
glibc-common	2.17	2.34
glibc-gconv-extra		2.34
glibc-locale-source		2.34
gmp	6.0.0	6.2.1
gnupg2	2.0.28	
gnupg2-minimal		2.3.7
gnutls		3.8.0
go-srpm-macros		3.2.0
gpgme	1.4.3	1.15.1
gpm-libs	1.20.6	1.20.7

패키지	AL1 AMI	2023년 AMI
grep	2.20	3.8
groff	1.22.2	
groff-base	1.22.2	1.22.4
grub	0.97	
grub2-common		2.06
grub2-efi-x64-ec2		2.06
grub2-pc-modules		2.06
grub2-tools		2.06
grub2-tools-minimal		2.06
grubby	7.0.15	8.40
gssproxy		0.8.4
gzip	1.5	1.12
hesiod	3.1.0	
hibagent	1.0.0	
hmacalc	0.9.12	
hostname		3.23
hunspell		1.7.0
hunspell-en		0.20140811.1
hunspell-en-GB		0.20140811.1
hunspell-en-US		0.20140811.1

패키지	AL1 AMI	2023년 AMI
hunspell-filesystem		1.7.0
hwdata	0.233	0.353
info	5.1	6.7
inih		49
initscripts	9.03.58	10.09
iproute	4.4.0	5.10.0
iptables	1.4.21	
iputils	20121221	20210202
irqbalance	1.5.0	1.9.0
jansson		2.14
java-1.7.0-openjdk	1.7.0.321	
javapackages-tools	0.9.1	
jitterentropy		3.4.1
jpackage-utils	1.7.5	
jq		1.7.1
json-c		0.14
kbd	1.15	2.4.0
kbd-misc	1.15	2.4.0
kernel	4.14.336	6.1.90
kernel-livepatch-r epo-s3		2023.4.20240513

패키지	AL1 AMI	2023년 AMI
kernel-srpm-macros		1.0
kernel-tools	4.14.336	6.1.90
keyutils	1.5.8	1.6.3
keyutils-libs	1.5.8	1.6.3
kmod	14	29
kmod-libs	14	29
kpartx	0.4.9	
kpatch-runtime		0.9.7
krb5-libs	1.15.1	1.21
lcms2	2.6	
less	436	608
libacl	2.2.49	2.3.1
libaio	0.3.109	0.3.111
libarchive		3.5.3
libargon2		20171227
libassuan	2.0.3	2.5.5
libattr	2.4.46	2.5.1
libbasicobjects		0.1.1
libblkid	2.23.2	2.37.4
libcap	2.16	2.48

패키지	AL1 AMI	2023년 AMI
libcap54	2.54	
libcap-ng	0.7.5	0.8.2
libcbor		0.7.0
libcgroup	0.40.rc1	
libcollection		0.7.0
libcom_err	1.43.5	1.46.5
libcomps		0.1.20
libconfig		1.7.2
libcurl	7.61.1	
libcurl-minimal		8.5.0
libdb		5.3.28
libdhash		0.5.0
libdnf		0.69.0
libeconf		0.4.0
libedit	2.11	3.1
libev		4.33
libevent	2.0.21	2.1.12
libfdisk		2.37.4
libffi	3.0.13	3.4.4
libfido2		1.10.0

패키지	AL1 AMI	2023년 AMI
libfontenc	1.0.5	
libgcc		11.4.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.11	1.42
libgssglue	0.1	
libibverbs		48.0
libICE	1.0.6	
libicu	50.2	
libidn	1.18	
libidn2	2.3.0	2.3.2
libini_config		1.3.1
libjpeg-turbo	1.2.90	
libkcapi		1.4.0
libkcapi-hmaccalc		1.4.0
libldb		2.6.2
libmaxminddb		1.5.2
libmetalink		0.1.3
libmnl	1.0.3	1.0.4

패키지	AL1 AMI	2023년 AMI
libmodulemd		2.13.0
libmount	2.23.2	2.37.4
libnetfilter_contrack	1.0.4	
libnfnetworking	1.0.1	
libnfsidmap	0.25	2.5.4
libnghttp2	1.33.0	1.59.0
libnih	1.0.1	
libnl	1.1.4	
libnl3		3.5.0
libpath_utils		0.2.1
libpcap		1.10.1
libpipeline	1.2.3	1.5.3
libpkgconf		1.8.0
libpng	1.2.49	
libpsl	0.6.2	0.21.1
libpwquality	1.2.3	1.4.4
libref_array		0.1.5
librepo		1.14.5
libreport-filesystem		2.15.2
libseccomp		2.5.3

패키지	AL1 AMI	2023년 AMI
libselinux	2.1.10	3.4
libselinux-utils	2.1.10	3.4
libsemanage	2.1.6	3.4
libsepol	2.1.7	3.4
libsigsegv		2.13
libSM	1.2.1	
libsmartcols	2.23.2	2.37.4
libsolv		0.7.22
libss	1.43.5	1.46.5
libssh2	1.4.2	
libsss_certmap		2.9.4
libsss_idmap		2.9.4
libsss_nss_idmap		2.9.4
libsss_sudo		2.9.4
libstdc++		11.4.1
libstdc++72	7.2.1	
libstoragemgmt		1.9.4
libsysfs	2.1.0	
libtalloc		2.3.4
libtasn1	2.3	4.19.0

패키지	AL1 AMI	2023년 AMI
libtdb		1.4.7
libtevent		0.13.0
libtextstyle		0.21
libtirpc	0.2.4	1.3.3
libudev	173	
libunistring	0.9.3	0.9.10
libuser	0.60	0.63
libutempter	1.1.5	1.2.1
libuuid	2.23.2	2.37.4
libuv		1.47.0
libverto	0.2.5	0.3.2
libverto-libev		0.3.2
libX11	1.6.0	
libX11-common	1.6.0	
libXau	1.0.6	
libxcb	1.11	
libXcomposite	0.4.3	
libxcrypt		4.4.33
libXext	1.3.2	
libXfont	1.4.5	

패키지	AL1 AMI	2023년 AMI
libXi	1.7.2	
libxml2	2.9.1	2.10.4
libxml2-python27	2.9.1	
libXrender	0.9.8	
libxslt	1.1.28	
libXtst	1.2.2	
libyaml	0.1.6	0.2.5
libzstd		1.5.5
lm_sensors-libs		3.6.0
lmdb-libs		0.9.29
log4j-cve-2021-44228-hotpatch	1.3	
logrotate	3.7.8	3.20.1
lsof	4.82	4.94.0
lua	5.1.4	
lua-libs		5.4.4
lua-srpm-macros		1
lvm2	2.02.166	
lvm2-libs	2.02.166	
lz4-libs		1.9.4
mailcap	2.1.31	

패키지	AL1 AMI	2023년 AMI
make	3.82	
man-db	2.6.3	2.9.3
man-pages	4.10	5.10
mdadm	3.2.6	
microcode_ctl	2.1	2.1
mingetty	1.08	
mpfr		4.1.0
nano	2.5.3	5.8
nc	1.84	
ncurses	5.7	6.2
ncurses-base	5.7	6.2
ncurses-libs	5.7	6.2
nettle		3.8
net-tools	1.60	2.0
newt	0.52.11	0.52.21
newt-python27	0.52.11	
nfs-utils	1.3.0	2.5.4
npth		1.6
nspr	4.25.0	4.35.0
nss	3.53.1	3.90.0

패키지	AL1 AMI	2023년 AMI
nss-pem	1.0.3	
nss-softokn	3.53.1	3.90.0
nss-softokn-freebl	3.53.1	3.90.0
nss-sysinit	3.53.1	3.90.0
nss-tools	3.53.1	
nss-util	3.53.1	3.90.0
ntp	4.2.8p15	
ntpdate	4.2.8p15	
ntsysv	1.3.49.3	1.15
numactl	2.0.7	
numactl-libs		2.0.14
ocaml-srpm-macros		6
oniguruma		6.9.7.1
openblas-srpm-macros		2
openldap	2.4.40	2.4.57
openssh	7.4p1	8.7p1
openssh-clients	7.4p1	8.7p1
openssh-server	7.4p1	8.7p1
openssl	1.0.2k	3.0.8
openssl-libs		3.0.8

패키지	AL1 AMI	2023년 AMI
openssl-pkcs11		0.4.12
os-prober		1.77
p11-kit	0.18.5	0.24.1
p11-kit-trust	0.18.5	0.24.1
package-notes-srpm-macros		0.4
pam	1.1.8	1.5.1
pam_ccreds	10	
pam_krb5	2.3.11	
pam_passwdqc	1.0.5	
parted	2.1	3.4
passwd	0.79	0.80
pciutils	3.1.10	3.7.0
pciutils-libs	3.1.10	3.7.0
pcre	8.21	
pcre2		10.40
pcre2-syntax		10.40
perl	5.16.3	
perl-Carp	1.26	1.50
perl-Class-Struct		0.66
perl-constant	1.27	1.33

패키지	AL1 AMI	2023년 AMI
perl-Digest	1.17	
perl-Digest-HMAC	1.03	
perl-Digest-MD5	2.52	
perl-Digest-SHA	5.85	
perl-DynaLoader		1.47
perl-Encode	2.51	3.15
perl-Errno		1.30
perl-Exporter	5.68	5.74
perl-Fcntl		1.13
perl-File-Basename		2.85
perl-File-Path	2.09	2.18
perl-File-stat		1.09
perl-File-Temp	0.23.01	0.231.100
perl-Filter	1.49	
perl-Getopt-Long	2.40	2.52
perl-Getopt-Std		1.12
perl-HTTP-Tiny	0.033	0.078
perl-if		0.60.800
perl-interpreter		5.32.1
perl-IO		1.43

패키지	AL1 AMI	2023년 AMI
perl-IPC-Open3		1.21
perl-libs	5.16.3	5.32.1
perl-macros	5.16.3	
perl-MIME-Base64		3.16
perl-mro		1.23
perl-overload		1.31
perl-overloading		0.02
perl-parent	0.225	0.238
perl-PathTools	3.40	3.78
perl-Pod-Escapes	1.04	1.07
perl-podlators	2.5.1	4.14
perl-Pod-Perldoc	3.20	3.28.01
perl-Pod-Simple	3.28	3.42
perl-Pod-Usage	1.63	2.01
perl-POSIX		1.94
perl-Scalar-List-Utils	1.27	1.56
perl-SelectSaver		1.02
perl-Socket	2.010	2.032
perl-srpm-macros		1
perl-Storable	2.45	3.21

패키지	AL1 AMI	2023년 AMI
perl-subst		1.03
perl-Symbol		1.08
perl-Term-ANSIColor		5.01
perl-Term-Cap		1.17
perl-Text-ParseWords	3.29	3.30
perl-Text-Tabs+Wrap		2021.0726
perl-threads	1.87	
perl-threads-shared	1.43	
perl-Time-HiRes	1.9725	
perl-Time-Local	1.2300	1.300
perl-vars		1.05
pinentry	0.7.6	
pkgconf		1.8.0
pkgconfig	0.27.1	
pkgconf-m4		1.8.0
pkgconf-pkg-config		1.8.0
pm-utils	1.4.1	
policycoreutils	2.1.12	3.4
policycoreutils-python-utils		3.4
popt	1.13	1.18

패키지	AL1 AMI	2023년 AMI
procmail	3.22	
procps	3.2.8	
procps-ng		3.3.17
protobuf-c		1.4.1
psacct	6.3.2	6.6.4
psmisc	22.20	23.4
pth	2.0.7	
publicsuffix-list-dafsa		20240212
python27	2.7.18	
python27-babel	0.9.4	
python27-backports	1.0	
python27-backports-ssl_match_hostname	3.4.0.2	
python27-boto	2.48.0	
python27-botocore	1.17.31	
python27-chardet	2.0.1	
python27-colorama	0.4.1	
python27-configobj	4.7.2	
python27-crypto	2.6.1	
python27-daemon	1.5.2	

패키지	AL1 AMI	2023년 AMI
python27-dateutil	2.1	
python27-devel	2.7.18	
python27-docutils	0.11	
python27-ecdsa	0.11	
python27-futures	3.0.3	
python27-imaging	1.1.6	
python27-iniparse	0.3.1	
python27-jinja2	2.7.2	
python27-jmespath	0.9.2	
python27-jsonpatch	1.2	
python27-jsonpointer	1.0	
python27-kitchen	1.1.1	
python27-libs	2.7.18	
python27-lockfile	0.8	
python27-markupsafe	0.11	
python27-paramiko	1.15.1	
python27-pip	9.0.3	
python27-ply	3.4	
python27-pyasn1	0.1.7	
python27-pycurl	7.19.0	

패키지	AL1 AMI	2023년 AMI
python27-pygpme	0.3	
python27-pyliblzma	0.5.3	
python27-pystache	0.5.3	
python27-pyxattr	0.5.0	
python27-PyYAML	3.10	
python27-requests	1.2.3	
python27-rsa	3.4.1	
python27-setuptools	36.2.7	
python27-simplejson	3.6.5	
python27-six	1.8.0	
python27-urlgrabber	3.10	
python27-urllib3	1.24.3	
python27-virtualenv	15.1.0	
python3		3.9.16
python3-attrs		20.3.0
python3-audit		3.0.6
python3-awscli		0.19.19
python3-babel		2.9.1
python3-cffi		1.14.5
python3-chardet		4.0.0

패키지	AL1 AMI	2023년 AMI
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36.0.1
python3-daemon		2.3.0
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils		0.16
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-idna		2.10
python3-jinja2		2.11.3
python3-jmespath		0.10.0
python3-jsonpatch		1.21
python3-jsonpointer		2.0
python3-jsonschemata		3.2.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0

패키지	AL1 AMI	2023년 AMI
python3-libs		3.9.16
python3-libselinux		3.4
python3-libsemanage		3.4
python3-libstorage mgmt		1.9.4
python3-lockfile		0.12.2
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip-wheel		21.3.1
python3-ply		3.11
python3-policycore utils		3.4
python3-prettytable		0.7.2
python3-prompt-too lkit		3.0.24
python3-pycparser		2.20
python3-pyrsistent		0.17.3
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pytz		2022.7.1

패키지	AL1 AMI	2023년 AMI
python3-pyyaml		5.4.1
python3-requests		2.25.1
python3-rpm		4.16.1.3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools		59.6.0
python3-setuptools-wheel		59.6.0
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1.25.10
python3-wcwidth		0.2.5
python-chevront		0.13.1
python-srpm-macros		3.9
quota	4.00	4.06
quota-nls	4.00	4.06
readline	6.2	8.1
rmt	0.4	
rng-tools	5	6.14

패키지	AL1 AMI	2023년 AMI
rootfiles	8.1	8.1
rpcbind	0.2.0	1.2.6
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit		4.16.1.3
rpm-python27	4.11.3	
rpm-sign-libs		4.16.1.3
rsync	3.0.6	3.2.6
rsyslog	5.8.10	
ruby	2.0	
ruby20	2.0.0.648	
ruby20-irb	2.0.0.648	
ruby20-libs	2.0.0.648	
rubygem20-bigdecimal	1.2.0	
rubygem20-json	1.8.3	
rubygem20-psych	2.0.0	
rubygem20-rdoc	4.2.2	
rubygems20	2.0.14.1	

패키지	AL1 AMI	2023년 AMI
rust-srpm-macros		21
sbsigntools		0.9.4
screen	4.0.3	4.8.0
sed	4.2.1	4.8
selinux-policy		37.22
selinux-policy-targeted		37.22
sendmail	8.14.4	
setserial	2.17	
setup	2.8.14	2.13.7
sgpio	1.2.0.10	
shadow-utils	4.1.4.2	4.9
shared-mime-info	1.1	
slang	2.2.1	2.3.2
sqlite	3.7.17	
sqlite-libs		3.40.0
sssd-client		2.9.4
sssd-common		2.9.4
sssd-kcm		2.9.4
sssd-nfs-idmap		2.9.4
strace		6.8

패키지	AL1 AMI	2023년 AMI
sudo	1.8.23	1.9.15
sysctl-defaults	1.0	1.0
sysfsutils	2.1.0	
sysstat		12.5.6
systemd		252.16
systemd-libs		252.16
systemd-networkd		252.16
systemd-pam		252.16
systemd-resolved		252.16
systemd-udev		252.16
system-release	2018.03	2023.4.20240513
systemtap-runtime		4.8
sysvinit	2.87	
tar	1.26	1.34
tbb		2020.3
tcp_wrappers	7.6	
tcp_wrappers-libs	7.6	
tcpdump		4.99.1
tcsh		6.24.07
time	1.7	1.9

패키지	AL1 AMI	2023년 AMI
tmpwatch	2.9.16	
traceroute	2.0.14	2.1.3
ttmkfdir	3.0.9	
tzdata	2023c	2024a
tzdata-java	2023c	
udev	173	
unzip	6.0	6.0
update-motd	1.0.1	2.2
upstart	0.6.5	
userspace-rcu		0.12.1
ustr	1.0.4	
util-linux	2.23.2	2.37.4
util-linux-core		2.37.4
vim-common	9.0.2120	9.0.2153
vim-data	9.0.2120	9.0.2153
vim-enhanced	9.0.2120	9.0.2153
vim-filesystem	9.0.2120	9.0.2153
vim-minimal	9.0.2120	9.0.2153
wget	1.18	1.21.3
which	2.19	2.21

패키지	AL1 AMI	2023년 AMI
words	3.0	3.0
xfsdump		3.1.11
xfspgrog		5.18.0
xorg-x11-fonts-Type1	7.2	
xorg-x11-font-utis	7.2	
xxd	9.0.2120	9.0.2153
xxhash-libs		0.8.0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	1.1.4	
yum-plugin-priorities	1.1.31	
yum-plugin-upgrade-helper	1.1.31	
yum-utils	1.1.31	
zip	3.0	3.0
zlib	1.2.8	1.2.11
zram-generator		1.1.2
zram-generator-defaults		1.1.2

패키지	AL1 AMI	2023년 AMI
zstd		1.5.5

Amazon Linux 1(AL1)과 Amazon Linux 2023 미니멀 AMI에 설치된 패키지 비교

AL1 및 AL2023 최소 AMI에 있는 RPM을 비교한 것입니다.

패키지	AL1 미니멀	AL2023 미니멀
acpid	2.0.19	
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.4.1
amazon-linux-repo-s3		2023.4.20240513
amazon-linux-sb-keys		2023.1
audit	2.6.5	3.0.6
audit-libs	2.6.5	3.0.6
authconfig	6.2.8	
awscli-2		2.15.30
basesystem	10.0	11
bash	4.2.46	5.2.15
binutils	2.27	
bzip2	1.0.6	

패키지	AL1 미니멀	AL2023 미니멀
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023.2.62	2023.2.64
checkpolicy	2.1.10	3.4
chkconfig	1.3.49.3	
chrony		4.3
cloud-disk-utils	0.27	
cloud-init	0.7.6	22.2.2
cloud-init-cfg-ec2		22.2.2
cloud-utils-growpart		0.31
coreutils	8.22	8.32
coreutils-common		8.32
cpio	2.10	2.13
cracklib	2.8.16	2.9.6
cracklib-dicts	2.8.16	2.9.6
cronie	1.4.4	
cronie-anacron	1.4.4	
crontabs	1.10	
crypto-policies		20220428
cryptsetup-libs		2.6.1
curl	7.61.1	

패키지	AL1 미니멀	AL2023 미니멀
<u>curl-minimal</u>		8.5.0
cyrus-sasl	2.1.23	
cyrus-sasl-lib	2.1.23	2.1.27
dash	0.5.5.1	
db4	4.7.25	
db4-utils	4.7.25	
dbus		1.12.28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.6.12	1.12.28
device-mapper		1.02.185
device-mapper-libs		1.02.185
dhclient	4.1.1	
dhcp-common	4.1.1	
diffutils	3.3	3.8
dnf		4.14.0
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0

패키지	AL1 미니멀	AL2023 미니멀
dnf-plugin-support-info		1.2
dracut	004	055
dracut-config-ec2		3.0
dracut-config-generic		055
dracut-modules-growroot	0.20	
e2fsprogs	1.43.5	1.46.5
e2fsprogs-libs	1.43.5	1.46.5
ec2-utils	0.7	2.2.0
ed	1.1	
efi-filesystem		5
efivar		38
efivar-libs		38
elfutils-default-yama-scope		0.188
elfutils-libelf	0.168	0.188
elfutils-libs		0.188
ethtool	3.15	
expat	2.1.0	2.5.0
file	5.37	5.39

패키지	AL1 미니멀	AL2023 미니멀
file-libs	5.37	5.39
filesystem	2.4.30	3.14
findutils	4.4.2	4.8.0
fipscheck	1.3.1	
fipscheck-lib	1.3.1	
fuse-libs	2.9.4	2.9.9
gawk	3.1.7	5.1.0
gdbm	1.8.0	
gdbm-libs		1.19
gdisk	0.8.10	1.0.8
generic-logos	17.0.0	
get_reference_source	1.2	
gettext		0.21
gettext-libs		0.21
glib2	2.36.3	2.74.7
glibc	2.17	2.34
glibc-all-langpacks		2.34
glibc-common	2.17	2.34
glibc-locale-source		2.34
gmp	6.0.0	6.2.1

패키지	AL1 미니멀	AL2023 미니멀
gnupg2	2.0.28	
gnupg2-minimal		2.3.7
gnutls		3.8.0
gpgme	1.4.3	1.15.1
grep	2.20	3.8
groff	1.22.2	
groff-base	1.22.2	1.22.4
grub	0.97	
grub2-common		2.06
grub2-efi-x64-ec2		2.06
grub2-pc-modules		2.06
grub2-tools		2.06
grub2-tools-minimal		2.06
grubby	7.0.15	8.40
gzip	1.5	1.12
hesiod	3.1.0	
hmacalc	0.9.12	
hostname		3.23
hwdata	0.233	0.353
info	5.1	

패키지	AL1 미니멀	AL2023 미니멀
inih		49
initscripts	9.03.58	10.09
iproute	4.4.0	5.10.0
iptables	1.4.21	
iputils	20121221	20210202
irqbalance		1.9.0
jansson		2.14
jitterentropy		3.4.1
jq		1.7.1
json-c		0.14
kbd	1.15	2.4.0
kbd-misc	1.15	2.4.0
kernel	4.14.336	6.1.90
kernel-livepatch-r epo-s3		2023.4.20240513
keyutils-libs	1.5.8	1.6.3
kmod	14	29
kmod-libs	14	29
krb5-libs	1.15.1	1.21
less	436	608
libacl	2.2.49	2.3.1

패키지	AL1 미니멀	AL2023 미니멀
libarchive		3.5.3
libargon2		20171227
libassuan	2.0.3	2.5.5
libattr	2.4.46	2.5.1
libblkid	2.23.2	2.37.4
libcap	2.16	2.48
libcap54	2.54	
libcap-ng	0.7.5	0.8.2
libcbor		0.7.0
libcgroup	0.40.rc1	
libcom_err	1.43.5	1.46.5
libcomps		0.1.20
libcurl	7.61.1	
libcurl-minimal		8.5.0
libdb		5.3.28
libdnf		0.69.0
libeconf		0.4.0
libedit	2.11	3.1
libfdisk		2.37.4
libffi	3.0.13	3.4.4

패키지	AL1 미니멀	AL2023 미니멀
libfido2		1.10.0
libgcc		11.4.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.11	1.42
libicu	50.2	
libidn	1.18	
libidn2	2.3.0	2.3.2
libkcapi		1.4.0
libkcapi-hmaccalc		1.4.0
libmnl	1.0.3	1.0.4
libmodulemd		2.13.0
libmount	2.23.2	2.37.4
libnetfilter_contrack	1.0.4	
libnfnetlink	1.0.1	
libnghttp2	1.33.0	1.59.0
libnih	1.0.1	
libpipeline		1.5.3
libpsl	0.6.2	0.21.1

패키지	AL1 미니멀	AL2023 미니멀
libpwquality	1.2.3	1.4.4
librepo		1.14.5
libreport-filessystem		2.15.2
libseccomp		2.5.3
libselinux	2.1.10	3.4
libselinux-utils	2.1.10	3.4
libsemanage	2.1.6	3.4
libsepol	2.1.7	3.4
libsigsegv		2.13
libsmartcols	2.23.2	2.37.4
libsolv		0.7.22
libss	1.43.5	1.46.5
libssh2	1.4.2	
libstdc++		11.4.1
libstdc++72	7.2.1	
libsysfs	2.1.0	
libtasn1	2.3	4.19.0
libtextstyle		0.21
libudev	173	
libunistring	0.9.3	0.9.10

패키지	AL1 미니멀	AL2023 미니멀
libuser	0.60	0.63
libutempter	1.1.5	1.2.1
libuuid	2.23.2	2.37.4
libverto	0.2.5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libyaml	0.1.6	0.2.5
libzstd		1.5.5
logrotate	3.7.8	3.20.1
lua	5.1.4	
lua-libs		5.4.4
lz4-libs		1.9.4
make	3.82	
man-db		2.9.3
microcode_ctl	2.1	2.1
mingetty	1.08	
mpfr		4.1.0
ncurses	5.7	6.2
ncurses-base	5.7	6.2
ncurses-libs	5.7	6.2

패키지	AL1 미니멀	AL2023 미니멀
nettle		3.8
net-tools	1.60	2.0
newt	0.52.11	
newt-python27	0.52.11	
npth		1.6
nspr	4.25.0	
nss	3.53.1	
nss-pem	1.0.3	
nss-softokn	3.53.1	
nss-softokn-freebl	3.53.1	
nss-sysinit	3.53.1	
nss-tools	3.53.1	
nss-util	3.53.1	
ntp	4.2.8p15	
ntpdate	4.2.8p15	
numactl-libs		2.0.14
oniguruma		6.9.7.1
openldap	2.4.40	2.4.57
openssh	7.4p1	8.7p1
openssh-clients		8.7p1

패키지	AL1 미니멀	AL2023 미니멀
openssh-server	7.4p1	8.7p1
openssl	1.0.2k	3.0.8
openssl-lib		3.0.8
openssl-pkcs11		0.4.12
os-prober		1.77
p11-kit	0.18.5	0.24.1
p11-kit-trust	0.18.5	0.24.1
pam	1.1.8	1.5.1
passwd	0.79	0.80
pciutils	3.1.10	3.7.0
pciutils-lib	3.1.10	3.7.0
pcre	8.21	
pcre2		10.40
pcre2-syntax		10.40
pinentry	0.7.6	
pkgconfig	0.27.1	
policycoreutils	2.1.12	3.4
popt	1.13	1.18
procmail	3.22	
procps	3.2.8	

패키지	AL1 미니멀	AL2023 미니멀
procps-ng		3.3.17
psmisc	22.20	23.4
pth	2.0.7	
publicsuffix-list-dafsa		20240212
python27	2.7.18	
python27-babel	0.9.4	
python27-backports	1.0	
python27-backports-ssl_match_hostname	3.4.0.2	
python27-chardet	2.0.1	
python27-configobj	4.7.2	
python27-iniparse	0.3.1	
python27-jinja2	2.7.2	
python27-jsonpatch	1.2	
python27-jsonpointer	1.0	
python27-libs	2.7.18	
python27-markupsafe	0.11	
python27-pycurl	7.19.0	
python27-pygpme	0.3	
python27-pyliblzma	0.5.3	

패키지	AL1 미니멀	AL2023 미니멀
python27-pyxattr	0.5.0	
python27-PyYAML	3.10	
python27-requests	1.2.3	
python27-setuptools	36.2.7	
python27-six	1.8.0	
python27-urlgrabber	3.10	
python27-urllib3	1.24.3	
python3		3.9.16
python3-attrs		20.3.0
python3-audit		3.0.6
python3-awscli		0.19.19
python3-babel		2.9.1
python3-cffi		1.14.5
python3-chardet		4.0.0
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36.0.1
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0

패키지	AL1 미니멀	AL2023 미니멀
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils		0.16
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-idna		2.10
python3-jinja2		2.11.3
python3-jmespath		0.10.0
python3-jsonpatch		1.21
python3-jsonpointer		2.0
python3-jjsonschema		3.2.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0
python3-libs		3.9.16
python3-libselenium		3.4
python3-libsemanage		3.4
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip-wheel		21.3.1

패키지	AL1 미니멀	AL2023 미니멀
python3-ply		3.11
python3-policycore utils		3.4
python3-prettytable		0.7.2
python3-prompt-too lkit		3.0.24
python3-pycparser		2.20
python3-pyrsistent		0.17.3
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pytz		2022.7.1
python3-pyyaml		5.4.1
python3-requests		2.25.1
python3-rpm		4.16.1.3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml- clib		0.1.2
python3-setools		4.4.1
python3-setuptools		59.6.0
python3-setuptools- wheel		59.6.0
python3-six		1.15.0

패키지	AL1 미니멀	AL2023 미니멀
python3-systemd		235
python3-urllib3		1.25.10
python3-wcwidth		0.2.5
readline	6.2	8.1
rng-tools		6.14
rootfiles	8.1	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit		4.16.1.3
rpm-python27	4.11.3	
rpm-sign-libs		4.16.1.3
rsyslog	5.8.10	
sbsigntools		0.9.4
sed	4.2.1	4.8
selinux-policy		37.22
selinux-policy-targeted		37.22
sendmail	8.14.4	

패키지	AL1 미니멀	AL2023 미니멀
setserial	2.17	
setup	2.8.14	2.13.7
shadow-utils	4.1.4.2	4.9
shared-mime-info	1.1	
slang	2.2.1	
sqlite	3.7.17	
sqlite-libs		3.40.0
sudo	1.8.23	1.9.15
sysctl-defaults	1.0	1.0
sysfsutils	2.1.0	
systemd		252.16
systemd-libs		252.16
systemd-networkd		252.16
systemd-pam		252.16
systemd-resolved		252.16
systemd-udev		252.16
system-release	2018.03	2023.4.20240513
sysvinit	2.87	
tar	1.26	1.34
tcp_wrappers-libs	7.6	

패키지	AL1 미니멀	AL2023 미니멀
tzdata	2023c	2024a
udev	173	
update-motd	1.0.1	2.2
upstart	0.6.5	
userspace-rcu		0.12.1
ustr	1.0.4	
util-linux	2.23.2	2.37.4
util-linux-core		2.37.4
vim-data	9.0.2120	9.0.2153
vim-minimal	9.0.2120	9.0.2153
which	2.19	2.21
xfspgrog		5.18.0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	1.1.4	
yum-plugin-priorities	1.1.31	
yum-plugin-upgrade-helper	1.1.31	
zlib	1.2.8	1.2.11

패키지	AL1 미니멀	AL2023 미니멀
zram-generator		1.1.2
zram-generator-def aults		1.1.2
zstd		1.5.5

Amazon Linux 1(AL1) 과 Amazon Linux 2023의 기본 컨테이너 이미지에 설치된 패키지 비교

AL1 및 AL2023 기본 컨테이너 이미지에 있는 RPM을 비교한 것입니다.

패키지	AL1 컨테이너	AL2023 컨테이너
alternatives		1.15
amazon-linux-repo- cdn		2023.4.20240513
audit-libs		3.0.6
basesystem	10.0	11
bash	4.2.46	5.2.15
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023.2.62	2023.2.64
chkconfig	1.3.49.3	
coreutils	8.22	
coreutils-single		8.32
crypto-policies		20220428

패키지	AL1 컨테이너	AL2023 컨테이너
curl	7.61.1	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.23	
db4	4.7.25	
db4-utils	4.7.25	
dnf		4.14.0
dnf-data		4.14.0
elfutils-default-yama-scope		0.188
elfutils-libelf	0.168	0.188
elfutils-libs		0.188
expat	2.1.0	2.5.0
file-libs	5.37	5.39
filesystem	2.4.30	3.14
gawk	3.1.7	5.1.0
gdbm	1.8.0	
gdbm-libs		1.19
glib2	2.36.3	2.74.7
glibc	2.17	2.34
glibc-common	2.17	2.34

패키지	AL1 컨테이너	AL2023 컨테이너
<code>glibc-minimal-langpack</code>		2.34
<code>gmp</code>	6.0.0	6.2.1
<u>gnupg2</u>	2.0.28	
<u>gnupg2-minimal</u>		2.3.7
<code>gpgme</code>	1.4.3	1.15.1
<code>grep</code>	2.20	3.8
<code>gzip</code>	1.5	
<code>info</code>	5.1	
<code>json-c</code>		0.14
<code>keyutils-libs</code>	1.5.8	1.6.3
<code>krb5-libs</code>	1.15.1	1.21
<code>libacl</code>	2.2.49	2.3.1
<code>libarchive</code>		3.5.3
<code>libassuan</code>	2.0.3	2.5.5
<code>libattr</code>	2.4.46	2.5.1
<code>libblkid</code>		2.37.4
<code>libcap</code>	2.16	2.48
<code>libcap-ng</code>		0.8.2
<code>libcom_err</code>	1.43.5	1.46.5
<code>libcomps</code>		0.1.20

패키지	AL1 컨테이너	AL2023 컨테이너
libcurl	7.61.1	
libcurl-minimal		8.5.0
libdnf		0.69.0
libffi	3.0.13	3.4.4
libgcc		11.4.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.11	1.42
libicu	50.2	
libidn2	2.3.0	2.3.2
libmodulemd		2.13.0
libmount		2.37.4
libnghttp2	1.33.0	1.59.0
libpsl	0.6.2	0.21.1
librepo		1.14.5
libreport-filessystem		2.15.2
libselinux	2.1.10	3.4
libsepol	2.1.7	3.4
libsigsegv		2.13

패키지	AL1 컨테이너	AL2023 컨테이너
libsmartcols		2.37.4
libsolv		0.7.22
libssh2	1.4.2	
libstdc++		11.4.1
libstdc++72	7.2.1	
libtasn1	2.3	4.19.0
libunistring	0.9.3	0.9.10
libuuid		2.37.4
libverto	0.2.5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libxml2-python27	2.9.1	
libyaml		0.2.5
libzstd		1.5.5
lua	5.1.4	
lua-libs		5.4.4
lz4-libs		1.9.4
make	3.82	
mpfr		4.1.0
ncurses	5.7	

패키지	AL1 컨테이너	AL2023 컨테이너
ncurses-base	5.7	6.2
ncurses-libs	5.7	6.2
npth		1.6
nspr	4.25.0	
nss	3.53.1	
nss-pem	1.0.3	
nss-softokn	3.53.1	
nss-softokn-freebl	3.53.1	
nss-sysinit	3.53.1	
nss-tools	3.53.1	
nss-util	3.53.1	
openldap	2.4.40	
openssl	1.0.2k	
openssl-libs		3.0.8
p11-kit	0.18.5	0.24.1
p11-kit-trust	0.18.5	0.24.1
pcre	8.21	
pcre2		10.40
pcre2-syntax		10.40
pinentry	0.7.6	

패키지	AL1 컨테이너	AL2023 컨테이너
pkgconfig	0.27.1	
popt	1.13	1.18
pth	2.0.7	
publicsuffix-list-dafsa		20240212
python27	2.7.18	
python27-chardet	2.0.1	
python27-iniparse	0.3.1	
python27-kitchen	1.1.1	
python27-libs	2.7.18	
python27-pycurl	7.19.0	
python27-pygpme	0.3	
python27-pyliblzma	0.5.3	
python27-pyattr	0.5.0	
python27-urlgrabber	3.10	
python3		3.9.16
python3-dnf		4.14.0
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0

패키지	AL1 컨테이너	AL2023 컨테이너
python3-libs		3.9.16
python3-pip-wheel		21.3.1
python3-rpm		4.16.1.3
python3-setuptools-wheel		59.6.0
readline	6.2	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-python27	4.11.3	
rpm-sign-libs		4.16.1.3
sed	4.2.1	4.8
setup	2.8.14	2.13.7
shared-mime-info	1.1	
sqlite	3.7.17	
sqlite-libs		3.40.0
sysctl-defaults	1.0	
system-release	2018.03	2023.4.20240513
tar	1.26	
tzdata	2023c	2024a
xz-libs	5.2.2	5.2.5

패키지	AL1 컨테이너	AL2023 컨테이너
yum	3.4.3	4.14.0
yum-metadata-parser	1.1.4	
yum-plugin-ovl	1.1.31	
yum-plugin-priorities	1.1.31	
yum-utils	1.1.31	
zlib	1.2.8	1.2.11

AL2023 시스템 요구 사항

이 섹션에서는 AL2023 사용을 위한 시스템 요구 사항에 대해 설명합니다.

주제

- [AL2023 실행을 위한 CPU 요구 사항](#)
- [AL2023 실행을 위한 메모리 \(RAM\) 요구 사항](#)

AL2023 실행을 위한 CPU 요구 사항

AL2023 코드를 실행하려면 사용된 프로세서가 특정 최소 요구 사항을 충족해야 합니다. 이러한 요구 사항을 충족하지 않는 CPU에서 AL2023 실행을 시도하면 코드 실행 초기에 잘못된 명령 오류가 발생할 수 있습니다.

[Amazon EC2 AL2023AL2023 컨테이너](#), 및 [Amazon EC2 외부 AL2023](#)에는 최소 요구 사항이 적용됩니다.

AL2023 ARM CPU 사양

모든 AL2023 aarch64 (ARM) 바이너리는 64비트용으로 빌드됩니다. 32비트 ARM 바이너리를 사용할 수 없으므로 64비트 CPU가 필요합니다. ARM

Note

ARM 기반 인스턴스에서 AL2023은 Graviton2 이상 프로세서를 사용하는 인스턴스 유형만 지원합니다. AL2023은 A1 인스턴스를 지원하지 않습니다.

AL2023을 사용하려면 암호화 확장(ARMv8.2+crypto)을 갖춘 ARMv8.2 호환 프로세서가 필요합니다. 의 모든 AL2023 패키지는 컴파일러 플래그를 aarch64 사용하여 빌드됩니다. -march=armv8.2-a+crypto 구형 ARM 프로세서에서 AL2023 코드를 실행하려고 할 때 정상적인 오류 메시지를 출력하려고 하지만 첫 번째 오류 메시지가 잘못된 명령 오류일 수 있습니다.

Note

AL2023 aarch64 기본 CPU 요구 사항으로 인해 이전 Raspberry Pi 시스템은 최소 CPU 요구 사항을 충족하지 Raspberry Pi 5 못합니다.

AL2023 x86-64 CPU 사양

모든 AL2023 x86-64 바이너리는 컴파일러에 x86-64v2 전달되어 x86-64 아키텍처를 수정할 - march=x86-64-v2 수 있도록 빌드됩니다.

아키텍처 x86-64v2 개정판은 기본 x86-64 아키텍처 위에 다음과 같은 CPU 기능을 추가합니다.

- CMPXCHG16B
- LAHF-SAHF
- POPCNT
- SSE3
- SSE4_1
- SSE4_2
- SSSE3

이는 2009년 이후에 출시된 x86-64 프로세서와 대략 일치합니다. 예로는 Intel Nehalem,, AMD JaguarAtom Silvermont, VIA Nano 및 Eden C 마이크로아키텍처가 있습니다.

Amazon EC2의 모든 x86-64 인스턴스 유형은 M1, C1 및 M2 인스턴스 패밀리 등 x86-64v2를 지원합니다.

32비트 x86 (i686) AL2023 바이너리는 빌드되지 않았습니다. Amazon Linux는 32비트 사용자 공간 바이너리를 계속 지원하지만 이 기능은 더 이상 사용되지 않으며 향후 Amazon Linux 메이저 버전에서 제거될 수 있습니다. AL2023 자세한 정보는 [32비트 x86 \(i686\) 패키지](#)를 참조하세요.

AL2023 실행을 위한 메모리 (RAM) 요구 사항

Amazon EC2 인스턴스 유형 .nano 패밀리 (t2.nano, t3.nanot3a.nano, 및t4g.nano) 에는 AL2023 최소 요구 사항인 512MB의 RAM이 있습니다.

Note

최소 요구 사항은 512MB이지만 이러한 인스턴스 유형은 메모리가 제한되며 기능과 성능이 제한될 수 있습니다.

AL2023 이미지는 RAM이 512MB 미만인 시스템에서 테스트되지 않았습니다. 512MB 미만의 RAM에서 AL2023 기반 컨테이너 이미지를 실행하는 것은 컨테이너식 워크로드에 따라 달라집니다.

일부 AL2023 릴리스 `dnf update` 간과 같은 일부 워크로드에는 512MB 이상의 RAM이 필요할 수 있습니다. 이러한 이유로 [AL2023.3](#) 릴리스에는 RAM이 800MB 미만인 인스턴스에 대해 `zram` 기본적으로 활성화가 도입되었습니다. 컨테이너식 워크로드의 경우 이 정도의 메모리를 사용하는 AL2023 인스턴스에서는 일부 워크로드가 정상적으로 실행되지만 메모리 사용량이 이 정도로 제한된 컨테이너에서 실행하면 실패할 수 있습니다.

RAM이 800MB 미만인 인스턴스 유형의 경우 AL2023 ([AL2023.3](#) 이상) 는 기본적으로 `zram` 기반 스왑을 활성화합니다. 메모리가 800MB 미만인 Amazon EC2 인스턴스 유형의 예로는 `t4g.nano`,, `t3a.nano` `t3.nanot2.nano`, 등이 있습니다. `t1.micro` 즉, AL2023 명령으로 메모리 페이지를 온 디맨드 방식으로 압축 및 압축 해제하므로 이러한 인스턴스 유형의 메모리 부족 현상이 줄어듭니다. 이렇게 하면 압축을 수행하는 데 필요한 CPU 사용량을 희생시키면서 더 많은 메모리가 있는 인스턴스 유형이 필요한 워크로드도 가능해집니다.

AL2023 켜기 사용 AWS

다른 AWS 서비스사람과 함께 사용하도록 AL2023 설정을 할 수 있습니다. 예를 들어, [Amazon Elastic Compute Cloud \(Amazon EC2\)](#) 인스턴스를 시작할 때 AL2023 AMI를 선택할 수 있습니다.

이러한 설정 절차에서는 AWS Identity and Access Management (IAM) 서비스를 사용합니다. IAM에 대한 자세한 내용은 다음 참조 자료를 참조하세요.

- [AWS Identity and Access Management \(IAM\)](#)
- [IAM 사용 설명서](#)

주제

- [시작하기 AWS](#)
- [Amazon EC2 AL2023](#)
- [컨테이너에서 AL2023 사용하기](#)
- [AL2023 켜기 AWS Elastic Beanstalk](#)
- [에서 AL2023 사용하기 AWS CloudShell](#)
- [AL2023 기반 Amazon ECS AMI를 사용하여 컨테이너식 워크로드를 호스팅합니다.](#)
- [AL2023 기반 아마존 Elastic File System 사용](#)
- [AL2023 기반으로 구축된 Amazon EMR 사용](#)
- [에서 AL2023 사용하기 AWS Lambda](#)

시작하기 AWS

등록해 보세요 AWS 계정

계정이 없는 경우 다음 단계를 완료하여 계정을 만드세요. AWS 계정

가입하려면 AWS 계정

1. <https://portal.aws.amazon.com/billing/signup>을 여세요.
2. 온라인 지시 사항을 따르세요.

등록 절차 중에는 전화를 받고 키패드로 인증 코드를 입력하는 과정이 있습니다.

에 AWS 계정가입하면 AWS 계정 루트 사용자a가 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스 액세스 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 [루트 사용자 액세스 권한이 필요한 작업을 수행하는 것](#)입니다.

AWS 가입 절차가 완료된 후 확인 이메일을 보냅니다. 언제든지 <https://aws.amazon.com/>으로 가서 내 계정(My Account)을 선택하여 현재 계정 활동을 보고 계정을 관리할 수 있습니다.

관리자 액세스 권한이 있는 사용자 생성

등록한 AWS 계정후에는 일상적인 작업에 루트 사용자를 사용하지 않도록 관리 사용자를 보호하고 AWS IAM Identity Center활성화하고 생성하십시오 AWS 계정 루트 사용자.

보안을 유지하세요 AWS 계정 루트 사용자

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 계정 [AWS Management Console](#)소유자로 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

루트 사용자를 사용하여 로그인하는 데 도움이 필요하면AWS 로그인 사용 설명서의 [루트 사용자 로 로그인](#)을 참조하세요.

2. 루트 사용자의 다중 인증(MFA)을 활성화합니다.

지침은 IAM [사용 설명서의 AWS 계정 루트 사용자 \(콘솔\)에 대한 가상 MFA 디바이스 활성화를 참조](#)하십시오.

관리자 액세스 권한이 있는 사용자 생성

1. IAM Identity Center를 활성화합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [AWS IAM Identity Center설정](#)을 참조하세요.

2. IAM Identity Center에서 사용자에게 관리 액세스 권한을 부여합니다.

를 ID 소스로 사용하는 방법에 대한 자습서는 사용 [설명서의 기본값으로 IAM Identity Center 디렉터리사용자 액세스 구성](#)을 참조하십시오. IAM Identity Center 디렉터리 AWS IAM Identity Center

관리 액세스 권한이 있는 사용자로 로그인

- IAM IDentity Center 사용자로 로그인하려면 IAM IDentity Center 사용자를 생성할 때 이메일 주소로 전송된 로그인 URL을 사용합니다.

IAM IDentity Center 사용자를 사용하여 [로그인하는 데 도움이 필요하면 사용 설명서의 AWS 액세스 포털에 로그인](#)을 참조하십시오. AWS 로그인

추가 사용자에게 액세스 권한 할당

1. IAM IDentity Center에서 최소 권한 적용 모범 사례를 따르는 권한 세트를 생성합니다.

지침은 AWS IAM IDentity Center 사용 설명서의 [Create a permission set](#)를 참조하세요.

2. 사용자를 그룹에 할당하고, 그룹에 Single Sign-On 액세스 권한을 할당합니다.

지침은 AWS IAM IDentity Center 사용 설명서의 [Add groups](#)를 참조하세요.

프로그래밍 방식 액세스 권한 부여

사용자가 AWS 외부 사용자와 상호 작용하려는 경우 프로그래밍 방식의 액세스가 필요합니다. AWS Management Console 프로그래밍 방식의 액세스 권한을 부여하는 방법은 액세스하는 사용자 유형에 따라 다릅니다. AWS

사용자에게 프로그래밍 방식 액세스 권한을 부여하려면 다음 옵션 중 하나를 선택합니다.

프로그래밍 방식 액세스가 필요한 사용자는 누구인가요?	To	액세스 권한을 부여하는 사용자
작업 인력 ID (IAM IDentity Center가 관리하는 사용자)	임시 자격 증명을 사용하여 AWS CLI, AWS SDK 또는 API에 대한 프로그래밍 요청에서 명할 수 있습니다. AWS	<p>사용하고자 하는 인터페이스에 대한 지침을 따릅니다.</p> <ul style="list-style-type: none"> • AWS CLI에 대한 내용은 사용 설명서의 AWS CLI 사용을 AWS IAM IDentity Center위 한 구성을 참조하십시오. AWS Command Line Interface • AWS SDK, 도구 및 AWS API의 경우 AWS SDK 및 도

<p>프로그래밍 방식 액세스가 필요한 사용자는 누구인가요?</p>	<p>To</p>	<p>액세스 권한을 부여하는 사용자</p>
		<p>구 참조 안내서의 IAM ID 센터 인증을 참조하십시오.</p>
<p>IAM</p>	<p>임시 자격 증명을 사용하여 AWS CLI, AWS SDK 또는 API에 대한 프로그래밍 방식 요청에 서명할 수 있습니다. AWS</p>	<p>IAM 사용 설명서의 AWS 리소스와 함께 임시 자격 증명 사용의 지침을 따르십시오.</p>
<p>IAM</p>	<p>(권장되지 않음) 장기 자격 증명을 사용하여 AWS CLI, AWS SDK 또는 API에 대한 프로그래밍 요청에 서명할 수 있습니다. AWS</p>	<p>사용하고자 하는 인터페이스에 대한 지침을 따릅니다.</p> <ul style="list-style-type: none"> • 에 대한 내용은 사용 설명서의 IAM 사용자 자격 증명을 사용한 인증을 참조하십시오. AWS CLI AWS Command Line Interface • AWS SDK 및 도구의 경우 SDK 및 도구 참조 안내서의 장기 자격 증명을 사용한 인증을 참조하십시오. AWS • AWS API의 경우 IAM 사용 설명서의 IAM 사용자의 액세스 키 관리를 참조하십시오.

Amazon EC2 AL2023

다음 절차 중 하나를 사용하여 AL2023 AMI를 사용하여 Amazon EC2 인스턴스를 시작할 수 있습니다. 표준 AMI 또는 미니멀 AMI를 선택합니다. 표준 AMI와 미니멀 AMI 차이에 대한 자세한 내용은 [AL2023 표준\(기본\) AMI와 미니멀 AMI 비교](#)을 참조하세요.

주제

- [아마존 EC2 콘솔을 사용하여 AL2023 실행](#)
- [SSM 파라미터를 사용하여 AL2023 시작 및 AWS CLI](#)

- [를 사용하여 최신 AL2023 AMI 실행 AWS CloudFormation](#)
- [특정 AMI ID를 사용하여 AL2023 실행](#)
- [AL2023 AMI 지원 중단 및 수명 주기](#)
- [AL2023 인스턴스에 연결](#)
- [AL2023 표준 AMI와 최소 AMI 비교](#)

아마존 EC2 콘솔을 사용하여 AL2023 실행

Amazon EC2 콘솔에서 AL2023 AMI를 시작합니다.

Note

ARM 기반 인스턴스에서 AL2023은 Graviton2 이상 프로세서를 사용하는 인스턴스 유형만 지원됩니다. AL2023은 A1 인스턴스를 지원하지 않습니다.

다음 단계를 거쳐 Amazon EC2 콘솔의 AL2023 AMI로 Amazon EC2 인스턴스를 시작합니다.

AL2023 AMI를 사용하여 EC2 인스턴스를 시작하려면

1. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창에서 AMI를 선택합니다.
3. 드롭다운 메뉴에서 공개 이미지(Public images)를 선택합니다.
4. 검색 필드에 **a12023-ami**를 입력합니다.

Note

소유자 별칭에 amazon이 표시되는지 확인하세요.

5. 목록에서 이미지를 선택합니다. 소스에서 AMI가 표준인지 미니멀인지 확인할 수 있습니다. AL2023 AMI 이름은 다음 형식으로 해석할 수 있습니다.

```
'a12023-[ami || ami-minimal]-2023.0.[release build date].[build number]-kernel-[version number]-[arm64 || x86_64]'
```

6. 다음 이미지는 AL2023 AMI 일부 목록입니다.

	Name	AMI ID	AMI name	Source	Owner	Owner alias
<input type="checkbox"/>	-	ami-000a4d9c6067d5d0d	al2023-ami-2023.0.20230222.1...	amazon/al2023-ami-2023.0.20230222.1-kernel-6.1-arm64	137112412989	amazon
<input type="checkbox"/>	-	ami-0a409f3927bd2662f	al2023-ami-2023.0.20230222.1...	amazon/al2023-ami-2023.0.20230222.1-kernel-6.1-x86_64	137112412989	amazon
<input type="checkbox"/>	-	ami-043e11d11db3d437e	al2023-ami-minimal-2023.0.20...	amazon/al2023-ami-minimal-2023.0.20230222.1-kernel-6.1-ar...	137112412989	amazon
<input type="checkbox"/>	-	ami-0d19aa82c9a61ef2c	al2023-ami-minimal-2023.0.20...	amazon/al2023-ami-minimal-2023.0.20230222.1-kernel-6.1-x8...	137112412989	amazon

Amazon EC2 인스턴스 시작에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [Amazon EC2 Linux 인스턴스 시작하기](#)를 참조하십시오.

SSM 파라미터를 사용하여 AL2023 시작 및 AWS CLI

에서 AMI의 SSM 파라미터 값을 사용하여 새 AL2023 인스턴스를 시작할 수 있습니다. AWS CLI구체적으로 설명하면, 다음 목록의 동적 SSM 파라미터 값 중 하나를 SSM 파라미터 `/aws/service/ami-amazon-linux-latest/` 값/ 앞에 넣으세요. 이렇게 하면 AWS CLI인스턴스를 시작할 수 있습니다.

- arm64 아키텍처 `al2023-ami-kernel-default-arm64`
- arm64 아키텍처 (미니멀 AMI) `al2023-ami-minimal-kernel-default-arm64`
- x86_64 아키텍처 `al2023-ami-kernel-default-x86_64`
- x86_64 아키텍처 (미니멀 AMI) `al2023-ami-minimal-kernel-default-x86_64`

Note

항목은 예제 파라미터입니다. 이를 사용자의 정보로 대체합니다.

```
$ aws ec2 run-instances \
  --image-id \
    resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-x86_64 \
  --instance-type m5.xlarge \
  --region us-east-1 \
  --key-name aws-key-us-east-1 \
  --security-group-ids sg-004a7650
```

`--image-id` 플래그가 SSM 파라미터의 값을 지정합니다.

--instance-type 플래그는 인스턴스의 유형과 크기를 지정합니다. 이 플래그는 선택한 AMI 유형과 호환되어야 합니다.

--region 플래그는 인스턴스를 생성하는 AWS 리전 위치를 지정합니다.

--key-name 플래그는 인스턴스에 연결하는 데 사용되는 키를 지정합니다. AWS 리전 인스턴스를 만든 리전에 있는 키를 사용하지 않으면 SSH를 사용하여 인스턴스에 연결할 수 없습니다.

--security-group-ids 플래그는 인바운드 및 아웃바운드 네트워크 트래픽에 대한 액세스 권한을 결정하는 보안 그룹을 지정합니다.

Important

이를 AWS CLI 위해서는 포트를 통해 원격 시스템에서 인스턴스에 액세스할 수 있는 기존 보안 그룹을 지정해야 TCP:22 합니다. 지정된 보안 그룹이 없으면 새 인스턴스는 기본 보안 그룹에 배치됩니다. 기본 보안 그룹의 인스턴스는 VPC 내의 다른 인스턴스만 연결할 수 있습니다.

자세한 내용을 알아보려면 AWS Command Line Interface 사용 설명서의 [Amazon EC2 인스턴스 시작, 목록 작성 및 종료](#)를 참조하세요.

를 사용하여 최신 AL2023 AMI 실행 AWS CloudFormation

를 사용하여 AWS CloudFormation AL2023 AMI를 시작하려면 다음 템플릿 중 하나를 사용하십시오.

Note

x86_64 및 Arm64 AMI는 각각 다른 인스턴스 유형이 필요합니다. 자세한 내용은 [Amazon EC2 인스턴스 유형](#)을 참조하세요.

JSON 템플릿:

```
{
  "Parameters": {
    "LatestAmiId": {
      "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
      "Default": "/aws/service/ami-amazon-linux-latest/al2023-ami-minimal-kernel-
default-x86_64"
    }
  },
}
```

```

"Resources": {
  "MyEC2Instance": {
    "Type": "AWS::EC2::Instance",
    "Properties": {
      "InstanceType": "t2.large",
      "ImageId": {
        "Ref": "LatestAmiId"
      }
    }
  }
}
}
}
}

```

YAML 템플릿:

```

Parameters:
  LatestAmiId:
    Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
    Default: '/aws/service/ami-amazon-linux-latest/al2023-ami-minimal-kernel-default-x86_64'

Resources:
  Instance:
    Type: 'AWS::EC2::Instance'
    Properties:
      InstanceType: 't2.large'
      ImageId: !Ref LatestAmiId

```

필요한 경우 “Default” 섹션 마지막에 있는 AMI 파라미터를 교체하세요. 다음과 같은 파라미터 값을 사용할 수 있습니다.

- arm64 아키텍처 al2023-ami-kernel-6.1-arm64
- arm64 아키텍처 (미니멀 AMI) al2023-ami-minimal-kernel-6.1-arm64
- x86_64 아키텍처 al2023-ami-kernel-6.1-x86_64
- x86_64 아키텍처 (미니멀 AMI) al2023-ami-minimal-kernel-6.1-x86_64

다음은 동적 커널 사양입니다. 기본 커널 버전은 주요 커널 버전이 업데이트될 때마다 자동으로 변경됩니다.

- arm64 아키텍처 al2023-ami-kernel-default-arm64

- arm64 아키텍처 (미니멀 AMI) `al2023-ami-minimal-kernel-default-arm64`
- x86_64 아키텍처 `al2023-ami-kernel-default-x86_64`
- x86_64 아키텍처 (미니멀 AMI) `al2023-ami-minimal-kernel-default-x86_64`

특정 AMI ID를 사용하여 AL2023 실행

AMI ID를 사용하여 특정 AL2023 AMI를 시작할 수 있습니다. Amazon EC2 콘솔의 AMI 목록을 보고 어떤 AL2023 AMI ID가 필요한지 확인할 수 있습니다. 또는 사용할 AWS Systems Manager 수도 있습니다. Systems Manager를 사용하는 경우 이전 섹션에 나온 AMI 별칭 중에서 선택하세요. 자세한 내용은 [AWS Systems Manager 파라미터 스토어를 사용하여 최신 Amazon Linux AMI ID 쿼리](#)를 참조하십시오.

AL2023 AMI 지원 중단 및 수명 주기

새 AL2023 릴리스에는 새 AMI가 포함됩니다. AMI를 등록하면 지원 중단 날짜가 표시됩니다. AL2023 AMI 지원 중단일은 각 개별 커널 릴리스의 [AL2023 기반 커널 라이브 패칭](#) 기간과 일치하는 출시된 시점으로부터 90일입니다.

Note

90일 개별 AMI 지원 종단을 의미하며 AL2023 [릴리스 케이던스](#) 또는 제품 지원 기간을 의미하지는 않습니다.

AMI 지원 중단에 대한 자세한 내용은 Amazon EC2 사용 [설명서의 AMI](#) 지원 종단을 참조하십시오.

업데이트된 AMI를 정기적으로 사용하여 인스턴스를 시작하면 업데이트된 커널을 포함한 최신 보안으로 업데이트된 인스턴스를 시작할 수 있습니다. 구 AMI 버전을 실행하여 업데이트한다면 인스턴스에 최신 보안 업데이트가 적용되지 않는 기간이 있습니다. 최신 AMI를 사용 중인지 확인하려면 SSM 파라미터를 사용하는 것이 좋습니다.

SSM 파라미터를 사용하여 인스턴스를 시작하는 방법에 대한 자세한 내용은 다음을 참조하세요.

- [SSM 파라미터를 사용하여 AL2023 시작 및 AWS CLI](#)
- [를 사용하여 최신 AL2023 AMI 실행 AWS CloudFormation](#)

AL2023 인스턴스에 연결

SSH 또는 AL2023 인스턴스에 AWS Systems Manager 연결하려면 SSH 또는 사용하십시오.

SSH를 사용하여 인스턴스에 연결

SSH를 사용하여 인스턴스에 연결하는 방법에 대한 지침은 Amazon EC2 사용 설명서의 [SSH를 사용하여 Linux 인스턴스에 연결을](#) 참조하십시오.

를 사용하여 인스턴스에 연결 AWS Systems Manager

AL2023 인스턴스에 연결하는 AWS Systems Manager 데 사용하는 방법에 대한 지침은 Amazon EC2 사용 설명서의 [세션 관리자를 사용하여 Linux 인스턴스에 연결을](#) 참조하십시오.

Amazon EC2 인스턴스 연결 사용

최소 AMI를 제외한 AL2023 AMI에는 EC2 인스턴스 연결 에이전트가 기본적으로 설치되어 있습니다. 최소 AMI에서 시작된 AL2023 인스턴스와 함께 EC2 인스턴스 연결을 사용하려면 `ec2-instance-connect` 패키지를 설치해야 합니다. EC2 인스턴스 연결 사용에 대한 지침은 Amazon EC2 사용 설명서의 [EC2 인스턴스 연결을 사용하여 Linux 인스턴스에 연결을](#) 참조하십시오.

AL2023 표준 AMI와 최소 AMI 비교

표준 (기본) 또는 최소 AL2023 AMI를 사용하여 Amazon EC2 인스턴스를 시작할 수 있습니다. 표준 또는 최소 AMI 유형으로 Amazon EC2 인스턴스를 시작하는 방법에 대한 지침은 을 참조하십시오.

[Amazon EC2 AL2023](#)

표준 AL2023 AMI에는 가장 일반적으로 사용되는 애플리케이션과 도구가 모두 설치되어 있습니다. 빠르게 시작하고 싶어서 AMI 사용자 지정을 사용하지 않는다면 표준 AMI를 사용하는 것이 좋습니다.

최소 AL2023 AMI는 운영 체제 (OS) 를 실행하는 데 필요한 가장 기본적인 도구와 유틸리티만 포함하는 간소화된 기본 버전입니다. OS 설치 공간을 최대한 줄이려면 미니멀 AMI를 사용하는 것이 좋습니다. 미니멀 AMI로 디스크 공간 사용률을 조금 줄이고 장기적인 비용 효율성을 향상시킬 수 있습니다. 크기가 작은 OS를 사용하고 도구와 애플리케이션을 수동으로 설치해도 좋다면 미니멀 AMI가 적합합니다.

컨테이너 이미지는 AL2023 미니멀 AMI 패키지 세트에 더 가깝습니다.

Amazon Linux 2023 Images에 설치된 패키지 비교

AL2023 AMI, 최소 AMI 및 컨테이너 이미지에 있는 RPM을 비교한 것입니다.

패키지	AMI	미니멀 AMI	컨테이너
acl	2.3.1		
acpid	2.0.32		
alternatives	1.15	1.15	1.15
amazon-chrony-config	4.3	4.3	
amazon-ec2-net-utils	2.4.1	2.4.1	
amazon-linux-repo-cdn			2023.4.20240513
amazon-linux-repo-s3	2023.4.20240513	2023.4.20240513	
amazon-linux-sb-keys	2023.1	2023.1	
amazon-rpm-config	228		
amazon-ssm-agent	3.3.380.0		
at	3.1.23		
attr	2.5.1		
audit	3.0.6	3.0.6	
audit-libs	3.0.6	3.0.6	3.0.6
aws-cfn-bootstrap	2.0		

패키지	AMI	미니멀 AMI	컨테이너
awscli-2	2.15.30	2.15.30	
basesystem	11	11	11
bash	5.2.15	5.2.15	5.2.15
bash-completion	2.11		
bc	1.07.1		
bind-libs	9.16.48		
bind-license	9.16.48		
bind-utils	9.16.48		
binutils	2.39		
boost-fil esystem	1.75.0		
boost-system	1.75.0		
boost-thread	1.75.0		
bzip2	1.0.8		
bzip2-libs	1.0.8	1.0.8	1.0.8
ca-certificates	2023.2.64	2023.2.64	2023.2.64
c-ares	1.19.0		
checkpolicy	3.4	3.4	
chkconfig	1.15		
chrony	4.3	4.3	
cloud-init	22.2.2	22.2.2	

패키지	AMI	미니멀 AMI	컨테이너
cloud-init-cfg-ec2	22.2.2	22.2.2	
cloud-utils-growpart	0.31	0.31	
coreutils	8.32	8.32	
coreutils-common	8.32	8.32	
coreutils-single			8.32
cpio	2.13	2.13	
cracklib	2.9.6	2.9.6	
cracklib-dicts	2.9.6	2.9.6	
crontabs	1.11		
crypto-policies	20220428	20220428	20220428
crypto-policies-scripts	20220428		
cryptsetup	2.6.1		
cryptsetup-libs	2.6.1	2.6.1	
curl-minimal	8.5.0	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27	
cyrus-sasl-plain	2.1.27		
dbus	1.12.28	1.12.28	

패키지	AMI	미니멀 AMI	컨테이너
dbus-broker	32	32	
dbus-common	1.12.28	1.12.28	
dbus-libs	1.12.28	1.12.28	
device-mapper	1.02.185	1.02.185	
device-mapper-libs	1.02.185	1.02.185	
diffutils	3.8	3.8	
dnf	4.14.0	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2	
dnf-plugins-core	4.3.0	4.3.0	
dnf-plugin-support-info	1.2	1.2	
dnf-utils	4.3.0		
dosfstools	4.2		
dracut	055	055	
dracut-config-ec2	3.0	3.0	
dracut-config-generic	055	055	

패키지	AMI	미니멀 AMI	컨테이너
dwz	0.14		
dyninst	10.2.1		
e2fsprogs	1.46.5	1.46.5	
e2fsprogs-libs	1.46.5	1.46.5	
ec2-hibinit-agent	1.0.8		
ec2-instance-connect	1.1		
ec2-instance-connect-selinux	1.1		
ec2-utils	2.2.0	2.2.0	
ed	1.14.2		
efi-filesystem	5	5	
efi-srpm-macros	5		
efivar	38	38	
efivar-libs	38	38	
elfutils-debuginfod-client	0.188		
elfutils-default-yama-scope	0.188	0.188	0.188

패키지	AMI	미니멀 AMI	컨테이너
elfutils-libelf	0.188	0.188	0.188
elfutils-libs	0.188	0.188	0.188
ethtool	5.15		
expat	2.5.0	2.5.0	2.5.0
file	5.39	5.39	
file-libs	5.39	5.39	5.39
filesystem	3.14	3.14	3.14
findutils	4.8.0	4.8.0	
fonts-srpm-macros	2.0.5		
fstrm	0.6.1		
fuse-libs	2.9.9	2.9.9	
gawk	5.1.0	5.1.0	5.1.0
gdbm-libs	1.19	1.19	1.19
gdisk	1.0.8	1.0.8	
gettext	0.21	0.21	
gettext-libs	0.21	0.21	
ghc-srpm-macros	1.5.0		
glib2	2.74.7	2.74.7	2.74.7
glibc	2.34	2.34	2.34

패키지	AMI	미니멀 AMI	컨테이너
glibc-all-langpacks	2.34	2.34	
glibc-common	2.34	2.34	2.34
glibc-gconv-extra	2.34		
glibc-locale-source	2.34	2.34	
glibc-minimal-langpack			2.34
gmp	6.2.1	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7	2.3.7
gnutls	3.8.0	3.8.0	
go-srpm-macros	3.2.0		
gpgme	1.15.1	1.15.1	1.15.1
gpm-libs	1.20.7		
grep	3.8	3.8	3.8
groff-base	1.22.4	1.22.4	
grub2-common	2.06	2.06	
grub2-efi-aa64-ec2	2.06 (aarch64)	2.06 (아키텍처64)	
grub2-efi-x64-ec2	2.06 (x86_64)	2.06 (x86_64)	

패키지	AMI	미니멀 AMI	컨테이너
grub2-pc-modules	2.06	2.06	
grub2-tools	2.06	2.06	
grub2-tools-minimal	2.06	2.06	
grubby	8.40	8.40	
gssproxy	0.8.4		
gzip	1.12	1.12	
hostname	3.23	3.23	
hunspell	1.7.0		
hunspell-en	0.20140811.1		
hunspell-en-GB	0.20140811.1		
hunspell-en-US	0.20140811.1		
hunspell-filesystem	1.7.0		
hwdata	0.353	0.353	
info	6.7		
inih	49	49	
initscripts	10.09	10.09	
iproute	5.10.0	5.10.0	
iputils	20210202	20210202	

패키지	AMI	미니멀 AMI	컨테이너
irqbalance	1.9.0	1.9.0	
jansson	2.14	2.14	
jitterentropy	3.4.1	3.4.1	
jq	1.7.1	1.7.1	
json-c	0.14	0.14	0.14
kbd	2.4.0	2.4.0	
kbd-misc	2.4.0	2.4.0	
kernel	6.1.90	6.1.90	
kernel- li vepatch-repo- s3	2023.4.20240513	2023.4.20240513	
kernel-srpm- macros	1.0		
kernel-tools	6.1.90		
keyutils	1.6.3		
keyutils-libs	1.6.3	1.6.3	1.6.3
kmod	29	29	
kmod-libs	29	29	
kpatch-runtime	0.9.7		
krb5-libs	1.21	1.21	1.21
less	608	608	

패키지	AMI	미니멀 AMI	컨테이너
libacl	2.3.1	2.3.1	2.3.1
libaio	0.3.111		
libarchive	3.5.3	3.5.3	3.5.3
libargon2	20171227	20171227	
libassuan	2.5.5	2.5.5	2.5.5
libattr	2.5.1	2.5.1	2.5.1
libbasicobjects	0.1.1		
libblkid	2.37.4	2.37.4	2.37.4
libcap	2.48	2.48	2.48
libcap-ng	0.8.2	0.8.2	0.8.2
libcbor	0.7.0	0.7.0	
libcollection	0.7.0		
libcom_err	1.46.5	1.46.5	1.46.5
libcomps	0.1.20	0.1.20	0.1.20
libconfig	1.7.2		
libcurl-minimal	8.5.0	8.5.0	8.5.0
libdb	5.3.28	5.3.28	
libdhash	0.5.0		
libdnf	0.69.0	0.69.0	0.69.0
libeconf	0.4.0	0.4.0	

패키지	AMI	미니멀 AMI	컨테이너
libedit	3.1	3.1	
libev	4.33		
libevent	2.1.12		
libfdisk	2.37.4	2.37.4	
libffi	3.4.4	3.4.4	3.4.4
libfido2	1.10.0	1.10.0	
libgcc	11.4.1	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2	1.10.2
libgomp	11.4.1	11.4.1	11.4.1
libgpg-error	1.42	1.42	1.42
libibverbs	48.0		
libidn2	2.3.2	2.3.2	2.3.2
libini_config	1.3.1		
libkcapi	1.4.0	1.4.0	
libkcapi-hmacalc	1.4.0	1.4.0	
libldb	2.6.2		
libmaxminddb	1.5.2		
libmetalink	0.1.3		
libmnl	1.0.4	1.0.4	
libmodulemd	2.13.0	2.13.0	2.13.0

패키지	AMI	미니멀 AMI	컨테이너
libmount	2.37.4	2.37.4	2.37.4
libnfsidmap	2.5.4		
libnghttp2	1.59.0	1.59.0	1.59.0
libnl3	3.5.0		
libpath_utils	0.2.1		
libpcap	1.10.1		
libpipeline	1.5.3	1.5.3	
libpkgconf	1.8.0		
libpsl	0.21.1	0.21.1	0.21.1
libpwquality	1.4.4	1.4.4	
libref_array	0.1.5		
librepo	1.14.5	1.14.5	1.14.5
libreport-filesystem	2.15.2	2.15.2	2.15.2
libseccomp	2.5.3	2.5.3	
libselenium	3.4	3.4	3.4
libselenium-utils	3.4	3.4	
libsemanage	3.4	3.4	
libsepol	3.4	3.4	3.4
libsigsegv	2.13	2.13	2.13

패키지	AMI	미니멀 AMI	컨테이너
libsmartcols	2.37.4	2.37.4	2.37.4
libsolv	0.7.22	0.7.22	0.7.22
libss	1.46.5	1.46.5	
libsss_certmap	2.9.4		
libsss_idmap	2.9.4		
libsss_nss_idmap	2.9.4		
libsss_sudo	2.9.4		
libstdc++	11.4.1	11.4.1	11.4.1
libstoragemgmt	1.9.4		
libtalloc	2.3.4		
libtasn1	4.19.0	4.19.0	4.19.0
libtcl	1.4.7		
libtevent	0.13.0		
libtextstyle	0.21	0.21	
libtirpc	1.3.3		
libunistring	0.9.10	0.9.10	0.9.10
libuser	0.63	0.63	
libutempter	1.2.1	1.2.1	
libuuid	2.37.4	2.37.4	2.37.4
libuv	1.47.0		

패키지	AMI	미니멀 AMI	컨테이너
libverto	0.3.2	0.3.2	0.3.2
libverto-libev	0.3.2		
libxcrypt	4.4.33	4.4.33	4.4.33
libxml2	2.10.4	2.10.4	2.10.4
libyaml	0.2.5	0.2.5	0.2.5
libzstd	1.5.5	1.5.5	1.5.5
lm_sensors-libs	3.6.0		
lmbd-libs	0.9.29		
logrotate	3.20.1	3.20.1	
lsof	4.94.0		
lua-libs	5.4.4	5.4.4	5.4.4
lua-srpm-macros	1		
lz4-libs	1.9.4	1.9.4	1.9.4
man-db	2.9.3	2.9.3	
man-pages	5.10		
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)	
mpfr	4.1.0	4.1.0	4.1.0
nano	5.8		
ncurses	6.2	6.2	
ncurses-base	6.2	6.2	6.2

패키지	AMI	미니멀 AMI	컨테이너
ncurses-libs	6.2	6.2	6.2
nettle	3.8	3.8	
net-tools	2.0	2.0	
newt	0.52.21		
nfs-utils	2.5.4		
npth	1.6	1.6	1.6
nspr	4.35.0		
nss	3.90.0		
nss-softokn	3.90.0		
nss-softokn-freebl	3.90.0		
nss-sysinit	3.90.0		
nss-util	3.90.0		
ntsysv	1.15		
numactl-libs	2.0.14	2.0.14	
ocaml-srpm-macros	6		
oniguruma	6.9.7.1	6.9.7.1	
openblas-srpm-macros	2		
openldap	2.4.57	2.4.57	

패키지	AMI	미니멀 AMI	컨테이너
openssh	8.7p1	8.7p1	
openssh-clients	8.7p1	8.7p1	
openssh-server	8.7p1	8.7p1	
openssl	3.0.8	3.0.8	
openssl-lib	3.0.8	3.0.8	3.0.8
openssl-pkcs11	0.4.12	0.4.12	
os-prober	1.77	1.77	
p11-kit	0.24.1	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1	0.24.1
package-notes-srpm-macros	0.4		
pam	1.5.1	1.5.1	
parted	3.4		
passwd	0.80	0.80	
pciutils	3.7.0	3.7.0	
pciutils-lib	3.7.0	3.7.0	
pcre2	10.40	10.40	10.40
pcre2-syntax	10.40	10.40	10.40
perl-Carp	1.50		
perl-Class-Struct	0.66		

패키지	AMI	미니멀 AMI	컨테이너
perl-constant	1.33		
perl-DynaLoader	1.47		
perl-Encode	3.15		
perl-Errno	1.30		
perl-Exporter	5.74		
perl-Fcntl	1.13		
perl-File-Basename	2.85		
perl-File-Path	2.18		
perl-File-stat	1.09		
perl-File-Temp	0.231.100		
perl-Getopt-Long	2.52		
perl-Getopt-Std	1.12		
perl-HTTP-Tiny	0.078		
perl-if	0.60.800		
perl-integerpreter	5.32.1		
perl-IO	1.43		
perl-IPC-Open3	1.21		
perl-libs	5.32.1		

패키지	AMI	미니멀 AMI	컨테이너
perl-MIME-Base64	3.16		
perl-mro	1.23		
perl-overload	1.31		
perl-overloading	0.02		
perl-parent	0.238		
perl-PathTools	3.78		
perl-Pod-Escapes	1.07		
perl-podlators	4.14		
perl-Pod-Perldoc	3.28.01		
perl-Pod-Simple	3.42		
perl-Pod-Usage	2.01		
perl-POSIX	1.94		
perl-Scalar-List-Utills	1.56		
perl-SelectSaver	1.02		
perl-Socket	2.032		
perl-srpm-macros	1		

패키지	AMI	미니멀 AMI	컨테이너
perl-Storable	3.21		
perl-subst	1.03		
perl-Symbol	1.08		
perl-Term-ANSIColor	5.01		
perl-Term-Cap	1.17		
perl-Text-ParseWords	3.30		
perl-Text-Tabs+Wrap	2021.0726		
perl-Time-Local	1.300		
perl-vars	1.05		
pkgconf	1.8.0		
pkgconf-m4	1.8.0		
pkgconf-pkg-config	1.8.0		
policycoreutils	3.4	3.4	
policycoreutils-python-utils	3.4		
popt	1.18	1.18	1.18
procps-ng	3.3.17	3.3.17	
protobuf-c	1.4.1		

패키지	AMI	미니멀 AMI	컨테이너
psacct	6.6.4		
psmisc	23.4	23.4	
publicsuffix- list-dafsa	20240212	20240212	20240212
python3	3.9.16	3.9.16	3.9.16
python3-attrs	20.3.0	20.3.0	
python3-audit	3.0.6	3.0.6	
python3-awscli	0.19.19	0.19.19	
python3-babel	2.9.1	2.9.1	
python3-cffi	1.14.5	1.14.5	
python3-chardet	4.0.0	4.0.0	
python3-c olorama	0.4.4	0.4.4	
python3-c onfigobj	5.0.6	5.0.6	
python3-c ryptography	36.0.1	36.0.1	
python3-daemon	2.3.0		
python3-d ateutil	2.8.1	2.8.1	
python3-dbus	1.2.18	1.2.18	
python3-distro	1.5.0	1.5.0	

패키지	AMI	미니멀 AMI	컨테이너
python3-dnf	4.14.0	4.14.0	4.14.0
python3-dnf-plugins-core	4.3.0	4.3.0	
python3-d ocutils	0.16	0.16	
python3-gpg	1.15.1	1.15.1	1.15.1
python3-hawkey	0.69.0	0.69.0	0.69.0
python3-idna	2.10	2.10	
python3-jinja2	2.11.3	2.11.3	
python3-j mespath	0.10.0	0.10.0	
python3-j sonpatch	1.21	1.21	
python3-j sonpointer	2.0	2.0	
python3-j sonschema	3.2.0	3.2.0	
python3-l ibcomps	0.1.20	0.1.20	0.1.20
python3-libdnf	0.69.0	0.69.0	0.69.0
python3-libs	3.9.16	3.9.16	3.9.16
python3-l ibselinux	3.4	3.4	

패키지	AMI	미니멀 AMI	컨테이너
python3-l ibsemanage	3.4	3.4	
python3-l ibstoragemgmt	1.9.4		
python3-l ockfile	0.12.2		
python3-m arkupsafe	1.1.1	1.1.1	
python3-n etifaces	0.10.6	0.10.6	
python3-o authlib	3.0.2	3.0.2	
python3-pip- wheel	21.3.1	21.3.1	21.3.1
python3-ply	3.11	3.11	
python3-p olicycoreutils	3.4	3.4	
python3-p rettytable	0.7.2	0.7.2	
python3-prompt- toolkit	3.0.24	3.0.24	
python3-p ycparser	2.20	2.20	
python3-p yrsistent	0.17.3	0.17.3	

패키지	AMI	미니멀 AMI	컨테이너
python3-pyserial	3.4	3.4	
python3-pysocks	1.7.1	1.7.1	
python3-pytz	2022.7.1	2022.7.1	
python3-pyyaml	5.4.1	5.4.1	
python3-requests	2.25.1	2.25.1	
python3-rpm	4.16.1.3	4.16.1.3	4.16.1.3
python3-ruamel-yaml	0.16.6	0.16.6	
python3-ruamel-yaml-clib	0.1.2	0.1.2	
python3-setools	4.4.1	4.4.1	
python3-setuptools	59.6.0	59.6.0	
python3-setuptools-wheel	59.6.0	59.6.0	59.6.0
python3-six	1.15.0	1.15.0	
python3-systemd	235	235	
python3-urllib3	1.25.10	1.25.10	
python3-wcwidth	0.2.5	0.2.5	
python-chevron	0.13.1		

패키지	AMI	미니멀 AMI	컨테이너
python-srpm-macros	3.9		
quota	4.06		
quota-nls	4.06		
readline	8.1	8.1	8.1
rng-tools	6.14	6.14	
rootfiles	8.1	8.1	
rpcbind	1.2.6		
rpm	4.16.1.3	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3	4.16.1.3
rpm-libs	4.16.1.3	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3	4.16.1.3	
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3	
rpm-sign-libs	4.16.1.3	4.16.1.3	4.16.1.3
rsync	3.2.6		
rust-srpm-macros	21		
sbsigntools	0.9.4	0.9.4	
screen	4.8.0		
sed	4.8	4.8	4.8

패키지	AMI	미니멀 AMI	컨테이너
selinux-policy	37.22	37.22	
selinux-policy-targeted	37.22	37.22	
setup	2.13.7	2.13.7	2.13.7
shadow-utils	4.9	4.9	
slang	2.3.2		
sqlite-libs	3.40.0	3.40.0	3.40.0
sssd-client	2.9.4		
sssd-common	2.9.4		
sssd-kcm	2.9.4		
sssd-nfs-idmap	2.9.4		
strace	6.8		
sudo	1.9.15	1.9.15	
sysctl-defaults	1.0	1.0	
sysstat	12.5.6		
systemd	252.16	252.16	
systemd-libs	252.16	252.16	
systemd-networkd	252.16	252.16	
systemd-pam	252.16	252.16	

패키지	AMI	미니멀 AMI	컨테이너
systemd-resolved	252.16	252.16	
systemd-udev	252.16	252.16	
system-release	2023.4.20240513	2023.4.20240513	2023.4.20240513
systemtap-runtime	4.8		
tar	1.34	1.34	
tbb	2020.3		
tcpdump	4.99.1		
tcsh	6.24.07		
time	1.9		
traceroute	2.1.3		
tzdata	2024a	2024a	2024a
unzip	6.0		
update-motd	2.2	2.2	
userspace-rcu	0.12.1	0.12.1	
util-linux	2.37.4	2.37.4	
util-linux-core	2.37.4	2.37.4	
vim-common	9.0.2153		
vim-data	9.0.2153	9.0.2153	
vim-enhanced	9.0.2153		

패키지	AMI	미니멀 AMI	컨테이너
vim-filesystem	9.0.2153		
vim-minimal	9.0.2153	9.0.2153	
wget	1.21.3		
which	2.21	2.21	
words	3.0		
xfsdump	3.1.11		
xfspgrog	5.18.0	5.18.0	
xxd	9.0.2153		
xxhash-libs	0.8.0		
xz	5.2.5	5.2.5	
xz-libs	5.2.5	5.2.5	5.2.5
yum	4.14.0	4.14.0	4.14.0
zip	3.0		
zlib	1.2.11	1.2.11	1.2.11
zram-generator	1.1.2	1.1.2	
zram-generator-defaults	1.1.2	1.1.2	
zstd	1.5.5	1.5.5	

컨테이너에서 AL2023 사용하기

Note

AL2023 를 사용하여 Amazon ECS에서 컨테이너식 워크로드를 호스팅하는 방법에 대한 자세한 내용은 [참조하십시오. AL2023 Amazon ECS 컨테이너 호스트](#)

사용 사례에 따라 컨테이너 내에서 AL2023 을 사용할 수 있는 몇 가지 방법이 있습니다. 아마존 리눅스 2 컨테이너 이미지 및 AL2023 최소 AMI와 가장 비슷합니다. [AL2023 기본 컨테이너 이미지](#)

고급 사용자를 위해 AL2023.2 릴리스에 도입된 최소 컨테이너 이미지와 함께 [베어](#) 본 컨테이너를 구축하는 방법을 설명하는 설명서를 제공합니다.

AL2023에서도 AL2023 기반 컨테이너 이미지 또는 Linux 배포판 기반 컨테이너 등 컨테이너 워크로드를 호스팅할 수 있습니다. [AL2023 Amazon ECS 컨테이너 호스트](#) 또는 설치된 컨테이너 런타임 패키지를 사용할 수 있습니다. docker, containerd, 및 nerdctl 패키지를 AL2023에 설치하고 사용할 수 있습니다.

주제

- [AL2023 기본 컨테이너 이미지 사용](#)
- [AL2023 최소 컨테이너 이미지](#)
- [기본적인 AL2023 컨테이너 이미지 구축](#)
- [Amazon Linux 2023 컨테이너 이미지에 설치된 패키지 비교](#)
- [Amazon Linux 2023 미니멀 AMI와 컨테이너 이미지에 설치된 패키지 비교](#)

AL2023 기본 컨테이너 이미지 사용

AL2023 컨테이너 이미지는 AL2023 AMI에 포함된 것과 동일한 소프트웨어 구성 요소로 구축됩니다. 이는 Docker 워크로드의 기본 이미지로 어느 환경에나 사용할 수 있습니다. [Amazon Elastic Compute Cloud](#) Amazon EC2) 애플리케이션에 Amazon Linux AMI를 이미 사용하고 있는 경우, Amazon Linux 컨테이너 이미지를 사용하여 애플리케이션을 컨테이너화할 수 있습니다.

로컬 개발 환경에서 Amazon Linux 컨테이너 이미지를 AWS 사용한 다음 Amazon [Elastic Container Service](#) (Amazon ECS) 를 사용하도록 애플리케이션을 푸시하십시오. 자세한 정보는 Amazon Elastic Container Registry 사용 설명서의 [Amazon ECS에서 Amazon ECR 이미지 사용](#)을 참조하세요.

Amazon Linux 컨테이너 이미지는 Amazon ECR Public에서 사용할 수 있습니다. 지정 AWS 담당자를 통해 또는 [amazon-linux-2023](#) 리포지토리에 문제를 제출하여 AL2023 관련 피드백을 제공할 수 있습니다. GitHub

Amazon ECR Public에서 Amazon Linux 컨테이너 이미지를 가져오려면

1. Docker 클라이언트를 Amazon Linux Public 레지스트리에 인증합니다. 인증 토큰은 12시간 동안 유효합니다. 자세한 내용은 Amazon Elastic Container Registry 사용 설명서의 [프라이빗 레지스트리 권한](#)을 참조하세요.

Note

이 `get-login-password` 명령은 최신 버전의 버전 2를 사용하여 지원됩니다. AWS CLI 자세한 내용은 AWS Command Line Interface 사용 설명서에서 [AWS Command Line Interface 설치](#)를 참조하세요.

```
$ aws ecr-public get-login-password --region us-east-1 | docker login --username
AWS --password-stdin public.ecr.aws
```

출력값은 다음과 같습니다.

```
Login succeeded
```

2. `docker pull` 명령을 실행하여 Amazon Linux 컨테이너 이미지를 가져옵니다. Amazon ECR Public Gallery에서 Amazon Linux 컨테이너 이미지를 보려면 [Amazon ECR Public Gallery - amazonlinux](#)를 참조하세요.

Note

AL2023 Docker 컨테이너 이미지를 가져올 때 다음 형식 중 하나로 태그를 사용할 수 있습니다.

- 최신 AL2023 컨테이너 이미지 버전을 다운로드하려면 `:2023` 태그를 사용하세요.
- 다음 형식으로 AL2023 특정 버전을 다운로드하세요.
 - `:2023.[0-7 release quarter].[release date].[build number]`

다음은 :2023 태그를 사용하여 가장 최근의 AL2023 컨테이너 이미지를 가져오는 예시입니다.

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023
```

3. (선택 사항) 로컬에서 컨테이너를 실행합니다.

```
$ docker run -it --security-opt seccomp=unconfined public.ecr.aws/amazonlinux/amazonlinux:2023 /bin/bash
```

Docker Hub에서 AL2023 컨테이너 이미지를 가져오려면

1. docker pull 명령을 사용하여 AL2023 컨테이너 이미지를 가져옵니다.

```
$ docker pull amazonlinux:2023
```

2. (선택 사항) 로컬에서 컨테이너를 실행합니다.

```
$ docker run -it amazonlinux:2023 /bin/bash
```

Note

AL2023 컨테이너 이미지는 dnf 패키지 관리자만 소프트웨어 패키지를 설치할 수 있습니다. 즉, 다른 소프트웨어에 사용할 수 있는 amazon-linux-extras 명령이나 이와 비슷한 명령이 없습니다.

AL2023 최소 컨테이너 이미지

Note

표준 AL2023 컨테이너 이미지는 대부분의 사용 사례에 적합하며, 최소 컨테이너 이미지에 맞게 조정하는 것이 AL2023 기본 컨테이너 이미지에 맞게 조정하는 것보다 더 많은 작업이 필요할 수 있습니다.

AL2023.2에 도입된 AL2023 최소 컨테이너 이미지는 다른 패키지를 설치하는 데 필요한 최소한의 패키지만 포함한다는 점에서 기본 컨테이너 이미지와 다릅니다. 최소 컨테이너 이미지는 편리한 패키지 집합이 아닌 최소 패키지 집합으로 설계되었습니다.

AL2023 미니멀 컨테이너 이미지는 AL2023에 이미 설치된 소프트웨어 구성 요소를 기반으로 구축되었습니다. 최소 컨테이너 이미지의 주요 차이점은 모든 기능을 갖춘 Python 기반 이미지가 아닌 dnf 패키지 관리자를 제공하는 microdnf 데 사용한다는 dnf 것입니다. 이를 통해 AL2023 AMI와 기본 컨테이너 이미지에 포함된 dnf 패키지 관리자의 전체 기능 세트를 갖추지 못한다는 단점이 있지만 최소 컨테이너 이미지는 더 작아질 수 있습니다.

AL2023 최소 컨테이너 이미지는 `provided.al2023` AWS Lambda 런타임 환경의 기반을 형성합니다.

최소 컨테이너 이미지에 포함된 패키지의 세부 목록은 을 참조하십시오. [Amazon Linux 2023 컨테이너 이미지에 설치된 패키지 비교](#)

미니멀 컨테이너 이미지 사이즈

AL2023 최소 컨테이너 이미지는 AL2023 기본 컨테이너 이미지보다 패키지 수가 적기 때문에 크기가 훨씬 작습니다. 다음 표는 Amazon Linux의 현재 및 이전 릴리스의 컨테이너 이미지 옵션을 비교합니다.

Note

이미지 크기는 [Amazon Linux Amazon ECR 퍼블릭 갤러리](#)에 있는 것과 같습니다.

이미지	버전	이미지 크기	참고
Amazon Linux 1(AL1)	2018.03.0.20230918.0	62.3MB	x86-64 전용
Amazon Linux 2	2.0.20230926.0	64.2MB	aarch64은 x86-64보다 1.6MB 큼니다.
Amazon Linux 기본 컨테이너 이미지	2023.2.20231002.0	52.4MB	

이미지	버전	이미지 크기	참고
Amazon Linux 2023 미니멀 컨테이너 이미 지	2023.2.20231002.0-미 니멀	35.2MB	

AL2023 미니멀 컨테이너 이미지 사용

에서 AL2023 최소 컨테이너 이미지를 사용할 수 ECR 있으며 2023-minimal 태그는 항상 최신 AL2023 기반 최소 컨테이너 이미지를 가리키며, minimal 태그는 AL2023 이후 Amazon Linux의 최신 버전으로 업데이트될 수 있습니다.

다음 예제를 docker 사용하여 이러한 태그를 가져올 수 있습니다.

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:minimal
```

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023-minimal
```

다음 예제는 최소 컨테이너 이미지를 가져와 Dockerfile 그 위에 GCC를 설치하는 a를 보여줍니다.

```
FROM public.ecr.aws/amazonlinux/amazonlinux:2023-minimal
RUN dnf install -y gcc && dnf clean all
```

기본적인 AL2023 컨테이너 이미지 구축

AL2023 컨테이너 이미지는 AL2023 AMI에 포함된 것과 동일한 소프트웨어 구성 요소로 구축됩니다. 여기에는 기본 컨테이너 계층이 패키지 관리자와 같은 Amazon EC2 인스턴스에서 실행되는 것과 유사하게 작동하도록 하는 소프트웨어가 포함됩니다. dnf 이 섹션에서는 애플리케이션에 필요한 최소한의 종속성만 포함하는 컨테이너를 처음부터 구성하는 방법을 설명합니다.

Note

표준 AL2023 컨테이너 이미지는 대부분의 사용 사례에 적합합니다. 표준 컨테이너 이미지를 사용하면 이미지 위에 쉽게 빌드할 수 있습니다. 기본 컨테이너 이미지는 이미지 위에 구축하기가 더 어렵습니다.

애플리케이션에 대한 종속성을 최소화한 컨테이너를 만들려면

1. 런타임 종속 항목을 확인하세요. 이는 애플리케이션에 따라 다릅니다.
2. FROM scratch을 빌드하는 Dockerfile / Containerfile를 구성하세요. 다음 Dockerfile 예제를 사용하여 bash 셸과 그 종속성만 포함하는 컨테이너를 빌드할 수 있습니다.

```
FROM public.ecr.aws/amazonlinux/amazonlinux:2023 as build
RUN mkdir /sysroot
RUN dnf --releasever=$(rpm -q system-release --qf '%{VERSION}') \
  --installroot /sysroot \
  -y \
  --setopt=install_weak_deps=False \
  install bash

FROM scratch
COPY --from=build /sysroot /
WORKDIR /
ENTRYPOINT ["/bin/bash"]
```

- Dockerfile은 다음과 같은 방법으로 실행합니다.
 1. build라는 이름의 AL2023 컨테이너를 실행합니다. 이 컨테이너는 기본 컨테이너를 부트스트랩하는 데 사용되며, 자체 배포되지 않고 배포할 컨테이너를 생성합니다.
 2. /sysroot 디렉터리 생성 이 디렉터리에서 build 컨테이너가 기본 컨테이너에 필요한 종속성을 설치합니다. 다음 단계는 /sysroot 경로가 기본 이미지의 루트 디렉터리가 되도록 패키징됩니다.

이 방식으로 --installroot 옵션을 dnf로 사용하면 다른 AL2023 이미지가 만들어집니다. dnf 기능으로 설치 프로그램과 이미지 생성 도구를 사용할 수 있습니다.

3. dnf를 호출하여 /sysroot에 패키지를 설치합니다.

이 rpm -q system-release --qf '%{VERSION}' 명령은 system-release 패키지를 쿼리(-q)하고 쿼리 형식(--qf)을 설정하여 쿼리되는 패키지 버전(%{VERSION} 변수는 RPM 버전의 rpm 변수)을 출력합니다.

build 컨테이너에서 system-release 버전의 --releasever 인수를 dnf로 설정하면 업데이트된 Amazon Linux 컨테이너 기본 이미지가 릴리스될 때마다 Dockerfile을 사용하여 기본 컨테이너를 다시 빌드할 수 있습니다.

--releasever2023.4.20240513과 같은 모든 아마존 리눅스 2023 버전으로 설정할 수 있습니다. 이렇게 하면 build 컨테이너는 최신 AL2023 버전으로 실행되지만 현재 AL2023 릴리스가 무엇이든 상관없이 2023.4.20240513부터 베어본 컨테이너를 빌드하게 됩니다.

--setopt=install_weak_deps=False 구성 옵션으로 권장하거나 권장되는 종속성이 아닌 필수 종속성만 dnf에 설치할 수 있습니다.

4. 설치된 시스템을 빈 (FROM scratch) 컨테이너의 루트에 복사합니다.
 5. /bin/bash 경우 ENTRYPOINT를 원하는 바이너리로 설정합니다.
3. 빈 디렉터리를 만들고 2단계의 예제 내용을 Dockerfile 파일에 추가합니다.

```
$ mkdir al2023-barebones-bash-example
$ cd al2023-barebones-bash-example
$ cat > Dockerfile <<EOF
FROM public.ecr.aws/amazonlinux/amazonlinux:2023 as build
RUN mkdir /sysroot
RUN dnf --releasever=$(rpm -q system-release --qf '%{VERSION}') \
  --installroot /sysroot \
  -y \
  --setopt=install_weak_deps=False \
  install bash && dnf --installroot /sysroot clean all

FROM scratch
COPY --from=build /sysroot /
WORKDIR /
ENTRYPOINT ["/bin/bash"]
EOF
```

4. 다음 명령을 실행하여 컨테이너를 빌드합니다.

```
$ docker build -t al2023-barebones-bash-example
```

5. 다음 명령으로 컨테이너를 실행하면 bash 전용 컨테이너가 얼마나 작은지 알 수 있습니다.

```
$ docker run -it --rm al2023-barebones-bash-example
bash-5.2# rpm
bash: rpm: command not found
bash-5.2# du -sh /usr/
bash: du: command not found
bash-5.2# ls
```

```
bash: ls: command not found
bash-5.2# echo /bin/*
/bin/alias /bin/bash /bin/bashbug /bin/bashbug-64 /bin/bg /bin/catchsegv /bin/cd /
bin/command /bin/fc /bin/fg /bin/gencat /bin/getconf /bin/getent /bin/getopts /
bin/hash /bin/iconv /bin/jobs /bin/ld.so /bin/ldd /bin/locale /bin/localedef /
bin/pldd /bin/read /bin/sh /bin/sotruss /bin/sprof /bin/type /bin/tzselect /bin/
ulimit /bin/umask /bin/unalias /bin/wait /bin/zdump
```

좀 더 실용적인 예를 들면, 다음 방법으로 Hello World!를 표시하는 C 응용 프로그램 컨테이너를 빌드합니다.

1. 빈 디렉터리를 만들고 C 소스 코드 및 Dockerfile을 추가합니다.

```
$ mkdir al2023-barebones-c-hello-world-example
$ cd al2023-barebones-c-hello-world-example
$ cat > hello-world.c <<EOF
#include <stdio.h>
int main(void)
{
    printf("Hello World!\n");
    return 0;
}
EOF

$ cat > Dockerfile <<EOF
FROM public.ecr.aws/amazonlinux/amazonlinux:2023 as build
COPY hello-world.c /
RUN dnf -y install gcc
RUN gcc -o hello-world hello-world.c
RUN mkdir /sysroot
RUN mv hello-world /sysroot/
RUN dnf --releasever=$(rpm -q system-release --qf '%{VERSION}') \
    --installroot /sysroot \
    -y \
    --setopt=install_weak_deps=False \
    install glibc && dnf --installroot /sysroot clean all

FROM scratch
COPY --from=build /sysroot /
WORKDIR /
ENTRYPOINT ["/hello-world"]
EOF
```

2. 다음 명령을 실행하여 컨테이너를 빌드합니다.

```
$ docker build -t al2023-barebones-c-hello-world-example .
```

3. 다음 명령을 실행하여 컨테이너를 실행합니다.

```
$ docker run -it --rm al2023-barebones-c-hello-world-example
Hello World!
```

Amazon Linux 2023 컨테이너 이미지에 설치된 패키지 비교

AL2023 기본 컨테이너 이미지에 있는 RPM을 AL2023 최소 컨테이너 이미지에 있는 RPM과 비교한 것입니다.

패키지	컨테이너	미니멀 컨테이너
alternatives	1.15	1.15
amazon-linux-repo-cdn	2023.4.20240513	2023.4.20240513
audit-libs	3.0.6	3.0.6
basesystem	11	11
bash	5.2.15	5.2.15
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023.2.64	2023.2.64
coreutils-single	8.32	8.32
crypto-policies	20220428	20220428
curl-minimal	8.5.0	8.5.0

패키지	컨테이너	미니멀 컨테이너
dnf	4.14.0	
dnf-data	4.14.0	4.14.0
elfutils-default-yama-scope	0.188	
elfutils-libelf	0.188	
elfutils-libs	0.188	
expat	2.5.0	
file-libs	5.39	5.39
filesystem	3.14	3.14
gawk	5.1.0	5.1.0
gdbm-libs	1.19	
glib2	2.74.7	2.74.7
glibc	2.34	2.34
glibc-common	2.34	2.34
glibc-minimal-langpack	2.34	2.34
gmp	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7
gobject-introspection		1.73.0
gpgme	1.15.1	1.15.1

패키지	컨테이너	미니멀 컨테이너
grep	3.8	3.8
json-c	0.14	0.14
keyutils-libs	1.6.3	1.6.3
krb5-libs	1.21	1.21
libacl	2.3.1	2.3.1
libarchive	3.5.3	3.5.3
libassuan	2.5.5	2.5.5
libattr	2.5.1	2.5.1
libblkid	2.37.4	2.37.4
libcap	2.48	2.48
libcap-ng	0.8.2	0.8.2
libcom_err	1.46.5	1.46.5
libcomps	0.1.20	
libcurl-minimal	8.5.0	8.5.0
libdnf	0.69.0	0.69.0
libffi	3.4.4	3.4.4
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	
libgpg-error	1.42	1.42

패키지	컨테이너	미니멀 컨테이너
libidn2	2.3.2	2.3.2
libmodulemd	2.13.0	2.13.0
libmount	2.37.4	2.37.4
libnghttp2	1.59.0	1.59.0
libpeas		1.32.0
libpsl	0.21.1	0.21.1
librepo	1.14.5	1.14.5
libreport-filessystem	2.15.2	2.15.2
libselinux	3.4	3.4
libsepol	3.4	3.4
libsigsegv	2.13	2.13
libsmartcols	2.37.4	2.37.4
libsolv	0.7.22	0.7.22
libstdc++	11.4.1	11.4.1
libtasn1	4.19.0	4.19.0
libunistring	0.9.10	0.9.10
libuuid	2.37.4	2.37.4
libverto	0.3.2	0.3.2
libxcrypt	4.4.33	
libxml2	2.10.4	2.10.4

패키지	컨테이너	미니멀 컨테이너
libyam1	0.2.5	0.2.5
libzstd	1.5.5	1.5.5
lua-libs	5.4.4	5.4.4
lz4-libs	1.9.4	1.9.4
microdnf		3.8.1
microdnf-dnf		3.8.1
mpfr	4.1.0	4.1.0
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
npth	1.6	1.6
openssl-libs	3.0.8	3.0.8
p11-kit	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1
pcre2	10.40	10.40
pcre2-syntax	10.40	10.40
popt	1.18	1.18
publicsuffix-list-dafsa	20240212	20240212
python3	3.9.16	
python3-dnf	4.14.0	
python3-gpg	1.15.1	

패키지	컨테이너	미니멀 컨테이너
python3-hawkey	0.69.0	
python3-libcomps	0.1.20	
python3-libdnf	0.69.0	
python3-libs	3.9.16	
python3-pip-wheel	21.3.1	
python3-rpm	4.16.1.3	
python3-setuptools-wheel	59.6.0	
readline	8.1	8.1
rpm	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	
rpm-libs	4.16.1.3	4.16.1.3
rpm-sign-libs	4.16.1.3	
sed	4.8	4.8
setup	2.13.7	2.13.7
sqlite-libs	3.40.0	3.40.0
system-release	2023.4.20240513	2023.4.20240513
tzdata	2024a	
xz-libs	5.2.5	5.2.5
yum	4.14.0	
zlib	1.2.11	1.2.11

Amazon Linux 2023 미니멀 AMI와 컨테이너 이미지에 설치된 패키지 비교

AL2023 최소 AMI에 있는 RPM을 AL2023 기반 및 최소 컨테이너 이미지에 있는 RPM과 비교합니다.

패키지	미니멀 AMI	컨테이너	미니멀 컨테이너
alternatives	1.15	1.15	1.15
amazon-chrony-config	4.3		
amazon-ec2-net-utils	2.4.1		
amazon-linux-repo-cdn		2023.4.20240513	2023.4.20240513
amazon-linux-repo-s3	2023.4.20240513		
amazon-linux-sb-keys	2023.1		
audit	3.0.6		
audit-libs	3.0.6	3.0.6	3.0.6
awscli-2	2.15.30		
basesystem	11	11	11
bash	5.2.15	5.2.15	5.2.15
bzip2-libs	1.0.8	1.0.8	1.0.8
ca-certificates	2023.2.64	2023.2.64	2023.2.64
checkpolicy	3.4		
chrony	4.3		

패키지	미니멀 AMI	컨테이너	미니멀 컨테이너
cloud-init	22.2.2		
cloud-init-cfg-ec2	22.2.2		
cloud-utils-growpart	0.31		
coreutils	8.32		
coreutils-common	8.32		
coreutils-single		8.32	8.32
cpio	2.13		
cracklib	2.9.6		
cracklib-dicts	2.9.6		
crypto-policies	20220428	20220428	20220428
cryptsetup-libs	2.6.1		
curl-minimal	8.5.0	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27		
dbus	1.12.28		
dbus-broker	32		
dbus-common	1.12.28		
dbus-libs	1.12.28		
device-mapper	1.02.185		

패키지	미니멀 AMI	컨테이너	미니멀 컨테이너
device-mapper-libs	1.02.185		
diffutils	3.8		
dnf	4.14.0	4.14.0	
dnf-data	4.14.0	4.14.0	4.14.0
dnf-plugin-release-notification	1.2		
dnf-plugins-core	4.3.0		
dnf-plugin-support-info	1.2		
dracut	055		
dracut-config-ec2	3.0		
dracut-config-generic	055		
e2fsprogs	1.46.5		
e2fsprogs-libs	1.46.5		
ec2-utils	2.2.0		
efi-filesystem	5		
efivar	38		
efivar-libs	38		

패키지	미니멀 AMI	컨테이너	미니멀 컨테이너
elfutils-default-yama-scope	0.188	0.188	
elfutils-libelf	0.188	0.188	
elfutils-libs	0.188	0.188	
expat	2.5.0	2.5.0	
file	5.39		
file-libs	5.39	5.39	5.39
filesystem	3.14	3.14	3.14
findutils	4.8.0		
fuse-libs	2.9.9		
gawk	5.1.0	5.1.0	5.1.0
gdbm-libs	1.19	1.19	
gdisk	1.0.8		
gettext	0.21		
gettext-libs	0.21		
glib2	2.74.7	2.74.7	2.74.7
glibc	2.34	2.34	2.34
glibc-all-langpacks	2.34		
glibc-common	2.34	2.34	2.34

패키지	미니멀 AMI	컨테이너	미니멀 컨테이너
glibc-locale-source	2.34		
glibc-minimal-langpack		2.34	2.34
gmp	6.2.1	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7	2.3.7
gnutls	3.8.0		
gobject-introspection			1.73.0
gpgme	1.15.1	1.15.1	1.15.1
grep	3.8	3.8	3.8
groff-base	1.22.4		
grub2-common	2.06		
grub2-efi-aa64-ec2	2.06 (aarch64)		
grub2-efi-x64-ec2	2.06 (x86_64)		
grub2-pc-modules	2.06		
grub2-tools	2.06		
grub2-tools-minimal	2.06		
grubby	8.40		

패키지	미니멀 AMI	컨테이너	미니멀 컨테이너
gzip	1.12		
hostname	3.23		
hwdata	0.353		
inih	49		
initscripts	10.09		
iproute	5.10.0		
iputils	20210202		
irqbalance	1.9.0		
jansson	2.14		
jitterentropy	3.4.1		
jq	1.7.1		
json-c	0.14	0.14	0.14
kbd	2.4.0		
kbd-misc	2.4.0		
kernel	6.1.90		
kernel-li vepatch-repo- s3	2023.4.20240513		
keyutils-libs	1.6.3	1.6.3	1.6.3
kmod	29		
kmod-libs	29		

패키지	미니멀 AMI	컨테이너	미니멀 컨테이너
krb5-libs	1.21	1.21	1.21
less	608		
libacl	2.3.1	2.3.1	2.3.1
libarchive	3.5.3	3.5.3	3.5.3
libargon2	20171227		
libassuan	2.5.5	2.5.5	2.5.5
libattr	2.5.1	2.5.1	2.5.1
libblkid	2.37.4	2.37.4	2.37.4
libcap	2.48	2.48	2.48
libcap-ng	0.8.2	0.8.2	0.8.2
libcbor	0.7.0		
libcom_err	1.46.5	1.46.5	1.46.5
libcomps	0.1.20	0.1.20	
libcurl-minimal	8.5.0	8.5.0	8.5.0
libdb	5.3.28		
libdnf	0.69.0	0.69.0	0.69.0
libeconf	0.4.0		
libedit	3.1		
libfdisk	2.37.4		
libffi	3.4.4	3.4.4	3.4.4

패키지	미니멀 AMI	컨테이너	미니멀 컨테이너
libfido2	1.10.0		
libgcc	11.4.1	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2	1.10.2
libgomp	11.4.1	11.4.1	
libgpg-error	1.42	1.42	1.42
libidn2	2.3.2	2.3.2	2.3.2
libkcapi	1.4.0		
libkcapi-hmaccalc	1.4.0		
libmnl	1.0.4		
libmodulemd	2.13.0	2.13.0	2.13.0
libmount	2.37.4	2.37.4	2.37.4
libnghttp2	1.59.0	1.59.0	1.59.0
libpeas			1.32.0
libpipeline	1.5.3		
libpsl	0.21.1	0.21.1	0.21.1
libpwquality	1.4.4		
librepo	1.14.5	1.14.5	1.14.5
libreport-filesystem	2.15.2	2.15.2	2.15.2
libseccomp	2.5.3		

패키지	미니멀 AMI	컨테이너	미니멀 컨테이너
libselinux	3.4	3.4	3.4
libselinux- utils	3.4		
libsemanage	3.4		
libsepol	3.4	3.4	3.4
libsigsegv	2.13	2.13	2.13
libsmartcols	2.37.4	2.37.4	2.37.4
libsolv	0.7.22	0.7.22	0.7.22
libss	1.46.5		
libstdc++	11.4.1	11.4.1	11.4.1
libtasn1	4.19.0	4.19.0	4.19.0
libtextstyle	0.21		
libunistring	0.9.10	0.9.10	0.9.10
libuser	0.63		
libutempter	1.2.1		
libuuid	2.37.4	2.37.4	2.37.4
libverto	0.3.2	0.3.2	0.3.2
libxcrypt	4.4.33	4.4.33	
libxml2	2.10.4	2.10.4	2.10.4
libyaml	0.2.5	0.2.5	0.2.5
libzstd	1.5.5	1.5.5	1.5.5

패키지	미니멀 AMI	컨테이너	미니멀 컨테이너
logrotate	3.20.1		
lua-libs	5.4.4	5.4.4	5.4.4
lz4-libs	1.9.4	1.9.4	1.9.4
man-db	2.9.3		
microcode_ctl	2.1 (x86_64)		
microdnf			3.8.1
microdnf-dnf			3.8.1
mpfr	4.1.0	4.1.0	4.1.0
ncurses	6.2		
ncurses-base	6.2	6.2	6.2
ncurses-libs	6.2	6.2	6.2
nettle	3.8		
net-tools	2.0		
npth	1.6	1.6	1.6
numactl-libs	2.0.14		
oniguruma	6.9.7.1		
openldap	2.4.57		
openssh	8.7p1		
openssh-clients	8.7p1		
openssh-server	8.7p1		

패키지	미니멀 AMI	컨테이너	미니멀 컨테이너
openssl	3.0.8		
openssl-lib	3.0.8	3.0.8	3.0.8
openssl-pkcs11	0.4.12		
os-prober	1.77		
p11-kit	0.24.1	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1	0.24.1
pam	1.5.1		
passwd	0.80		
pciutils	3.7.0		
pciutils-lib	3.7.0		
pcre2	10.40	10.40	10.40
pcre2-syntax	10.40	10.40	10.40
policycoreutils	3.4		
popt	1.18	1.18	1.18
procps-ng	3.3.17		
psmisc	23.4		
publicsuffix-list-dafsa	20240212	20240212	20240212
python3	3.9.16	3.9.16	
python3-attrs	20.3.0		
python3-audit	3.0.6		

패키지	미니멀 AMI	컨테이너	미니멀 컨테이너
python3-awscrt	0.19.19		
python3-babel	2.9.1		
python3-cffi	1.14.5		
python3-chardet	4.0.0		
python3-colorama	0.4.4		
python3-configobj	5.0.6		
python3-cryptography	36.0.1		
python3-dateutil	2.8.1		
python3-dbus	1.2.18		
python3-distro	1.5.0		
python3-dnf	4.14.0	4.14.0	
python3-dnf-plugins-core	4.3.0		
python3-docutils	0.16		
python3-gpg	1.15.1	1.15.1	
python3-hawkey	0.69.0	0.69.0	
python3-idna	2.10		
python3-jinja2	2.11.3		

패키지	미니멀 AMI	컨테이너	미니멀 컨테이너
python3-j mespath	0.10.0		
python3-j sonpatch	1.21		
python3-j sonpointer	2.0		
python3-j sonschema	3.2.0		
python3-l ibcomps	0.1.20	0.1.20	
python3-libdnf	0.69.0	0.69.0	
python3-libs	3.9.16	3.9.16	
python3-l ibselinux	3.4		
python3-l ibsemanage	3.4		
python3-m arkupsafe	1.1.1		
python3-n etifaces	0.10.6		
python3-o authlib	3.0.2		
python3-pip- wheel	21.3.1	21.3.1	
python3-ply	3.11		

패키지	미니멀 AMI	컨테이너	미니멀 컨테이너
python3-policoreutils	3.4		
python3-prettytable	0.7.2		
python3-prompt-toolkit	3.0.24		
python3-pyparser	2.20		
python3-pyrsistent	0.17.3		
python3-pyserial	3.4		
python3-pysocks	1.7.1		
python3-pytz	2022.7.1		
python3-pyyaml	5.4.1		
python3-requests	2.25.1		
python3-rpm	4.16.1.3	4.16.1.3	
python3-ruamel-yaml	0.16.6		
python3-ruamel-yaml-clib	0.1.2		
python3-setools	4.4.1		

패키지	미니멀 AMI	컨테이너	미니멀 컨테이너
python3-s etuptools	59.6.0		
python3-s etuptools- wheel	59.6.0	59.6.0	
python3-six	1.15.0		
python3-systemd	235		
python3-urllib3	1.25.10		
python3-wcwidth	0.2.5		
readline	8.1	8.1	8.1
rng-tools	6.14		
rootfiles	8.1		
rpm	4.16.1.3	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3	
rpm-libs	4.16.1.3	4.16.1.3	4.16.1.3
rpm-plugin- selinux	4.16.1.3		
rpm-plugin- systemd-inhibit	4.16.1.3		
rpm-sign-libs	4.16.1.3	4.16.1.3	
sbsigntools	0.9.4		
sed	4.8	4.8	4.8

패키지	미니멀 AMI	컨테이너	미니멀 컨테이너
selinux-policy	37.22		
selinux-policy-targeted	37.22		
setup	2.13.7	2.13.7	2.13.7
shadow-utils	4.9		
sqlite-libs	3.40.0	3.40.0	3.40.0
sudo	1.9.15		
sysctl-defaults	1.0		
systemd	252.16		
systemd-libs	252.16		
systemd-networkd	252.16		
systemd-pam	252.16		
systemd-resolved	252.16		
systemd-udev	252.16		
system-release	2023.4.20240513	2023.4.20240513	2023.4.20240513
tar	1.34		
tzdata	2024a	2024a	
update-motd	2.2		
userspace-rcu	0.12.1		

패키지	미니멀 AMI	컨테이너	미니멀 컨테이너
util-linux	2.37.4		
util-linux-core	2.37.4		
vim-data	9.0.2153		
vim-minimal	9.0.2153		
which	2.21		
xfspg	5.18.0		
xz	5.2.5		
xz-libs	5.2.5	5.2.5	5.2.5
yum	4.14.0	4.14.0	
zlib	1.2.11	1.2.11	1.2.11
zram-generator	1.1.2		
zram-generator-defaults	1.1.2		
zstd	1.5.5		

AL2023 크기 AWS Elastic Beanstalk

AWS Elastic Beanstalk 웹 애플리케이션 및 서비스를 배포하고 확장하기 위한 서비스입니다. 코드를 업로드하면 Elastic Beanstalk가 용량 프로비저닝, 로드 밸런싱, Auto Scaling부터 애플리케이션 상태 모니터링에 이르기까지 자동으로 배포합니다. 자세한 내용은 [AWS Elastic Beanstalk](#)를 참조하세요.

Elastic Beanstalk를 사용하려면 애플리케이션을 생성하고, 애플리케이션 소스 번들 포맷(예: Java .war 파일)으로 애플리케이션 버전을 업로드한 다음, 애플리케이션에 대한 몇 가지 정보를 입력합니다. Elastic Beanstalk는 환경을 자동으로 시작하고 코드 실행에 필요한 리소스를 생성 및 AWS 구성합니다. 자세한 내용은 [AWS Elastic Beanstalk 개발자 안내서](#)를 참조하세요.

Amazon EC2 인스턴스에서 Elastic Beanstalk Linux 플랫폼이 실행되며 이 인스턴스는 Amazon Linux에서 실행됩니다. 2023년 8월 4일 현재, Elastic Beanstalk는 Amazon Linux 2023 기반에서 Docker, Tomcat, Java SE, Node.js, PHP 및 Python 등의 플랫폼 브랜치를 지원합니다. Elastic Beanstalk는 앞으로 더 많은 Elastic Beanstalk 플랫폼에 AL2023을 지원하기 위해 노력하고 있습니다.

Elastic Beanstalk 플랫폼 지원 및 AL2023에 구축된 현재 플랫폼의 전체 목록은 [Elastic Beanstalk 개발자 가이드](#)의 [Elastic Beanstalk 리눅스 플랫폼](#) 섹션에서 확인할 수 있습니다.

최신 Elastic Beanstalk 플랫폼 및 기존 플랫폼 버전에 대한 릴리스 정보는 [Elastic Beanstalk 릴리스 정보](#)에 있습니다.

에서 AL2023 사용하기 AWS CloudShell

AWS CloudShell 에서 직접 실행할 수 있는 브라우저 기반의 사전 인증된 셸입니다. AWS Management Console 몇 가지 방법으로 탐색할 CloudShell 수 AWS Management Console 있습니다. 자세한 내용은 [시작하는 방법 AWS CloudShell?](#) 을 참조하십시오.

AWS CloudShell는 현재 아마존 리눅스 2를 기반으로 하고 있으며 AL2023 버전으로 마이그레이션될 예정입니다. AL2023 로의 마이그레이션은 2023년 12월 AWS 리전 4일부터 전면 시행될 예정입니다. AL2023 CloudShell 마이그레이션에 대한 자세한 내용은 아마존 [리눅스 2에서 아마존 리눅스 2023으로 AWS CloudShell 마이그레이션](#)을 참조하십시오.

AL2023 기반 Amazon ECS AMI를 사용하여 컨테이너식 워크로드를 호스팅합니다.

Note

컨테이너 내에서 AL2023 사용 방법에 대한 자세한 내용은 을 참조하십시오. [AL2023 컨테이너](#)

Amazon Elastic Container Service(Amazon ECS)는 컨테이너 애플리케이션을 쉽게 배포, 관리 및 확장할 수 있도록 도와주는 완전 관리형 컨테이너 오케스트레이션 서비스입니다. 완전관리형 서비스인 Amazon ECS에는 AWS 구성 및 운영 모범 사례가 내장되어 있습니다. Amazon Elastic Container Registry (Amazon ECR) AWS 및 Docker와 같은 타사 도구 모두와 통합됩니다. 이러한 통합을 통해 환경이 아닌 애플리케이션 구축에 더욱 집중할 수 있습니다. 컨트롤 플레인을 관리하는 복잡한 과정 없이 클라우드의 AWS 리전에서 컨테이너 워크로드를 실행하고 확장할 수 있습니다.

AL2023 기반 Amazon ECS에 최적화된 AMI를 사용하여 AL2023 상에서 컨테이너식 워크로드를 호스팅할 수 있습니다. 자세한 내용은 [Amazon ECS에 최적화된 AMI](#)를 참조하십시오.

AL2와 비교한 Amazon ECS용 AL2023 변경 사항

AL2와 마찬가지로 AL2023 역시 Amazon ECS Linux 인스턴스로 실행하는 데 필요한 기본 패키지를 제공합니다. AL2에서는 containerd, 및 ecs-init 패키지를 통해 amazon-linux-extras 사용할 수 있었지만 AL2023에서는 이러한 패키지를 코어 리포지토리에 포함합니다.

버전이 지정된 리포지토리를 통한 결정적 업그레이드 기능을 사용하면 모든 AL2023 AMI는 기본적으로 특정 리포지토리 버전에 고정됩니다. 이는 AL2023 Amazon ECS에 최적화된 AMI의 경우에도 마찬가지입니다. 환경에 대한 모든 업데이트는 배포 전에 신중하게 관리하고 테스트할 수 있으며, 문제 발생 시 이전 AMI의 콘텐츠로 쉽게 되돌릴 수 있는 방법을 제공합니다. 이 AL2023 기능에 대한 자세한 내용은 [AL2023 버전 지정 저장소를 통한 결정적 업그레이드 사용](#)를 참조하세요.

AL2023은 AL2에서 지원되는 cgroup v1 인터페이스를 통해 cgroup v2로 전환됩니다. 자세한 정보는 [통합 제어 그룹 계층 구조 \(cgroup v2\)](#)을 참조하세요.

Note

[2023.2.20230920 이전의 AL2023 버전 \(최초의 AL2023.2 릴리스\)](#)에는 cgroup 내부의 메모리 부족 (OOM) 처리에 대한 버그가 포함되어 있습니다. systemd OOM 킬러가 한 번에 하나의 프로세스를 선택하는 대신 cgroup의 모든 프로세스가 항상 종료되었는데, 이는 의도된 동작입니다.

이는 AL2 동작과 비교했을 때 회귀 현상이었으며, AL2023 2023.2.20230920 릴리스를 기준으로 수정되었습니다.

[Amazon ECS에 최적화된 AMI를 구축하는 코드는 프로젝트에서 사용할 수 있습니다.](#) [amazon-ecs-ami GitHub 릴리스 노트](#)에는 어떤 AL2023 버전이 어떤 Amazon ECS AMI 버전에 매핑되는지가 설명되어 있습니다.

AL2023 Amazon ECS에 최적화된 AMI 사용자 지정

Important

Amazon ECS에 최적화된 AL2023 AMI를 사용하는 것이 좋습니다. 자세한 내용은 [Amazon Elastic 컨테이너 서비스 개발자 안내서의 Amazon ECS에 최적화된 AMI](#)를 참조하십시오.

Amazon ECS에서 사용자 지정 AMI를 생성할 때 사용하는 것과 동일한 빌드 스크립트를 사용할 수 있습니다. 자세한 내용은 [Amazon ECS에 최적화된 Linux AMI 빌드](#) 스크립트를 참조하십시오.

AL2023 기반 아마존 Elastic File System 사용

Amazon Elastic File System(Amazon EFS)은 완전히 탄력적인 서버리스 파일 스토리지를 제공하므로 스토리지 용량과 성능을 프로비저닝하거나 관리하지 않고도 파일 데이터를 공유할 수 있습니다. Amazon EFS는 애플리케이션을 중단하지 않고 페타바이트급까지 온디맨드 규모로 확장할 수 있도록 구축되었으며 파일을 추가하고 제거할 때 확장 및 축소됩니다. Amazon EFS에 간단한 웹 서비스 인터페이스가 있으므로 파일 시스템을 빠르고 쉽게 생성하고 구성할 수 있습니다. 모든 파일 스토리지 인프라를 관리하는 서비스이므로 사용자는 복잡한 파일 시스템 구성을 배포, 패치 및 유지 보수하는 번잡함에서 벗어날 수 있습니다.

Amazon EFS는 네트워크 파일 시스템 버전 4(NFSv4.1 및 NFSv4.0) 프로토콜을 지원하므로 현재 사용하는 애플리케이션 및 도구도 Amazon EFS에서 원활하게 작동합니다. Amazon EC2, Amazon ECS 등을 비롯한 여러 컴퓨팅 인스턴스가 동시에 Amazon EFS 파일 시스템에 액세스할 수 있습니다. AWS Lambda따라서, EFS 파일 시스템은 다수의 인스턴스 또는 서버에서 실행되는 워크로드와 애플리케이션의 공통 데이터 소스를 제공합니다.

AL2023에 `amazon-efs-utils` 설치

이 `amazon-efs-utils` 패키지는 AL2023 리포지토리에서 제공되며, 설치하여 Amazon EFS 파일 시스템에 액세스하는 데 사용할 수 있습니다.

AL2023에 `amazon-efs-utils` 패키지를 설치하세요.

- 다음 `amazon-efs-utils` 명령을 사용하여 설치합니다.

```
$ dnf -y install amazon-efs-utils
```

AL2023에 Amazon EFS 파일 시스템 마운트

를 설치한 후 `amazon-efs-utils` AL2023 인스턴스에 Amazon EFS 파일 시스템을 마운트할 수 있습니다.

AL2023에 Amazon EFS 파일 시스템 마운트

- 파일 시스템 ID를 사용하여 마운트하려면 다음 명령을 사용합니다.

```
sudo mount -t efs file-system-id efs-mount-point/
```

전송 중인 데이터가 TLS를 사용하거나 파일 시스템 ID 대신 DNS 이름 또는 마운트 대상 IP를 사용하여 암호화되도록 파일 시스템을 마운트할 수도 있습니다. 자세한 내용은 [EFS 마운트 도우미로 Amazon Linux 인스턴스에 마운트](#)를 참조하세요.

AL2023 기반으로 구축된 Amazon EMR 사용

Amazon EMR은 Apache Hadoop 및 AWS가 제공하는 서비스를 사용하여 많은 데이터를 쉽고 효율적으로 처리할 수 있도록 지원하는 웹 서비스입니다.

AL2023 기반 아마존 EMR 릴리스

Amazon EMR 릴리스 7.0.0은 AL2023 기반으로 구축된 첫 번째 릴리스입니다. 이번 릴리스부터 AL2023은 Amazon EMR의 기본 운영 체제가 되어 AL2023의 모든 이점을 Amazon EMR에 제공합니다. 자세한 내용은 [Amazon EMR 7.0.0](#) 릴리스 노트를 참조하십시오.

EKS 기반 Amazon EMR에 AL2023 지원

EKS 6.13 기반 Amazon EMR은 AL2023을 옵션으로 도입한 첫 번째 릴리스입니다. 이번 릴리스에서는 Java 17 런타임과 함께 AL2023 운영 체제로 Spark를 실행할 수 있습니다. 자세한 내용은 [EKS 6.13용 아마존 EMR 출시 노트 및 EKS 기반 아마존 EMR의 모든 릴리스 노트](#)를 참조하십시오.

에서 AL2023 사용하기 AWS Lambda

를 사용하면 AWS Lambda서버를 프로비저닝하거나 관리하지 않고도 코드를 실행할 수 있습니다. 사용한 컴퓨팅 시간에 대해서만 비용을 지불하고 코드가 실행되지 않을 때는 요금이 부과되지 않습니다. 관리 부담 없이 거의 모든 유형의 애플리케이션 또는 백엔드 서비스를 위한 코드를 실행할 수 있습니다. 코드를 업로드하기만 하면고가용성을 유지한 채로 코드를 실행하고 확장하는 데 필요한 모든 것을 Lambda가 알아서 처리해 줍니다.

AL2023 **provided.al2023** 관리형 런타임 및 컨테이너 이미지

provided.al2023기본 런타임은 [AL2023 최소 컨테이너 이미지를](#) 기반으로 하며 AL2023 기반 Lambda 관리형 런타임 및 컨테이너 기본 이미지를 제공합니다. provided.al2023런타임은 AL2023

최소 컨테이너 이미지를 기반으로 하기 때문에 런타임은 약 109MB의 `provided.a12` 런타임보다 40MB 미만으로 상당히 작습니다.

자세한 내용은 [Lambda 런타임 및 Lambda 컨테이너 이미지 작업을](#) 참조하십시오.

AL2023 기반 람다 런타임

Node.js20, Python 3.12, Java 21, .NET 8과 같은 관리형 언어 런타임의 향후 릴리스는 AL2023 기반이며 [AL2023 기반 런타임 발표에](#) 설명된 대로 기본 이미지로 사용될 예정입니다. `provided.a12023`

AL2023 기반 람다 함수

- [다음과 같이 작성된 AL2023 람다 함수 Go](#)
- [다음과 같이 작성된 AL2023 람다 함수 Rust](#)

자세한 내용은 [개발자 가이드](#)의 AWS Lambda Lambda 런타임을 참조하세요.

튜토리얼

다음 자습서는 Amazon Linux 2023 (AL2023) 를 실행하는 Amazon EC2 인스턴스를 사용하여 일반적인 작업을 수행하는 방법을 보여줍니다. 동영상 자습서는 [AWS 교육용](#) 동영상 및 실습을 참조하십시오.

AL2 지침은 Amazon EC2 사용 설명서에서 Linux를 [실행하는 Amazon EC2 인스턴스에 대한 자습서를](#) 참조하십시오.

튜토리얼

- [튜토리얼: AL2023 기반 램프 서버 설치](#)
- [튜토리얼: AL2023 기반 SSL/TLS 구성](#)
- [튜토리얼: AL2023 WordPress 블로그 호스팅](#)

튜토리얼: AL2023 기반 램프 서버 설치

다음 절차는 PHP 및 [MariaDB](#) (커뮤니티에서 개발한 MySQL 포크) 지원이 포함된 Apache 웹 서버를 AL2023 인스턴스 (LAMP 웹 서버 또는 LAMP 스택이라고도 함) 에 설치하는 데 도움이 됩니다. 이 서버를 사용해서 고정 웹사이트를 호스팅하거나 데이터베이스에서 정보를 읽고 쓰는 동적 PHP 애플리케이션을 배포할 수 있습니다.

Important

이 절차는 AL2023 사용을 위한 것입니다. Ubuntu 또는 Red Hat Enterprise, Linux와 같은 다른 배포에서 LAMP 웹 서버를 설정하려는 경우 이 자습서를 이용할 수 없습니다. [우분투의 경우 다음 우분투 커뮤니티 설명서인 SQLPHP를 참조하십시오.](#) [ApacheMy](#) 다른 배포는 관련 설명서를 참조하세요.

Tasks

- [1단계: LAMP 서버 준비](#)
- [2단계: LAMP 서버 테스트](#)
- [3단계: 데이터베이스 서버 보안 설정](#)
- [4단계: \(선택 사항\) 설치 phpMyAdmin](#)
- [문제 해결](#)

- [관련 주제](#)

1단계: LAMP 서버 준비

필수 조건

- 이 자습서에서는 인터넷에서 연결할 수 있는 퍼블릭 DNS 이름을 가진 AL2023 기반의 새 인스턴스를 이미 시작했다고 가정합니다. 자세한 정보는 [Amazon EC2 AL2023](#)을 참조하세요. SSH(포트 22), HTTP(포트 80), HTTPS(포트 443) 연결을 허용할 수 있도록 보안 그룹을 구성해야 합니다. 이러한 사전 요구 사항에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [Linux 인스턴스용 인바운드 트래픽 승인을](#) 참조하십시오.
- 다음 절차는 AL2023 (현재 8.1) 에서 사용할 수 있는 최신 PHP 버전을 설치합니다. 이 자습서에서 명시한 애플리케이션이 아닌 PHP 애플리케이션을 사용하려는 경우 8.1과의 호환성을 확인해야 합니다.

LAMP 서버를 준비하려면

1. 인스턴스에 연결합니다. 자세한 정보는 [AL2023 인스턴스에 연결](#)을 참조하세요.
2. 모든 소프트웨어 패키지가 최신 상태로 업데이트되어 있는지 확인하기 위해, 인스턴스에서 쿼크 소프트웨어 업데이트를 실행합니다. 이 과정은 몇 분 정도 시간이 소요될 수 있지만, 최신 보안 업데이트와 버그 수정이 있는지 확인하는 것이 중요합니다.

-y 옵션을 사용하면 확인 여부를 묻지 않고 업데이트를 설치합니다. 설치 전에 업데이트 정보를 확인하려면 이 옵션을 생략합니다.

```
[ec2-user ~]$ sudo dnf update -y
```

3. 최신 버전의 아파치 웹 서버와 AL2023 용 PHP 패키지를 설치합니다.

```
[ec2-user ~]$ sudo dnf install -y httpd wget php-fpm php-mysql php-json php php-devel
```

4. 소프트웨어 패키지 MariaDB를 설치합니다. dnf install 명령을 사용하여 여러 소프트웨어 패키지와 모든 관련 종속 프로그램을 동시에 설치합니다.

```
[ec2-user ~]$ sudo dnf install mariadb105-server
```

다음 명령을 사용하여 이러한 패키지의 현재 버전을 볼 수 있습니다.

```
[ec2-user ~]$ sudo dnf info package_name
```

예제

```
[root@ip-172-31-25-170 ec2-user]# dnf info mariadb105
Last metadata expiration check: 0:00:16 ago on Tue Feb 14 21:35:13 2023.
Installed Packages
Name           : mariadb105
Epoch         : 3
Version        : 10.5.16
Release        : 1.amzn2023.0.6
Architecture   : x86_64
Size           : 18 M
Source         : mariadb105-10.5.16-1.amzn2023.0.6.src.rpm
Repository     : @System
From repo      : amazonlinux
Summary        : A very fast and robust SQL database server
URL            : http://mariadb.org
License        : GPLv2 and LGPLv2
Description    : MariaDB is a community developed fork from MySQL - a multi-user,
multi-threaded
                : SQL database server. It is a client/server implementation consisting
of
                : a server daemon (mariadb) and many different client programs and
libraries.
                : The base package contains the standard MariaDB/MySQL client programs
and
                : utilities.
```

5. Apache 웹 서버를 시작합니다.

```
[ec2-user ~]$ sudo systemctl start httpd
```

6. systemctl 명령을 사용하여 Apache 웹 서버가 매번 시스템이 부팅할 때마다 시작되도록 합니다.

```
[ec2-user ~]$ sudo systemctl enable httpd
```

다음 명령을 실행하여 httpd가 실행되고 있는지 확인할 수 있습니다.

```
[ec2-user ~]$ sudo systemctl is-enabled httpd
```

7. 인스턴스에 대해 인바운드 HTTP(포트 80) 연결을 허용하는 보안 규칙이 없는 경우 추가합니다. 기본적으로 시작하는 동안 인스턴스에 대해 `launch-wizard-N` 보안 그룹이 생성되었습니다. 보안 그룹 규칙을 추가하지 않은 경우 이 그룹에는 SSH 연결을 허용하는 단일 규칙만 포함됩니다.
 - a. <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
 - b. 탐색 창에서 인스턴스(Instances)를 선택하고 인스턴스를 선택합니다.
 - c. 보안 탭에서 인바운드 규칙을 확인합니다. 다음과 같은 규칙이 표시되어야 합니다.

Port range	Protocol	Source
22	tcp	0.0.0.0/0

⚠ Warning

0.0.0.0/0을 사용하면 모든 IP 주소에서 SSH를 사용하여 인스턴스에 액세스할 수 있습니다. 테스트 환경에서 잠시 사용하는 것은 괜찮지만 프로덕션 환경에서는 안전하지 않습니다. 프로덕션에서는 특정 IP 주소나 주소 범위만 인스턴스에 액세스하도록 허용하세요.

- d. HTTP(포트 80) 연결을 허용하는 인바운드 규칙이 없는 경우 지금 규칙을 추가해야 합니다. 보안 그룹에 대한 링크를 선택합니다. [Linux 인스턴스의 인바운드 트래픽 권한 부여의](#) 절차를 사용하여 다음 값이 포함된 새 인바운드 보안 규칙을 추가합니다.
 - 유형: HTTP
 - 프로토콜: TCP
 - 포트 범위: 80
 - 소스: 사용자 지정

8. 웹 서버를 테스트합니다. 웹 브라우저에서 인스턴스의 공용 DNS 주소(또는 공용 IP 주소)를 입력합니다. `/var/www/html`에 콘텐츠가 없으면 "It works!"라는 메시지를 보여 주는 Apache 테스트 페이지가 표시됩니다.

Amazon EC2 콘솔을 사용하여 인스턴스에 대한 퍼블릭 DNS를 가져올 수 있습니다(퍼블릭IPv4 DNS 열 확인, 이 열이 숨겨진 경우 기본 설정(기어 모양 아이콘)을 선택하고 퍼블릭 IPv4 DNS를 켭니다).

인스턴스의 보안 그룹에 포트 80에서 HTTP 트래픽을 허용하는 규칙이 포함되어 있는지 확인합니다. 자세한 내용은 보안 [그룹에 규칙 추가](#)를 참조하십시오.

⚠ Important

Amazon Linux를 사용하지 않는 경우, 이러한 연결을 허용하도록 인스턴스에서 방화벽을 구성해야 할 수도 있습니다. 방화벽 구성 방법에 대한 자세한 내용은 사용자의 특정 배포에 대한 문서를 참조하세요.

Apache httpd는 Apache document root라는 디렉터리에 보관된 파일을 처리합니다. Amazon Linux Apache 문서 루트는 `/var/www/html`이며, 기본적으로 루트에서 소유합니다.

ec2-user 계정에서 이 디렉터리의 파일을 조작할 수 있게 하려면 디렉터리의 소유권과 권한을 변경해야 합니다. 이 작업을 수행하는 방법에는 여러 가지가 있습니다. 본 자습서에서는 ec2-user를 apache 그룹에 추가하여 apache 그룹에 `/var/www` 디렉터리의 소유권을 부여하고 쓰기 권한을 할당합니다.

파일 권한 설정

1. 사용자(이 경우는 ec2-user)를 apache 그룹에 추가합니다.

```
[ec2-user ~]$ sudo usermod -a -G apache ec2-user
```

2. 로그아웃하고 다시 로그인한 다음, 새 그룹을 선택하고 멤버십을 확인합니다.
 - a. 로그아웃합니다(exit 명령을 사용하거나 터미널 창 닫기).

```
[ec2-user ~]$ exit
```

- b. apache 그룹의 멤버십을 확인하려면 인스턴스에 다시 연결한 후 다음 명령을 실행합니다.

```
[ec2-user ~]$ groups
ec2-user adm wheel apache systemd-journal
```

3. `/var/www` 및 그 콘텐츠의 그룹 소유권을 apache 그룹으로 변경합니다.

```
[ec2-user ~]$ sudo chown -R ec2-user:apache /var/www
```

4. 그룹 쓰기 권한을 추가하여 나중에 하위 디렉터리에 대한 그룹 ID를 설정하려면 `/var/www`와 그 하위 디렉터리의 디렉터리 권한을 변경합니다.

```
[ec2-user ~]$ sudo chmod 2775 /var/www && find /var/www -type d -exec sudo chmod 2775 {} \;
```

5. 그룹 쓰기 권한을 추가하려면 /var/www 및 그 하위 디렉터리의 파일 권한을 반복하여 변경합니다.

```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;
```

이제 ec2-user와 apache 그룹의 향후 멤버는 Apache document root에서 파일 추가, 삭제, 편집을 할 수 있고, 이를 통해 사용자는 정적 웹 사이트 또는 PHP 애플리케이션과 같은 콘텐츠를 추가할 수 있습니다.

웹 서버를 보호하려면(선택 사항)

HTTP 프로토콜을 실행하는 웹 서버는 송신하거나 수신하는 데이터에 대해 아무런 전송 보안 기능도 제공하지 않습니다. 웹 브라우저를 사용하여 HTTP 서버에 연결할 때 방문하는 URL, 수신하는 웹 페이지의 내용, 제출하는 HTML 양식의 내용(암호 포함)이 모두 네트워크 경로를 따라 어디서든 엿보려는 사람들에게 보입니다. 웹 서버를 안전하게 보호하기 위한 최선의 방법은 SSL/TLS 암호화로 데이터를 보호하는 HTTPS(HTTP Secure) 지원 기능을 설치하는 것입니다.

서버에서 HTTPS를 활성화하는 방법에 대한 자세한 내용은 [튜토리얼: AL2023 기반 SSL/TLS 구성](#) 섹션을 참조하세요.

2단계: LAMP 서버 테스트

서버가 설치되어 실행 중이고 파일 권한이 올바르게 설정되었다면 사용자의 ec2-user 계정을 통해 인터넷에서 사용 가능한 /var/www/html 디렉터리에서 PHP 파일을 생성할 수 있어야 합니다.

LAMP 서버를 테스트하려면

1. Apache 문서 루트에서 PHP 파일을 생성합니다.

```
[ec2-user ~]$ echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

이 명령을 실행하는 동안 "Permission denied" 오류가 발생하면, 로그아웃하고 다시 로그인한 다음, [파일 권한 설정](#)에서 구성한 적절한 그룹 권한을 선택합니다.

2. 웹 브라우저에서는 방금 생성한 파일의 URL을 입력합니다. 이 URL은 인스턴스의 퍼블릭 DNS 주소에 슬래시(/)와 파일 이름이 추가된 형태입니다. 다음 예를 참조하세요.

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

PHP 정보 페이지가 표시되어야 합니다:

PHP Version 8.1.7

System	Linux ip-172-31-16-77.ec2.internal 5.15.57-28.127.amzn2022.aarch64 #1 SMP Thu Aug 4 17:06:57 UTC 2022 aarch64
Build Date	Jun 7 2022 18:21:38
Build System	Linux
Build Provider	Amazon Linux
Compiler	gcc (GCC) 11.3.1 20220421 (Red Hat 11.3.1-2)
Architecture	aarch64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/10-opcache.ini, /etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gd.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-mbstring.ini, /etc/php.d/20-mysqli.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-simplexml.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/20-xml.ini, /etc/php.d/20-xmlwriter.ini, /etc/php.d/20-xsl.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini, /etc/php.d/30-xmlreader.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902,NTS
PHP Extension Build	API20210902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, bzip2.*, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
 Zend Engine v4.1.7, Copyright (c) Zend Technologies
 with Zend OPcache v8.1.7, Copyright (c), by Zend Technologies

이 페이지가 보이지 않을 경우 이전 단계에서 /var/www/html/phpinfo.php 파일이 제대로 생성되었는지 확인하세요. 또한 다음 명령을 사용하여 필수 패키지가 모두 설치되었는지도 확인할 수 있습니다.

```
[ec2-user ~]$ sudo dnf list installed httpd mariadb-server php-mysqli
```

출력에서 필요한 패키지가 하나라도 나열되지 않으면, `sudo yum install package` 명령을 사용하여 패키지를 설치합니다.

3. `phpinfo.php` 파일을 삭제합니다. 이 파일은 유용한 정보를 포함하고 있지만 보안상 이유로 인터넷에 공개되어서는 안 됩니다.

```
[ec2-user ~]$ rm /var/www/html/phpinfo.php
```

이제 LAMP 웹 서버가 완전히 동작하는 상태가 됩니다. `/var/www/html`의 Apache document root에 콘텐츠를 추가하면 인스턴스에 대한 퍼블릭 DNS 주소에서 그 콘텐츠를 볼 수 있습니다.

3단계: 데이터베이스 서버 보안 설정

MariaDB 서버의 기본 설치에는 테스트 및 개발 기능에 유용한 여러 기능을 포함하고 있지만, 이 기능들은 프로덕션 서버에서는 비활성화되거나 제거되어야 합니다. `mysql_secure_installation` 명령을 통해 루트 암호를 설정하고 설치 패키지에서 보안성이 낮은 기능을 제거하는 과정을 수행할 수 있습니다. MariaDB 서버를 사용할 계획이 없더라도 이 절차를 수행하는 것이 좋습니다.

MariaDB 서버의 보안을 유지하려면

1. MariaDB 서버를 시작합니다.

```
[ec2-user ~]$ sudo systemctl start mariadb
```

2. `mysql_secure_installation`를 실행합니다.

```
[ec2-user ~]$ sudo mysql_secure_installation
```

- a. 암호를 입력하라는 메시지가 표시되면 루트 계정의 암호를 입력합니다.
 - i. 현재 루트 암호를 입력합니다. 기본적으로 root 계정에는 암호가 없습니다. Enter를 누릅니다.
 - ii. 암호를 설정하려면 **Y**를 누른 후 안전한 암호를 두 번 입력합니다. 보안 암호 생성에 대한 자세한 내용은 <https://identitysafe.norton.com/password-generator/> 섹션을 참조하세요. 이 암호를 안전한 장소에 보관하시기 바랍니다.

MariaDB에 대한 루트 암호를 설정하는 것은 데이터베이스를 보호하는 가장 기초적인 방법일 뿐입니다. 데이터베이스 기반 애플리케이션을 빌드하거나 설치할 때, 일반적으로

그 애플리케이션의 데이터베이스 서비스 사용자를 만들고 데이터베이스 관리 이외의 어떤 목적으로도 루트 계정을 사용하지 못하게 합니다.

- b. **Y**를 눌러서 익명 사용자 계정을 제거합니다.
 - c. **Y**를 입력하여 원격 루트 로그인을 비활성화합니다.
 - d. **Y**를 눌러서 테스트 데이터베이스를 제거합니다.
 - e. **Y**를 눌러서 권한 테이블을 다시 로드하고 변경사항을 저장합니다.
3. (선택 사항) 지금 바로 사용할 계획이 아니라면 MariaDB 서버를 중지합니다. 필요할 때 다시 시작할 수 있습니다.

```
[ec2-user ~]$ sudo systemctl stop mariadb
```

4. (선택 사항) 부팅 시 MariaDB 서버가 시작되도록 하려면 다음 명령을 입력합니다.

```
[ec2-user ~]$ sudo systemctl enable mariadb
```

4단계: (선택 사항) 설치 phpMyAdmin

[phpMyAdmin](#) EC2 인스턴스의 MySQL 데이터베이스를 보고 편집하는 데 사용할 수 있는 웹 기반 데이터베이스 관리 도구입니다. Amazon Linux 인스턴스에서 phpMyAdmin을 설치 및 구성하려면 다음 단계를 따르세요.

Important

Apache에서 SSL/TLS를 활성화하지 않은 경우 phpMyAdmin을 사용하여 LAMP 서버에 액세스하지 않는 것이 좋습니다. 이 상태에서 액세스하면 데이터베이스 관리자 암호와 기타 데이터가 인터넷을 통해 안전하지 못한 상태로 전송됩니다. 개발자의 보안 권장 사항은 설치 [보안을](#) 참조하십시오. phpMyAdmin EC2 인스턴스에서의 웹 서버 보안에 대한 일반적인 정보는 [튜토리얼: AL2023 기반 SSL/TLS 구성](#) 섹션을 참조하세요.

설치하려면 phpMyAdmin

1. 필요한 종속 항목을 설치합니다.

```
[ec2-user ~]$ sudo dnf install php-mbstring php-xml -y
```

2. Apache를 다시 시작합니다.

```
[ec2-user ~]$ sudo systemctl restart httpd
```

3. php-fpm을 다시 시작합니다.

```
[ec2-user ~]$ sudo systemctl restart php-fpm
```

4. Apache 문서 루트(/var/www/html)로 이동합니다.

```
[ec2-user ~]$ cd /var/www/html
```

5. <https://www.phpmyadmin.net/downloads> 에서 최신 phpMyAdmin 릴리스의 소스 패키지를 선택합니다. 인스턴스로 파일을 직접 다운로드하려면 다음 예제와 같이 링크를 복사한 후 wget 명령에 붙여 넣습니다.

```
[ec2-user html]$ wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.tar.gz
```

6. phpMyAdmin 폴더를 생성하고 다음 명령을 사용하여 해당 폴더로 패키지의 압축을 풉니다.

```
[ec2-user html]$ mkdir phpMyAdmin && tar -xvzf phpMyAdmin-latest-all-languages.tar.gz -C phpMyAdmin --strip-components 1
```

7. `phpMyAdminlatest-all-languages-.tar.gz #####` 삭제합니다.

```
[ec2-user html]$ rm phpMyAdmin-latest-all-languages.tar.gz
```

8. (선택 사항) MySQL 서버가 실행 중이지 않으면 지금 시작합니다.

```
[ec2-user ~]$ sudo systemctl start mariadb
```

9. 웹 브라우저에 설치 URL을 입력합니다. phpMyAdmin 아래의 예와 같이 이 URL은 인스턴스의 퍼블릭 DNS 주소(또는 퍼블릭 IP 주소)에 슬래시(/)와 설치 디렉터리의 이름이 추가된 형태입니다.
예:

```
http://my.public.dns.amazonaws.com/phpMyAdmin
```

phpMyAdmin 로그인 페이지가 표시되어야 합니다.

phpMyAdmin

Welcome to phpMyAdmin

Language

English

Log in

Username: root

Password:

Go

10. 이전에 만든 root 사용자 이름과 MySQL 루트 암호를 사용하여 phpMyAdmin 설치에 로그인합니다.

작동하려면 먼저 설치를 구성해야 합니다. 먼저 다음과 같이 구성 파일을 수동으로 작성하는 것이 좋습니다.

- a. 최소 구성 파일로 시작하려면 자주 사용하는 텍스트 편집기를 사용하여 새 파일을 생성한 후에 config.sample.inc.php 내용을 파일에 복사합니다.
- b. 가 들어 index.php 있는 config.inc.php phpMyAdmin 디렉터리에 파일을 저장합니다.
- c. 추가 [설정에 대한 내용은 설치 지침의 phpMyAdmin 설치 스크립트 사용](#) 섹션에 있는 사후 파일 작성 지침을 참조하십시오.

사용에 대한 자세한 내용은 사용 phpMyAdmin [phpMyAdmin 설명서](#)를 참조하십시오.

문제 해결

이 섹션에서는 새 LAMP 서버를 설정하는 동안 발생할 수 있는 일반적인 문제 해결을 위한 제안을 제공합니다.

웹 브라우저를 사용하여 서버에 연결할 수 없음

다음을 확인하여 Apache 웹 서버가 실행 중이고 액세스 가능한지 확인합니다.

- 웹 서버가 실행되고 있습니까?

다음 명령을 실행하여 httpd가 실행되고 있는지 확인할 수 있습니다.

```
[ec2-user ~]$ sudo systemctl is-enabled httpd
```

httpd 프로세스가 실행되지 않는 경우 [LAMP 서버를 준비하려면](#)에 설명된 단계를 반복합니다.

- 방화벽이 올바르게 구성되었습니까?

인스턴스의 보안 그룹에 포트 80에서 HTTP 트래픽을 허용하는 규칙이 포함되어 있는지 확인합니다. 자세한 내용은 [보안 그룹에 규칙 추가](#)를 참조하십시오.

HTTPS를 사용하여 서버에 연결할 수 없음

다음 확인을 수행하여 Apache 웹 서버가 HTTPS를 지원하도록 구성되어 있는지 확인합니다.

- 웹 서버가 올바르게 구성되었습니까?

Apache를 설치한 후 서버는 HTTP 트래픽에 대해 구성됩니다. HTTPS를 지원하려면 서버에서 TLS를 활성화하고 SSL 인증서를 설치합니다. 자세한 정보는 [튜토리얼: AL2023 기반 SSL/TLS 구성](#) 섹션을 참조하세요.

- 방화벽이 올바르게 구성되었습니까?

인스턴스의 보안 그룹에 포트 443에서 HTTP 트래픽을 허용하는 규칙이 포함되어 있는지 확인합니다. 자세한 내용은 [Linux 인스턴스의 인바운드 트래픽 인증](#)을 참조하십시오.

관련 주제

인스턴스로 파일을 전송하거나 웹 서버에 WordPress 블로그를 설치하는 방법에 대한 자세한 내용은 다음 문서를 참조하십시오.

- Amazon EC2 사용 설명서의 [WinSCP를 사용하여 Linux 인스턴스로 파일을 전송하십시오.](#)
- Amazon EC2 사용 설명서의 [SCP 클라이언트를 사용하여 Linux 인스턴스로 파일을 전송하십시오.](#)
- [튜토리얼: AL2023 WordPress 블로그 호스팅](#)

이 자습서에서 사용되는 명령과 소프트웨어에 대한 자세한 내용은 다음 웹 페이지를 확인하세요.

- Apache 웹 서버: <http://httpd.apache.org/>
- MariaDB 데이터베이스 서버: <https://mariadb.org/>
- PHP 프로그래밍 언어: <http://php.net/>

웹 서버에 대한 도메인 이름을 등록하거나 기존 도메인 이름을 현재 호스트로 이전하는 것에 대한 자세한 내용은 Amazon Route 53 개발자 안내서의 [Amazon Route 53에서 도메인 및 하위 도메인 생성 및 마이그레이션](#)을 참조하세요.

튜토리얼: AL2023 기반 SSL/TLS 구성

Secure Sockets Layer/Transport Layer Security(SSL/TLS)는 웹 서버와 웹 클라이언트 간 암호화된 채널을 만들어 전송 중인 데이터가 도청되지 않도록 보호합니다. 이 자습서에서는 AL2023 및 Apache 웹 서버를 사용하는 EC2 인스턴스에 SSL/TLS 지원을 수동으로 추가하는 방법을 설명합니다. 이 자습서에서는 로드 밸런서를 사용하고 있지 않다고 가정합니다. Elastic Load Balancing을 사용하는 경우 [AWS Certificate Manager](#)의 인증서를 대신 사용하여 로드 밸런서에 SSL 오프로드를 구성하도록 선택할 수 있습니다.

일반적으로 웹 암호화를 단순히 SSL이라고 부릅니다. 웹 브라우저에서 여전히 SSL을 지원하지만, 후속 프로토콜인 TLS가 공격에 덜 취약합니다. AL2023에서는 기본적으로 모든 버전의 SSL에 대한 서버 측 지원을 비활성화합니다. [보안 표준 본문](#)에서는 TLS 1.0이 안전하지 않다고 간주합니다. TLS 1.0 및 TLS 1.1은 2021년 3월에 공식적으로 [사용 중단](#)되었습니다. 이 자습서에서는 TLS 1.2 활성화만을 기반으로 하여 지침을 제공합니다. TLS 1.3은 2018년에 완성되었으며 기본 TLS 라이브러리(이 자습서에서는 OpenSSL)가 지원되고 활성화되어 있는 한 AL2에서 사용할 수 있습니다. [클라이언트는 2023년 6월 28일까지 TLS 1.2 이상을 지원해야 합니다.](#) 업데이트된 암호화 표준에 대한 자세한 내용은 [RFC 7568](#) 및 [RFC 8446](#)을 참조하세요.

이 자습서는 현대 웹 암호화를 단순히 TLS로 언급합니다.

⚠ Important

이 절차는 AL2023 사용을 위한 것입니다. 다른 배포를 실행하는 EC2 인스턴스 또는 이전 Amazon Linux 버전을 실행하는 인스턴스를 설정하려는 경우 이 자습서의 일부 절차가 적용하지 않을 수 있습니다. Ubuntu의 경우 Ubuntu 커뮤니티 설명서: [Ubuntu에서 SSL 열기](#)를 참조하십시오. Red Hat Enterprise Linux의 경우 [Apache HTTP 웹 서버 설정](#)을 참조하십시오. 다른 배포는 관련 설명서를 참조하십시오.

ℹ Note

또는 Nitro 엔클레이브용 AWS Certificate Manager (ACM) 을 사용할 수도 있습니다. 이 엔클레이브 애플리케이션은 AWS Nitro 엔클레이브가 설치된 Amazon EC2 인스턴스에서 실행되는 웹 애플리케이션 및 서버에서 퍼블릭 및 프라이빗 SSL/TLS 인증서를 사용할 수 있는 엔클레이브 애플리케이션입니다. AWS Nitro Enclaves는 격리된 컴퓨팅 환경을 생성하여 SSL/TLS 인증서 및 프라이빗 키와 같은 매우 민감한 데이터를 보호하고 안전하게 처리할 수 있도록 지원하는 Amazon EC2 기능입니다.

Nitro Enclaves용 ACM은 Amazon EC2 Linux 인스턴스에서 실행되는 nginx와 함께 작동하여 프라이빗 키를 생성하고, 인증서와 프라이빗 키를 배포하며, 인증서 갱신을 관리합니다.

Nitro Enclaves용 ACM을 사용하려면 엔클레이브 지원 Linux 인스턴스를 사용해야 합니다.

자세한 내용은 [Nitro 엔클레이브란 무엇입니까?](#) 를 AWS 참조하십시오. 그리고 니트로 엔클레이브AWS Certificate Manager 사용 설명서의 [니트로 엔클레이브에 대해서도 읽어보세요](#).AWS

내용

- [필수 조건](#)
- [1단계: 서버에서 TLS 활성화](#)
- [2단계: CA가 서명한 인증서 가져오기](#)
- [3단계: 보안 구성 테스트 및 강화](#)
- [문제 해결](#)

필수 조건

이 자습서를 시작하기 전에 다음 단계를 완료합니다.

- EBS 지원 AL2023 인스턴스를 시작합니다. 자세한 정보는 [Amazon EC2 AL2023](#)을 참조하십시오.

- 인스턴스가 다음 TCP 포트에서 연결을 허용하도록 보안 그룹을 구성합니다.
 - SSH(포트 22)
 - HTTP(포트 80)
 - HTTPS(포트 443)

자세한 내용은 Amazon EC2 [사용 설명서의 Linux 인스턴스의 인바운드 트래픽 승인을](#) 참조하십시오.

- Apache 웹 서버를 설치합니다. step-by-step 지침은 을 참조하십시오. [튜토리얼: AL2023 기반 램프 서버 설치](#) httpd 패키지와 그 종속 프로그램만 필요합니다. PHP 및 MariaDB와 관련된 지침은 무시해도 됩니다.
- 웹 사이트를 식별하고 인증하려면 TLS 퍼블릭 키 인프라(PKI)는 도메인 이름 시스템(DNS)을 사용합니다. EC2 인스턴스를 사용하여 퍼블릭 웹 사이트를 호스팅하려는 경우, 웹 서버의 도메인 이름을 등록하거나 Amazon EC2 호스트로 기존 도메인 이름을 전송해야 합니다. 수많은 타사 도메인 등록 및 DNS 호스팅 서비스를 이에 사용할 수 있습니다. 또는 [Amazon Route 53](#)을 사용할 수도 있습니다.

1단계: 서버에서 TLS 활성화

이 절차는 자체 서명된 디지털 인증서를 사용하여 AL2023 기반 TLS를 설정하는 프로세스를 안내합니다.

Note

자체 서명된 인증서는 테스트에는 허용되지만 프로덕션에는 허용되지 않습니다. 자체 서명된 인증서를 인터넷에 노출하면 사이트 방문자에게 인사말로 보안 경고가 표시됩니다.

서버에서 TLS를 활성화하려면

1. 인스턴스에 연결한 다음 Apache가 실행되는지 확인합니다. 자세한 정보는 [AL2023 인스턴스에 연결](#)을 참조하세요.

```
[ec2-user ~]$ sudo systemctl is-enabled httpd
```

반환된 값이 "enabled"가 아닌 경우 Apache를 시작한 다음 시스템 부팅 시마다 시작하도록 설정합니다.

```
[ec2-user ~]$ sudo systemctl start httpd && sudo systemctl enable httpd
```

- 모든 소프트웨어 패키지가 최신 상태로 업데이트되어 있는지 확인하기 위해, 인스턴스에서 쿼크 소프트웨어 업데이트를 실행합니다. 이 업데이트 과정은 몇 분 정도 시간이 소요될 수 있지만, 최신 보안 업데이트와 버그 수정을 위해 수행할 필요가 있습니다.

Note

-y 옵션을 사용하면 확인 여부를 묻지 않고 업데이트를 설치합니다. 설치 전에 업데이트 정보를 확인하려면 이 옵션을 생략합니다.

```
[ec2-user ~]$ sudo dnf install openssl mod_ssl
```

- 다음 명령을 입력하면 사이트에 대한 정보를 입력할 수 있는 프롬프트가 표시됩니다.

```
[ec2-user ~]$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/pki/tls/private/apache-selfsigned.key -out /etc/pki/tls/certs/apache-selfsigned.crt
```

그러면 새 파일인 `apache-selfsigned.crt` 파일이 `/etc/pki/tls/certs/` 디렉터리에 생성됩니다. 지정된 파일의 이름은 `/etc/httpd/conf.d/ssl.conf`의 `SSLCertificateFile` 명령에 할당된 기본값과 일치합니다.

이제 인스턴스에는 보안 서버 구성과 테스트를 위한 인증서 생성에 사용할 다음 파일이 포함됩니다.

- `/etc/httpd/conf.d/ssl.conf`

`mod_ssl`의 구성 파일입니다. 여기에는 Apache에 암호화 키 및 인증서의 위치, 허용하는 TLS 프로토콜 버전, 허용하는 암호화 암호를 알려주는 명령이 포함되어 있습니다. 다음은 로컬 인증서 파일입니다.

- `/etc/pki/tls/certs/apache-selfsigned.crt`

이 파일에는 자체 서명된 인증서와 인증서의 프라이빗 키가 모두 포함됩니다. Apache에서는 인증서와 키를 PEM 형식으로 요구합니다. 이 형식은 아래의 축약된 예제와 같이 "BEGIN" 및 "END" 라인으로 프레임 처리된 Base64 인코딩 ASCII 문자로 구성됩니다.

```

-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBCkgwggSkAgEAAoIBAQD2KKx/8Zk94m1q
3gQMZF9ZN66Ls19+3tHAgQ5Fpo9KJDhzLj00CI8u1PTcGmAah5kEitCEc0wzmNeo
BC10wYR6G0rGaKtK9Dn7CuIjvubtUysVyQoMVPQ971deakHWeRMiEJFXg6kZZ0vr
GvwnKoMh3D1K44D9dX7IDua2P1Yx5+eroA+1Lqf32ZSaA00bBIMIYTHigwbHMZoT
...
56tE7THvH7v0Ef4/iU0sIrEzaMaJ0mqkmY1A70qQGQKBgBF3H1qNRNHuyMcP0DFs
27hDzPDinrquSEvoZIGgkDM1h2irTiipJ/GhkvTpoQ1v0fK/VXw8vSgeaBuhwJvS
LXU9HvYq0U604FgD3nAyB9hI0BE13r1HjUvbjT7moH+RhnNz6eeqqdscCS09VtRAO
4QQvAq0a8UheYeoXLdWcHaLP
-----END PRIVATE KEY-----

-----BEGIN CERTIFICATE-----
MIIEazCCA10gAwIBAgICWxQwDQYJKoZIhvcNAQELBQAwbGExCzAJBgNVBAYTAi0t
MRIwEAYDVQQIDAlTb211U3RhdGUxETAPBgNVBACMFNvbWVwDaXR5MRkwFwYDVQQK
DBBTb211T3JnYW5pemF0aW9uMR8wHQYDVQQLDBZTb211T3JnYW5pemF0aW9uYXV
bm10MRkwFwYDVQQDDDBBpcC0xNzItMzEtMjAtMjMMSQwIgwYJKoZIhvcNAQkBFhVy
...
z5rRUE/XzxRLBZ0oWZpNWTXJkQ3uFYH6s/sBwtHpKKZMz0vDedREjNKAvk4ws6F0
CuIjvubtUysVyQoMVPQ971deakHWeRMiEJFXg6kZZ0vrGvwnKoMh3D1K44D9d1U3
WanXWehT6FiSZvB4sTEXXJN2jdw8g+sHGnZ8zC0sc1knYhHrCVD2vnB1ZJKSZvak
3ZazhBxtQSukFM0nWPP2a0DMMFGYUHOd0BQE8sBJxg==
-----END CERTIFICATE-----

```

파일 이름과 확장명은 편의상 사용되며 기능에 영향을 미치지 않습니다. 예를 들어 `ssl.conf` 파일에서 관련 명령에 동일한 이름을 사용하는 한, 인증서 이름을 `cert.crt`, `cert.pem` 또는 다른 파일 이름으로 지정할 수 있습니다.

Note

기본 TLS 파일을 고유의 사용자 지정 파일로 대체하는 경우 파일이 PEM 형식인지 확인하세요.

4. Apache를 다시 시작합니다.

```
[ec2-user ~]$ sudo systemctl restart httpd
```

Note

앞에서 설명한 대로 EC2 인스턴스에서 TCP 포트 443에 액세스할 수 있는지 확인하세요.

5. Apache 웹 서버가 현재 포트 443에 대해 HTTPS(보안 HTTP)를 지원해야 합니다. 접두사가 **https://**인 브라우저 URL 표시줄에 IP 주소 또는 EC2 인스턴스의 정규화된 도메인 이름을 입력하여 이를 테스트합니다.

신뢰할 수 없는 자체 서명된 호스트 인증서를 사용하여 사이트에 연결하기 때문에 브라우저에 보안 경고가 연속으로 표시될 수 있습니다. 경고를 무시하고 계속 진행합니다.

Apache 기본 테스트 페이지가 열리면 서버에 TLS가 구성되었다는 것입니다. 브라우저와 서버 사이를 통과하는 모든 데이터가 이제 암호화됩니다.

Note

사이트 방문자에게 경고 화면이 표시되는 것을 방지하려면 암호화뿐만 아니라 해당 사이트의 소유자라는 것을 공개적으로 인증하는 신뢰할 수 있는 CA 서명 인증서를 가져와야 합니다.

2단계: CA가 서명한 인증서 가져오기

CA가 서명한 인증서를 가져오려면 다음 절차를 사용할 수 있습니다.

- 프라이빗 키에서 인증서 서명 요청(CSR)을 생성합니다.
- 인증 기관(CA)에 CSR 제출
- 단일 호스트 인증서 가져오기
- Apache를 수정하여 인증서 사용

자체 서명된 TLS X.509 호스트 인증서는 CA가 서명한 인증서와 암호적으로 동일합니다. 그 차이는 수학적이지가 아니라 사회적입니다. CA는 신청자에게 인증서를 발급하기 전에 도메인의 소유권을 최소한으로 검사합니다. 각 웹 브라우저에는 이를 하도록 브라우저 공급업체에서 신뢰한 CA 목록이 포함되어 있습니다. X.509 인증서는 프라이빗 서버 키에 해당하는 퍼블릭 키와 퍼블릭 키에 암호화 방식으로 연결된 CA의 서명으로 주로 구성되어 있습니다. 브라우저가 HTTPS를 통해 웹 서버에 연결되면 서버는 브라우저에서 신뢰할 수 있는 CA 목록을 확인하도록 인증서를 제공합니다. 서명자가 목록에 있거나

신뢰할 수 있는 다른 서명자로 구성되는 신뢰 체인을 통해 서명자에 액세스할 수 있는 경우, 브라우저는 서버와 암호화된 빠른 데이터 채널을 협상하고 페이지를 로드합니다.

요청 확인 절차로 인해 인증서에는 일반적으로 비용이 발생하므로 여러 인증 기관을 알아봐야 합니다. 일부 CA는 기본 수준 인증서를 무료로 제공합니다. 이러한 CA 중 가장 주목할 만한 것은 [Let's Encrypt](#) 프로젝트인데, 이것은 인증서 생성 및 갱신 프로세스의 자동화도 지원합니다. Let's Encrypt 인증서 사용에 대한 자세한 내용은 [Certbot 받기](#)를 참조하세요.

상업용 서비스를 제공할 계획이라면 [AWS Certificate Manager](#)가 좋은 옵션입니다.

호스트 인증서의 기본을 이루는 것은 키입니다. 2019년 현재 [정부](#) 및 [산업](#) 그룹에서는 2030년까지 문서를 보호하기 위해 마련된 RSA 키에 대해 최소 2048비트의 키(모듈러스) 크기를 사용할 것을 권장합니다. AL2023 OpenSSL에서 생성되는 기본 모듈러스 크기는 2048비트로, CA 서명 인증서에 사용하기에 적합합니다. 다음 절차는 사용자 지정된 키(예: 더 큰 모듈러스 또는 다른 암호화 알고리즘 사용)를 원하는 사용자를 위해 제공되는 선택적 단계입니다.

Important

이 CA 서명 호스트 인증서 획득 지침은 등록과 호스팅이 완료된 DNS 도메인을 소유하지 않을 경우 제대로 적용하기 어렵습니다.

CA가 서명한 인증서를 가져오려면

1. 인스턴스에 연결한 다음 `/etc/pki/tls/private/`으로 이동합니다. 여기는 TLS에 대한 서버의 프라이빗 키를 저장하는 디렉터리입니다. 기존 호스트 키를 사용하여 CSR을 생성하려면 3단계로 건너뛰니다. 인스턴스 연결에 대한 자세한 내용은 [AL2023 인스턴스에 연결](#)을 참조하십시오.
2. (선택 사항) 새 프라이빗 키를 생성합니다. 다음은 몇 가지 키 구성 샘플입니다. 어떤 결과 키도 웹 서버에서 사용할 수 있지만 보안 구현의 정도와 유형은 각각 다릅니다.
 - 예 1: 기본 RSA 호스트 키를 만듭니다. 결과 파일인 **custom.key**는 2048비트 RSA 프라이빗 키입니다.

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key
```

- 예 2: 더 큰 모듈러스로 더 강력한 RSA 키를 만듭니다. 결과 파일인 **custom.key**는 4096비트 RSA 프라이빗 키입니다.

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key 4096
```

- 예 3: 암호로 보호되는 4096비트 암호화 RSA 키를 생성합니다. 그러면 AES-128 암호화로 암호화된 4096비트 RSA 프라이빗 키인 **custom.key** 파일이 생성됩니다.

Important

키 암호화를 통해 보안을 강화할 수 있지만, 암호화된 키에는 암호가 필요하기 때문에 이를 사용하는 서비스는 자동으로 시작할 수 없습니다. 이 키를 사용할 때마다 SSH 연결을 통해 암호(위의 예에서는 "abcde12345")를 제공해야 합니다.

```
[ec2-user ~]$ sudo openssl genrsa -aes128 -passout pass:abcde12345 -out
custom.key 4096
```

- 예 4: 비 RSA 암호를 사용하여 키를 생성합니다. RSA 암호화는 두 개의 라지 소수의 결과를 기반으로 하는 공개 키의 크기 때문에 상대적으로 느릴 수 있습니다. 그러나 RSA 암호화 이외의 암호화를 사용하는 TLS의 키를 생성할 수 있습니다. 타원 곡선 수학을 기반으로 하는 키는 동등한 보안 수준을 제공할 때보다 작고 산술적으로 빠릅니다.

```
[ec2-user ~]$ sudo openssl ecparam -name prime256v1 -out custom.key -genkey
```

그 결과는 OpenSSL에서 지원하는 "명명된 곡선"인 prime256v1을 사용하는 256비트 타원 곡선 프라이빗 키입니다. [NIST](#)에 따르면 이 키의 암호화 강도는 2048비트 RSA 키보다 약간 더 높습니다.

Note

모든 CA가 RSA elliptic-curve-based 키와 동일한 수준의 키 지원을 제공하는 것은 아닙니다.

새 프라이빗 키의 소유권 및 권한은 매우 제한적(소유자=루트, 그룹=루트, 소유자 전용 읽기/쓰기)이어야 합니다. 명령은 다음 예에서와 같습니다.

```
[ec2-user ~]$ sudo chown root:root custom.key
[ec2-user ~]$ sudo chmod 600 custom.key
[ec2-user ~]$ ls -al custom.key
```

이 명령의 결과는 다음과 같아야 합니다.

```
-rw----- root root custom.key
```

만족스러운 키를 생성 및 구성한 후 CSR을 생성할 수 있습니다.

- 원하는 키를 사용하여 CSR을 생성합니다. 다음 예에는 **custom.key**가 사용됩니다.

```
[ec2-user ~]$ sudo openssl req -new -key custom.key -out csr.pem
```

OpenSSL은 대화 상자를 열고 아래 표의 정보를 입력하라는 메시지를 표시합니다. 도메인에서 확인된 기본 호스트 인증서의 경우 Common Name을 제외한 모든 필드는 선택 사항입니다.

명칭	설명	예
국가 이름	해당 국가의 두 자리 ISO 약자.	US(=미국)
주 또는 지방 이름	해당 조직이 위치한 주 또는 지방의 이름. 이 이름은 약어로 사용할 수 없음.	워싱턴
시 이름	조직의 위치(예: 도시).	시애틀
조직 이름	해당 조직의 정식 이름. 조직 이름의 약칭을 사용하지 마세요.	Example Corporation
조직 단위 이름	조직에 대한 추가 정보(있는 경우).	부서 예
일반 이름	이 값은 사용자가 브라우저에 입력해야 하는 웹 주소와 정확히 일치해야 합니다. 일반적으로 이는 www.example.com 의 형식으로, 호스트 이름 또는 별칭이 앞에 붙는 도메인 이름을 뜻합니다. 자체 서명된 인증서로 DNS 확인 없이 테스트하는 경우, 일반 이름은 호스트 이름만으로 구성될 수 있습니다. CA는 *.example.com 과 같이 와일드 카드 이름을 허용하는 비싼 인증서도 제공합니다.	www.example.com
이메일 주소	서버 관리자의 이메일 주소.	someone@example.com

마지막으로 OpenSSL은 챌린지 암호(선택 사항)를 입력하라는 메시지를 표시합니다. 이 암호는 해당 CSR 및 사용자와 해당 CA 간의 트랜잭션에만 적용되므로, 암호 및 기타 선택적 필드(선택적 회사 이름)에 대한 해당 CA의 권장 사항을 따릅니다. CSR 챌린지 암호는 서버 작업에 영향을 미치지 않습니다.

결과 파일인 `csr.pem`에는 퍼블릭 키, 퍼블릭 키의 디지털 서명 및 입력한 메타데이터가 포함되어 있습니다.

4. CA에 CSR을 제출합니다. 이는 보통 텍스트 편집기에서 CSR 파일을 열고 웹 양식에 내용을 복사하는 것으로 구성됩니다. 이때 인증서에 추가할 하나 이상의 주체 대체 이름(SAN)을 입력하라는 메시지가 나타날 수 있습니다. `www.example.com`이 일반 이름일 경우, `example.com`은 좋은 SAN이며, 그 반대의 경우도 마찬가지입니다. 사이트 방문자는 이 이름 중 하나를 입력하면 오류 없이 연결됩니다. CA 웹 양식에서 이를 허용하는 경우, SAN 목록에 일반 이름을 포함시킵니다. 일부 CA는 이를 자동으로 포함시킵니다.

요청이 승인되면 CA에서 서명한 새 호스트 인증서를 받게 됩니다. CA의 신뢰 체인을 완료하는 데 필요한 추가 인증서가 포함된 중간 인증서 파일을 다운로드하라는 안내를 받을 수도 있습니다.

Note

CA는 다양한 목적을 위해 마련된 여러 형식의 파일을 보낼 수 있습니다. 본 자습서에서는 PEM 형식의 인증서 파일만 사용해야 하는데, 이는 보통 `.pem` 또는 `.crt` 파일 확장명으로 표시되지만 항상 그런 것은 아닙니다. 어떤 파일을 사용할지 확실하지 않은 경우 텍스트 편집기로 파일을 열고 다음 라인으로 시작되는 블록 하나 이상이 포함되는 파일을 찾습니다.

```
- - - - -BEGIN CERTIFICATE - - - - -
```

또한 파일은 다음 라인으로 끝나야 합니다.

```
- - - - -END CERTIFICATE - - - - -
```

또한 명령줄의 파일을 다음과 같이 테스트할 수 있습니다.

```
[ec2-user certs]$ openssl x509 -in certificate.crt -text
```

이 줄이 파일에 나타나는지 확인하세요. .p7b, .p7c, 또는 유사한 파일 확장명으로 끝나는 파일을 사용하지 않습니다.

5. /etc/pki/tls/certs 디렉터리에 CA가 서명한 새 인증서와 모든 중간 인증서를 배치합니다.

Note

여러 가지 방법으로 새 인증서를 EC2 인스턴스에 업로드할 수 있지만, 가장 간편하고 유익한 방법은 텍스트 편집기(예: vi, nano, 메모장)를 로컬 컴퓨터와 인스턴스에 모두 열고 두 편집기 간에 파일 콘텐츠를 복사하여 붙이는 것입니다. EC2 인스턴스에서 이러한 작업을 수행할 때 루트 [sudo] 권한이 필요합니다. 이렇게 하면 권한 또는 경로 문제가 있는 경우 즉시 확인할 수 있습니다. 하지만 콘텐츠를 복사하는 동안 라인을 추가하거나 어떤 식으로든 콘텐츠를 변경하지 않도록 주의하세요.

/etc/pki/tls/certs 디렉터리 내에서 파일 소유권, 그룹 및 권한 설정이 매우 제한적인 AL2023 기본값 (소유자=루트, 그룹=루트, 소유자 전용 읽기/쓰기) 과 일치하는지 확인합니다. 다음 예제는 사용하는 명령을 보여 줍니다.

```
[ec2-user certs]$ sudo chown root:root custom.crt
[ec2-user certs]$ sudo chmod 600 custom.crt
[ec2-user certs]$ ls -al custom.crt
```

이 명령의 결과는 다음과 같아야 합니다.

```
-rw----- root root custom.crt
```

중간 인증서 파일에 대한 권한은 덜 엄격합니다(소유자=루트, 그룹=루트, 소유자 쓰기 가능, 그룹 읽기 가능, 모든 사용자 읽기 가능). 다음 예제는 사용하는 명령을 보여 줍니다.

```
[ec2-user certs]$ sudo chown root:root intermediate.crt
[ec2-user certs]$ sudo chmod 644 intermediate.crt
[ec2-user certs]$ ls -al intermediate.crt
```

이 명령의 결과는 다음과 같아야 합니다.

```
-rw-r--r-- root root intermediate.crt
```

6. `/etc/pki/tls/private/` 디렉터리에서 CSR을 생성할 때 사용한 프라이빗 키를 배치합니다.

Note

여러 가지 방법으로 사용자 지정 키를 EC2 인스턴스에 업로드할 수 있지만, 가장 간편하고 유익한 방법은 텍스트 편집기(예: vi, nano, 메모장)를 로컬 컴퓨터와 인스턴스에 모두 열고 두 편집기 간에 파일 콘텐츠를 복사하여 붙이는 것입니다. EC2 인스턴스에서 이러한 작업을 수행할 때 루트 [sudo] 권한이 필요합니다. 이렇게 하면 권한 또는 경로 문제가 있는 경우 즉시 확인할 수 있습니다. 하지만 콘텐츠를 복사하는 동안 라인을 추가하거나 어떤 식으로든 콘텐츠를 변경하지 않도록 주의하세요.

`/etc/pki/tls/private` 디렉터리 내에서 다음 명령을 사용하여 파일 소유권, 그룹 및 권한 설정이 매우 제한적인 AL2023 기본값 (소유자=루트, 그룹=루트, 소유자 전용 읽기/쓰기) 과 일치하는지 확인하십시오.

```
[ec2-user private]$ sudo chown root:root custom.key
[ec2-user private]$ sudo chmod 600 custom.key
[ec2-user private]$ ls -al custom.key
```

이 명령의 결과는 다음과 같아야 합니다.

```
-rw----- root root custom.key
```

7. 새 인증서 및 키 파일을 반영하기 위해 `/etc/httpd/conf.d/ssl.conf`를 편집합니다.
- Apache의 `SSLCertificateFile` 명령에 CA가 서명한 호스트 인증서의 경로와 파일 이름을 입력합니다.

```
SSLCertificateFile /etc/pki/tls/certs/custom.crt
```

- 중간 인증서 파일을 받은 경우(이 예에서는 `intermediate.crt`), Apache의 `SSLCACertificateFile` 명령을 사용하여 경로 및 파일 이름을 입력합니다.

```
SSLCACertificateFile /etc/pki/tls/certs/intermediate.crt
```

Note

일부 CA는 호스트 인증서와 중간 인증서를 단일 파일로 결합하기 때문에 `SSLCertificateFile` 명령이 불필요합니다. CA가 제공한 지침을 참조하세요.

- c. Apache의 `SSLCertificateKeyFile` 명령에 프라이빗 키(이 예에서는 `custom.key`)의 경로와 파일 이름을 입력합니다.

```
SSLCertificateKeyFile /etc/pki/tls/private/custom.key
```

8. `/etc/httpd/conf.d/ssl.conf`를 저장하고 Apache를 다시 시작합니다.

```
[ec2-user ~]$ sudo systemctl restart httpd
```

9. `https://` 접두사가 포함된 브라우저 URL 막대에 도메인 이름을 입력하여 서버를 테스트합니다. 브라우저에서는 테스트 페이지가 오류 생성 없이 HTTPS를 통해 로드되어야 합니다.

3단계: 보안 구성 테스트 및 강화

TLS이 작동되고 일반에 공개된 후 이의 실제 보안 수준을 테스트해야 합니다. 보안 설정을 무료로 완벽하게 분석해 주는 [Qualys SSL Labs](#)와 같은 온라인 서비스를 사용하면 이를 손쉽게 수행할 수 있습니다. 그 결과에 따라 수용할 프로토콜, 원하는 암호 및 제외할 암호를 관리하여 기본 보안 구성을 강화할 수 있습니다. 자세한 내용은 [how Qualys formulates its scores](#) 섹션을 참조하세요.

Important

실제 테스트는 서버 보안에 매우 중요합니다. 구성상의 작은 오류가 심각한 보안 침해 및 데이터 손실로 이어질 수 있습니다. 권장되는 보안 사례는 연구 및 새롭게 생겨나는 위협에 대처하기 위해 끊임없이 변화하므로 보안 감사를 주기적으로 실시하는 것이 서버 관리에 필수적입니다.

[Qualys SSL Labs](#) 사이트에 `www.example.com` 형식으로 서버의 정규화된 도메인 이름을 입력합니다. 약 2분 후 사이트 등급(A - F) 및 확인된 상세 분석 결과를 받게 됩니다. 다음 표에는 AL2023 기본 Apache 구성과 동일한 설정과 기본 Certbot 인증서를 사용하는 도메인에 대한 보고서가 요약되어 있습니다.

종합 등급	B
인증서	100%
프로토콜 지원	95%
키 교환	70%
암호화 수준	90%

개요에서 구성이 대체로 문제가 없어 보여도 세부 정보 보고서에서는 몇몇 잠재적 문제를 여기에 심각도 순서로 나열하여 표시합니다.

x RC4 암호는 이전 버전의 특정 브라우저에서 사용하도록 지원됩니다. 암호는 암호화 알고리즘의 수학적 핵심입니다. TLS 데이터 스트림을 암호화하는 데 사용하는 빠른 암호인 RC4에는 몇 가지 [심각한 취약점](#)이 있는 것으로 알려져 있습니다. 타당한 레거시 브라우저 지원 사유가 없다면 비활성화해야 합니다.

x 이전 TLS 버전이 지원됩니다. 구성에서는 TLS 1.0(이미 사용 중지 상태)과 TLS 1.1(사용 중지 절차 진행 중)을 지원합니다. 2018년부터는 TLS 1.2만 권장됩니다.

x 전방향 보안은 부분적으로 지원됩니다. [전방향 보안](#)은 프라이빗 키에서 파생된 임시(사용 후 삭제) 세션 키를 사용하여 암호화하는 알고리즘의 기능입니다. 이는 실제 공격자가 웹 서버의 장기 프라이빗 키를 보유하고 있더라도 HTTPS 데이터의 암호를 해독할 수 없다는 것을 뜻합니다.

TLS 구성을 수정하고 향후에 대비하려면

1. 텍스트 편집기에서 `/etc/httpd/conf.d/ssl.conf` 구성 파일을 열고 다음 줄의 시작 부분에 `"#"`을 입력하여 해당 줄을 주석으로 처리합니다.

```
#SSLProtocol all -SSLv3
```

2. 다음 명령을 추가합니다.

```
#SSLProtocol all -SSLv3
SSLProtocol -SSLv2 -SSLv3 -TLSv1 -TLSv1.1 +TLSv1.2
```

이러한 명령은 SSL 버전 2 및 3과 TLS 버전 1.0 및 1.1을 명시적으로 비활성화합니다. 이제 이 서버는 TLS 1.2 이외의 프로토콜을 사용하는 클라이언트와의 암호화된 연결을 허용하지 않습니다. 명령의 상세 내용은 서버의 구성 내용을 사람에게 더욱 명확히 전달합니다.

Note

이러한 방식으로 TLS 버전 1.0 및 1.1을 비활성화하면 적은 비율의 오래된 웹 브라우저가 사이트에 액세스하지 못하도록 차단합니다.

허용된 암호의 목록을 수정하려면

1. 구성 파일인 `/etc/httpd/conf.d/ssl.conf`에서 **SSLCipherSuite** 명령이 포함된 섹션을 찾고 기존의 줄을 줄의 시작에 `#`을 입력하여 주석으로 처리합니다.

```
#SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5
```

2. 명시적 암호 그룹과 전방향 보안을 우선순위에 두고 부정확한 암호를 방지하는 암호 오더를 지정합니다. 여기에 사용된 **SSLCipherSuite** 명령은 서버에서 실행되는 특정 소프트웨어에 맞게 TLS 구성을 조정하는 [Mozilla SSL Configuration Generator](#)의 출력에 기반합니다. (자세한 내용은 Mozilla의 [Security/Server Side TLS\(보안/서버 측 TLS\)](#)를 참조하세요.) 먼저 다음 명령의 출력을 사용하여 Apache와 OpenSSL의 버전을 확인합니다.

```
[ec2-user ~]$ yum list installed | grep httpd
```

```
[ec2-user ~]$ yum list installed | grep openssl
```

예를 들어, 반환된 정보가 Apache 2.4.34 및 OpenSSL 1.0.2인 경우 이를 생성기에 입력합니다. "현대" 호환성 모델을 선택하면 적극적으로 보안을 적용하지만 대부분의 브라우저에서 여전히 작동하는 **SSLCipherSuite** 명령을 생성합니다. 소프트웨어가 최신 구성을 지원하지 않으면 소프트웨어를 업데이트하거나 대신 "중간" 구성을 선택할 수 있습니다.

```
SSLCipherSuite ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:
ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:
ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256
```

선택된 암호에는 이름에 Elliptic Curve Diffie-Hellman Ephemeral의 약자인 ECDHE가 포함되어 있습니다. ephemeral은 전방향 보안을 나타냅니다. 부차적 결과로서 해당 암호는 RC4를 지원하지 않습니다.

내용이 표시되지 않는 기본값 또는 terse 명령 대신 명시적 암호 목록을 사용하는 것이 좋습니다.

생성된 명령을 `/etc/httpd/conf.d/ss1.conf`에 복사합니다.

Note

여기에서는 가독성을 위해 여러 줄로 표시했지만, 이 명령은 `/etc/httpd/conf.d/ss1.conf`에 복사할 때 암호 이름 사이에 공백 없이 콜론만을 추가하여 한 줄에 입력해야 합니다.

3. 마지막으로 줄 시작 부분에 있는 "#"을 제거하여 다음 줄의 주석 처리를 해제합니다.

```
#SSLHonorCipherOrder on
```

이 명령은 (이 예에서는) 전방향 보안을 지원하는 암호를 포함하여 서버에서 순위가 높은 암호를 선호하도록 합니다. 이 명령이 설정되면 서버는 먼저 강력한 보안 연결 설정을 시도해 본 후 보안이 더 약한 허용된 암호로 대체합니다.

이 두 절차를 모두 완료한 다음에는 변경 사항을 `/etc/httpd/conf.d/ss1.conf`에 저장하고 Apache를 재시작합니다.

[Qualys SSL Labs](#)에서 도메인을 다시 테스트하면 RC4 취약성과 다른 경고 문제가 해결되고 요약은 다음과 같을 것입니다.

종합 등급	A
인증서	100%
프로토콜 지원	100%
키 교환	90%
암호화 수준	90%

OpenSSL을 업데이트할 때마다 새 암호가 사용되고 이전 암호에 대한 지원은 제거됩니다. EC2 AL2023 인스턴스를 유지하고 up-to-date, [OpenSSL의](#) 보안 발표를 주시하고, 기술 언론에서 보도되는 새로운 보안 침해 사례에 주의를 기울이십시오.

문제 해결

- 암호를 입력하지 않으면 Apache 웹 서버가 시작되지 않음

암호화되고 암호로 보호되는 프라이빗 서버 키를 설치한 경우 이는 예상된 동작입니다.

키에서 암호화 및 암호 요구 사항을 제거할 수 있습니다. 기본 디렉터리에 custom.key라는 암호화된 프라이빗 RSA 키가 있고 이 키의 암호가 **abcde12345**라고 가정하면, EC2 인스턴스에서 다음 명령을 실행하여 이 키의 암호화되지 않은 버전을 생성합니다.

```
[ec2-user ~]$ cd /etc/pki/tls/private/
[ec2-user private]$ sudo cp custom.key custom.key.bak
[ec2-user private]$ sudo openssl rsa -in custom.key -passin pass:abcde12345 -out
custom.key.nocrypt
[ec2-user private]$ sudo mv custom.key.nocrypt custom.key
[ec2-user private]$ sudo chown root:root custom.key
[ec2-user private]$ sudo chmod 600 custom.key
[ec2-user private]$ sudo systemctl restart httpd
```

이제 Apache가 암호를 묻지 않고 시작할 것입니다.

- sudo dnf install -y mod_ssl을 실행할 때 오류가 발생합니다.

SSL에 필요한 패키지를 설치하려 할 때 다음과 같은 오류가 표시될 수 있습니다.

```
Error: httpd24-tools conflicts with httpd-tools-2.2.34-1.16.amzn1.x86_64
Error: httpd24 conflicts with httpd-2.2.34-1.16.amzn1.x86_64
```

이는 일반적으로 EC2 인스턴스가 AL2023 인스턴스를 실행하고 있지 않음을 의미합니다. 이 자습서는 공식 AL2023 AMI에서 새로 생성한 인스턴스만 지원합니다.

튜토리얼: AL2023 WordPress 블로그 호스팅

다음 절차는 AL2023 인스턴스에 WordPress 블로그를 설치, 구성 및 보호하는 데 도움이 됩니다. 이 자습서는 WordPress 블로그를 호스팅하는 웹 서버를 완전히 제어할 수 있다는 점에서 Amazon EC2 사용에 대한 좋은 입문서입니다. 이는 기존 호스팅 서비스에서는 일반적이지 않습니다.

사용자는 서버에 대한 소프트웨어 패키지를 업데이트하고 보안 패치를 유지관리할 책임이 있습니다. 웹 서버 구성과 직접 상호 작용할 필요가 없는 보다 자동화된 WordPress 설치를 위해 이 AWS CloudFormation 서비스는 빠르게 시작할 수 있는 WordPress 템플릿을 제공합니다. 자세한 내용은 AWS CloudFormation 사용 설명서에서 [시작하기](#)를 참조하세요. Windows 인스턴스에서 WordPress 블로그를 호스팅하려면 Amazon EC2 [사용 설명서의 Amazon EC2 Windows 인스턴스에 WordPress 블로그 배포](#)를 참조하십시오. 데이터베이스를 분리한고가용성 솔루션이 필요한 경우 개발자 안내서의고가용성 웹 사이트 [배포](#)를 참조하십시오. WordPress AWS Elastic Beanstalk

Important

이 절차는 AL2023 사용을 위한 것입니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하세요. 본 자습서에 있는 단계의 상당수가 Ubuntu 인스턴스에서 작동하지 않습니다. Ubuntu 인스턴스에 WordPress 설치하는 데 도움이 필요하다면 Ubuntu [WordPress](#) 설명서를 참조하십시오. Amazon Linux, macOS 또는 Unix 시스템에서 [CodeDeploy](#) 사용하여 이 작업을 수행할 수도 있습니다.

주제

- [필수 조건](#)
- [설치 WordPress](#)
- [다음 단계](#)
- [도움말! 내 퍼블릭 DNS 이름이 변경되어 블로그를 사용할 수 없습니다.](#)

필수 조건

블로그를 호스팅하는 데 사용하는 인스턴스에 엘라스틱 IP 주소 (EIP) 를 연결하는 것이 좋습니다. WordPress 인스턴스의 퍼블릭 DNS 주소가 설치 위치를 바꾸거나 위반하는 것을 방지할 수 있기 때문입니다. 자신이 소유하고 있는 도메인 이름을 블로그에 사용하고 싶다면 도메인 이름의 DNS 레코드가 EIP 주소를 가리키도록 업데이트할 수 있습니다(이와 관련하여 도움이 필요하다면 도메인 이름 등록 기관에게 문의하세요). 실행 중인 인스턴스와 연결되어 있는 EIP 주소는 한 개까지 무료로 사용할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서에서 [탄력적 IP 주소](#)를 참조하세요. [튜토리얼: AL2023 기반 램프 서버 설치](#) 자습서는 보안 그룹이 HTTP 및 HTTPS 트래픽을 허용하도록 구성하는 단계와 파일 권한이 웹 서버에 맞게 적절하게 설정되어 있는지 확인하는 여러 단계를 포함하고 있습니다. 보안 그룹에 규칙을 추가하는 방법에 대한 자세한 내용은 보안 그룹에 [규칙 추가](#)를 참조하십시오.

블로그에 사용할 도메인 이름이 아직 없는 경우 Route 53에 도메인 이름을 등록하고 인스턴스의 EIP 주소를 도메인 이름에 연결할 수 있습니다. 자세한 내용은 Amazon Route 53 개발자 안내서의 [Amazon Route 53을 사용하여 도메인 이름 등록](#)을 참조하세요.

설치 WordPress

인스턴스에 연결하고 WordPress 설치 패키지를 다운로드합니다. 인스턴스 연결에 대한 자세한 내용은 [AL2023 인스턴스에 연결](#) 주제를 참조하세요.

1. 다음 명령을 사용하여 이러한 패키지를 다운로드하고 설치합니다.

```
dnf install wget php-mysqlnd httpd php-fpm php-mysqli mariadb105-server php-json
php php-devel -y
```

2. 출력에 다음과 비슷한 문구의 경고가 표시될 수 있습니다(버전은 시간에 따라 다를 수 있음).

```
WARNING:
  A newer release of "Amazon Linux" is available.

  Available Versions:

dnf update --releasever=2023.0.20230202

  Release notes:
  https://aws.amazon.com

  Version 2023.0.20230204:
  Run the following command to update to 2023.0.20230204:

  dnf update --releasever=2023.0.20230204 ... etc
```

가장 좋은 방법은 OS를 up-to-date 최대한 유지하는 것이 좋지만, 환경에 충돌이 없도록 각 버전을 반복해서 살펴보는 것이 좋습니다. 1단계에서 설명한 이전 패키지 설치에 실패할 경우 나열된 최신 릴리스 중 하나로 업데이트하고 다시 시도해야 할 수 있습니다.

3. 명령을 사용하여 최신 WordPress 설치 패키지를 다운로드합니다. wget 다음 명령을 사용할 경우 언제나 최신 릴리스를 다운로드합니다.

```
[ec2-user ~]$ wget https://wordpress.org/latest.tar.gz
```

4. 설치 패키지의 압축 및 아카이빙을 해제합니다. 설치 폴더는 wordpress라는 폴더로 압축 해제됩니다.

```
[ec2-user ~]$ tar -xzf latest.tar.gz
```

WordPress 설치에 사용할 데이터베이스 사용자 및 데이터베이스를 만들려면

WordPress 설치 시 블로그 게시물 및 사용자 의견과 같은 정보를 데이터베이스에 저장해야 합니다. 다음 절차를 통해 블로그의 데이터베이스와 이 데이터베이스에 대해 정보 읽기 및 저장 권한이 있는 사용자를 생성할 수 있습니다.

1. 데이터베이스와 웹 서버를 시작합니다.

```
[ec2-user ~]$ sudo systemctl start mariadb httpd
```

2. 데이터베이스 서버에 root 사용자로 로그인합니다. 메시지가 표시되면 데이터베이스 root 암호를 입력합니다. 이 암호는 사용자의 root 시스템 암호와 다를 수 있으며, 데이터베이스 서버를 보안 설정하지 않은 경우 암호가 비어 있을 수도 있습니다.

데이터베이스 서버를 보안 설정하지 않았다면 반드시 설정하시기 바랍니다. 자세한 내용은 [3단계: 데이터베이스 서버 보안 설정 \(AL2023\)](#) 를 참조하십시오.

```
[ec2-user ~]$ mysql -u root -p
```

3. MySQL 데이터베이스에 대한 사용자 및 암호를 생성합니다. WordPress 설치에서는 이러한 값을 사용하여 MySQL 데이터베이스와 통신합니다. 고유한 사용자 이름과 암호로 해당 부분을 대체하여 다음 명령을 입력합니다.

```
CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY 'your_strong_password';
```

사용자에 대해 보안이 강력한 암호를 생성하시기 바랍니다. 작은따옴표(')는 각 명령을 구별하는 구분자로 기능하기 때문에, 암호에는 사용하지 마세요. 기존 암호를 재사용하지 마세요. 새로 설정한 암호는 안전한 장소에 보관하세요.

4. 데이터베이스를 생성합니다. 데이터베이스에 이를 설명할 수 있는 유의미한 이름을 붙입니다(예: wordpress-db.).

Note

아래 명령에서 데이터베이스 이름을 앞 뒤로 묶는 기호(')를 백틱(backtick)이라고 합니다. 백틱(`) 키는 일반적으로 표준 키보드에서 Tab 키 위에 있습니다. 백틱이 항상 필요하지는

않지만, 이를 통해 데이터베이스 이름에 하이픈(-) 등 허용되지 않는 문자를 사용할 수 있습니다.

```
CREATE DATABASE `wordpress-db`;
```

- 이전에 생성한 WordPress 사용자에게 데이터베이스의 전체 권한을 부여하십시오.

```
GRANT ALL PRIVILEGES ON `wordpress-db`.* TO "wordpress-user"@"localhost";
```

- 데이터베이스 권한을 새로고침(flush)해서 모든 변경사항이 적용되도록 합니다.

```
FLUSH PRIVILEGES;
```

- mysql 클라이언트를 종료합니다.

```
exit
```

wp-config.php 파일 생성 및 편집 방법

WordPress 설치 폴더에는 라는 샘플 구성 파일이 들어 wp-config-sample.php 있습니다. 본 절차에서는 이 파일을 복사하고 특정 구성에 맞도록 편집합니다.

- wp-config-sample.php 파일을 wp-config.php라는 파일에 복사합니다. 이를 통해 새 구성 파일을 생성하고 원본 샘플 파일을 이전 상태 그대로 백업으로 보존할 수 있습니다.

```
[ec2-user ~]$ cp wordpress/wp-config-sample.php wordpress/wp-config.php
```

- wp-config.php 파일을 원하는 텍스트 편집기(nano, vim 등)로 편집하고 설치에 대한 값을 입력합니다. 원하는 텍스트 편집기가 없는 경우 초보자에게는 nano가 적합합니다.

```
[ec2-user ~]$ nano wordpress/wp-config.php
```

- DB_NAME을(를) 정의하는 줄을 찾고 database_name_here을(를) [Step 4의 WordPress 설치에 사용할 데이터베이스 사용자 및 데이터베이스를 만들려면](#)에서 생성한 데이터베이스 이름으로 변경합니다.

```
define('DB_NAME', 'wordpress-db');
```

- b. DB_USER을(를) 정의하는 줄을 찾고 username_here을(를) [Step 3의 WordPress 설치에 사용할 데이터베이스 사용자 및 데이터베이스를 만들려면](#)에서 생성한 데이터베이스 사용자로 변경합니다.

```
define('DB_USER', 'wordpress-user');
```

- c. DB_PASSWORD을(를) 정의하는 줄을 찾고 password_here을(를) [Step 3의 WordPress 설치에 사용할 데이터베이스 사용자 및 데이터베이스를 만들려면](#)에서 생성한 보안성이 강력한 암호로 변경합니다.

```
define('DB_PASSWORD', 'your_strong_password');
```

- d. Authentication Unique Keys and Salts라는 섹션을 검색합니다. 이러한 KEY SALT 값과 값은 WordPress 사용자가 로컬 컴퓨터에 저장하는 브라우저 쿠키에 대한 암호화 계층을 제공합니다. 기본적으로 긴 무작위 값을 추가해서 사이트의 보안성을 강화할 수 있습니다. <https://api.wordpress.org/secret-key/1.1/salt/>을 방문해서 키 값의 세트를 무작위로 생성하고 이를 wp-config.php 파일로 복사해서 붙여 넣을 수 있습니다. PuTTY 터미널로 텍스트를 붙여넣기 하기 위해, PuTTY 터미널 내부에서 텍스트를 붙여넣기하려는 위치에 커서를 놓고 마우스를 오른쪽 클릭합니다.

보안 키에 대한 자세한 내용은 <https://wordpress.org/support/article/editing-wp-config-php/#security-keys>를 참조하십시오.

Note

아래 값은 예시 목적만을 위한 것입니다. 설치 시 이 값을 사용하지는 마세요.

```
define('AUTH_KEY',          ' #U$$+[RXN8:b^-L 0(WU_+ c+WFkI~c]o)-bHw+)/
Aj[wTwSiZ<Qb[mghEXcRh-');
define('SECURE_AUTH_KEY',  'Zsz._P=l/|y.Lq)XjlkwS1y5NJ76E6EJ.AV0pCKZZB,*~*r ?
60P$eJT@;+(ndLg');
define('LOGGED_IN_KEY',    'ju}qwre3V*+8f_z0Wf?{LlGsQ]Ye@2Jh^,8x>)Y |;(^[Iw]Pi
+LG#A4R?7N`YB3');
define('NONCE_KEY',        'P(g62HeZxEes|LnI^i=H,[XwK9I&[2s|:~0N]VJM%?:v2v]v+;
+^9eXUahg@::Cj');
define('AUTH_SALT',        'C$DpB4Hj[JK:~{qL`sRvA:{:7yShy(9A@5wg+`JJVb1fk%-
Bx*M4(qc[Qg%JT!h');
```

```
define('SECURE_AUTH_SALT', 'd!uRu#}+q#{f$Z?Z9uFPG.${+S{n~1M&%@~gL>U>NV<zpD-@2-
Es7Q10-bp28EKv'});
define('LOGGED_IN_SALT', ' ;j{00P*owZf)kVD+FVLn-~ >.|Y%Ug4#I^*LVd9QeZ^&XmK|
e(76miC+&W&+^0P/');
define('NONCE_SALT', ' -97r*V/cgxLmp?Zy4zUU4r99QQ_rGs2LTd%P;|
_e1tS)8_B/, .6[=UK<J_y9?JWG');
```

- e. 파일을 저장하고 텍스트 편집기를 종료합니다.

Apache WordPress 문서 루트 아래에 파일을 설치하려면

- 설치 폴더의 압축을 풀고 MySQL 데이터베이스 및 사용자를 만들고 WordPress 구성 파일을 사용자 지정했으므로 이제 설치 파일을 웹 서버 문서 루트에 복사하여 설치를 완료하는 설치 스크립트를 실행할 수 있습니다. 이러한 파일의 위치는 WordPress 블로그를 웹 서버의 실제 루트 (예:) 에서 사용할 수 있는지 또는 루트 아래의 하위 디렉토리나 폴더 (예:my.public.dns.amazonaws.com) 에서 사용할 수 있도록 할지에 따라 달라집니다. my.public.dns.amazonaws.com/blog

- 문서 루트에서 WordPress 실행하려면 다음과 같이 워드프레스 설치 디렉터리의 내용 (디렉터리 자체는 복사하지 않음) 을 복사합니다.

```
[ec2-user ~]$ cp -r wordpress/* /var/www/html/
```

- 문서 루트 아래의 대체 디렉터리에서 WordPress 실행하려면 먼저 해당 디렉터리를 만든 다음 파일을 해당 디렉터리에 복사하세요. 이 예제에서는 디렉터리에서 WordPress 실행됩니다 `blog`.

```
[ec2-user ~]$ mkdir /var/www/html/blog
[ec2-user ~]$ cp -r wordpress/* /var/www/html/blog/
```

Important

다음 프로시저로 즉시 이동하지 않는 경우는 보안상 문제가 발생할 수 있으므로 Apache 웹 서버(httpd)를 중단하세요. 설치를 Apache 문서 루트로 이동하면 WordPress 설치 스크립트가 보호되지 않으며 Apache 웹 서버가 실행 중인 경우 공격자가 블로그에 액세스할 수 있습니다. Apache 웹 서버를 중지하려면 `sudo service httpd stop` 명령을 입력합니다. 다음 절차로 즉시 이동하는 경우는 Apache 웹 서버를 중단시킬 필요가 없습니다.

퍼머링크 사용을 허용하려면 WordPress

WordPress 퍼머링크가 제대로 작동하려면 Apache `.htaccess` 파일을 사용해야 하지만 Amazon Linux에서는 기본적으로 활성화되어 있지 않습니다. 따라서 아래 방법에 따라 Apache 문서 루트에서 모든 재정의의 허용해야 합니다.

1. 자주 사용하는 텍스트 편집기(`httpd.conf` 또는 `nano`)로 `vim` 파일을 엽니다. 원하는 텍스트 편집기가 없는 경우 초보자에게는 `nano`가 적합합니다.

```
[ec2-user ~]$ sudo vim /etc/httpd/conf/httpd.conf
```

2. 다음과 같이 시작하는 영역을 찾습니다. `<Directory "/var/www/html">`

```
<Directory "/var/www/html">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.4/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks

#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   Options FileInfo AuthConfig Limit
#
AllowOverride None

#
# Controls who can get stuff from this server.
#
Require all granted
</Directory>
```

3. 위 영역에서 `AllowOverride None` 라인을 `AllowOverride ALL`로 변경합니다.

Note

이 파일에는 AllowOverride 라인이 많기 때문에 <Directory "/var/www/html"> 영역의 라인을 변경할 때는 주의해야 합니다.

```
AllowOverride All
```

4. 파일을 저장하고 텍스트 편집기를 종료합니다.

PHP 그래픽 드로잉 라이브러리를 AL2023 상에 설치하려면

PHP용 GD 라이브러리를 사용하면 이미지를 수정할 수 있습니다. 블로그의 헤더 이미지를 잘라야 하는 경우 이 라이브러리를 설치합니다. 설치하는 버전에는 이 라이브러리의 특정 최소 버전 (예: 버전 8.1) 이 필요할 수 있습니다. phpMyAdmin

다음 명령을 사용하여 PHP 그래픽 드로잉 라이브러리를 AL2023 상에 설치합니다. 예를 들어, LAMP 스택 설치의 일부로 소스에서 php8.1을 설치한 경우 이 명령은 PHP 그래픽 그리기 라이브러리의 버전 8.1을 설치합니다.

```
[ec2-user ~]$ sudo dnf install php-gd
```

설치된 버전을 확인하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo dnf list installed | grep php-gd
```

다음은 예제 출력입니다.

```
php-gd.x86_64                8.1.30-1.amzn2                @amazonlinux
```

Amazon Linux AMI에서 PHP 그래픽 그리기 라이브러리를 설치하려면

PHP용 GD 라이브러리를 사용하면 이미지를 수정할 수 있습니다. 블로그의 헤더 이미지를 잘라야 하는 경우 이 라이브러리를 설치합니다. 설치하는 버전에는 이 라이브러리의 특정 최소 버전 (예: 버전 8.1) 이 필요할 수 있습니다. phpMyAdmin

사용 가능한 버전을 확인하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ dnf list | grep php
```

다음은 PHP 그래픽 그리기 라이브러리(버전 8.1) 출력의 예입니다.

```
php8.1.aarch64                                8.1.7-1.amzn2023.0.1
                                                @amazonlinux
php8.1-cli.aarch64                            8.1.7-1.amzn2023.0.1
                                                @amazonlinux
php8.1-common.aarch64                        8.1.7-1.amzn2023.0.1
                                                @amazonlinux
php8.1-devel.aarch64                          8.1.7-1.amzn2023.0.1
                                                @amazonlinux
php8.1-fpm.aarch64                            8.1.7-1.amzn2023.0.1
                                                @amazonlinux
php8.1-gd.aarch64                             8.1.7-1.amzn2023.0.1
                                                @amazonlinux
```

Amazon Linux AMI에서 특정 버전의 PHP 그래픽 그리기 라이브러리(예: 버전 php8.1)를 설치하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo dnf install -y php8.1-gd
```

Apache 웹 서버에 대한 파일 권한 수정 방법

에서 사용할 수 있는 일부 기능을 사용하려면 Apache 문서 루트에 대한 쓰기 권한이 WordPress 필요합니다 (예: 관리 화면을 통한 미디어 업로드). 이미 적용하지 않았다면 다음의 그룹 멤버십 및 권한 ([LAMP 웹 서버 자습서](#)에서 자세히 설명)을 적용합니다.

1. `/var/www`의 파일 소유권 및 그 콘텐츠를 apache 사용자에게 허용합니다.

```
[ec2-user ~]$ sudo chown -R apache /var/www
```

2. `/var/www` 및 그 콘텐츠의 그룹 소유권을 apache 그룹에 허용합니다.

```
[ec2-user ~]$ sudo chgrp -R apache /var/www
```

3. `/var/www` 및 그 하위 디렉터리의 디렉터리 권한을 변경해서 그룹 쓰기 권한을 추가하고 미래 하위 디렉터리에서 그룹 ID를 설정합니다.

```
[ec2-user ~]$ sudo chmod 2775 /var/www
```

```
[ec2-user ~]$ find /var/www -type d -exec sudo chmod 2775 {} \;
```

4. /var/www 및 그 하위 디렉터리의 파일 권한을 재귀적으로 변경합니다.

```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod 0644 {} \;
```

Note

FTP WordPress 서버로도 사용하려는 경우 여기에 좀 더 허용적인 그룹 설정이 필요합니다. 이 작업을 수행하려면 의 권장 [단계 및 보안 설정을](#) 검토하십시오 WordPress.

5. Apache 웹 서버를 재시작해서 새 그룹 및 권한을 가져옵니다.

```
[ec2-user ~]$ sudo systemctl restart httpd
```

AL2023 WordPress 설치 스크립트를 실행하려면

설치할 준비가 되었습니다 WordPress. 사용하는 명령은 운영 체제에 따라 다릅니다. 이 절차의 명령은 AL2023 버전에서만 사용할 수 있습니다. 다음 절차를 AL2023 AMI와 함께 사용하십시오.

1. systemctl 명령을 사용하여 시스템이 부팅될 때마다 httpd 및 데이터베이스 서비스가 시작되도록 합니다.

```
[ec2-user ~]$ sudo systemctl enable httpd && sudo systemctl enable mariadb
```

2. 데이터베이스 서버가 실행되는지 확인합니다.

```
[ec2-user ~]$ sudo systemctl status mariadb
```

데이터베이스 서비스가 실행 중이지 않은 경우, 이를 시작합니다.

```
[ec2-user ~]$ sudo systemctl start mariadb
```

3. Apache 웹 서버(httpd)가 실행 중인지 확인합니다.

```
[ec2-user ~]$ sudo systemctl status httpd
```

httpd 서비스가 실행 중이지 않은 경우, 이를 시작합니다.

```
[ec2-user ~]$ sudo systemctl start httpd
```

4. 웹 브라우저에서 WordPress 블로그의 URL (인스턴스의 퍼블릭 DNS 주소 또는 해당 주소 뒤에 오는 blog 폴더) 을 입력합니다. WordPress 설치 스크립트가 표시되어야 합니다. WordPress 설치에 필요한 정보를 제공하십시오. 설치를 WordPress 선택하여 설치를 완료합니다. 자세한 내용은 [5단계: WordPress 웹 사이트에서 설치 스크립트 실행](#)을 참조하십시오.

AL2023 AMI를 사용하여 WordPress 설치 스크립트를 실행하려면

1. chkconfig 명령을 사용하여 시스템이 부팅될 때마다 httpd 및 데이터베이스 서비스가 시작되도록 합니다.

```
[ec2-user ~]$ sudo chkconfig httpd on && sudo chkconfig mariadb on
```

2. 데이터베이스 서버가 실행되는지 확인합니다.

```
[ec2-user ~]$ sudo service mariadb status
```

데이터베이스 서비스가 실행 중이지 않은 경우, 이를 시작합니다.

```
[ec2-user ~]$ sudo service mariadb start
```

3. Apache 웹 서버(httpd)가 실행 중인지 확인합니다.

```
[ec2-user ~]$ sudo service httpd status
```

httpd 서비스가 실행 중이지 않은 경우, 이를 시작합니다.

```
[ec2-user ~]$ sudo service httpd start
```

4. 웹 브라우저에서 WordPress 블로그의 URL (인스턴스의 퍼블릭 DNS 주소 또는 해당 주소 뒤에 blog 폴더) 을 입력합니다. WordPress 설치 스크립트가 표시되어야 합니다. WordPress 설치에 필요한 정보를 제공하십시오. 설치를 WordPress 선택하여 설치를 완료합니다. 자세한 내용은 [5단계: WordPress 웹 사이트에서 설치 스크립트 실행](#)을 참조하십시오.

다음 단계

WordPress 블로그를 테스트한 후 구성을 업데이트해 보세요.

사용자 지정 도메인 이름 사용

EC2 인스턴스의 EIP 주소와 연결되어 있는 도메인 이름이 있는 경우에는 EC2 퍼블릭 DNS 주소 대신에 해당 이름을 사용하여 블로그를 구성할 수 있습니다. 자세한 내용은 WordPress 웹 [사이트의 사이트 URL 변경을](#) 참조하십시오.

블로그 구성

다른 [테마](#)와 [플러그인](#)을 사용하여 더욱 풍부한 맞춤형 경험을 독자에게 제공하도록 블로그를 구성할 수도 있습니다. 하지만 설치 프로세스가 역효과를 낳아 전체 블로그를 잃는 경우가 발생할 수도 있습니다. 따라서 테마나 플러그인을 설치하기 전에 인스턴스의 백업 Amazon Machine Image(AMI)를 생성하여 설치 중 오류가 발생하더라도 블로그를 복구할 수 있도록 대비하는 것이 좋습니다. 자세한 내용은 Amazon EC2 사용 설명서의 [자체 AMI 생성을](#) 참조하십시오.

용량 증가

WordPress 블로그가 유명해지면서 더 많은 컴퓨팅 파워나 스토리지가 필요하다면 다음 단계를 고려해 보십시오.

- 인스턴스에서 스토리지 공간을 확장합니다. 자세한 내용은 [Amazon EBS 엘라스틱 볼륨을](#) 참조하십시오.
- MySQL 데이터베이스를 [Amazon RDS](#)로 이동하여 이 서비스의 간편한 조정 기능을 이용합니다.

인터넷 트래픽의 네트워크 성능 향상

블로그가 전 세계에 위치한 사용자로부터 트래픽을 유도할 것으로 예상되는 경우 [AWS Global Accelerator](#) 사용을 고려해 보세요. Global Accelerator를 사용하면 사용자의 클라이언트 디바이스와 실행 중인 WordPress 애플리케이션 간의 인터넷 트래픽 성능을 개선하여 지연 시간을 줄일 수 있습니다. AWS Global Accelerator는 [AWS 글로벌 네트워크](#)를 사용하여 트래픽을 클라이언트와 가장 가까운 AWS 지역의 정상 애플리케이션 엔드포인트로 전달합니다.

에 대해 자세히 알아보십시오. WordPress

다음 링크에는 에 대한 자세한 정보가 포함되어 WordPress 있습니다.

- [에 대한 WordPress 자세한 내용은 Codex의 WordPress Codex 도움말 설명서를 참조하십시오.](#)
- 설치 문제 해결에 대한 자세한 내용은 [일반적인 설치](#) 문제를 참조하세요.

- WordPress [블로그의 보안을 강화하는 방법에 대한 자세한 내용은 강화를 참조하십시오. WordPress](#)
- WordPress 블로그 up-to-date 유지에 대한 자세한 내용은 [업데이트를 WordPress](#) 참조하십시오.

도움말! 내 퍼블릭 DNS 이름이 변경되어 블로그를 사용할 수 없습니다.

EC2 인스턴스의 퍼블릭 DNS 주소를 사용하여 WordPress 설치가 자동으로 구성됩니다. 이때 인스턴스를 중단했다가 다시 시작하면 퍼블릭 DNS 주소가 바뀌어(탄력적 IP 주소와 연결되어 있지 않은 경우) 블로그를 더 이상 사용할 수 없게 됩니다. 리소스를 참조해야 할 주소가 더 이상 존재하지 않거나 다른 EC2 인스턴스에 할당되었기 때문입니다. 문제에 대한 자세한 설명과 몇 가지 가능한 해결 방법은 <https://wordpress.org/support/article/changing-the-site-url/>에 요약되어 있습니다.

WordPress 설치 중에 이런 일이 발생한 경우 wp-cli 명령줄 인터페이스를 사용하는 아래 절차에 따라 블로그를 복구할 수 있습니다. WordPress

를 사용하여 WordPress 사이트 URL을 변경하려면 wp-cli

1. SSH를 통해 EC2 인스턴스에 연결합니다.
2. 인스턴스의 이전 사이트 URL과 새로운 사이트 URL을 기록합니다. 이전 사이트 URL은 설치 WordPress 당시 EC2 인스턴스의 퍼블릭 DNS 이름일 가능성이 높습니다. 그리고 새로운 사이트 URL은 EC2 인스턴스의 현재 퍼블릭 DNS 이름입니다. 이전 사이트 URL을 잘 모르더라도 아래와 같이 curl 명령을 사용하여 찾을 수 있습니다.

```
[ec2-user ~]$ curl localhost | grep wp-content
```

명령을 실행하여 출력되는 화면에서 이전 퍼블릭 DNS 이름의 참조를 확인해야 합니다. 출력 화면은 다음과 같습니다(빨간색의 이전 사이트 URL).

```
<script type='text/javascript' src='http://ec2-52-8-139-223.us-west-1.compute.amazonaws.com/wp-content/themes/twentyfifteen/js/functions.js?ver=20150330'></script>
```

3. 다음 명령으로 wp-cli를 다운로드합니다.

```
[ec2-user ~]$ curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar
```

4. WordPress 설치 시 이전 사이트 URL을 검색하고 다음 명령으로 바꾸십시오. 기존 사이트 URL과 새 사이트 URL을 EC2 인스턴스와 WordPress 설치 경로 (일반적으로 /var/www/html 또는 /var/www/html/blog)로 대체합니다.

```
[ec2-user ~]$ php wp-cli.phar search-replace 'old_site_url' 'new_site_url' --path=/  
path/to/wordpress/installation --skip-columns=guid
```

5. 웹 브라우저에 WordPress 블로그의 새 사이트 URL을 입력하여 사이트가 다시 제대로 작동하는지 확인합니다. 그렇지 않은 경우 자세한 내용은 [사이트 URL 변경](#) 및 [일반적인 설치 문제를](#) 참조하십시오.

Amazon EC2 외부에서 Amazon Linux 2023 사용

Amazon Linux 2023 컨테이너 이미지는 호환되는 컨테이너 런타임 환경에서 실행됩니다. 컨테이너에서 Amazon Linux 2023을 사용하는 방법에 대한 자세한 내용은 [AL2023 컨테이너](#) 섹션을 참조하세요.

Amazon Linux 2023(AL2023)은 Amazon EC2에서 실행되지 않고 가상 게스트로 실행할 수도 있습니다. 현재 KVM (qcow2), VMware () 및 Hyper-V (OVA) 이미지를 사용할 수 있습니다. vhd/x

Note

Amazon Linux 2023 이미지 구성은 Amazon Linux 2와 다릅니다.

[온프레미스에서 가상 머신으로 Amazon Linux 2 실행](#)하는 경우 AL2023 호환되도록 구성을 수정해야 합니다.

KVM, VMware 및 Hyper-V와 함께 사용할 Amazon Linux 2023 이미지를 다운로드하십시오.

[KVM, VMware 및 Hyper-V와 함께 사용할 아마존 리눅스 2023 디스크 이미지는 `cdn.amazonlinux.com`에서 다운로드할 수 있습니다.](#)

Amazon EC2가 아닌 가상 환경에서 사용하기 위해 지원되는 Amazon Linux 2023 구성

이 섹션에서는 KVM, VMware 또는 Hyper-V와 같은 비 Amazon EC2 가상화 환경에서 Amazon Linux 2023을 실행하기 위한 요구 사항을 다룹니다.

이 기본 [AL2023 시스템 요구 사항](#)은 Amazon EC2 가상 환경이 아닌 환경을 지원합니다. 다음 항목의 각 하이퍼바이저 환경은 지원되는 디바이스 모델 목록에 자세히 설명되어 있습니다.

KVM, VMware 및 Hyper-V는 다양한 구성 옵션을 제공하므로 보안, 성능 및 안정성 요구 사항에 맞게 구성하려면 주의를 기울여야 합니다. 자세한 내용은 하이퍼바이저가 제공하는 설명서를 참조하세요.

주제

- [KVM에서 AL2023 실행을 위한 요구 사항](#)
- [에서 AL2023 실행을 위한 요구 사항 VMware](#)
- [Hyper-V에서 아마존 리눅스 2023을 실행하기 위한 요구 사항](#)

KVM에서 AL2023 실행을 위한 요구 사항

이 섹션에서는 KVM에서 AL2023 실행을 위한 요구 사항을 설명합니다. AL2023 KVM 이미지는 aarch64 및 x86-64 아키텍처에서 모두 사용할 수 있습니다. 이러한 요구 사항은 KVM 이미지의 [AL2023 시스템 요구 사항](#) 기본에 추가됩니다.

주제

- [KVM에서 AL2023 실행을 위한 KVM 호스트 요구 사항](#)
- [KVM 기반 AL2023 기기 지원](#)
- [KVM의 부팅 모드 \(UEFI 및 BIOS\) AL2023 지원](#)
- [KVM에서 AL2023 실행 시 제한 사항](#)

KVM에서 AL2023 실행을 위한 KVM 호스트 요구 사항

KVM 이미지는 현재 이 Ubuntu 버전에서 제공하는 qemu 버전이 6.2+dfsg-2ubuntu6.15 있는 Ubuntu 22.04.3 LTS를 실행하는 호스트에서 시스템 유형과 시스템 유형을 사용하여 검증되었습니다. q35 x86-64 virt aarch64

KVM 기반 AL2023 기기 지원

AL2023 KVM 이미지(aarch64 및 x86-64 모두)로 테스트된 **qemu** 장치 모델은 다음과 같습니다.

- virtio-blk (virtio 블록 디바이스)
- virtio-scsi (디스크 장치가 있는 virtio SCSI 컨트롤러)
- virtio-net (virtio 네트워크 디바이스)
- ahci (가상 CD-ROM 드라이브에만 사용 가능)
- usb-storage (over xhci)

AL2023 KVM 이미지 검증에서 사용할 수 있지만 많이 사용되지는 않는 추가 **qemu** 장치 모델은 다음과 같습니다.

- x86-64 VGA (qemu VGA)만 해당
- virtio-rng (가상 난수 생성기)
- 레거시 AT 키보드 및 PS/2 마우스 디바이스
- 레거시 시리얼 디바이스

KVM의 부팅 모드 (UEFI 및 BIOS) AL2023 지원

x86-64 이미지는 레거시 BIOS 및 UEFI 부팅 모드에서 테스트했습니다. aarch64 이미지는 UEFI 부팅 모드에서 테스트했습니다.

Note

기본적으로 UEFI 부팅 모드를 사용할 때 일부 가상 컴퓨터 관리자는 보안 부팅을 지원하는 Microsoft 보안 부팅 키를 VM에 제공합니다. 이 구성으로는 AL2023 부팅은 불가능합니다. AL2023 부트 로더는 Microsoft의 서명을 받지 않았으므로 가상 머신은 UEFI 키 없이 프로비저닝하거나 보안 부팅을 위한 AL2023 키를 사용하여 프로비저닝해야 합니다.

Important

KVM 이미지에 대한 보안 부팅 지원은 아직 검증되지 않았습니다.

KVM에서 AL2023 실행 시 제한 사항

KVM에서 AL2023 실행에는 몇 가지 알려진 제한 사항이 있습니다.

Note

나열된 지원되지 않는 기능 중 일부를 구현하는 코드가 AL2023 내에 존재하여 제대로 작동할 수 있습니다. 지원되지 않는 기능 목록은 현재 작업에 의존해야 할 기능과 Amazon Linux 팀이 향후 업데이트의 일부로 사용할 수 있는 작업에 대해 정보에 입각한 결정을 내릴 수 있도록 하기 위한 것입니다.

KVM에서 AL2023 실행과 관련된 알려진 제한 사항

- KVM 게스트 에이전트는 현재 패키지로 제공되거나 지원되지 않습니다.
- CPU, 메모리 또는 기타 디바이스 유형의 핫 플러그 및 언플러그는 지원되지 않습니다.
- VM 하이버네이션은 지원되지 않습니다.
- VM 마이그레이션은 지원되지 않습니다.
- PCI 패스스루 또는 USB 패스스루와 같은 디바이스 패스스루는 지원되지 않습니다.

에서 AL2023 실행을 위한 요구 사항 VMware

이 섹션에서는 AL2023 실행을 위한 요구 사항을 설명합니다. VMware. AL2023 VMware 이미지는 해당 x86-64 아키텍처에서만 사용할 수 있습니다. VMware의 이미지는 사용할 수 없거나 aarch64 지원되지 않습니다. 이러한 요구 [AL2023 시스템 요구 사항](#) 사항은 VMware 이미지의 기본 요건에 추가됩니다.

주제

- [VMware AL2023 실행을 위한 호스트 요구 사항 VMware](#)
- [AL2023 지원 장치 지원 커짐 VMware](#)
- [부팅 모드 \(UEFI 및 BIOS\) AL2023 지원 커짐 VMware](#)
- [AL2023 실행 시 제한 사항 VMware](#)

VMware AL2023 실행을 위한 호스트 요구 사항 VMware

AL2023 VMware OVA 이미지는 현재 다음 항목에 대해 검증되었습니다.

- VMware 인텔 (R) 제온 (R) 플래티넘 8124M 프로세서를 사용하는 호스트에서 실행되는 워크스테이션 17.5.0
- VMware 인텔 (R) 제온 (R) 플래티넘 8275CL 프로세서를 사용하는 vSphere 8.0

AL2023 VMware OVA 이미지는 시스템 하드웨어 버전 13을 지정합니다.

VMware 컴퓨터 하드웨어 버전 13은 다음 기관에서 지원됩니다.

- ESXi 6.5 이상
- VMware 워크스테이션 14 이상

AL2023 지원 장치 지원 커짐 VMware

다음 VMware 장치 모델은 AL2023 VMware OVA 이미지와 함께 사용하도록 테스트되었습니다 (**x86-64 전용**).

- vmw_pvscsi (VMware 반가상화 컨트롤러 SCSI)
- vmxnet3 (VMware 반가상화 네트워크 장치)
- ata_piix (가상 CD-ROM 드라이브에만 사용 가능한 레거시 IDE)

AL2023 VMware 이미지 검증이 지원되지만 많이 사용되지는 않은 추가 VMware 장치 모델:

- vmw_vmci 및 관련 vsock 인터페이스 (VMware 게스트 에이전트를 위한 가상 소켓 전송)
- vmw_balloon 메모리 벌룬 디바이스
- VMware SVGA 컨트롤러
- 레거시 AT 키보드 및 PS/2 마우스 디바이스

VMware 게스트 에이전트 패키지 (open-vm-tools) 는 AL2023 VMware OVA 이미지에서 기본적으로 사용 및 설치됩니다.

부팅 모드 (UEFI 및 BIOS) AL2023 지원 커짐 VMware

2023.3.20231211 릴리스부터 AL2023 VMware OVA 이미지는 레거시 모드와 부팅 모드 모두에서 검증되었습니다. BIOS UEFI OVA 기본 구성은 여전히 BIOS 레거시이지만 사용자가 변경할 수 있습니다.

Important

보안 부팅 지원이 필요하며 UEFI, 아직 AL2023 실행 시 유효성이 검증되지 않았습니다. VMware

AL2023 실행 시 제한 사항 VMware

AL2023 실행에는 몇 가지 알려진 제한 사항이 있습니다 VMware.

Note

나열되어 있는 지원되지 않는 기능 중 일부를 구현하는 코드가 AL2023 내에 존재하며 올바르게 작동합니다. 지원되지 않는 기능 목록을 통해 고객은 현재 작업 시 사용 가능한 기능과 향후 Amazon Linux 팀이 업데이트할 기능을 확인할 수 있습니다.

AL2023 실행 시 발생하는 알려진 제한 사항 VMware

- UEFI 현재 AL2023 켜진 상태에서는 보안 부팅이 검증되지 않았습니다. VMware
- CPU, 메모리 또는 기타 디바이스 유형의 핫 플러그 및 언플러그는 지원되지 않습니다.

- VM 하이버네이션은 지원되지 않습니다.
- VM 마이그레이션은 지원되지 않습니다.
- PCI 패스스루 또는 USB 패스스루와 같은 디바이스 패스스루는 지원되지 않습니다.

Hyper-V에서 아마존 리눅스 2023을 실행하기 위한 요구 사항

이 섹션에서는 Hyper-V에서 Amazon Linux 2023을 실행하기 위한 요구 사항을 다룹니다. AL2023 Hyper-V 이미지는 해당 아키텍처에서만 사용할 수 있습니다. x86-64 에 대한 Hyper-V 이미지는 aarch64 현재 사용할 수 없거나 지원되지 않습니다.

이 섹션에서는 Hyper-V 이미지에 [AL2023 시스템 요구 사항](#) 대한 기본 요구 사항 외에 추가 요구 사항을 다룹니다.

주제

- [Hyper-V에서 Amazon Linux 2023을 실행하기 위한 Hyper-V 호스트 요구 사항](#)
- [Hyper-V 기반 아마존 리눅스 2023에 대한 장치 지원](#)
- [Hyper-V에서 아마존 리눅스 2023을 실행하는 데 따른 제한](#)

Hyper-V에서 Amazon Linux 2023을 실행하기 위한 Hyper-V 호스트 요구 사항

Hyper-V용 Amazon Linux 2023의 주요 자격은 EC2 c5.meta1 인스턴스에서 실행되는 윈도우 서버 2022에서 이루어집니다.

Hyper-V 기반 아마존 리눅스 2023에 대한 장치 지원

Amazon Linux 2023은 다음과 같은 가상화된 하드웨어 세트를 갖춘 1세대 및 2세대 Hyper-V 가상 머신 모두에서 테스트되었습니다.

- 1세대 (레거시 BIOS 부트) VM
- 2세대 (UEFI 부팅 - 보안 부팅 없음) VM
- 다음 장치 모델은 AL2023 Hyper-V 이미지와 함께 사용하도록 테스트되었습니다.
 - 루트 `hv_storvsc` 디스크용 Hyper-V 가상 스토리지 및 2세대 VM의 에뮬레이션된 CD-ROM 드라이브
 - 1세대 가상 머신의 가상 CD-ROM `ata_piix` 드라이브용 에뮬레이트된 PIIX IDE
 - Hyper-V 가상 이더넷 `hv_netvsc`

- 다음 장치 모델은 활성화되었지만 테스트를 거치지 않았습니다.
 - 1세대 VM의 레거시 VGA 텍스트 모드
 - 2세대 VM의 UEFI 펌웨어 기반 프레임 *simplifiedrmfb* 버퍼
 - 하이퍼-V 벌룬 hv_balloon
 - 하이퍼-V 벌룬 hv_balloon
 - 하이퍼-V HID/마우스 hid_hyperv
- 현재 AL2023 모드에서는 다음 장치 모드가 활성화되지 않습니다.
 - 하이퍼-V PCI 패스스루
 - 하이퍼-V DRM 그래픽

Important

2세대 가상 머신의 경우, 보안 부팅은 지원되지 않으므로 Amazon Linux 2023을 성공적으로 부팅하려면 가상 머신을 시작하기 전에 보안 부팅을 비활성화해야 합니다. Hyper-V는 현재 Microsoft 자체 키로 소프트웨어 구성 요소가 서명된 보안 부팅만 지원하는 반면, Amazon Linux 부트로더는 Amazon 개인 키로 서명됩니다. Hyper-V는 현재 타사 키 가져오기를 지원하지 않습니다.

Hyper-V에서 아마존 리눅스 2023을 실행하는 데 따른 제한

다음은 Hyper-V에서 Amazon Linux 2023을 실행할 때 알려진 몇 가지 제한 사항입니다.

Note

나열되어 있는 지원되지 않는 기능 중 일부를 구현하는 코드가 AL2023 내에 존재하며 올바르게 작동합니다. 지원되지 않는 기능 목록을 통해 고객은 현재 작업 시 사용 가능한 기능과 향후 Amazon Linux 팀이 업데이트할 기능을 확인할 수 있습니다.

Hyper-V에서 AL2023 실행에 대한 알려진 제한 사항

- UEFI 보안 부팅 모드는 현재 Hyper-V의 AL2023 버전에서 지원되거나 작동하지 않습니다.
- CPU, 메모리 또는 기타 디바이스 유형의 핫 플러그 및 언플러그는 지원되지 않습니다.
- 가상 머신(VM) 하이버네이션은 지원되지 않습니다.

- 가상 머신(VM) 마이그레이션은 지원되지 않습니다.
- PCI 패스스루 또는 USB 패스스루와 같은 디바이스 패스스루는 지원되지 않습니다.

Amazon EC2 외부에서 사용 시 Amazon Linux 2023 설정 및 **cloud-init** 구성

이 섹션에서는 KVM, VMware 또는 Hyper-V와 같이 Amazon EC2에서 직접 실행되지 않을 때 Amazon Linux 2023 가상 머신을 설정하고 구성하는 방법을 설명합니다.

기본적으로 Amazon Linux 2023 가상 머신 이미지에 사용자 암호 또는 ssh 키를 쓸 수 없으며 처음 검색된 네트워크 인터페이스에서 DHCP를 통해 네트워크 구성을 가져옵니다. 즉, 추가 구성 없이는 생성된 가상 머신에 연결할 방법이 없습니다.

따라서 가상 머신에 어떤 형태로든 구성을 제공해야 합니다. Amazon Linux에서 이를 수행하는 표준 메커니즘은 cloud-init 데이터 소스를 통해 이루어집니다.

Amazon Linux 2023은 다음 데이터 소스로 검증되었습니다.

NoCloud

cloud-init 구성 파일이 포함된 시드 ISO9660 이미지가 들어 있는 가상 CD-ROM을 통해 온프레미스 이미지를 구성하는 기존 방법입니다.

VMware

Amazon Linux 2023은 VMware guestinfo.userdata 및 guestinfo.metadata의 VMware 전용 데이터 소스를 통해 vSphere에서 실행되는 VMware 이미지를 구성할 수 있도록 추가 지원합니다.

Note

데이터 소스 구성은 Amazon Linux 2와 다를 수 있습니다. 구체적으로 설명하면, Amazon Linux 2023에서 systemd-networkd를 구성할 수 있으며 이를 위해 [cloud-init 네트워크 구성 설명서](#)에서 설명한 cloud-init “네트워킹 구성 버전 2”를 따라야 합니다.

Amazon Linux 2023에 포함된 cloud-init 버전의 cloud-init 구성 메커니즘 전체 설명서는 [업스트림 cloud-init 설명서](#)에서 있습니다.

NoCloud (**seed.iso**) KVM 및 VMware의 아마존 리눅스 2023용 **cloud-init** 구성

이 섹션에서는 `seed.iso` 이미지를 생성하고 사용하여 KVM 또는 에서 실행되는 Amazon Linux 2023을 구성하는 방법을 다룹니다. VMware, KVM 및 VMware 환경에는 [Amazon EC2 인스턴스 메타 데이터 서비스 \(IMDS\)](#)가 없으므로 Amazon Linux 2023을 구성하는 다른 방법이 필요하며 `seed.iso` 이미지를 제공하는 것도 이러한 방법 중 하나입니다.

`seed.iso` 부팅 이미지에는 네트워크 구성, 호스트 이름, 사용자 데이터 등 새 VM을 부팅하고 구성하는 데 필요한 기본 정보가 포함되어 있습니다.

Note

`seed.iso` 이미지에는 VM을 부팅하는 데 필요한 구성 정보만 있습니다. Amazon Linux 2023 운영 파일은 없습니다.

`seed.iso` 부팅 이미지를 생성하려면 최소 구성 파일 두 개가 필요하며 세 개가 필요한 경우도 있습니다.

meta-data

이 파일에는 일반적으로 가상 머신의 호스트 이름이 있습니다.

user-data

이 파일로 사용자 계정, 암호, ssh 키 페어 및/또는 액세스 메커니즘을 구성합니다. 기본적으로 Amazon Linux 2 KVM 및 VMware 이미지로 `ec2-user` 사용자 계정을 생성합니다. `user-data` 구성 파일을 사용하여 기본 사용자 계정의 암호 및/또는 키를 설정합니다.

network-config (선택 사항)

이 파일은 일반적으로 가상 시스템의 네트워크 구성을 제공하여 기본 구성을 재정의합니다. 기본 구성은 사용 가능한 첫 번째 네트워크 인터페이스의 DHCP를 사용하는 것입니다.

seed.iso 디스크 이미지 생성

1. Linux 또는 macOS 컴퓨터에서 `seedconfig`라는 이름의 새 폴더를 만들어 이 폴더로 이동합니다.

Note

Windows 또는 다른 운영 체제에서도 가능하지만 `seed.iso` 이미지를 생성하려면 `mkisofs`와 동일한 도구가 있어야 합니다.

2. meta-data 구성 파일을 생성합니다.

- a. meta-data라는 이름의 새로운 파일을 만듭니다.
- b. 원하는 편집기에서 meta-data 파일을 열어 다음 구문을 추가합니다. 여기서 `vm-hostname`을 VM 호스트 이름으로 바꾸어야 합니다.

```
local-hostname: vm-hostname
```

- c. meta-data 구성 파일을 저장하고 닫습니다.

3. user-data 구성 파일을 생성합니다.

- a. user-data라는 이름의 새로운 파일을 만듭니다.
- b. 원하는 편집기에서 user-data 파일을 열고 필요에 따라 변경한 후 추가합니다.

```
#cloud-config
#vim:syntax=yaml
users:
# A user by the name 'ec2-user' is created in the image by default.
- default
- name: ec2-user
ssh_authorized_keys:
- ssh-rsa ssh-key
# In the above line, replace ssh key with the content of your ssh public key.
```

- c. 선택적으로 user-data 구성 파일에 사용자 계정을 더 추가할 수 있습니다.

추가 사용자 계정, 계정 액세스 메커니즘, 암호 및 키 페어를 지정할 수 있습니다. 지원하는 지시어 대한 자세한 내용은 [업스트림 cloud-init 설명서](#)를 참조하세요.

- d. user-data 구성 파일을 저장하고 닫습니다.

4. (선택 사항) network-config 구성 파일 생성

- a. network-config라는 명칭의 새로운 파일을 만듭니다.

- b. 원하는 편집기에서 `network-config` 파일을 열고 다음 구문을 추가합니다. 여기서 다양한 IP 주소를 설정에 적합한 주소로 바꾸어야 합니다.

```
version: 2
ethernets:
  enp1s0:
    addresses:
      - 192.168.122.161/24
    gateway4: 192.168.122.1
    nameservers:
      addresses: 192.168.122.1
```

Note

cloud-init 네트워크 구성은 VM 구성에 따라 변경될 수 있는 인터페이스 이름을 지정하는 대신 인터페이스의 MAC 주소와 일치시키는 메커니즘을 제공합니다. 네트워크 구성을 위한 이 cloud-init 기능(및 기타 기능)은 [업스트림 cloud-init 네트워크 구성 버전2 설명서](#)에 자세히 설명되어 있습니다.

- c. `network-config` 구성 파일을 저장하고 닫습니다.
5. 이전 단계에서 만든 `meta-data`, `user-data` 및 `network-config`(옵션) 구성 파일로 `seed.iso` 디스크 이미지를 생성합니다.

`seed.iso` 디스크 이미지를 만들 운영 체제에 따라 다음 중 하나를 수행합니다.

- Linux 시스템에서 **mkisofs** 또는 **genisoimage** 같은 도구를 사용하여 `seed.iso` 파일을 생성합니다. `seedconfig` 폴더로 이동하여 다음 명령을 실행합니다.

```
$ mkisofs -output seed.iso -volid cidata -joliet -rock user-data meta-data
```

- `network-config`를 사용하는 경우 **mkisofs** 간접 호출 시 포함시키세요.

```
$ mkisofs -output seed.iso -volid cidata -joliet -rock user-data meta-data
network-config
```

- macOS 시스템에서 **hdiutil** 같은 도구를 사용하여 완성된 `seed.iso` 파일을 만들 수 있습니다. **hdiutil**은 파일 목록이 아닌 경로 이름을 사용하므로 `network-config` 구성 파일의 생성 여부에 관계없이 동일한 간접 호출을 사용할 수 있습니다.

```
$ hdiutil makehybrid -o seed.iso -hfs -joliet -iso -default-volume-name cidata
seedconfig/
```

- 이제 생성된 `seed.iso` 파일을 가상 CD-ROM 드라이브를 통해 새 Amazon Linux 2023 가상 머신에 첨부하면 `cloud-init`가 처음 부팅할 곳을 찾고 시스템에 구성에 적용됩니다.

VMwareAL2023 게스트 정보 **cloud-init** 구성 커짐 VMware

VMware환경에는 [Amazon EC2 인스턴스 메타 데이터 서비스 \(IMDS\)](#) 가 없으므로 AL2023 구성을 위한 다른 방법이 필요합니다. 이 섹션에서는 VMware vSphere에서 사용할 수 있는 `seed.iso` 가상 CD-ROM 드라이브에 대한 대체 구성 메커니즘을 사용하는 방법에 대해 설명합니다.

이 구성 방법은 VMware `extraconfig` 메커니즘을 사용하여 구성 데이터를 제공합니다. `cloud-init` 다음 각 키에 대해 해당 **keyname.encoding** 속성을 제공해야 합니다.

다음 키를 VMware `extraconfig` 메커니즘에 제공할 수 있습니다.

guestinfo.metadata

cloud-init 메타데이터가 포함된 JSON 또는 YAML

guestinfo.userdata

cloud-config 형식의 cloud-init 사용자 데이터가 포함된 YAML 설명서.

guestinfo.vendordata (선택 사항)

YAMLcloud-init공급업체 데이터 포함

해당 인코딩 속성(`guestinfo.metadata.encoding`, `guestinfo.userdata.encoding` 및 `guestinfo.vendordata.encoding`)은 다음을 포함합니다.

base64

속성의 콘텐츠는 base64로 인코딩되어 있습니다.

gzip+base64

base64로 인코딩된 후 속성 콘텐츠가 gzip으로 압축됩니다.

Note

이 `seed.iso` 메서드는 별도의 (선택 사항) `network-config` 구성 파일을 지원합니다. VMwareguestinfo네트워킹 구성이 제공되는 방식이 다릅니다. 추가 정보는 다음 섹션에 제공됩니다.

명시적 네트워크 구성이 필요한 경우 다음 두 가지 YAML 또는 JSON 속성의 형태로 metadata에 포함되어야 합니다.

network

JSON 또는 YAML 형식으로 인코딩된 네트워크 구성을 포함합니다.

network.encoding

위 네트워크 구성 데이터의 인코딩을 포함합니다. `cloud-init`에서 지원되는 인코딩은 `guestinfo` 데이터에 지원되는 것과 동일한 `base64` 및 `gzip+base64`입니다.

Example VMwarevSphere **govc** CLI 도구를 사용하여 다음과 같은 구성을 전달하는 방법 **guestinfo**

1. 에 설명된 대로 `meta-data`, `user-data`, 및 선택적 `network-config` 구성 파일을 준비하십시오. [NoCloud \(seed.iso\) KVM 및 VMware의 아마존 리눅스 2023용 cloud-init 구성](#)
2. 구성 파일을 에서 사용할 수 있는 형식으로 VMware `guestinfo` 변환합니다.

```
# 'meta-data', `user-data` and `network-config` are the configuration
# files in the same format that would be used by a NoCloud (seed.iso)
# data source, read-them and convert them to VMware guestinfo
#
# The VM_NAME variable is assumed to be set to the name of the VM
# It is assumed that the necessary govc environment (credentials etc...) are
# already set

metadata=$(cat "meta-data")
userdata=$(cat "user-data")
if [ -e "network-config" ] ; then
    # We need to embed the network config inside the meta-data
    netconf=$(base64 -w0 "network-config")
    metadata=$(printf "%s\nnetwork: %s\nnetwork.encoding: base64" "$metadata"
"$netconf")
fi
```

```

metadata=$(base64 -w0 <<< "$metadata")
govc vm.change -vm "$VM_NAME" \
  -e guestinfo.metadata="$metadata" \
  -e guestinfo.metadata.encoding="base64"
userdata=$(base64 -w0 <<< "$userdata")
govc vm.change -vm "$VM_NAME" \
  -e guestinfo.userdata="$userdata" \
  -e guestinfo.userdata.encoding="base64"

```

아마존 리눅스 2023 표준 AMI에 설치된 패키지와 AL2023 KVM 이미지 비교

AL2023 표준 AMI에 있는 RPM을 AL2023 KVM 이미지에 있는 RPM과 비교한 것입니다.

패키지	AMI	KVM
acl	2.3.1	2.3.1
acpid	2.0.32	
alternatives	1.15	1.15
amazon-chrony-config	4.3	
amazon-ec2-net-utils	2.4.1	
amazon-linux-onprem		1.2
amazon-linux-repo-cdn		2023.4.20240513
amazon-linux-repo-s3	2023.4.20240513	
amazon-linux-sb-keys	2023.1	2023.1
amazon-onprem-netw ork		1.2
amazon-rpm-config	228	228

패키지	AMI	KVM
amazon-ssm-agent	3.3.380.0	3.3.380.0
at	3.1.23	3.1.23
attr	2.5.1	2.5.1
audit	3.0.6	3.0.6
audit-libs	3.0.6	3.0.6
aws-cfn-bootstrap	2.0	
awscli-2	2.15.30	2.15.30
basesystem	11	11
bash	5.2.15	5.2.15
bash-completion	2.11	2.11
bc	1.07.1	1.07.1
bind-libs	9.16.48	9.16.48
bind-license	9.16.48	9.16.48
bind-utils	9.16.48	9.16.48
binutils	2.39	2.39
boost-filesystem	1.75.0	1.75.0
boost-system	1.75.0	1.75.0
boost-thread	1.75.0	1.75.0
bzip2	1.0.8	1.0.8
bzip2-libs	1.0.8	1.0.8

패키지	AMI	KVM
ca-certificates	2023.2.64	2023.2.64
c-ares	1.19.0	
checkpolicy	3.4	3.4
chkconfig	1.15	1.15
chrony	4.3	4.3
cloud-init	22.2.2	22.2.2
cloud-init-cfg-ec2	22.2.2	
cloud-init-cfg-onpre m		22.2.2
cloud-utils-growpart	0.31	0.31
coreutils	8.32	8.32
coreutils-common	8.32	8.32
cpio	2.13	2.13
cracklib	2.9.6	2.9.6
cracklib-dicts	2.9.6	2.9.6
crontabs	1.11	1.11
crypto-policies	20220428	20220428
crypto-policies-sc ripts	20220428	20220428
cryptsetup	2.6.1	2.6.1
cryptsetup-libs	2.6.1	2.6.1

패키지	AMI	KVM
curl-minimal	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27
cyrus-sasl-plain	2.1.27	2.1.27
dbus	1.12.28	1.12.28
dbus-broker	32	32
dbus-common	1.12.28	1.12.28
dbus-libs	1.12.28	1.12.28
device-mapper	1.02.185	1.02.185
device-mapper-libs	1.02.185	1.02.185
diffutils	3.8	3.8
dnf	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2
dnf-plugins-core	4.3.0	4.3.0
dnf-plugin-support-info	1.2	1.2
dnf-utils	4.3.0	4.3.0
dosfstools	4.2	4.2
dracut	055	055
dracut-config-ec2	3.0	

패키지	AMI	KVM
dracut-config-generic	055	055
dwz	0.14	0.14
dyninst	10.2.1	10.2.1
e2fsprogs	1.46.5	1.46.5
e2fsprogs-libs	1.46.5	1.46.5
ec2-hibinit-agent	1.0.8	
ec2-instance-connect	1.1	
ec2-instance-connect-selinux	1.1	
ec2-utils	2.2.0	
ed	1.14.2	1.14.2
efi-filesystem	5	5
efi-srpm-macros	5	5
efivar	38	38
efivar-libs	38	38
elfutils-debuginfod-client	0.188	0.188
elfutils-default-yama-scope	0.188	0.188
elfutils-libelf	0.188	0.188
elfutils-libs	0.188	0.188

패키지	AMI	KVM
ethtool	5.15	5.15
expat	2.5.0	2.5.0
file	5.39	5.39
file-libs	5.39	5.39
filesystem	3.14	3.14
findutils	4.8.0	4.8.0
fonts-srpm-macros	2.0.5	2.0.5
fstrm	0.6.1	0.6.1
fuse-libs	2.9.9	2.9.9
gawk	5.1.0	5.1.0
gdbm-libs	1.19	1.19
gdisk	1.0.8	1.0.8
gettext	0.21	0.21
gettext-libs	0.21	0.21
ghc-srpm-macros	1.5.0	1.5.0
glib2	2.74.7	2.74.7
glibc	2.34	2.34
glibc-all-langpacks	2.34	2.34
glibc-common	2.34	2.34
glibc-gconv-extra	2.34	2.34

패키지	AMI	KVM
glibc-locale-source	2.34	2.34
gmp	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7
gnutls	3.8.0	3.8.0
go-srpm-macros	3.2.0	3.2.0
gpgme	1.15.1	1.15.1
gpm-libs	1.20.7	1.20.7
grep	3.8	3.8
groff-base	1.22.4	1.22.4
grub2-common	2.06	2.06
grub2-efi-aa64-ec2	2.06 (aarch64)	2.06 (아키텍처64)
grub2-efi-x64-ec2	2.06 (x86_64)	2.06 (x86_64)
grub2-pc		2.06 (x86_64)
grub2-pc-modules	2.06	2.06 (노아크)
grub2-tools	2.06	2.06
grub2-tools-minimal	2.06	2.06
grubby	8.40	8.40
gssproxy	0.8.4	0.8.4
gzip	1.12	1.12
hostname	3.23	3.23

패키지	AMI	KVM
hunspell	1.7.0	1.7.0
hunspell-en	0.20140811.1	0.20140811.1
hunspell-en-GB	0.20140811.1	0.20140811.1
hunspell-en-US	0.20140811.1	0.20140811.1
hunspell-filesystem	1.7.0	1.7.0
hwdata	0.353	0.353
info	6.7	6.7
inih	49	49
initscripts	10.09	10.09
iproute	5.10.0	5.10.0
iputils	20210202	20210202
irqbalance	1.9.0	1.9.0
jansson	2.14	2.14
jitterentropy	3.4.1	3.4.1
jq	1.7.1	
json-c	0.14	0.14
kbd	2.4.0	2.4.0
kbd-misc	2.4.0	2.4.0
kernel	6.1.90	6.1.90
kernel-livepatch-r epo-cdn		2023.4.20240513

패키지	AMI	KVM
kernel-livepatch-r epo-s3	2023.4.20240513	
kernel-modules-extra		6.1.90
kernel-modules-ext ra-common		6.1.90
kernel-srpm-macros	1.0	1.0
kernel-tools	6.1.90	6.1.90
keyutils	1.6.3	1.6.3
keyutils-libs	1.6.3	1.6.3
kmod	29	29
kmod-libs	29	29
kpatch-runtime	0.9.7	0.9.7
krb5-libs	1.21	1.21
less	608	608
libacl	2.3.1	2.3.1
libaio	0.3.111	0.3.111
libarchive	3.5.3	3.5.3
libargon2	20171227	20171227
libassuan	2.5.5	2.5.5
libattr	2.5.1	2.5.1
libbasicobjects	0.1.1	0.1.1

패키지	AMI	KVM
libblkid	2.37.4	2.37.4
libcap	2.48	2.48
libcap-ng	0.8.2	0.8.2
libcbor	0.7.0	0.7.0
libcollection	0.7.0	0.7.0
libcom_err	1.46.5	1.46.5
libcomps	0.1.20	0.1.20
libconfig	1.7.2	1.7.2
libcurl-minimal	8.5.0	8.5.0
libdb	5.3.28	5.3.28
libdhash	0.5.0	
libdnf	0.69.0	0.69.0
libeconf	0.4.0	0.4.0
libedit	3.1	3.1
libev	4.33	4.33
libevent	2.1.12	2.1.12
libfdisk	2.37.4	2.37.4
libffi	3.4.4	3.4.4
libfido2	1.10.0	1.10.0
libgcc	11.4.1	11.4.1

패키지	AMI	KVM
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	11.4.1
libgpg-error	1.42	1.42
libibverbs	48.0	48.0
libidn2	2.3.2	2.3.2
libini_config	1.3.1	1.3.1
libkcapi	1.4.0	1.4.0
libkcapi-hmacalc	1.4.0	1.4.0
libldb	2.6.2	
libmaxminddb	1.5.2	1.5.2
libmetalink	0.1.3	0.1.3
libmnl	1.0.4	1.0.4
libmodulemd	2.13.0	2.13.0
libmount	2.37.4	2.37.4
libnfsidmap	2.5.4	2.5.4
libnghttp2	1.59.0	1.59.0
libnl3	3.5.0	3.5.0
libpath_utils	0.2.1	0.2.1
libpcap	1.10.1	1.10.1
libpipeline	1.5.3	1.5.3

패키지	AMI	KVM
libpkgconf	1.8.0	1.8.0
libpsl	0.21.1	0.21.1
libpwquality	1.4.4	1.4.4
libref_array	0.1.5	0.1.5
librepo	1.14.5	1.14.5
libreport-filessystem	2.15.2	2.15.2
libseccomp	2.5.3	2.5.3
libselinux	3.4	3.4
libselinux-utils	3.4	3.4
libsemanage	3.4	3.4
libsepol	3.4	3.4
libsigsegv	2.13	2.13
libsmartcols	2.37.4	2.37.4
libsolv	0.7.22	0.7.22
libss	1.46.5	1.46.5
libsss_certmap	2.9.4	
libsss_idmap	2.9.4	2.9.4
libsss_nss_idmap	2.9.4	2.9.4
libsss_sudo	2.9.4	
libstdc++	11.4.1	11.4.1

패키지	AMI	KVM
libstoragemgmt	1.9.4	1.9.4
libtalloc	2.3.4	
libtasn1	4.19.0	4.19.0
libtdb	1.4.7	
libtevent	0.13.0	
libtextstyle	0.21	0.21
libtirpc	1.3.3	1.3.3
libunistring	0.9.10	0.9.10
libuser	0.63	0.63
libutempter	1.2.1	1.2.1
libuuid	2.37.4	2.37.4
libuv	1.47.0	1.47.0
libverto	0.3.2	0.3.2
libverto-libev	0.3.2	0.3.2
libxcrypt	4.4.33	4.4.33
libxml2	2.10.4	2.10.4
libyaml	0.2.5	0.2.5
libzstd	1.5.5	1.5.5
lm_sensors-libs	3.6.0	3.6.0
lmdb-libs	0.9.29	0.9.29

패키지	AMI	KVM
logrotate	3.20.1	3.20.1
lsyf	4.94.0	4.94.0
lua-libs	5.4.4	5.4.4
lua-srpm-macros	1	1
lz4-libs	1.9.4	1.9.4
man-db	2.9.3	2.9.3
man-pages	5.10	5.10
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)
mpfr	4.1.0	4.1.0
nano	5.8	5.8
ncurses	6.2	6.2
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
nettle	3.8	3.8
net-tools	2.0	2.0
newt	0.52.21	0.52.21
nfs-utils	2.5.4	2.5.4
npth	1.6	1.6
nspr	4.35.0	4.35.0
nss	3.90.0	3.90.0

패키지	AMI	KVM
nss-softokn	3.90.0	3.90.0
nss-softokn-freebl	3.90.0	3.90.0
nss-sysinit	3.90.0	3.90.0
nss-util	3.90.0	3.90.0
ntsysv	1.15	1.15
numactl-libs	2.0.14	2.0.14
ocaml-srpm-macros	6	6
oniguruma	6.9.7.1	
openblas-srpm-macros	2	2
openldap	2.4.57	2.4.57
openssh	8.7p1	8.7p1
openssh-clients	8.7p1	8.7p1
openssh-server	8.7p1	8.7p1
openssl	3.0.8	3.0.8
openssl-libs	3.0.8	3.0.8
openssl-pkcs11	0.4.12	0.4.12
os-prober	1.77	1.77
p11-kit	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1
package-notes-srpm-macros	0.4	0.4

패키지	AMI	KVM
pam	1.5.1	1.5.1
parted	3.4	3.4
passwd	0.80	0.80
pciutils	3.7.0	3.7.0
pciutils-libs	3.7.0	3.7.0
pcre2	10.40	10.40
pcre2-syntax	10.40	10.40
perl-Carp	1.50	1.50
perl-Class-Struct	0.66	0.66
perl-constant	1.33	1.33
perl-DynaLoader	1.47	1.47
perl-Encode	3.15	3.15
perl-Errno	1.30	1.30
perl-Exporter	5.74	5.74
perl-Fcntl	1.13	1.13
perl-File-Basename	2.85	2.85
perl-File-Path	2.18	2.18
perl-File-stat	1.09	1.09
perl-File-Temp	0.231.100	0.231.100
perl-Getopt-Long	2.52	2.52

패키지	AMI	KVM
perl-Getopt-Std	1.12	1.12
perl-HTTP-Tiny	0.078	0.078
perl-if	0.60.800	0.60.800
perl-interpreter	5.32.1	5.32.1
perl-IO	1.43	1.43
perl-IPC-Open3	1.21	1.21
perl-libs	5.32.1	5.32.1
perl-MIME-Base64	3.16	3.16
perl-mro	1.23	1.23
perl-overload	1.31	1.31
perl-overloading	0.02	0.02
perl-parent	0.238	0.238
perl-PathTools	3.78	3.78
perl-Pod-Escapes	1.07	1.07
perl-podlators	4.14	4.14
perl-Pod-Perldoc	3.28.01	3.28.01
perl-Pod-Simple	3.42	3.42
perl-Pod-Usage	2.01	2.01
perl-POSIX	1.94	1.94
perl-Scalar-List-Utils	1.56	1.56

패키지	AMI	KVM
perl-SelectSaver	1.02	1.02
perl-Socket	2.032	2.032
perl-srpm-macros	1	1
perl-Storable	3.21	3.21
perl-subst	1.03	1.03
perl-Symbol	1.08	1.08
perl-Term-ANSIColor	5.01	5.01
perl-Term-Cap	1.17	1.17
perl-Text-ParseWords	3.30	3.30
perl-Text-Tabs+Wrap	2021.0726	2021.0726
perl-Time-Local	1.300	1.300
perl-vars	1.05	1.05
pkgconf	1.8.0	1.8.0
pkgconf-m4	1.8.0	1.8.0
pkgconf-pkg-config	1.8.0	1.8.0
policycoreutils	3.4	3.4
policycoreutils-python-utils	3.4	
popt	1.18	1.18
procps-ng	3.3.17	3.3.17
protobuf-c	1.4.1	1.4.1

패키지	AMI	KVM
psacct	6.6.4	6.6.4
psmisc	23.4	23.4
publicsuffix-list-dafsa	20240212	20240212
python3	3.9.16	3.9.16
python3-attrs	20.3.0	20.3.0
python3-audit	3.0.6	3.0.6
python3-awscrt	0.19.19	0.19.19
python3-babel	2.9.1	2.9.1
python3-cffi	1.14.5	1.14.5
python3-chardet	4.0.0	4.0.0
python3-colorama	0.4.4	0.4.4
python3-configobj	5.0.6	5.0.6
python3-cryptography	36.0.1	36.0.1
python3-daemon	2.3.0	
python3-dateutil	2.8.1	2.8.1
python3-dbus	1.2.18	1.2.18
python3-distro	1.5.0	1.5.0
python3-dnf	4.14.0	4.14.0
python3-dnf-plugins-core	4.3.0	4.3.0

패키지	AMI	KVM
python3-docutils	0.16	0.16
python3-gpg	1.15.1	1.15.1
python3-hawkey	0.69.0	0.69.0
python3-idna	2.10	2.10
python3-jinja2	2.11.3	2.11.3
python3-jmespath	0.10.0	0.10.0
python3-jsonpatch	1.21	1.21
python3-jsonpointer	2.0	2.0
python3-jsonschema	3.2.0	3.2.0
python3-libcomps	0.1.20	0.1.20
python3-libdnf	0.69.0	0.69.0
python3-libs	3.9.16	3.9.16
python3-libselenium	3.4	3.4
python3-libsemanage	3.4	3.4
python3-libstorage mgmt	1.9.4	1.9.4
python3-lockfile	0.12.2	
python3-markupsafe	1.1.1	1.1.1
python3-netifaces	0.10.6	0.10.6
python3-oauthlib	3.0.2	3.0.2
python3-pip-wheel	21.3.1	21.3.1

패키지	AMI	KVM
python3-ply	3.11	3.11
python3-policycore utils	3.4	3.4
python3-prettytable	0.7.2	0.7.2
python3-prompt-too lkit	3.0.24	3.0.24
python3-pycparser	2.20	2.20
python3-pyrsistent	0.17.3	0.17.3
python3-pyserial	3.4	3.4
python3-pysocks	1.7.1	1.7.1
python3-pytz	2022.7.1	2022.7.1
python3-pyyaml	5.4.1	5.4.1
python3-requests	2.25.1	2.25.1
python3-rpm	4.16.1.3	4.16.1.3
python3-ruamel-yaml	0.16.6	0.16.6
python3-ruamel-yaml- clib	0.1.2	0.1.2
python3-setools	4.4.1	4.4.1
python3-setuptools	59.6.0	59.6.0
python3-setuptools- wheel	59.6.0	59.6.0
python3-six	1.15.0	1.15.0

패키지	AMI	KVM
python3-systemd	235	235
python3-urllib3	1.25.10	1.25.10
python3-wcwidth	0.2.5	0.2.5
python-chevron	0.13.1	
python-srpm-macros	3.9	3.9
quota	4.06	4.06
quota-nls	4.06	4.06
readline	8.1	8.1
rng-tools	6.14	6.14
rootfiles	8.1	8.1
rpcbind	1.2.6	1.2.6
rpm	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3
rpm-libs	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3	4.16.1.3
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3
rpm-sign-libs	4.16.1.3	4.16.1.3
rsync	3.2.6	3.2.6
rust-srpm-macros	21	21
sbsigntools	0.9.4	0.9.4

패키지	AMI	KVM
screen	4.8.0	4.8.0
sed	4.8	4.8
selinux-policy	37.22	37.22
selinux-policy-targeted	37.22	37.22
setup	2.13.7	2.13.7
shadow-utils	4.9	4.9
slang	2.3.2	2.3.2
sqlite-libs	3.40.0	3.40.0
sssd-client	2.9.4	2.9.4
sssd-common	2.9.4	
sssd-kcm	2.9.4	
sssd-nfs-idmap	2.9.4	
strace	6.8	6.8
sudo	1.9.15	1.9.15
sysctl-defaults	1.0	1.0
sysstat	12.5.6	12.5.6
systemd	252.16	252.16
systemd-libs	252.16	252.16
systemd-networkd	252.16	252.16
systemd-pam	252.16	252.16

패키지	AMI	KVM
systemd-resolved	252.16	252.16
systemd-udev	252.16	252.16
system-release	2023.4.20240513	2023.4.20240513
systemtap-runtime	4.8	4.8
tar	1.34	1.34
tbb	2020.3	2020.3
tcpdump	4.99.1	4.99.1
tcsch	6.24.07	6.24.07
time	1.9	1.9
traceroute	2.1.3	2.1.3
tzdata	2024a	2024a
unzip	6.0	6.0
update-motd	2.2	2.2
userspace-rcu	0.12.1	0.12.1
util-linux	2.37.4	2.37.4
util-linux-core	2.37.4	2.37.4
vim-common	9.0.2153	9.0.2153
vim-data	9.0.2153	9.0.2153
vim-enhanced	9.0.2153	9.0.2153
vim-filesystem	9.0.2153	9.0.2153

패키지	AMI	KVM
vim-minimal	9.0.2153	9.0.2153
wget	1.21.3	1.21.3
which	2.21	2.21
words	3.0	3.0
xfsdump	3.1.11	3.1.11
xfspgrog	5.18.0	5.18.0
xxd	9.0.2153	9.0.2153
xxhash-libs	0.8.0	0.8.0
xz	5.2.5	5.2.5
xz-libs	5.2.5	5.2.5
yum	4.14.0	4.14.0
zip	3.0	3.0
zlib	1.2.11	1.2.11
zram-generator	1.1.2	
zram-generator-def aults	1.1.2	
zstd	1.5.5	1.5.5

아마존 리눅스 2023 표준 AMI에 설치된 패키지를 AL2023 VMware OVA 이미지와 비교

AL2023 표준 AMI에 있는 RPM을 AL2023 VMware OVA 이미지에 있는 RPM과 비교한 것입니다.

패키지	AMI	VMware OVA
acl	2.3.1	2.3.1
acpid	2.0.32	
alternatives	1.15	1.15
amazon-chrony-config	4.3	
amazon-ec2-net-utils	2.4.1	
amazon-linux-onprem		1.2
amazon-linux-repo-cdn		2023.4.20240513
amazon-linux-repo-s3	2023.4.20240513	
amazon-linux-sb-keys	2023.1	2023.1
amazon-onprem-network		1.2
amazon-rpm-config	228	228
amazon-ssm-agent	3.3.380.0	3.3.380.0
at	3.1.23	3.1.23
attr	2.5.1	2.5.1
audit	3.0.6	3.0.6
audit-libs	3.0.6	3.0.6
aws-cfn-bootstrap	2.0	
awscli-2	2.15.30	2.15.30
basesystem	11	11

패키지	AMI	VMware OVA
bash	5.2.15	5.2.15
bash-completion	2.11	2.11
bc	1.07.1	1.07.1
bind-libs	9.16.48	9.16.48
bind-license	9.16.48	9.16.48
bind-utils	9.16.48	9.16.48
binutils	2.39	2.39
boost-filesystem	1.75.0	1.75.0
boost-system	1.75.0	1.75.0
boost-thread	1.75.0	1.75.0
bzip2	1.0.8	1.0.8
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023.2.64	2023.2.64
c-ares	1.19.0	
checkpolicy	3.4	3.4
chkconfig	1.15	1.15
chrony	4.3	4.3
cloud-init	22.2.2	22.2.2
cloud-init-cfg-ec2	22.2.2	
cloud-init-cfg-onpre		22.2.2

패키지	AMI	VMware OVA
cloud-utils-growpart	0.31	0.31
coreutils	8.32	8.32
coreutils-common	8.32	8.32
cpio	2.13	2.13
cracklib	2.9.6	2.9.6
cracklib-dicts	2.9.6	2.9.6
crontabs	1.11	1.11
crypto-policies	20220428	20220428
crypto-policies-scripts	20220428	20220428
cryptsetup	2.6.1	2.6.1
cryptsetup-libs	2.6.1	2.6.1
curl-minimal	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27
cyrus-sasl-plain	2.1.27	2.1.27
dbus	1.12.28	1.12.28
dbus-broker	32	32
dbus-common	1.12.28	1.12.28
dbus-libs	1.12.28	1.12.28
device-mapper	1.02.185	1.02.185
device-mapper-libs	1.02.185	1.02.185

패키지	AMI	VMware OVA
diffutils	3.8	3.8
dnf	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2
dnf-plugins-core	4.3.0	4.3.0
dnf-plugin-support-info	1.2	1.2
dnf-utils	4.3.0	4.3.0
dosfstools	4.2	4.2
dracut	055	055
dracut-config-ec2	3.0	
dracut-config-generic	055	055
dwz	0.14	0.14
dyninst	10.2.1	10.2.1
e2fsprogs	1.46.5	1.46.5
e2fsprogs-libs	1.46.5	1.46.5
ec2-hibinit-agent	1.0.8	
ec2-instance-connect	1.1	
ec2-instance-connect-selinux	1.1	

패키지	AMI	VMware OVA
ec2-utils	2.2.0	
ed	1.14.2	1.14.2
efi-filesystem	5	5
efi-srpm-macros	5	5
efivar	38	38
efivar-libs	38	38
elfutils-debuginfod-client	0.188	0.188
elfutils-default-yama-scope	0.188	0.188
elfutils-libelf	0.188	0.188
elfutils-libs	0.188	0.188
ethtool	5.15	5.15
expat	2.5.0	2.5.0
file	5.39	5.39
file-libs	5.39	5.39
filesystem	3.14	3.14
findutils	4.8.0	4.8.0
fonts-srpm-macros	2.0.5	2.0.5
fstrm	0.6.1	0.6.1
fuse3		3.10.4

패키지	AMI	VMware OVA
fuse3-libs		3.10.4
fuse-common		3.10.4
fuse-libs	2.9.9	2.9.9
gawk	5.1.0	5.1.0
gdbm-libs	1.19	1.19
gdisk	1.0.8	1.0.8
gettext	0.21	0.21
gettext-libs	0.21	0.21
ghc-srpm-macros	1.5.0	1.5.0
glib2	2.74.7	2.74.7
glibc	2.34	2.34
glibc-all-langpacks	2.34	2.34
glibc-common	2.34	2.34
glibc-gconv-extra	2.34	2.34
glibc-locale-source	2.34	2.34
gmp	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7
gnutls	3.8.0	3.8.0
go-srpm-macros	3.2.0	3.2.0
gpgme	1.15.1	1.15.1

패키지	AMI	VMware OVA
gpm-libs	1.20.7	1.20.7
grep	3.8	3.8
groff-base	1.22.4	1.22.4
grub2-common	2.06	2.06
grub2-efi-x64-ec2	2.06	2.06
grub2-pc		2.06
grub2-pc-modules	2.06	2.06
grub2-tools	2.06	2.06
grub2-tools-minimal	2.06	2.06
grubby	8.40	8.40
gssproxy	0.8.4	0.8.4
gzip	1.12	1.12
hostname	3.23	3.23
hunspell	1.7.0	1.7.0
hunspell-en	0.20140811.1	0.20140811.1
hunspell-en-GB	0.20140811.1	0.20140811.1
hunspell-en-US	0.20140811.1	0.20140811.1
hunspell-filesystem	1.7.0	1.7.0
hwdata	0.353	0.353
info	6.7	6.7

패키지	AMI	VMware OVA
inih	49	49
initscripts	10.09	10.09
iproute	5.10.0	5.10.0
iputils	20210202	20210202
irqbalance	1.9.0	1.9.0
jansson	2.14	2.14
jitterentropy	3.4.1	3.4.1
jq	1.7.1	
json-c	0.14	0.14
kbd	2.4.0	2.4.0
kbd-misc	2.4.0	2.4.0
kernel	6.1.90	6.1.90
kernel-livepatch-r epo-cdn		2023.4.20240513
kernel-livepatch-r epo-s3	2023.4.20240513	
kernel-modules-extra		6.1.90
kernel-modules-ext ra-common		6.1.90
kernel-srpm-macros	1.0	1.0
kernel-tools	6.1.90	6.1.90

패키지	AMI	VMware OVA
keyutils	1.6.3	1.6.3
keyutils-libs	1.6.3	1.6.3
kmod	29	29
kmod-libs	29	29
kpatch-runtime	0.9.7	0.9.7
krb5-libs	1.21	1.21
less	608	608
libacl	2.3.1	2.3.1
libaio	0.3.111	0.3.111
libarchive	3.5.3	3.5.3
libargon2	20171227	20171227
libassuan	2.5.5	2.5.5
libattr	2.5.1	2.5.1
libbasicobjects	0.1.1	0.1.1
libblkid	2.37.4	2.37.4
libcap	2.48	2.48
libcap-ng	0.8.2	0.8.2
libcbor	0.7.0	0.7.0
libcollection	0.7.0	0.7.0
libcom_err	1.46.5	1.46.5

패키지	AMI	VMware OVA
libcomps	0.1.20	0.1.20
libconfig	1.7.2	1.7.2
libcurl-minimal	8.5.0	8.5.0
libdb	5.3.28	5.3.28
libdhash	0.5.0	
libdnf	0.69.0	0.69.0
libeconf	0.4.0	0.4.0
libedit	3.1	3.1
libev	4.33	4.33
libevent	2.1.12	2.1.12
libfdisk	2.37.4	2.37.4
libffi	3.4.4	3.4.4
libfido2	1.10.0	1.10.0
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	11.4.1
libgpg-error	1.42	1.42
libibverbs	48.0	48.0
libidn2	2.3.2	2.3.2
libini_config	1.3.1	1.3.1

패키지	AMI	VMware OVA
libkcapi	1.4.0	1.4.0
libkcapi-hmaccalc	1.4.0	1.4.0
libldb	2.6.2	
libmaxminddb	1.5.2	1.5.2
libmetalink	0.1.3	0.1.3
libmnl	1.0.4	1.0.4
libmodulemd	2.13.0	2.13.0
libmount	2.37.4	2.37.4
libmspack		0.10.1
libnfsidmap	2.5.4	2.5.4
libnghttp2	1.59.0	1.59.0
libnl3	3.5.0	3.5.0
libpath_utils	0.2.1	0.2.1
libpcap	1.10.1	1.10.1
libpipeline	1.5.3	1.5.3
libpkgconf	1.8.0	1.8.0
libpsl	0.21.1	0.21.1
libpwquality	1.4.4	1.4.4
libref_array	0.1.5	0.1.5
librepo	1.14.5	1.14.5

패키지	AMI	VMware OVA
libreport-filessystem	2.15.2	2.15.2
libseccomp	2.5.3	2.5.3
libselinux	3.4	3.4
libselinux-utils	3.4	3.4
libsemanage	3.4	3.4
libsepol	3.4	3.4
libsigsegv	2.13	2.13
libsmartcols	2.37.4	2.37.4
libsolv	0.7.22	0.7.22
libss	1.46.5	1.46.5
libsss_certmap	2.9.4	
libsss_idmap	2.9.4	2.9.4
libsss_nss_idmap	2.9.4	2.9.4
libsss_sudo	2.9.4	
libstdc++	11.4.1	11.4.1
libstoragegmt	1.9.4	1.9.4
libtalloc	2.3.4	
libtasn1	4.19.0	4.19.0
libttdb	1.4.7	
libtevent	0.13.0	

패키지	AMI	VMware OVA
libtextstyle	0.21	0.21
libtirpc	1.3.3	1.3.3
libtool-ltdl		2.4.7
libunistring	0.9.10	0.9.10
libuser	0.63	0.63
libutempter	1.2.1	1.2.1
libuuid	2.37.4	2.37.4
libuv	1.47.0	1.47.0
libverto	0.3.2	0.3.2
libverto-libev	0.3.2	0.3.2
libxcrypt	4.4.33	4.4.33
libxml2	2.10.4	2.10.4
libxslt		1.1.34
libyaml	0.2.5	0.2.5
libzstd	1.5.5	1.5.5
lm_sensors-libs	3.6.0	3.6.0
lmdb-libs	0.9.29	0.9.29
logrotate	3.20.1	3.20.1
lsof	4.94.0	4.94.0
lua-libs	5.4.4	5.4.4

패키지	AMI	VMware OVA
lua-srpm-macros	1	1
lz4-libs	1.9.4	1.9.4
man-db	2.9.3	2.9.3
man-pages	5.10	5.10
microcode_ctl	2.1	2.1
mpfr	4.1.0	4.1.0
nano	5.8	5.8
ncurses	6.2	6.2
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
nettle	3.8	3.8
net-tools	2.0	2.0
newt	0.52.21	0.52.21
nfs-utils	2.5.4	2.5.4
npth	1.6	1.6
nspr	4.35.0	4.35.0
nss	3.90.0	3.90.0
nss-softokn	3.90.0	3.90.0
nss-softokn-freebl	3.90.0	3.90.0
nss-sysinit	3.90.0	3.90.0

패키지	AMI	VMware OVA
nss-util	3.90.0	3.90.0
ntsysv	1.15	1.15
numactl-libs	2.0.14	2.0.14
ocaml-srpm-macros	6	6
oniguruma	6.9.7.1	
openblas-srpm-macros	2	2
openldap	2.4.57	2.4.57
openssh	8.7p1	8.7p1
openssh-clients	8.7p1	8.7p1
openssh-server	8.7p1	8.7p1
openssl	3.0.8	3.0.8
openssl-libs	3.0.8	3.0.8
openssl-pkcs11	0.4.12	0.4.12
open-vm-tools		12.3.0
os-prober	1.77	1.77
p11-kit	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1
package-notes-srpm-macros	0.4	0.4
pam	1.5.1	1.5.1
parted	3.4	3.4

패키지	AMI	VMware OVA
passwd	0.80	0.80
pciutils	3.7.0	3.7.0
pciutils-libs	3.7.0	3.7.0
pcre2	10.40	10.40
pcre2-syntax	10.40	10.40
perl-Carp	1.50	1.50
perl-Class-Struct	0.66	0.66
perl-constant	1.33	1.33
perl-DynaLoader	1.47	1.47
perl-Encode	3.15	3.15
perl-Errno	1.30	1.30
perl-Exporter	5.74	5.74
perl-Fcntl	1.13	1.13
perl-File-Basename	2.85	2.85
perl-File-Path	2.18	2.18
perl-File-stat	1.09	1.09
perl-File-Temp	0.231.100	0.231.100
perl-Getopt-Long	2.52	2.52
perl-Getopt-Std	1.12	1.12
perl-HTTP-Tiny	0.078	0.078

패키지	AMI	VMware OVA
perl-if	0.60.800	0.60.800
perl-interpreter	5.32.1	5.32.1
perl-IO	1.43	1.43
perl-IPC-Open3	1.21	1.21
perl-libs	5.32.1	5.32.1
perl-MIME-Base64	3.16	3.16
perl-mro	1.23	1.23
perl-overload	1.31	1.31
perl-overloading	0.02	0.02
perl-parent	0.238	0.238
perl-PathTools	3.78	3.78
perl-Pod-Escapes	1.07	1.07
perl-podlators	4.14	4.14
perl-Pod-Perldoc	3.28.01	3.28.01
perl-Pod-Simple	3.42	3.42
perl-Pod-Usage	2.01	2.01
perl-POSIX	1.94	1.94
perl-Scalar-List-Utils	1.56	1.56
perl-SelectSaver	1.02	1.02
perl-Socket	2.032	2.032

패키지	AMI	VMware OVA
perl-srpm-macros	1	1
perl-Storable	3.21	3.21
perl-subst	1.03	1.03
perl-Symbol	1.08	1.08
perl-Term-ANSIColor	5.01	5.01
perl-Term-Cap	1.17	1.17
perl-Text-ParseWords	3.30	3.30
perl-Text-Tabs+Wrap	2021.0726	2021.0726
perl-Time-Local	1.300	1.300
perl-vars	1.05	1.05
pkgconf	1.8.0	1.8.0
pkgconf-m4	1.8.0	1.8.0
pkgconf-pkg-config	1.8.0	1.8.0
policycoreutils	3.4	3.4
policycoreutils-python-utils	3.4	
popt	1.18	1.18
procps-ng	3.3.17	3.3.17
protobuf-c	1.4.1	1.4.1
psacct	6.6.4	6.6.4
psmisc	23.4	23.4

패키지	AMI	VMware OVA
publicsuffix-list-dafsa	20240212	20240212
python3	3.9.16	3.9.16
python3-attrs	20.3.0	20.3.0
python3-audit	3.0.6	3.0.6
python3-awscli	0.19.19	0.19.19
python3-babel	2.9.1	2.9.1
python3-cffi	1.14.5	1.14.5
python3-chardet	4.0.0	4.0.0
python3-colorama	0.4.4	0.4.4
python3-configobj	5.0.6	5.0.6
python3-cryptography	36.0.1	36.0.1
python3-daemon	2.3.0	
python3-dateutil	2.8.1	2.8.1
python3-dbus	1.2.18	1.2.18
python3-distro	1.5.0	1.5.0
python3-dnf	4.14.0	4.14.0
python3-dnf-plugins-core	4.3.0	4.3.0
python3-docutils	0.16	0.16
python3-gpg	1.15.1	1.15.1

패키지	AMI	VMware OVA
python3-hawkey	0.69.0	0.69.0
python3-idna	2.10	2.10
python3-jinja2	2.11.3	2.11.3
python3-jmespath	0.10.0	0.10.0
python3-jsonpatch	1.21	1.21
python3-jsonpointer	2.0	2.0
python3-jsonschema	3.2.0	3.2.0
python3-libcomps	0.1.20	0.1.20
python3-libdnf	0.69.0	0.69.0
python3-libs	3.9.16	3.9.16
python3-libselenium	3.4	3.4
python3-libsemanage	3.4	3.4
python3-libstorage mgmt	1.9.4	1.9.4
python3-lockfile	0.12.2	
python3-markupsafe	1.1.1	1.1.1
python3-netifaces	0.10.6	0.10.6
python3-oauthlib	3.0.2	3.0.2
python3-pip-wheel	21.3.1	21.3.1
python3-ply	3.11	3.11

패키지	AMI	VMware OVA
python3-policycore utils	3.4	3.4
python3-prettytable	0.7.2	0.7.2
python3-prompt-too lkit	3.0.24	3.0.24
python3-pycparser	2.20	2.20
python3-pyrsistent	0.17.3	0.17.3
python3-pyserial	3.4	3.4
python3-pysocks	1.7.1	1.7.1
python3-pytz	2022.7.1	2022.7.1
python3-pyyaml	5.4.1	5.4.1
python3-requests	2.25.1	2.25.1
python3-rpm	4.16.1.3	4.16.1.3
python3-ruamel-yaml	0.16.6	0.16.6
python3-ruamel-yaml- clib	0.1.2	0.1.2
python3-setools	4.4.1	4.4.1
python3-setuptools	59.6.0	59.6.0
python3-setuptools- wheel	59.6.0	59.6.0
python3-six	1.15.0	1.15.0
python3-systemd	235	235

패키지	AMI	VMware OVA
python3-urllib3	1.25.10	1.25.10
python3-wcwidth	0.2.5	0.2.5
python-chevron	0.13.1	
python-srpm-macros	3.9	3.9
quota	4.06	4.06
quota-nls	4.06	4.06
readline	8.1	8.1
rng-tools	6.14	6.14
rootfiles	8.1	8.1
rpcbind	1.2.6	1.2.6
rpm	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3
rpm-libs	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3	4.16.1.3
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3
rpm-sign-libs	4.16.1.3	4.16.1.3
rsync	3.2.6	3.2.6
rust-srpm-macros	21	21
sbsigntools	0.9.4	0.9.4
screen	4.8.0	4.8.0

패키지	AMI	VMware OVA
sed	4.8	4.8
selinux-policy	37.22	37.22
selinux-policy-targeted	37.22	37.22
setup	2.13.7	2.13.7
shadow-utils	4.9	4.9
slang	2.3.2	2.3.2
sqlite-libs	3.40.0	3.40.0
sssd-client	2.9.4	2.9.4
sssd-common	2.9.4	
sssd-kcm	2.9.4	
sssd-nfs-idmap	2.9.4	
strace	6.8	6.8
sudo	1.9.15	1.9.15
sysctl-defaults	1.0	1.0
sysstat	12.5.6	12.5.6
systemd	252.16	252.16
systemd-libs	252.16	252.16
systemd-networkd	252.16	252.16
systemd-pam	252.16	252.16
systemd-resolved	252.16	252.16

패키지	AMI	VMware OVA
systemd-udev	252.16	252.16
system-release	2023.4.20240513	2023.4.20240513
systemtap-runtime	4.8	4.8
tar	1.34	1.34
tbb	2020.3	2020.3
tcpdump	4.99.1	4.99.1
tcsch	6.24.07	6.24.07
time	1.9	1.9
traceroute	2.1.3	2.1.3
tzdata	2024a	2024a
unzip	6.0	6.0
update-motd	2.2	2.2
userspace-rcu	0.12.1	0.12.1
util-linux	2.37.4	2.37.4
util-linux-core	2.37.4	2.37.4
vim-common	9.0.2153	9.0.2153
vim-data	9.0.2153	9.0.2153
vim-enhanced	9.0.2153	9.0.2153
vim-filesystem	9.0.2153	9.0.2153
vim-minimal	9.0.2153	9.0.2153

패키지	AMI	VMware OVA
wget	1.21.3	1.21.3
which	2.21	2.21
words	3.0	3.0
xfsdump	3.1.11	3.1.11
xfspgrog	5.18.0	5.18.0
xmlsec1		1.2.33
xmlsec1-openssl		1.2.33
xxd	9.0.2153	9.0.2153
xxhash-libs	0.8.0	0.8.0
xz	5.2.5	5.2.5
xz-libs	5.2.5	5.2.5
yum	4.14.0	4.14.0
zip	3.0	3.0
zlib	1.2.11	1.2.11
zram-generator	1.1.2	
zram-generator-def aults	1.1.2	
zstd	1.5.5	1.5.5

아마존 리눅스 2023 표준 AMI에 설치된 패키지와 AL2023 하이퍼-V 이미지 비교

AL2023 표준 AMI에 있는 RPM을 AL2023 Hyper-V 이미지에 있는 RPM과 비교한 것입니다.

패키지	AMI	하이퍼-V VHDX
acl	2.3.1	2.3.1
acpid	2.0.32	
alternatives	1.15	1.15
amazon-chrony-config	4.3	
amazon-ec2-net-utils	2.4.1	
amazon-linux-onprem		1.2
amazon-linux-repo-cdn		2023.4.20240319
amazon-linux-repo-s3	2023.4.20240319	
amazon-linux-sb-keys	2023.1	2023.1
amazon-onprem-netw ork		1.2
amazon-rpm-config	228	228
amazon-ssm-agent	3.2.2303.0	3.2.2303.0
at	3.1.23	3.1.23
attr	2.5.1	2.5.1
audit	3.0.6	3.0.6
audit-libs	3.0.6	3.0.6

패키지	AMI	하이퍼-V VHDX
aws-cfn-bootstrap	2.0	
awscli-2	2.14.5	2.14.5
basesystem	11	11
bash	5.2.15	5.2.15
bash-completion	2.11	2.11
bc	1.07.1	1.07.1
bind-libs	9.16.48	9.16.48
bind-license	9.16.48	9.16.48
bind-utils	9.16.48	9.16.48
binutils	2.39	2.39
boost-filesystem	1.75.0	1.75.0
boost-system	1.75.0	1.75.0
boost-thread	1.75.0	1.75.0
bzip2	1.0.8	1.0.8
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023.2.64	2023.2.64
c-ares	1.19.0	
checkpolicy	3.4	3.4
chkconfig	1.15	1.15
chrony	4.3	4.3

패키지	AMI	하이퍼-V VHDX
cloud-init	22.2.2	22.2.2
cloud-init-cfg-ec2	22.2.2	
cloud-init-cfg-onpre		22.2.2
cloud-utils-growpart	0.31	0.31
coreutils	8.32	8.32
coreutils-common	8.32	8.32
cpio	2.13	2.13
cracklib	2.9.6	2.9.6
cracklib-dicts	2.9.6	2.9.6
crontabs	1.11	1.11
crypto-policies	20220428	20220428
crypto-policies-scripts	20220428	20220428
cryptsetup	2.6.1	2.6.1
cryptsetup-libs	2.6.1	2.6.1
curl-minimal	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27
cyrus-sasl-plain	2.1.27	2.1.27
dbus	1.12.28	1.12.28
dbus-broker	32	32

패키지	AMI	하이퍼-V VHDX
dbus-common	1.12.28	1.12.28
dbus-libs	1.12.28	1.12.28
device-mapper	1.02.185	1.02.185
device-mapper-libs	1.02.185	1.02.185
diffutils	3.8	3.8
dnf	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2
dnf-plugins-core	4.3.0	4.3.0
dnf-plugin-support-info	1.2	1.2
dnf-utils	4.3.0	4.3.0
dosfstools	4.2	4.2
dracut	055	055
dracut-config-ec2	3.0	
dracut-config-generic	055	055
dwz	0.14	0.14
dyninst	10.2.1	10.2.1
e2fsprogs	1.46.5	1.46.5

패키지	AMI	하이퍼-V VHDX
e2fsprogs-libs	1.46.5	1.46.5
ec2-hibinit-agent	1.0.8	
ec2-instance-connect	1.1	
ec2-instance-connect-selinux	1.1	
ec2-utils	2.2.0	
ed	1.14.2	1.14.2
efi-filesystem	5	5
efi-srpm-macros	5	5
efivar	38	38
efivar-libs	38	38
elfutils-debuginfod-client	0.188	0.188
elfutils-default-yama-scope	0.188	0.188
elfutils-libelf	0.188	0.188
elfutils-libs	0.188	0.188
ethtool	5.15	5.15
expat	2.5.0	2.5.0
file	5.39	5.39
file-libs	5.39	5.39

패키지	AMI	하이퍼-V VHDX
filesystem	3.14	3.14
findutils	4.8.0	4.8.0
fonts-srpm-macros	2.0.5	2.0.5
fstrm	0.6.1	0.6.1
fuse-libs	2.9.9	2.9.9
gawk	5.1.0	5.1.0
gdbm-libs	1.19	1.19
gdisk	1.0.8	1.0.8
gettext	0.21	0.21
gettext-libs	0.21	0.21
ghc-srpm-macros	1.5.0	1.5.0
glib2	2.74.7	2.74.7
glibc	2.34	2.34
glibc-all-langpacks	2.34	2.34
glibc-common	2.34	2.34
glibc-gconv-extra	2.34	2.34
glibc-locale-source	2.34	2.34
gmp	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7
gnutls	3.8.0	3.8.0

패키지	AMI	하이퍼-V VHDX
go-srpm-macros	3.2.0	3.2.0
gpgme	1.15.1	1.15.1
gpm-libs	1.20.7	1.20.7
grep	3.8	3.8
groff-base	1.22.4	1.22.4
grub2-common	2.06	2.06
grub2-efi-x64-ec2	2.06	2.06
grub2-pc		2.06
grub2-pc-modules	2.06	2.06
grub2-tools	2.06	2.06
grub2-tools-minimal	2.06	2.06
grubby	8.40	8.40
gssproxy	0.8.4	0.8.4
gzip	1.12	1.12
hostname	3.23	3.23
hunspell	1.7.0	1.7.0
hunspell-en	0.20140811.1	0.20140811.1
hunspell-en-GB	0.20140811.1	0.20140811.1
hunspell-en-US	0.20140811.1	0.20140811.1
hunspell-filesystem	1.7.0	1.7.0

패키지	AMI	하이퍼-V VHDX
hwdata	0.353	0.353
hyperv-daemons		0
hyperv-daemons-lic ense		0
hypervfcopyd		0
hypervkvpd		0
hyperv-tools		0
hypervvssd		0
info	6.7	6.7
inih	49	49
initscripts	10.09	10.09
iproute	5.10.0	5.10.0
iputils	20210202	20210202
irqbalance	1.9.0	1.9.0
jansson	2.14	2.14
jitterentropy	3.4.1	3.4.1
jq	1.7.1	1.7.1
json-c	0.14	0.14
kbd	2.4.0	2.4.0
kbd-misc	2.4.0	2.4.0
kernel	6.1.79	6.1.79

패키지	AMI	하이퍼-V VHDX
kernel-livepatch-r epo-cdn		2023.4.20240319
kernel-livepatch-r epo-s3	2023.4.20240319	
kernel-modules-extra		6.1.79
kernel-modules-ext ra-common		6.1.79
kernel-srpm-macros	1.0	1.0
kernel-tools	6.1.79	6.1.79
keyutils	1.6.3	1.6.3
keyutils-libs	1.6.3	1.6.3
kmod	29	29
kmod-libs	29	29
kpatch-runtime	0.9.7	0.9.7
krb5-libs	1.21	1.21
less	608	608
libacl	2.3.1	2.3.1
libaio	0.3.111	0.3.111
libarchive	3.5.3	3.5.3
libargon2	20171227	20171227
libassuan	2.5.5	2.5.5

패키지	AMI	하이퍼-V VHDX
libattr	2.5.1	2.5.1
libbasicobjects	0.1.1	0.1.1
libblkid	2.37.4	2.37.4
libcap	2.48	2.48
libcap-ng	0.8.2	0.8.2
libcbor	0.7.0	0.7.0
libcollection	0.7.0	0.7.0
libcom_err	1.46.5	1.46.5
libcomps	0.1.20	0.1.20
libconfig	1.7.2	1.7.2
libcurl-minimal	8.5.0	8.5.0
libdb	5.3.28	5.3.28
libdhash	0.5.0	
libdnf	0.69.0	0.69.0
libeconf	0.4.0	0.4.0
libedit	3.1	3.1
libev	4.33	4.33
libevent	2.1.12	2.1.12
libfdisk	2.37.4	2.37.4
libffi	3.4.4	3.4.4

패키지	AMI	하이퍼-V VHDX
libfido2	1.10.0	1.10.0
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	11.4.1
libgpg-error	1.42	1.42
libibverbs	48.0	48.0
libidn2	2.3.2	2.3.2
libini_config	1.3.1	1.3.1
libkcap	1.4.0	1.4.0
libkcap-hmacalc	1.4.0	1.4.0
libldb	2.6.2	
libmaxminddb	1.5.2	1.5.2
libmetalink	0.1.3	0.1.3
libmnl	1.0.4	1.0.4
libmodulemd	2.13.0	2.13.0
libmount	2.37.4	2.37.4
libnfsidmap	2.5.4	2.5.4
libnghttp2	1.57.0	1.57.0
libnl3	3.5.0	3.5.0
libpath_utils	0.2.1	0.2.1

패키지	AMI	하이퍼-V VHDX
libpcap	1.10.1	1.10.1
libpipeline	1.5.3	1.5.3
libpkgconf	1.8.0	1.8.0
libpsl	0.21.1	0.21.1
libpwquality	1.4.4	1.4.4
libref_array	0.1.5	0.1.5
librepo	1.14.5	1.14.5
libreport-filesystem	2.15.2	2.15.2
libseccomp	2.5.3	2.5.3
libselinux	3.4	3.4
libselinux-utils	3.4	3.4
libsemanage	3.4	3.4
libsepol	3.4	3.4
libsigsegv	2.13	2.13
libsmartcols	2.37.4	2.37.4
libsolv	0.7.22	0.7.22
libss	1.46.5	1.46.5
libsss_certmap	2.9.4	
libsss_idmap	2.9.4	2.9.4
libsss_nss_idmap	2.9.4	2.9.4

패키지	AMI	하이퍼-V VHDX
libsss_sudo	2.9.4	
libstdc++	11.4.1	11.4.1
libstoragemgmt	1.9.4	1.9.4
libtalloc	2.3.4	
libtasn1	4.19.0	4.19.0
libtdb	1.4.7	
libtevent	0.13.0	
libtextstyle	0.21	0.21
libtirpc	1.3.3	1.3.3
libunistring	0.9.10	0.9.10
libuser	0.63	0.63
libutempter	1.2.1	1.2.1
libuuid	2.37.4	2.37.4
libuv	1.47.0	1.47.0
libverto	0.3.2	0.3.2
libverto-libev	0.3.2	0.3.2
libxcrypt	4.4.33	4.4.33
libxml2	2.10.4	2.10.4
libyaml	0.2.5	0.2.5
libzstd	1.5.5	1.5.5

패키지	AMI	하이퍼-V VHDX
lm_sensors-libs	3.6.0	3.6.0
lmdb-libs	0.9.29	0.9.29
logrotate	3.20.1	3.20.1
lsof	4.94.0	4.94.0
lua-libs	5.4.4	5.4.4
lua-srpm-macros	1	1
lz4-libs	1.9.4	1.9.4
man-db	2.9.3	2.9.3
man-pages	5.10	5.10
microcode_ctl	2.1	2.1
mpfr	4.1.0	4.1.0
nano	5.8	5.8
ncurses	6.2	6.2
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
nettle	3.8	3.8
net-tools	2.0	2.0
newt	0.52.21	0.52.21
nfs-utils	2.5.4	2.5.4
npth	1.6	1.6

패키지	AMI	하이퍼-V VHDX
nspr	4.35.0	4.35.0
nss	3.90.0	3.90.0
nss-softokn	3.90.0	3.90.0
nss-softokn-freebl	3.90.0	3.90.0
nss-sysinit	3.90.0	3.90.0
nss-util	3.90.0	3.90.0
ntsysv	1.15	1.15
numactl-libs	2.0.14	2.0.14
ocaml-srpm-macros	6	6
oniguruma	6.9.7.1	6.9.7.1
openblas-srpm-macros	2	2
openldap	2.4.57	2.4.57
openssh	8.7p1	8.7p1
openssh-clients	8.7p1	8.7p1
openssh-server	8.7p1	8.7p1
openssl	3.0.8	3.0.8
openssl-libs	3.0.8	3.0.8
openssl-pkcs11	0.4.12	0.4.12
os-prober	1.77	1.77
p11-kit	0.24.1	0.24.1

패키지	AMI	하이퍼-V VHDX
p11-kit-trust	0.24.1	0.24.1
package-notes-srpm-macros	0.4	0.4
pam	1.5.1	1.5.1
parted	3.4	3.4
passwd	0.80	0.80
pciutils	3.7.0	3.7.0
pciutils-libs	3.7.0	3.7.0
pcre2	10.40	10.40
pcre2-syntax	10.40	10.40
perl-Carp	1.50	1.50
perl-Class-Struct	0.66	0.66
perl-constant	1.33	1.33
perl-DynaLoader	1.47	1.47
perl-Encode	3.15	3.15
perl-Errno	1.30	1.30
perl-Exporter	5.74	5.74
perl-Fcntl	1.13	1.13
perl-File-Basename	2.85	2.85
perl-File-Path	2.18	2.18
perl-File-stat	1.09	1.09

패키지	AMI	하이퍼-V VHDX
perl-File-Temp	0.231.100	0.231.100
perl-Getopt-Long	2.52	2.52
perl-Getopt-Std	1.12	1.12
perl-HTTP-Tiny	0.078	0.078
perl-if	0.60.800	0.60.800
perl-interpreter	5.32.1	5.32.1
perl-IO	1.43	1.43
perl-IPC-Open3	1.21	1.21
perl-libs	5.32.1	5.32.1
perl-MIME-Base64	3.16	3.16
perl-mro	1.23	1.23
perl-overload	1.31	1.31
perl-overloading	0.02	0.02
perl-parent	0.238	0.238
perl-PathTools	3.78	3.78
perl-Pod-Escapes	1.07	1.07
perl-podlators	4.14	4.14
perl-Pod-Perldoc	3.28.01	3.28.01
perl-Pod-Simple	3.42	3.42
perl-Pod-Usage	2.01	2.01

패키지	AMI	하이퍼-V VHDX
perl-POSIX	1.94	1.94
perl-Scalar-List-Utils	1.56	1.56
perl-SelectSaver	1.02	1.02
perl-Socket	2.032	2.032
perl-srpm-macros	1	1
perl-Storable	3.21	3.21
perl-subst	1.03	1.03
perl-Symbol	1.08	1.08
perl-Term-ANSIColor	5.01	5.01
perl-Term-Cap	1.17	1.17
perl-Text-ParseWords	3.30	3.30
perl-Text-Tabs+Wrap	2021.0726	2021.0726
perl-Time-Local	1.300	1.300
perl-vars	1.05	1.05
pkgconf	1.8.0	1.8.0
pkgconf-m4	1.8.0	1.8.0
pkgconf-pkg-config	1.8.0	1.8.0
policycoreutils	3.4	3.4
policycoreutils-python-utils	3.4	

패키지	AMI	하이퍼-V VHDX
popt	1.18	1.18
procps-ng	3.3.17	3.3.17
protobuf-c	1.4.1	1.4.1
psacct	6.6.4	6.6.4
psmisc	23.4	23.4
publicsuffix-list-dafsa	20240212	20240212
python3	3.9.16	3.9.16
python3-attrs	20.3.0	20.3.0
python3-audit	3.0.6	3.0.6
python3-awscrt	0.19.19	0.19.19
python3-babel	2.9.1	2.9.1
python3-cffi	1.14.5	1.14.5
python3-chardet	4.0.0	4.0.0
python3-colorama	0.4.4	0.4.4
python3-configobj	5.0.6	5.0.6
python3-cryptography	36.0.1	36.0.1
python3-daemon	2.3.0	
python3-dateutil	2.8.1	2.8.1
python3-dbus	1.2.18	1.2.18
python3-distro	1.5.0	1.5.0

패키지	AMI	하이퍼-V VHDX
python3-dnf	4.14.0	4.14.0
python3-dnf-plugins-core	4.3.0	4.3.0
python3-docutils	0.16	0.16
python3-gpg	1.15.1	1.15.1
python3-hawkey	0.69.0	0.69.0
python3-idna	2.10	2.10
python3-jinja2	2.11.3	2.11.3
python3-jmespath	0.10.0	0.10.0
python3-jsonpatch	1.21	1.21
python3-jsonpointer	2.0	2.0
python3-jjsonschema	3.2.0	3.2.0
python3-libcomps	0.1.20	0.1.20
python3-libdnf	0.69.0	0.69.0
python3-libs	3.9.16	3.9.16
python3-libselenium	3.4	3.4
python3-libsemanage	3.4	3.4
python3-libstorage mgmt	1.9.4	1.9.4
python3-lockfile	0.12.2	
python3-markupsafe	1.1.1	1.1.1

패키지	AMI	하이퍼-V VHDX
python3-netifaces	0.10.6	0.10.6
python3-oauthlib	3.0.2	3.0.2
python3-pip-wheel	21.3.1	21.3.1
python3-ply	3.11	3.11
python3-policycore utils	3.4	3.4
python3-prettytable	0.7.2	0.7.2
python3-prompt-too lkit	3.0.24	3.0.24
python3-pycparser	2.20	2.20
python3-pyrsistent	0.17.3	0.17.3
python3-pyserial	3.4	3.4
python3-pysocks	1.7.1	1.7.1
python3-pytz	2022.7.1	2022.7.1
python3-pyyaml	5.4.1	5.4.1
python3-requests	2.25.1	2.25.1
python3-rpm	4.16.1.3	4.16.1.3
python3-ruamel-yaml	0.16.6	0.16.6
python3-ruamel-yaml- clib	0.1.2	0.1.2
python3-setools	4.4.1	4.4.1

패키지	AMI	하이퍼-V VHDX
python3-setuptools	59.6.0	59.6.0
python3-setuptools-wheel	59.6.0	59.6.0
python3-six	1.15.0	1.15.0
python3-systemd	235	235
python3-urllib3	1.25.10	1.25.10
python3-wcwidth	0.2.5	0.2.5
python-chevron	0.13.1	
python-srpm-macros	3.9	3.9
quota	4.06	4.06
quota-nls	4.06	4.06
readline	8.1	8.1
rng-tools	6.14	6.14
rootfiles	8.1	8.1
rpcbind	1.2.6	1.2.6
rpm	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3
rpm-libs	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3	4.16.1.3
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3

패키지	AMI	하이퍼-V VHDX
rpm-sign-libs	4.16.1.3	4.16.1.3
rsync	3.2.6	3.2.6
rust-srpm-macros	21	21
sbsigntools	0.9.4	0.9.4
screen	4.8.0	4.8.0
sed	4.8	4.8
selinux-policy	37.22	37.22
selinux-policy-targeted	37.22	37.22
setup	2.13.7	2.13.7
shadow-utils	4.9	4.9
slang	2.3.2	2.3.2
sqlite-libs	3.40.0	3.40.0
sssd-client	2.9.4	2.9.4
sssd-common	2.9.4	
sssd-kcm	2.9.4	
sssd-nfs-idmap	2.9.4	
strace	5.16	5.16
sudo	1.9.14	1.9.14
sysctl-defaults	1.0	1.0
sysstat	12.5.6	12.5.6

패키지	AMI	하이퍼-V VHDX
systemd	252.16	252.16
systemd-libs	252.16	252.16
systemd-networkd	252.16	252.16
systemd-pam	252.16	252.16
systemd-resolved	252.16	252.16
systemd-udev	252.16	252.16
system-release	2023.4.20240319	2023.4.20240319
systemtap-runtime	4.8	4.8
tar	1.34	1.34
tbb	2020.3	2020.3
tcpdump	4.99.1	4.99.1
tcsh	6.24.07	6.24.07
time	1.9	1.9
traceroute	2.1.3	2.1.3
tzdata	2024a	2024a
unzip	6.0	6.0
update-motd	2.2	2.2
userspace-rcu	0.12.1	0.12.1
util-linux	2.37.4	2.37.4
util-linux-core	2.37.4	2.37.4

패키지	AMI	하이퍼-V VHDX
vim-common	9.0.2153	9.0.2153
vim-data	9.0.2153	9.0.2153
vim-enhanced	9.0.2153	9.0.2153
vim-filesystem	9.0.2153	9.0.2153
vim-minimal	9.0.2153	9.0.2153
wget	1.21.3	1.21.3
which	2.21	2.21
words	3.0	3.0
xfsdump	3.1.11	3.1.11
xfspgrog	5.18.0	5.18.0
xxd	9.0.2153	9.0.2153
xxhash-libs	0.8.0	0.8.0
xz	5.2.5	5.2.5
xz-libs	5.2.5	5.2.5
yum	4.14.0	4.14.0
zip	3.0	3.0
zlib	1.2.11	1.2.11
zram-generator	1.1.2	
zram-generator-defaults	1.1.2	
zstd	1.5.5	1.5.5

AL2023 업데이트

보안 업데이트와 새로운 기능을 활용하려면 AL2023 릴리스를 최신 상태로 유지하는 것이 중요합니다. AL2023을 사용하면 [AL2023 버전 지정 저장소를 통한 결정적 업그레이드 사용](#)을 통해 환경 전반에서 패키지 버전과 업데이트를 일치시킬 수 있습니다.

주제

- [새 업데이트에 대한 알림 받기](#)
- [AL2023 패키지 및 운영 체제 업데이트 관리](#)
- [AL2023 버전 지정 저장소를 통한 결정적 업그레이드 사용](#)
- [AL2023 기반 커널 라이브 패칭](#)

새 업데이트에 대한 알림 받기

새 AL2023 AMI가 릴리스될 때마다 알림을 받을 수 있습니다. 다음 주제에 대한 알림이 [Amazon SNS](#)에 게시됩니다.

```
arn:aws:sns:us-east-1:137112412989:amazon-linux-2023-ami-updates
```

새 AL2023 AMI가 게시되면 메시지가 여기에 게시됩니다. AMI 버전이 메시지에 포함됩니다.

이러한 메시지는 여러 방식으로 수신할 수 있습니다. 다음 방법을 사용하는 것이 좋습니다.

1. [Amazon SNS 콘솔](#)을 엽니다.
2. 필요한 경우 탐색 막대에서 미국 동부 (버지니아 북부) AWS 리전 로 변경합니다. 구독중인 SNS 알림을 만든 리전을 선택해야 합니다.
3. 탐색 창에서 구독과 구독 생성을 선택합니다.
4. 구독 생성 대화 상자에서 다음과 같이 수행합니다.
 - a. 주제 ARN의 경우 다음 Amazon 리소스 이름 (ARN) 을 복사하여 붙여넣습니다.
arn:aws:sns:us-east-1:137112412989:amazon-linux-2023-ami-updates
 - b. 프로토콜에서 이메일을 선택합니다.
 - c. 엔드포인트에 알림 받을 이메일 주소를 입력합니다.
 - d. Create subscription을 선택합니다.

5. 제목이 “AWS 알림 - 구독 확인”인 확인 이메일을 받게 됩니다. 이메일을 열고 구독 확인을 선택하여 구독 신청을 완료합니다.

AL2023 패키지 및 운영 체제 업데이트 관리

이전 버전의 아마존 리눅스와 달리 AL2023 AMI는 특정 버전의 아마존 리눅스 리포지토리에 고정되어 있습니다. AL2023 인스턴스 보안 및 버그를 모두 수정하려면 DNF 구성을 업데이트하세요. 또는 최신 AL2023 인스턴스를 실행하세요.

이 섹션에서는 실행 중인 인스턴스에서 DNF 패키지 및 리포지토리를 관리하는 방법에 대해 설명합니다. 또한 시작 시 사용 가능한 최신 Amazon Linux 리포지토리를 활성화하도록 사용자 데이터 스크립트에서 DNF를 구성하는 방법도 설명합니다. 자세한 내용은 [DNF 명령 레퍼런스](#)(DNF Command Reference)를 참조하세요.

주제

- [사용 가능한 패키지 업데이트 확인](#)
- [DNF 및 리포지토리 버전에서 보안 업데이트](#)
- [\(보안\) 업데이트 후 자동 서비스 재시작](#)
- [최신 리포지토리 버전이 활성화된 상태로 인스턴스 시작](#)
- [패키지 지원 정보 가져오기](#)
- [최신 리포지토리 버전 확인](#)
- [새 리포지토리 추가, 사용 또는 사용 중지](#)
- [cloud-init를 사용하여 리포지토리 추가](#)

사용 가능한 패키지 업데이트 확인

`dnf check-update` 명령을 사용하여 시스템 업데이트가 있는지 확인할 수 있습니다. AL2023의 경우 명령에 `--releasever=version-number` 옵션을 추가하는 것이 좋습니다.

이 옵션을 추가하면 DNF가 최신 저장소의 업데이트도 확인합니다. 예를 들어, `dnf check-update` 명령을 실행하면 반환된 최신 버전을 `version-number`의 값으로 사용합니다.

최신 버전의 리포지토리를 사용하도록 인스턴스를 업데이트하면 출력에는 업데이트할 모든 패키지 목록이 포함됩니다.

Note

dnf check-update 명령에 플래그 옵션으로 릴리스 버전을 지정하지 않으면 현재 구성된 리포지토리 버전만 확인합니다. 즉, 최신 버전의 리포지토리에 있는 패키지는 검사되지 않습니다.

```
$ sudo dnf check-update --releasever=2023.0.20230210
```

```
Last metadata expiration check: 0:06:13 ago on Mon 13 Feb 2023 10:39:32 PM UTC.
```

```
bind-libs.x86_64                32:9.16.27-1.amzn2023      amazonlinux
bind-license.noarch             32:9.16.27-1.amzn2023      amazonlinux
bind-utils.x86_64              32:9.16.27-1.amzn2023      amazonlinux
cloud-init.noarch              22.2.2-1.amzn2023.1.4      amazonlinux
dnf.noarch                      4.12.0-2.amzn2023.0.1      amazonlinux
dnf-data.noarch                4.12.0-2.amzn2023.0.1      amazonlinux
dracut.x86_64                  055-6.amzn2023.0.4         amazonlinux
dracut-config-generic.x86_64   055-6.amzn2023.0.4         amazonlinux
glib2.x86_64                   2.73.2-678.amzn2023        amazonlinux
gmp.x86_64                     1:6.2.1-2.amzn2023         amazonlinux
grep.x86_64                    3.8-1.amzn2023.0.1         amazonlinux
kpatch-runtime.noarch          0.9.4-7.amzn2023           amazonlinux
libgcc.x86_64                  11.3.1-2.amzn2023.0.6      amazonlinux
libgomp.x86_64                 11.3.1-2.amzn2023.0.6      amazonlinux
libpkgconf.x86_64              1.7.3-7.amzn2023.0.1       amazonlinux
libstdc++.x86_64               11.3.1-2.amzn2023.0.6      amazonlinux
lz4-libs.x86_64                1.9.4-1.amzn2023           amazonlinux
pkgconf.x86_64                 1.7.3-7.amzn2023.0.1       amazonlinux
pkgconf-m4.noarch              1.7.3-7.amzn2023.0.1       amazonlinux
pkgconf-pkg-config.x86_64      1.7.3-7.amzn2023.0.1       amazonlinux
python3-dnf.noarch             4.12.0-2.amzn2023.0.1      amazonlinux
python3-rpm.x86_64             4.16.1.3-12.amzn2023.0.2    amazonlinux
rpm.x86_64                     4.16.1.3-12.amzn2023.0.2    amazonlinux
rpm-build-libs.x86_64          4.16.1.3-12.amzn2023.0.2    amazonlinux
rpm-libs.x86_64                4.16.1.3-12.amzn2023.0.2    amazonlinux
rpm-plugin-selinux.x86_64      4.16.1.3-12.amzn2023.0.2    amazonlinux
rpm-plugin-systemd-inhibit.x86_64 4.16.1.3-12.amzn2023.0.2    amazonlinux
rpm-sign-libs.x86_64           4.16.1.3-12.amzn2023.0.2    amazonlinux
slang.x86_64                   2.3.2-9.amzn2023.0.1       amazonlinux
system-release.noarch          2023.0.20230210-0.amzn2023  amazonlinux
systemd.x86_64                 250.8-1.amzn2023.0.1       amazonlinux
systemd-libs.x86_64           250.8-1.amzn2023.0.1       amazonlinux
```

systemd-networkd.x86_64	250.8-1.amzn2023.0.1	amazonlinux
systemd-pam.x86_64	250.8-1.amzn2023.0.1	amazonlinux
systemd-resolved.x86_64	250.8-1.amzn2023.0.1	amazonlinux
systemd-udev.x86_64	250.8-1.amzn2023.0.1	amazonlinux
vim-common.x86_64	2:9.0.327-1.amzn2023.0.1	amazonlinux
vim-data.noarch	2:9.0.327-1.amzn2023.0.1	amazonlinux
vim-enhanced.x86_64	2:9.0.327-1.amzn2023.0.1	amazonlinux
vim-filessystem.noarch	2:9.0.327-1.amzn2023.0.1	amazonlinux
vim-minimal.x86_64	2:9.0.327-1.amzn2023.0.1	amazonlinux
wget.x86_64	1.21.3-1.amzn2023	amazonlinux
yum.noarch	4.12.0-2.amzn2023.0.1	amazonlinux

이 명령의 경우 새 패키지가 있다면 반환 코드는 100입니다. 새 패키지가 없다면 반환 코드는 0입니다. 또한 업데이트할 모든 패키지도 출력에 나열됩니다.

DNF 및 리포지토리 버전에서 보안 업데이트

최신 패키지 업데이트 및 보안 업데이트는 최신 저장소 버전에만 제공됩니다. AL2023 AMI 구 버전에서 실행한 인스턴스의 경우 보안 업데이트를 설치하기 전에 리포지토리 버전을 업데이트해야 합니다. 이 `dnf check-release-update` 명령에는 시스템에 설치된 모든 패키지를 최신 저장소 버전으로 업데이트하는 업데이트 명령이 포함되어 있습니다.

```
$ sudo dnf update --releasever=2023.0.20230210
```

```
Last metadata expiration check: 0:01:40 ago on Mon 13 Feb 2023 10:39:32 PM UTC.
Dependencies resolved.
```

```
=====
Package                Arch   Version                                Repository   Size
=====
Upgrading:
bind-libs               x86_64 32:9.16.27-1.amzn2023                amazonlinux 1.2 M
bind-license            noarch 32:9.16.27-1.amzn2023                amazonlinux 16 k
bind-utils              x86_64 32:9.16.27-1.amzn2023                amazonlinux 202 k
cloud-init              noarch 22.2.2-1.amzn2023.1.4                amazonlinux 1.1 M
dnf                     noarch 4.12.0-2.amzn2023.0.1                amazonlinux 454 k
dnf-data                noarch 4.12.0-2.amzn2023.0.1                amazonlinux 42 k
dracut                  x86_64 055-6.amzn2023.0.4                   amazonlinux 345 k
dracut-config-generic  x86_64 055-6.amzn2023.0.4                   amazonlinux 8.5 k
glib2                   x86_64 2.73.2-678.amzn2023                  amazonlinux 2.7 M
gmp                     x86_64 1:6.2.1-2.amzn2023                   amazonlinux 324 k
grep                    x86_64 3.8-1.amzn2023.0.1                   amazonlinux 316 k
kpatch-runtime          noarch 0.9.4-7.amzn2023                     amazonlinux 30 k
libgcc                  x86_64 11.3.1-2.amzn2023.0.6                amazonlinux 121 k
=====
```

```

libgomp                x86_64 11.3.1-2.amzn2023.0.6    amazonlinux 296 k
libpkgconf              x86_64 1.7.3-7.amzn2023.0.1          amazonlinux  37 k
libstdc++              x86_64 11.3.1-2.amzn2023.0.6        amazonlinux 758 k
lz4-libs               x86_64 1.9.4-1.amzn2023              amazonlinux  81 k
pkgconf                x86_64 1.7.3-7.amzn2023.0.1          amazonlinux  41 k
pkgconf-m4             noarch 1.7.3-7.amzn2023.0.1          amazonlinux  15 k
pkgconf-pkg-config     x86_64 1.7.3-7.amzn2023.0.1          amazonlinux  11 k
python3-dnf            noarch 4.12.0-2.amzn2023.0.1         amazonlinux 415 k
python3-rpm            x86_64 4.16.1.3-12.amzn2023.0.2      amazonlinux  89 k
rpm                    x86_64 4.16.1.3-12.amzn2023.0.2      amazonlinux 487 k
rpm-build-libs         x86_64 4.16.1.3-12.amzn2023.0.2      amazonlinux  92 k
rpm-libs               x86_64 4.16.1.3-12.amzn2023.0.2      amazonlinux 311 k
rpm-plugin-selinux     x86_64 4.16.1.3-12.amzn2023.0.2      amazonlinux  18 k
rpm-plugin-systemd-inhibit x86_64 4.16.1.3-12.amzn2023.0.2      amazonlinux  19 k
rpm-sign-libs          x86_64 4.16.1.3-12.amzn2023.0.2      amazonlinux  22 k
slang                  x86_64 2.3.2-9.amzn2023.0.1          amazonlinux 410 k
system-release         noarch 2023.0.20230210-0.amzn2023    amazonlinux  25 k
systemd                x86_64 250.8-1.amzn2023.0.1          amazonlinux 4.2 M
systemd-libs           x86_64 250.8-1.amzn2023.0.1          amazonlinux 615 k
systemd-networkd       x86_64 250.8-1.amzn2023.0.1          amazonlinux 614 k
systemd-pam            x86_64 250.8-1.amzn2023.0.1          amazonlinux 335 k
systemd-resolved       x86_64 250.8-1.amzn2023.0.1          amazonlinux 277 k
systemd-udev           x86_64 250.8-1.amzn2023.0.1          amazonlinux 1.9 M
vim-common             x86_64 2:9.0.327-1.amzn2023.0.1      amazonlinux 7.2 M
vim-data              noarch 2:9.0.327-1.amzn2023.0.1      amazonlinux  27 k
vim-enhanced          x86_64 2:9.0.327-1.amzn2023.0.1      amazonlinux 1.8 M
vim-filesystem        noarch 2:9.0.327-1.amzn2023.0.1      amazonlinux  21 k
vim-minimal           x86_64 2:9.0.327-1.amzn2023.0.1      amazonlinux 764 k
wget                  x86_64 1.21.3-1.amzn2023             amazonlinux 813 k
yum                   noarch 4.12.0-2.amzn2023.0.1         amazonlinux  39 k

```

Transaction Summary

```

=====
Upgrade 43 Packages
...

```

보안 기능만 포함된 패키지를 업데이트하는 `--security` 옵션을 추가할 수 있습니다.

```

$ sudo dnf update --releasever=2023.0.20230210 --security
Amazon Linux 2023 repository           18 MB/s | 11 MB    00:00
Last metadata expiration check: 0:00:02 ago on Mon 13 Feb 2023 10:39:32 PM UTC.
Dependencies resolved.
=====

```

Package	Arch	Version	Repository	Size
Upgrading:				
bind-libs	x86_64	32:9.16.27-1.amzn2023	amazonlinux	1.2 M
bind-license	noarch	32:9.16.27-1.amzn2023	amazonlinux	16 k
bind-utils	x86_64	32:9.16.27-1.amzn2023	amazonlinux	202 k
gmp	x86_64	1:6.2.1-2.amzn2023	amazonlinux	324 k
lz4-libs	x86_64	1.9.4-1.amzn2023	amazonlinux	81 k
vim-common	x86_64	2:9.0.327-1.amzn2023.0.1	amazonlinux	7.2 M
vim-data	noarch	2:9.0.327-1.amzn2023.0.1	amazonlinux	27 k
vim-enhanced	x86_64	2:9.0.327-1.amzn2023.0.1	amazonlinux	1.8 M
vim-filesystem	noarch	2:9.0.327-1.amzn2023.0.1	amazonlinux	21 k
vim-minimal	x86_64	2:9.0.327-1.amzn2023.0.1	amazonlinux	764 k
wget	x86_64	1.21.3-1.amzn2023	amazonlinux	813 k

Transaction Summary

Upgrade 11 Packages

...

AL2023 패키지 버전을 확인하려면 다음 중 하나 이상을 수행하세요.

- `dnf check-update` 명령을 실행합니다.
- Amazon Linux 리포지토리 업데이트 SNS 주제(`arn:aws:sns:us-east-1:137112412989:amazon-linux-2023-ami-updates`)를 구독하세요. 자세한 설명은 Amazon Simple Notification Service 개발자 안내서에서 [Amazon SNS 주제 구독](#)을 참조하세요.
- [AL2023 릴리스 정보](#)를 정기적으로 확인하세요.

Important

실행 중인 인스턴스에 보안 업데이트를 적용할 때는 DNF가 최신 리포지토리 버전을 가리키는 지 확인하십시오.

(보안) 업데이트 후 자동 서비스 재시작

Amazon Linux는 이제 [스마트 재시작](#) 패키지와 함께 제공됩니다. `Smart-restart` 시스템 패키지 관리자를 사용하여 패키지를 설치하거나 삭제할 때마다 시스템 업데이트 시 `systemd` 서비스를 다시 시작합니다. 이는 실행될 때마다 발생합니다 `dnf (update|upgrade|downgrade)`.

Smart-restart의 needs-restarting 패키지와 사용자 지정 리스팅 거부 메커니즘을 사용하여 재 시작해야 하는 서비스와 시스템 재부팅 권장 여부를 결정합니다. dnf-utils 시스템 재부팅을 권장하면 재부팅 힌트 마커 파일이 생성됩니다 (/run/smart-restart/reboot-hint-marker

smart-restart을(를) 설치하려면

다른 패키지와 마찬가지로 다음 DNF 명령을 실행합니다.

```
$ sudo dnf install smart-restart
```

설치 후 후속 트랜잭션이 smart-restart 로직을 트리거합니다.

거부 목록

Smart-restart특정 서비스가 다시 시작되지 않도록 차단하도록 지시할 수 있습니다. 차단된 서비스는 재부팅이 필요한지 여부를 결정하는 데 영향을 주지 않습니다. 추가 서비스를 차단하려면 다음 예와 /etc/smart-restart-conf.d/ 같이 -denylist 접미사가 포함된 파일을 추가합니다.

```
$ cat /etc/smart-restart-conf.d/custom-denylist
# Some comments
myservice.service
```

Note

재부팅이 필요한지 여부를 결정할 때 모든 *-denylist 파일을 읽고 평가합니다.

커스텀 후크

리스팅 거부 외에도 서비스 재시작을 시도하기 전후에 사용자 지정 스크립트를 실행하는 메커니즘을 smart-restart 제공합니다. 사용자 지정 스크립트를 사용하여 준비 단계를 수동으로 수행하거나 남은 재시작 또는 완료된 재시작을 다른 구성 요소에 알릴 수 있습니다.

접미사가 -pre-restart 또는 인 모든 스크립트가 실행됩니다-post-restart. /etc/smart-restart-conf.d/ 순서가 중요한 경우 다음 예와 같이 모든 스크립트 앞에 번호를 붙여서 실행 순서를 확인하십시오.

```
$ ls /etc/smart-restart-conf.d/*-pre-restart
001-my-script-pre-restart
002-some-other-script-pre-restart
```

최신 리포지토리 버전이 활성화된 상태로 인스턴스 시작

사용자 데이터 스크립트에 DNF 명령을 추가하면 Amazon Linux AMI 실행 시 설치할 RPM 패키지를 선택할 수 있습니다. 다음은 사용자 데이터 스크립트로 user-data 스크립트로 실행한 모든 인스턴스에 동일한 패키지 업데이트가 설치되어 있는지 확인하는 예시입니다.

```
#!/bin/bash
dnf update --releasever=2023.0.20230210
# Additional setup and install commands below
dnf install httpd php7.4 mysql80
```

이 스크립트는 슈퍼 유저(루트)로 실행해야 합니다. 다음 명령으로 실행하세요.

```
$ sudo sh -c "bash nameofscript.sh"
```

자세한 내용은 Amazon EC2 [사용 설명서의 사용자 데이터 및 셸 스크립트](#)를 참조하십시오.

Note

사용자 데이터 스크립트 대신 최신 Amazon Linux AMI 버전 또는 Amazon Linux AMI 기반의 사용자 지정 AMI를 실행하세요. 최신 Amazon Linux AMI에 필요한 업데이트가 모두 설치되어 있으며 특정 리포지토리 버전에서 실행되도록 구성되어 있습니다.

패키지 지원 정보 가져오기

다양한 오픈 소스 소프트웨어 프로젝트가 AL2023에 통합되어 있습니다. 각 프로젝트는 Amazon Linux와 독립적으로 관리되며 릴리스 및 end-of-support 일정이 다릅니다. DNF supportinfo 플러그인이 제공하는 패키지 메타데이터를 통해 이러한 패키지에 대한 Amazon Linux 정보를 알 수 있습니다. 다음은 **dnf supportinfo** 명령이 glibc 패키지의 메타데이터를 반환하는 예시입니다.

```
$ sudo dnf supportinfo --pkg glibc
Last metadata expiration check: 0:07:56 ago on Wed Mar 1 23:21:49 2023.
Name           : glibc
Version        : 2.34-52.amzn2023.0.2
State          : installed
Support Status  : supported
Support Periods : from 2023-03-15      : supported
                : from 2028-03-15      : unsupported
Support Statement : Amazon Linux 2023 End Of Life
```

```

Link           : https://aws.amazon.com/amazon-linux-ami/faqs/
Other Info     : This is the support statement for AL2023. The
                ...: end of life of Amazon Linux 2023 would be March 2028.
                ...: From this point, the Amazon Linux 2023 packages (listed
                ...: below) will no longer, receive any updates from AWS.

```

최신 리포지토리 버전 확인

AL2023 인스턴스는 DNF 유틸리티를 사용하여 리포지토리를 관리하고 RPM 패키지 업데이트를 할 수 있습니다. Amazon Linux 리포지토리에서 이 패키지를 사용할 수 있습니다. DNF 명령 `dnf check-release-update`로 최신 DNF 리포지토리 버전을 확인할 수 있습니다.

```

$ sudo dnf check-release-update
WARNING:
  A newer release of "Amazon Linux" is available.

  Available Versions:

  Version 2023.0.20230210:
    Run the following command to update to 2023.0.20230210:

      dnf update --releasever=2023.0.20230210

  Release notes:
    https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes.html

```

그러면 사용 가능한 모든 최신 DNF 리포지토리 버전을 알 수 있습니다. 결과가 나오지 않으면 DNF 이 현재 사용 가능한 최신 버전이라는 뜻입니다. 현재 설치된 `system-release` 패키지 버전은 `releasever` DNF 변수로 설정되어 있습니다. 다음 명령을 실행하여 현재 리포지토리 버전을 확인합니다.

```
$ rpm -q system-release --qf "%{VERSION}\n"
```

DNF 패키지 트랜잭션(예: 설치, 업데이트 또는 제거 명령)을 실행하면 최신 리포지토리 버전을 알리는 메시지가 나타납니다. 예를 들어, AL2023 구 버전으로 실행한 인스턴스에 `httpd` 패키지를 설치하면 다음과 같은 출력이 나타납니다.

```

$ sudo dnf install httpd -y
Last metadata expiration check: 0:16:52 ago on Wed Mar 1 23:21:49 2023.
Dependencies resolved.
=====

```

```

Package                Arch  Version                Repository  Size
=====
Installing:
httpd                  x86_64 2.4.54-3.amzn2023.0.4 amazonlinux 46 k
Installing dependencies:
apr                   x86_64 1.7.2-2.amzn2023.0.2  amazonlinux 129 k
apr-util              x86_64 1.6.3-1.amzn2023.0.1  amazonlinux 98 k
generic-logos-httpd
                        noarch 18.0.0-12.amzn2023.0.3 amazonlinux 19 k
httpd-core            x86_64 2.4.54-3.amzn2023.0.4 amazonlinux 1.3 M
httpd-filesystem     noarch 2.4.54-3.amzn2023.0.4 amazonlinux 13 k
httpd-tools          x86_64 2.4.54-3.amzn2023.0.4 amazonlinux 80 k
libbrotli             x86_64 1.0.9-4.amzn2023.0.2  amazonlinux 315 k
mailcap              noarch 2.1.49-3.amzn2023.0.3 amazonlinux 33 k
Installing weak dependencies:
apr-util-openssl     x86_64 1.6.3-1.amzn2023.0.1  amazonlinux 17 k
mod_http2            x86_64 1.15.24-1.amzn2023.0.3 amazonlinux 152 k
mod_lua              x86_64 2.4.54-3.amzn2023.0.4 amazonlinux 60 k

```

Transaction Summary

```

=====
Install 12 Packages

```

Total download size: 2.3 M

Installed size: 6.8 M

Downloading Packages:

```

(1/12): apr-util-openssl-1.6.3-1.am 212 kB/s | 17 kB    00:00
(2/12): apr-1.7.2-2.amzn2023.0.2.x8 1.1 MB/s | 129 kB    00:00
(3/12): httpd-core-2.4.54-3.amzn202 8.9 MB/s | 1.3 MB    00:00
(4/12): mod_http2-1.15.24-1.amzn202 1.9 MB/s | 152 kB    00:00
(5/12): apr-util-1.6.3-1.amzn2023.0 1.7 MB/s | 98 kB     00:00
(6/12): mod_lua-2.4.54-3.amzn2023.0 1.4 MB/s | 60 kB     00:00
(7/12): httpd-2.4.54-3.amzn2023.0.4 1.5 MB/s | 46 kB     00:00
(8/12): libbrotli-1.0.9-4.amzn2023. 4.4 MB/s | 315 kB    00:00
(9/12): mailcap-2.1.49-3.amzn2023.0 753 kB/s | 33 kB     00:00
(10/12): httpd-tools-2.4.54-3.amzn2 978 kB/s | 80 kB     00:00
(11/12): httpd-filesystem-2.4.54-3. 210 kB/s | 13 kB     00:00
(12/12): generic-logos-httpd-18.0.0 439 kB/s | 19 kB     00:00

```

```

-----
Total                                6.6 MB/s | 2.3 MB    00:00

```

Running transaction check

Transaction check succeeded.

Running transaction test

Transaction test succeeded.

Running transaction

```

Preparing          :                               1/12
Installing         : apr-1.7.2-2.amzn2023.0.2.x86_64 1/12
Installing         : apr-util-openssl-1.6.3-1.amzn2023.0.1. 2/12
Installing         : apr-util-1.6.3-1.amzn2023.0.1.x86_64 3/12
Installing         : mailcap-2.1.49-3.amzn2023.0.3.noarch 4/12
Installing         : httpd-tools-2.4.54-3.amzn2023.0.4.x86_ 5/12
Installing         : generic-logos-httpd-18.0.0-12.amzn2023 6/12
Running scriptlet: httpd-filesystem-2.4.54-3.amzn2023.0.4 7/12
Installing         : httpd-filesystem-2.4.54-3.amzn2023.0.4 7/12
Installing         : httpd-core-2.4.54-3.amzn2023.0.4.x86_6 8/12
Installing         : mod_http2-1.15.24-1.amzn2023.0.3.x86_6 9/12
Installing         : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 10/12
Installing         : mod_lua-2.4.54-3.amzn2023.0.4.x86_64 11/12
Installing         : httpd-2.4.54-3.amzn2023.0.4.x86_64 12/12
Running scriptlet: httpd-2.4.54-3.amzn2023.0.4.x86_64 12/12
Verifying         : apr-1.7.2-2.amzn2023.0.2.x86_64 1/12
Verifying         : apr-util-openssl-1.6.3-1.amzn2023.0.1. 2/12
Verifying         : httpd-core-2.4.54-3.amzn2023.0.4.x86_6 3/12
Verifying         : mod_http2-1.15.24-1.amzn2023.0.3.x86_6 4/12
Verifying         : apr-util-1.6.3-1.amzn2023.0.1.x86_64 5/12
Verifying         : mod_lua-2.4.54-3.amzn2023.0.4.x86_64 6/12
Verifying         : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 7/12
Verifying         : httpd-2.4.54-3.amzn2023.0.4.x86_64 8/12
Verifying         : httpd-tools-2.4.54-3.amzn2023.0.4.x86_ 9/12
Verifying         : mailcap-2.1.49-3.amzn2023.0.3.noarch 10/12
Verifying         : httpd-filesystem-2.4.54-3.amzn2023.0.4 11/12
Verifying         : generic-logos-httpd-18.0.0-12.amzn2023 12/12

```

Installed:

```

apr-1.7.2-2.amzn2023.0.2.x86_64
apr-util-1.6.3-1.amzn2023.0.1.x86_64
apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
httpd-2.4.54-3.amzn2023.0.4.x86_64
httpd-core-2.4.54-3.amzn2023.0.4.x86_64
httpd-filesystem-2.4.54-3.amzn2023.0.4.noarch
httpd-tools-2.4.54-3.amzn2023.0.4.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch
mod_http2-1.15.24-1.amzn2023.0.3.x86_64
mod_lua-2.4.54-3.amzn2023.0.4.x86_64

```

Complete!

새 리포지토리 추가, 사용 또는 사용 중지

DNF 패키지 관리 시스템으로 다른 리포지토리의 패키지를 설치하려면 `/etc/dnf/dnf.conf` 파일 또는 `/etc/yum.repos.d` 디렉터리의 `repository.repo` 파일에 리포지토리 정보를 추가해야 합니다. 이 작업을 수동으로 수행할 수 있습니다. 하지만 대부분의 DNF 리포지토리는 리포지토리 URL을 통해 자체 `repository.repo` 파일을 제공합니다.

Note

현재로서는 AL2023에 추가할 수 있는 리포지토리가 없습니다. 그러나 이 점은 추후 개선될 것입니다. 또한 패키지를 직접 작성하여 AL2023 엔터프라이즈 환경에서 사용할 수 있습니다. 패키지를 사용하려면 먼저 패키지를 저장한 리포지토리를 추가하고 활성화해야 합니다.

다음 명령을 실행하여 현재 활성화된 리포지토리를 확인할 수 있습니다.

```
$ dnf repolist all --verbose
```

```
Loaded plugins: builddep, changelog, config-manager, copr, debug, debuginfo-install,
download, generate_completion_cache, groups-manager, needs-restarting, playground,
release-notification, repoclosure, repodiff, repograph, repomanage, reposync,
supportinfo
```

```
DNF version: 4.12.0
```

```
cachedir: /var/cache/dnf
```

```
Last metadata expiration check: 0:00:02 ago on Wed Mar 1 23:40:15 2023.
```

```
Repo-id           : amazonlinux
```

```
Repo-name        : Amazon Linux 2023 repository
```

```
Repo-status      : enabled
```

```
Repo-revision    : 1677203368
```

```
Repo-updated     : Fri Feb 24 01:49:28 2023
```

```
Repo-pkgs       : 12632
```

```
Repo-available-pkgs: 12632
```

```
Repo-size       : 12 G
```

```
Repo-mirrors    : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/mirrors/2023.0.20230222/x86_64/mirror.list
```

```
Repo-baseurl   : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/guids/
```

```
cf9296325a6c46ff40c775a8e2d632c4c3fd9d9164014ce3304715d61b90ca8e/x86_64/
```

```
: (0 more)
```

```
Repo-expire    : 172800 second(s) (last: Wed Mar 1 23:40:15
```

```

                : 2023)
Repo-filename  : /etc/yum.repos.d/amazonlinux.repo

Repo-id        : amazonlinux-debuginfo
Repo-name      : Amazon Linux 2023 repository - Debug
Repo-status    : disabled
Repo-mirrors   : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/mirrors/2023.0.20230222/debuginfo/x86_64/mirror.list
Repo-expire    : 21600 second(s) (last: unknown)
Repo-filename  : /etc/yum.repos.d/amazonlinux.repo

Repo-id        : amazonlinux-source
Repo-name      : Amazon Linux 2023 repository - Source packages
Repo-status    : disabled
Repo-mirrors   : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/mirrors/2023.0.20230222/SRPMS/mirror.list
Repo-expire    : 21600 second(s) (last: unknown)
Repo-filename  : /etc/yum.repos.d/amazonlinux.repo

Repo-id        : kernel-livepatch
Repo-name      : Amazon Linux 2023 Kernel Livepatch repository
Repo-status    : disabled
Repo-mirrors   : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/kernel-livepatch/mirrors/al2023/x86_64/mirror.list
Repo-expire    : 172800 second(s) (last: unknown)
Repo-filename  : /etc/yum.repos.d/kernel-livepatch.repo

Repo-id        : kernel-livepatch-source
Repo-name      : Amazon Linux 2023 Kernel Livepatch repository -
                : Source packages
Repo-status    : disabled
Repo-mirrors   : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/kernel-livepatch/mirrors/al2023/SRPMS/mirror.list
Repo-expire    : 21600 second(s) (last: unknown)
Repo-filename  : /etc/yum.repos.d/kernel-livepatch.repo
Total packages: 12632

```

Note

--verbose 옵션 플래그를 추가하지 않으면 Repo-id, Repo-name 및 Repo-status 정보만 출력에 포함됩니다.

/etc/yum.repos.d 디렉터리에 yum 리포지토리 추가하기

1. .repo 파일의 위치를 찾습니다. 이 예시에서 .repo 파일은 <https://www.example.com/repository.repo>에 있습니다.
2. `dnf config-manager` 명령을 사용하여 리포지토리를 추가합니다.

```
$ sudo dnf config-manager --add-repo https://www.example.com/repository.repo
Loaded plugins: priorities, update-motd, upgrade-helper
adding repo from: https://www.example.com/repository.repo
grabbing file https://www.example.com/repository.repo to /etc/
yum.repos.d/repository.repo
repository.repo | 4.0 kB 00:00
repo saved to /etc/yum.repos.d/repository.repo
```

리포지토리를 설치한 후 다음 절차에 따라 리포지토리를 활성화합니다.

/etc/yum.repos.d에서 yum 리포지토리를 사용하려면 `--enable` 플래그 및 `#####` 이름과 함께 `dnf config-manager` 명령을 사용합니다.

```
$ sudo dnf config-manager --enable repository
```

Note

리포지토리 사용 중지하려면 동일한 명령 구문을 사용하되 명령을 `--enable`에서 `--disable`로 변경하세요.

cloud-init를 사용하여 리포지토리 추가

이 방법으로 리포지토리를 추가할 수 있지만 `cloud-init` 프레임워크를 사용하여 새 리포지토리를 추가할 수도 있습니다.

새 패키지 리포지토리를 추가하려면 다음 템플릿을 사용하는 것이 좋습니다. 이 파일을 로컬에 저장하는 것을 추천합니다.

```
#cloud-config
yum_repos:
  repository.repo:
    baseurl: https://www.example.com/
```

```
enabled: true
gpgcheck: true
gpgkey: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EXAMPLE
name: Example Repository
```

Note

cloud-init를 사용하면 좋은 점 한 가지는 구성 파일에 `packages:` 섹션을 추가할 수 있다는 것입니다. 설치하려는 패키지 이름을 이 섹션에 추가할 수 있습니다. 패키지는 기본 리포지토리 또는 `cloud-config` 파일에 추가한 새 리포지토리에서 설치할 수 있습니다. YAML 파일 구조에 대한 자세한 내용은 *cloud-init* 설명서의 [YUM 리포지토리 추가](#)(Adding a YUM repository)를 참조하세요.

YAML 형식 파일을 설정한 후에는 AWS CLI의 `cloud-init` 프레임워크에서 해당 파일을 실행할 수 있습니다. `--userdata` 옵션과 `.yaml` 파일 이름을 포함해야 원하는 작업을 호출할 수 있습니다.

```
$ aws ec2 run-instances \
  --image-id \
    resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-x86_64 \
  --instance-type m5.xlarge \
  --region us-east-1 \
  --key-name aws-key-us-east-1 \
  --security-group-ids sg-004a7650 \
  --user-data file:///cloud-config.yml
```

AL2023 버전 지정 저장소를 통한 결정적 업그레이드 사용

Note

기본적으로 AL2023 인스턴스에는 시작 시 심각하고 중요한 추가 보안 업데이트가 자동으로 수신되지 않습니다. 처음에는 AL2023 버전 및 선택한 AMI에서 사용할 수 있었던 업데이트가 포함되어 있습니다.

메이저 및 마이너 릴리스에 포함된 업데이트를 관리하세요.

AL2023을 통해 환경 전반에서 패키지 버전과 업데이트 간의 일관성을 유지할 수 있습니다. 또한 동일한 Amazon Machine Image (AMI)의 여러 인스턴스를 일관성있게 유지할 수 있습니다. 기본으로 설정

된 AL2023의 버전 지정 리포지토리 기능으로 결정적 업그레이드를 하면 특정 목적의 일정에 따라 업데이트할 수 있습니다.

새 패키지 업데이트를 릴리스할 때마다 사용할 새 버전과 해당 버전에서 사용할 새 AMI가 포함됩니다.

AL2023은 리포지토리의 특정 버전에서 사용 가능합니다. 이는 메이저 버전과 마이너 버전 모두에서 지원됩니다. SSM 파라미터에서 볼 수 있는 AL2023 AMI는 항상 최신 버전입니다. 중요하고 중요한 보안 업데이트를 포함하여 가장 많은 up-to-date 패키지와 업데이트를 제공합니다.

기존 AMI에서 인스턴스를 시작하는 경우 자동으로 업데이트되지 않습니다. 프로비저닝 기능으로 설치된 추가 패키지는 기존 AMI의 리포지토리 버전에 매핑됩니다.

이 기능을 사용하면 환경 전반에서 패키지 버전과 업데이트 간의 일관성을 유지할 수 있습니다. 특히 같은 AMI에서 여러 인스턴스를 실행하는 경우 그렇습니다. 필요에 맞는 일정에 따라 업데이트할 수 있습니다. 특정 리포지토리 버전으로 고정할 수 있어서 실행 시 특정 업데이트 세트를 적용할 수도 있습니다.

메이저 버전과 마이너 버전 업그레이드의 차이점은 다음과 같습니다.

AL2023 메이저 버전 릴리스에는 대규모 업데이트가 포함되며 패키지를 추가, 삭제 또는 업데이트할 수 있습니다. 호환성을 유지하려면 인스턴스를 새 메이저 버전으로 업그레이드하기 전에 해당 버전에서 애플리케이션을 테스트해야 합니다.

AL2023 마이너 버전 릴리스에는 기능 및 보안 업데이트가 포함되지만 패키지 변경 사항은 포함되지 않습니다. 이렇게 하면 새 버전에서도 Linux 기능 및 시스템 라이브러리 API를 계속 사용할 수 있습니다. 업데이트하기 전에 애플리케이션을 테스트하지 않아도 됩니다.

AL2023 리포지토리에서 사용할 수 있는 패키지 업데이트를 제어하십시오.

새 버전의 AL2023 리포지토리를 게시해도 이전 버전은 모두 계속 사용할 수 있습니다. 기본적으로 리포지토리 버전 관리용 플러그인은 AMI 구축에 사용된 것과 동일한 버전입니다. 패키지 업데이트를 관리하려면 다음 단계를 따르세요.

1. 다음 명령을 실행하여 사용 가능한 리포지토리 버전을 검색합니다.

```
$ sudo dnf check-release-update
```

2. 다음 명령을 실행하여 버전을 선택하세요.

```
$ sudo dnf --releasever=version update
```

이 명령으로 현재 Amazon Linux 릴리스 버전에서 명령줄에 지정된 릴리스 버전으로 업데이트할 수 있습니다. 패키지 업데이트 목록은 dnf에서 제공합니다. 업데이트하기 전에 업데이트를 확인해야 합니다. 업데이트가 완료되면 새 릴리스 버전은 향후 모든 dnf 활동의 기본 릴리스 버전이 됩니다.

자세한 내용은 [AL2023 패키지 및 운영 체제 업데이트 관리](#)(를) 참조하세요.

버전이 지정된 리포지토리를 통한 결정적 업그레이드

주제

- [결정적 업그레이드 시스템 사용](#)
- [결정적 업그레이드 시스템의 선택적 업데이트](#)
- [결정적 업그레이드와 함께 연속 재정의 사용](#)

결정적 업그레이드 시스템 사용

dnf upgrade 명령을 실행하면 시스템은 releasever 변수가 지정하는 저장소에서 업그레이드를 확인합니다. `### ### ## ##### ## ##### ## ## (#: 2023.4.20240513) releasever ## #.`

두 가지 방법 중 하나를 사용하여 releasever 값을 변경할 수 있습니다. 이러한 방법은 내림차순으로 시스템 우선 순위를 나열합니다. 즉, 방법 1은 방법 2와 3보다 우선하고 방법 2는 방법 3보다 우선합니다.

1. 명령줄 플래그를 사용했다면 그 값은 `--releasever=latest`입니다.
2. 오버라이드 변수 파일이 설정된 경우 지정된 값은 `/etc/dnf/vars/releasever`입니다.
3. 현재 설치된 system-release 패키지 버전입니다.

즉, 패키지 버전은 `2023.0.20230210`입니다.

```
$ rpm -q system-release
system-release-2023.0.20230210-0.amzn2023.noarch
```

새로 설치한 시스템에는 재정의 변수가 없습니다. 시스템이 설치된 system-release 버전을 제어하기 때문에 업그레이드가 불가능합니다.

```
$ cat /etc/dnf/vars/releasever
```

```
cat: /etc/dnf/vars/releasever: No such file or directory
```

```
$ sudo dnf upgrade
```

```
Last metadata expiration check: 0:00:02 ago on Wed 15 Feb 2023 06:14:12 PM UTC.
Dependencies resolved.
Nothing to do.
Complete!
```

releasever 플래그로 원하는 버전을 지정하면 특정 버전의 패키지를 가져올 수 있습니다.

```
$ rpm -q system-release
```

```
system-release-2023.0.20230222-0.amzn2023.noarch
```

```
$ sudo dnf upgrade --releasever=2023.0.20230329
```

```
Amazon Linux 2023 repository                26 MB/s | 12 MB      00:00
Dependencies resolved.
=====
Package                Arch    Version                                Repository    Size
=====
Installing:
kernel                  aarch64 6.1.21-1.45.amzn2023                  amazonlinux  26 M
Upgrading:
amazon-linux-repo-s3    noarch  2023.0.20230329-0.amzn2023            amazonlinux  18 k
ca-certificates        noarch  2023.2.60-1.0.amzn2023.0.1           amazonlinux  828 k
cloud-init              noarch  22.2.2-1.amzn2023.1.7                 amazonlinux  1.1 M

... [ list edited for clarity ]

system-release          noarch  2023.0.20230329-0.amzn2023            amazonlinux  29 k

... [ list edited for clarity ]

vim-data                noarch  2:9.0.1403-1.amzn2023.0.1            amazonlinux  25 k
vim-minimal             aarch64 2:9.0.1403-1.amzn2023.0.1            amazonlinux  753 k

Transaction Summary
=====
Install   1 Package
Upgrade  42 Packages

Total download size: 56 M
```

--releasever 옵션은 system-release와 /etc/dnf/vars/releasever 모두 재정의하므로 이 업그레이드의 결과는 다음과 같습니다.

1. 설치된 모든 패키지의 구 버전과 최신 버전 간의 변경 사항을 업그레이드합니다.
2. 업그레이드하면 시스템은 system-release 최신 버전의 리포지토리를 제어합니다.

결정적 업그레이드 시스템의 선택적 업데이트

정식 릴리스 버전을 사용하는 시스템에 최신 릴리스에서 선택한 패키지를 설치하는 것이 좋습니다.

dnf check-update로 업그레이드할 패키지를 식별할 수 있습니다.

```
$ sudo dnf check-update --releasever=latest --security
Amazon Linux 2023 repository                13 MB/s | 10 MB    00:00
Last metadata expiration check: 0:00:02 ago on Wed 15 Feb 2023 02:52:21 AM UTC.

bind-libs.aarch64                32:9.16.27-1.amzn2023.0.1    amazonlinux
bind-license.noarch              32:9.16.27-1.amzn2023.0.1    amazonlinux
bind-utils.aarch64              32:9.16.27-1.amzn2023.0.1    amazonlinux
cryptsetup.aarch64              2.4.3-2.amzn2023.0.1        amazonlinux
cryptsetup-libs.aarch64         2.4.3-2.amzn2023.0.1        amazonlinux
curl-minimal.aarch64            7.85.0-1.amzn2023.0.1       amazonlinux
glibc.aarch64                   2.34-40.amzn2023.0.2        amazonlinux
glibc-all-langpacks.aarch64     2.34-40.amzn2023.0.2        amazonlinux
glibc-common.aarch64           2.34-40.amzn2023.0.2        amazonlinux
glibc-locale-source.aarch64     2.34-40.amzn2023.0.2        amazonlinux
gmp.aarch64                     1:6.2.1-2.amzn2023.0.1      amazonlinux
gnupg2-minimal.aarch64         2.3.7-1.amzn2023.0.2        amazonlinux
gzip.aarch64                   1.10-5.amzn2023.0.1         amazonlinux
kernel.aarch64                 6.1.12-17.42.amzn2023       amazonlinux
kernel-tools.aarch64           6.1.12-17.42.amzn2023       amazonlinux
libarchive.aarch64             3.5.3-2.amzn2023.0.1        amazonlinux
libcurl-minimal.aarch64        7.85.0-1.amzn2023.0.1       amazonlinux
libsepol.aarch64               3.4-3.amzn2023.0.2          amazonlinux
libsolv.aarch64                0.7.22-1.amzn2023.0.1       amazonlinux
libxml2.aarch64                2.9.14-1.amzn2023.0.1       amazonlinux
logrotate.aarch64              3.20.1-2.amzn2023.0.2       amazonlinux
lua-libs.aarch64               5.4.4-3.amzn2023.0.1        amazonlinux
lz4-libs.aarch64               1.9.4-1.amzn2023.0.1        amazonlinux
openssl.aarch64                1:3.0.5-1.amzn2023.0.3      amazonlinux
openssl-libs.aarch64          1:3.0.5-1.amzn2023.0.3      amazonlinux
pcre2.aarch64                  10.40-1.amzn2023.0.1        amazonlinux
```

pcre2-syntax.noarch	10.40-1.amzn2023.0.1	amazonlinux
rsync.aarch64	3.2.6-1.amzn2023.0.2	amazonlinux
vim-common.aarch64	2:9.0.475-1.amzn2023.0.1	amazonlinux
vim-data.noarch	2:9.0.475-1.amzn2023.0.1	amazonlinux
vim-enhanced.aarch64	2:9.0.475-1.amzn2023.0.1	amazonlinux
vim-filesystem.noarch	2:9.0.475-1.amzn2023.0.1	amazonlinux
vim-minimal.aarch64	2:9.0.475-1.amzn2023.0.1	amazonlinux
xz.aarch64	5.2.5-9.amzn2023.0.1	amazonlinux
xz-libs.aarch64	5.2.5-9.amzn2023.0.1	amazonlinux
zlib.aarch64	1.2.11-32.amzn2023.0.3	amazonlinux

업그레이드하고 싶은 패키지를 설치합니다. `sudo dnf upgrade --releasever=latest` 및 패키지 이름을 사용하면 `system-release` 패키지가 변경되지 않습니다.

```
$ sudo dnf upgrade --releasever=latest openssl openssl-libs
Last metadata expiration check: 0:01:28 ago on Wed 15 Feb 2023 02:52:21 AM UTC.
Dependencies resolved.
=====
Package           Arch      Version                               Repository      Size
=====
Upgrading:
openssl           aarch64  1:3.0.5-1.amzn2023.0.3              amazonlinux     1.1 M
openssl-libs     aarch64  1:3.0.5-1.amzn2023.0.3              amazonlinux     2.1 M

Transaction Summary
=====
Upgrade 2 Packages

Total download size: 3.2 M
```

Note

`sudo dnf upgrade --releasever=latest`를 사용하면 `system-release`를 포함한 모든 패키지가 업데이트됩니다. 그러면 연속 재정의의 설정하지 않는 한 최신 `system-release` 버전으로 유지됩니다.

결정적 업그레이드와 함께 연속 재정의의 사용

`--releasever=latest`를 추가하는 대신 연속 재정의의 기능으로 변수 값을 `##`으로 설정하여 시스템 잠금 해제를 할 수 있습니다.

```
$ echo latest | sudo tee /etc/dnf/vars/releasever
latest
```

```
$ sudo dnf upgrade
```

```
Last metadata expiration check: 0:03:36 ago on Wed 15 Feb 2023 02:52:21 AM UTC.
Dependencies resolved.
```

```
=====
Package                               Arch    Version                               Repository    Size
=====
```

Package	Arch	Version	Repository	Size
Installing:				
kernel	aarch64	6.1.73-45.amzn2023	amazonlinux	24 M
Upgrading:				
acl	aarch64	2.3.1-2.amzn2023.0.1	amazonlinux	72 k
alternatives	aarch64	1.15-2.amzn2023.0.1	amazonlinux	36 k
amazon-ec2-net-utils	noarch	2.3.0-1.amzn2023.0.1	amazonlinux	16 k
at	aarch64	3.1.23-6.amzn2023.0.1	amazonlinux	60 k
attr	aarch64	2.5.1-3.amzn2023.0.1	amazonlinux	59 k
audit	aarch64	3.0.6-1.amzn2023.0.1	amazonlinux	249 k
audit-libs	aarch64	3.0.6-1.amzn2023.0.1	amazonlinux	116 k
aws-c-auth-libs	aarch64	0.6.5-6.amzn2023.0.2	amazonlinux	79 k
aws-c-cal-libs	aarch64	0.5.12-7.amzn2023.0.2	amazonlinux	34 k
aws-c-common-libs	aarch64	0.6.14-6.amzn2023.0.2	amazonlinux	119 k
aws-c-compression-libs	aarch64	0.2.14-5.amzn2023.0.2	amazonlinux	22 k
aws-c-event-stream-libs	aarch64	0.2.7-5.amzn2023.0.2	amazonlinux	47 k
aws-c-http-libs	aarch64	0.6.8-6.amzn2023.0.2	amazonlinux	147 k
aws-c-io-libs	aarch64	0.10.12-5.amzn2023.0.6	amazonlinux	109 k
aws-c-mqtt-libs	aarch64	0.7.8-7.amzn2023.0.2	amazonlinux	61 k
aws-c-s3-libs	aarch64	0.1.27-5.amzn2023.0.3	amazonlinux	54 k
aws-c-sdkutils-libs	aarch64	0.1.1-5.amzn2023.0.2	amazonlinux	26 k
aws-checksums-libs	aarch64	0.1.12-5.amzn2023.0.2	amazonlinux	50 k
awscli-2	noarch	2.7.8-1.amzn2023.0.4	amazonlinux	7.3 M
basesystem	noarch	11-11.amzn2023.0.1	amazonlinux	7.8 k
bash	aarch64	5.1.8-2.amzn2023.0.1	amazonlinux	1.6 M
bash-completion	noarch	1:2.11-2.amzn2023.0.1	amazonlinux	292 k
bc	aarch64	1.07.1-14.amzn2023.0.1	amazonlinux	120 k
bind-libs	aarch64	32:9.16.27-1.amzn2023.0.1	amazonlinux	1.2 M
bind-license	noarch	32:9.16.27-1.amzn2023.0.1	amazonlinux	14 k
bind-utils	aarch64	32:9.16.27-1.amzn2023.0.1	amazonlinux	206 k
binutils	aarch64	2.38-20.amzn2023.0.3	amazonlinux	4.6 M
boost-filesystem	aarch64	1.75.0-4.amzn2023.0.1	amazonlinux	55 k
boost-system	aarch64	1.75.0-4.amzn2023.0.1	amazonlinux	14 k
boost-thread	aarch64	1.75.0-4.amzn2023.0.1	amazonlinux	54 k
bzip2	aarch64	1.0.8-6.amzn2023.0.1	amazonlinux	53 k

```

bzip2-libs          aarch64 1.0.8-6.amzn2023.0.1      amazonlinux 44 k
c-ares              aarch64 1.17.2-1.amzn2023.0.1     amazonlinux 107 k
ca-certificates    noarch 2021.2.50-1.0.amzn2023.0.3 amazonlinux 343 k
checkpolicy        aarch64 3.4-3.amzn2023.0.1       amazonlinux 345 k
chkconfig          aarch64 1.15-2.amzn2023.0.1      amazonlinux 162 k
chrony             aarch64 4.2-7.amzn2023.0.4       amazonlinux 314 k
cloud-init         noarch 22.2.2-1.amzn2023.1.7    amazonlinux 1.1 M
cloud-utils-growpart aarch64 0.31-8.amzn2023.0.2     amazonlinux 31 k
coreutils          aarch64 8.32-30.amzn2023.0.2     amazonlinux 1.1 M
coreutils-common   aarch64 8.32-30.amzn2023.0.2     amazonlinux 2.0 M
cpio               aarch64 2.13-10.amzn2023.0.1     amazonlinux 269 k
cracklib           aarch64 2.9.6-27.amzn2023.0.1   amazonlinux 83 k
cracklib-dicts     aarch64 2.9.6-27.amzn2023.0.1   amazonlinux 3.6 M
crontabs           noarch 1.11-24.20190603git.amzn2023.0.1
                                                           amazonlinux 19 k

crypto-policies    noarch 20230128-1.gitdfb10ea.amzn2023.0.1
                                                           amazonlinux 61 k
crypto-policies-scripts noarch 20230128-1.gitdfb10ea.amzn2023.0.1
                                                           amazonlinux 81 k

...
Installing dependencies:
amazon-linux-repo-cdn noarch 2023.0.20230210-0.amzn2023 amazonlinux 16 k
xxhash-libs          aarch64 0.8.0-3.amzn2023.0.1     amazonlinux 32 k
Installing weak dependencies:
amazon-chrony-config noarch 4.2-7.amzn2023.0.4       amazonlinux 14 k
gawk-all-langpacks  aarch64 5.1.0-3.amzn2023.0.1     amazonlinux 207 k

```

Transaction Summary

```

=====
Install    5 Packages
Upgrade   413 Packages

```

Total download size: 199 M

Note

재정의 변수 `/etc/dnf/vars/releasever`를 사용한 경우 다음 명령으로 재정의 값을 삭제하고 기본 잠금 동작을 복원할 수 있습니다.

```
$ sudo rm /etc/dnf/vars/releasever
```

AL2023 기반 커널 라이브 패칭

AL2023 커널 라이브 패칭을 사용하면 실행 중인 애플리케이션을 재부팅하거나 중단하지 않고도 실행 중인 Linux 커널에 보안 취약성 및 중요 버그 패치를 적용할 수 있습니다. 또한 커널 라이브 패치는 애플리케이션의 가용성을 높이는 동시에 인프라를 최신 상태로 안전하게 유지할 수 있습니다.

AWS 두 가지 유형의 AL2023 커널 라이브 패치를 릴리스합니다.

- 보안 업데이트 - Linux의 공통 보안 취약성 및 노출(Common Vulnerabilities and Exposures) 목록 업데이트가 있습니다. 이러한 업데이트는 일반적으로 Amazon Linux 보안 권고 등급을 사용하여 중요 또는 매우 중요로 등급이 매겨집니다. 대개 공통 취약점 등급 시스템(Common Vulnerability Scoring System) 점수 7 이상에 매핑됩니다. 경우에 따라 CVE가 할당되기 전에 업데이트를 제공할 AWS 수도 있습니다. 이러한 경우 버그 수정 패치를 배포합니다.
- 버그 수정 - CVE와 관련없는 중요한 버그 및 안정성 문제에 대한 수정 사항이 포함되어 있습니다.

AWS AL2023 커널 버전 출시 후 최대 3개월 동안 커널 라이브 패치를 제공합니다. 3개월 후 커널 라이브 패치를 계속 받으려면 최신 커널 버전으로 업데이트해야 합니다.

AL2023 커널 라이브 패치는 기존 AL2023 리포지토리에서 서명된 RPM 패키지로 사용할 수 있으며, 패치는 기존 DNF 패키지 관리자 워크플로우를 사용하여 개별 인스턴스에 설치할 수 있습니다. 또는 AWS Systems Manager를 사용하여 관리형 인스턴스 그룹에 설치할 수도 있습니다.

AL2023 커널 라이브 패치는 무료로 제공됩니다.

주제

- [제한 사항](#)
- [지원되는 구성 및 필수 조건](#)
- [커널 라이브 패치 작업](#)

제한 사항

커널 라이브 패치를 적용하는 동안 최대 절전 모드를 실행하거나, 고급 디버깅 도구(예: SystemTap, kprobes 및 eBPF 기반 도구)를 사용하거나, 커널 라이브 패치 인프라에 사용되는 ftrace 출력 파일에 액세스할 수 없습니다.

지원되는 구성 및 필수 조건

커널 라이브 패치는 AL2023을 실행하는 Amazon EC2 인스턴스 및 온프레미스 가상 머신에서 사용할 수 있습니다.

AL2023 커널 라이브 패치를 사용하려면 다음 사양이 필요합니다.

- 64비트 x86_64 또는 ARM64 아키텍처
- 커널 버전 6.1

정책 요구 사항

AL2023 리포지토리에서 패키지를 다운로드하려면 Amazon EC2가 서비스 소유의 Amazon S3 버킷에 액세스할 수 있어야 합니다. 환경에서 Amazon S3용 Amazon VPC (가상 사설 클라우드) 엔드포인트를 사용하는 경우, VPC 엔드포인트 정책이 해당 퍼블릭 버킷에 대한 액세스를 허용하는지 확인하십시오. 다음 표에서는 Amazon EC2가 커널 라이브 패치를 위해 액세스해야 할 수 있는 Amazon S3 버킷에 대해 설명합니다.

S3 버킷 ARN	설명
<code>arn:aws:s3:::al2023-repos-##-de612dc2/*</code>	AL2023 리포지토리를 포함하는 Amazon S3 버킷

커널 라이브 패치 작업

인스턴스 자체 명령줄을 사용하여 개별 인스턴스에서 커널 라이브 패치를 활성화하고 사용할 수 있습니다. 또는 AWS Systems Manager를 사용하여 관리형 인스턴스 그룹에서 커널 라이브 패치를 활성화하고 사용할 수 있습니다.

다음 섹션에서는 명령줄을 사용하여 개별 인스턴스에서 커널 라이브 패치를 활성화하고 사용하는 방법에 대해 설명합니다.

관리형 인스턴스 그룹에서 커널 라이브 패치를 활성화하고 사용하는 방법에 대한 자세한 내용은 AWS Systems Manager 사용 설명서에서 [AL 2023 인스턴스에서 커널 라이브 패치 사용](#)을 참조하세요.

주제

- [커널 라이브 패치 활성화](#)

- [사용 가능한 커널 라이브 패치 보기](#)
- [커널 라이브 패치 적용](#)
- [설치된 커널 라이브 패치 보기](#)
- [커널 라이브 패치 비활성화](#)

커널 라이브 패치 활성화

커널 라이브 패치는 AL2023에서 기본적으로 비활성화되어 있습니다. 라이브 패치를 사용하려면 커널 라이브 패치 DNF 플러그인을 설치하고 라이브 패치 기능을 활성화해야 합니다.

커널 라이브 패치 활성화

1. 커널 라이브 패치는 커널 버전 6.1가 포함된 AL2023에 사용할 수 있습니다. 커널 버전을 확인하려면 다음 명령을 실행합니다.

```
$ sudo dnf list kernel
```

2. 커널 라이브 패치용 DNF 플러그인을 설치합니다.

```
$ sudo dnf install -y kpatch-dnf
```

3. 커널 라이브 패치용 DNF 플러그인을 활성화합니다.

```
$ sudo dnf kernel-livepatch -y auto
```

또한 이 명령으로 구성된 리포지토리에서 최신 버전의 커널 라이브 패치 RPM을 설치할 수 있습니다.

4. 커널 라이브 패치용 DNF 플러그인이 성공적으로 설치되었는지 확인하려면 다음 명령을 실행합니다.

커널 라이브 패치를 활성화하면 빈 커널 라이브 패치 RPM이 적용됩니다. 커널 라이브 패치가 성공적으로 활성화된 경우 초기 빈 커널 라이브 패치 RPM을 포함하는 목록이 나타납니다.

```
$ sudo rpm -qa | grep kernel-livepatch
dnf-plugin-kernel-livepatch-1.0-0.11.amzn2023.noarch
kernel-livepatch-6.1.12-17.42-1.0-0.amzn2023.x86_64
```

5. kpatch 패키지를 설치합니다.

```
$ sudo dnf install -y kpatch-runtime
```

6. kpatch 서비스가 이전에 설치된 경우 업데이트합니다.

```
$ sudo dnf update kpatch-runtime
```

7. kpatch 서비스를 시작합니다. 이 서비스로 초기화 또는 부팅 시 모든 커널 라이브 패치를 로드할 수 있습니다.

```
$ sudo systemctl enable kpatch.service && sudo systemctl start kpatch.service
```

사용 가능한 커널 라이브 패치 보기

Amazon Linux 보안 알림이 Amazon Linux 보안 센터에 게시됩니다. 커널 라이브 패치 알림을 비롯한 AL 2023 보안 알림에 대한 자세한 내용은 [Amazon Linux 보안 센터](#)를 참조하세요. 커널 라이브 패치 앞에 ALASLIVEPATCH가 붙습니다. Amazon Linux 보안 센터에는 버그를 해결하는 커널 라이브 패치가 나열되지 않을 수 있습니다.

명령줄을 사용하여 권고 및 CVE와 관련된 사용 가능한 커널 라이브 패치를 찾을 수도 있습니다.

권고와 관련된 사용 가능한 모든 커널 라이브 패치를 나열하려면

다음 명령을 사용합니다.

```
$ sudo dnf updateinfo list
```

```
Last metadata expiration check: 1:06:23 ago on Mon 13 Feb 2023 09:28:19 PM UTC.
ALAS2LIVEPATCH-2021-123    important/Sec. kernel-
livepatch-6.1.12-17.42-1.0-4.amzn2023.x86_64
ALAS2LIVEPATCH-2022-124    important/Sec. kernel-
livepatch-6.1.12-17.42-1.0-3.amzn2023.x86_64
```

CVE와 관련된 사용 가능한 모든 커널 라이브 패치를 나열하려면

다음 명령을 사용합니다.

```
$ sudo dnf updateinfo list cves
```

```
Last metadata expiration check: 1:07:26 ago on Mon 13 Feb 2023 09:28:19 PM UTC.
CVE-2022-0123    important/Sec. kernel-livepatch-6.1.12-17.42-1.0-4.amzn2023.x86_64
```

CVE-2022-3210 important/Sec. kernel-livepatch-6.1.12-17.42-1.0-3.amzn2023.x86_64

커널 라이브 패치 적용

DNF 패키지 관리자로 정기적인 업데이트를 하는 것과 같은 방식으로 커널 라이브 패치를 설치합니다. 커널 라이브 패치용 DNF 플러그인으로 설치할 커널 라이브 패치를 관리할 수 있으며 재부팅할 필요가 없습니다.

Tip

커널을 최신 상태로 안전하게 유지하려면 커널 라이브 패치를 사용하여 정기적으로 업데이트 하는 것이 좋습니다.

특정 커널 라이브 패치를 설치하거나 정기 보안 업데이트로 사용 가능한 커널 라이브 패치를 설치할 수 있습니다.

특정 커널 라이브 패치를 적용하려면

1. [사용 가능한 커널 라이브 패치 보기](#)에 설명된 명령 중 하나를 사용하여 커널 라이브 패치 버전을 설치합니다.
2. AL2023 커널에 커널 라이브 패치를 설치합니다.

```
$ sudo dnf install kernel-livepatch-kernel_version-package_version.amzn2023.x86_64
```

예를 들어, 다음 명령은 AL2023 커널 6.1.12-17.42 버전에 커널 라이브 패치를 설치합니다.

```
$ sudo dnf install kernel-livepatch-6.1.12-17.42-1.0-4.amzn2023.x86_64
```

정기 보안 업데이트와 함께 사용 가능한 커널 라이브 패치를 적용하려면

다음 명령을 사용합니다.

```
$ sudo dnf update --security
```

버그 수정을 포함하려면 --security 옵션을 생략하세요.

⚠ Important

- 커널 버전은 커널 라이브 패치를 설치한 후에 업데이트되지 않으며 인스턴스를 재부팅한 후에만 새 버전으로 업데이트됩니다.
- AL2023 커널은 3개월 동안 커널 라이브 패치를 받습니다. 3개월이 경과한 후에는 해당 커널 버전에 대한 새로운 커널 라이브 패치가 릴리스되지 않습니다.
- 3개월 후에도 계속 커널 라이브 패치를 받으려면 인스턴스를 재부팅하여 최신 커널 버전으로 업데이트해야 합니다. 업데이트 후 다음 3개월 동안 커널 라이브 패치가 인스턴스에 설치됩니다.
- 커널 버전에 대한 지원 기간을 확인하려면 다음 명령을 실행합니다.

```
$ sudo dnf kernel-livepatch support
```

설치된 커널 라이브 패치 보기

설치된 커널 라이브 패치를 보려면

다음 명령을 사용합니다.

```
$ sudo kpatch list
Loaded patch modules:
livepatch_CVE_2022_36946 [enabled]

Installed patch modules:
livepatch_CVE_2022_36946 (6.1.57-29.131.amzn2023.x86_64)
livepatch_CVE_2022_36946 (6.1.57-30.131.amzn2023.x86_64)
```

이 명령은 로드되고 설치된 보안 업데이트 커널 라이브 패치 목록을 반환합니다. 다음은 예시 출력입니다.

ℹ Note

단일 커널 라이브 패치에 여러 개의 라이브 패치가 포함되어 설치될 수 있습니다.

커널 라이브 패치 비활성화

커널 라이브 패치를 더 이상 사용할 필요가 없는 경우 언제든지 비활성화할 수 있습니다.

- livepatches 사용 비활성화:

1. 플러그인 비활성화:

```
$ sudo dnf kernel-livepatch manual
```

2. kpatch 서비스 비활성화:

```
$ sudo systemctl disable --now kpatch.service
```

- livepatch 도구 완전 삭제:

1. 플러그인 삭제:

```
$ sudo dnf remove kpatch-dnf
```

2. kpatch-runtime 삭제:

```
$ sudo dnf remove kpatch-runtime
```

3. 설치된 livepatches 삭제:

```
$ sudo dnf remove kernel-livepatch\*
```

AL2023 런타임 프로그래밍 시작하기

AL2023 언어는 일부 언어 런타임의 다양한 버전을 제공합니다. 우리는 동시에 여러 버전을 지원하는 업스트림 프로젝트를 사용합니다. 패키지를 검색하고 설치하는 `dnf` 명령을 사용하여 이름 버전이 지정된 패키지를 설치하고 관리하는 방법에 대한 정보를 찾아보세요.

다음 주제에서는 AL2023 내에서 각 언어 생태계가 어떻게 존재하는지 간략하게 설명합니다.

주제

- [AL2023의 C, C++, 및 Fortran](#)
- [AL2023에서 Go 사용](#)
- [AL2023에서 Java 사용](#)
- [AL2023에서 Perl 사용](#)
- [AL2023에서 PHP 사용](#)
- [AL2023에서 Python 사용](#)
- [AL2023에서 Rust 사용](#)

AL2023의 C, C++, 및 Fortran

AL2023에는 GNU 컴파일러 컬렉션 (GCC) 과 LLVM (저수준 가상 머신) 의 Clang 프론트엔드가 모두 포함되어 있습니다.

GCC 주요 버전은 AL2023이 종료되기 전까지 계속 사용할 수 있습니다. 마이너 릴리스에 버그 수정이 포함되어 있으며 이는 AL2023 릴리스에 포함될 수 있습니다. 기타 버그, 성능 및 보안 수정 사항은 AL2023에 설치된 GCC의 주요 버전으로 백포트될 수 있습니다.

AL2023 버전에는 C (`gcc`), C++ (`g++`) 및 포트란 (`gfortran`) 프론트엔드가 포함된 버전 11이 포함되어 있습니다.
GCC `g++ gfortran`

AL2023 프론트엔드는 (`gnat`), Ada (`gnat`), Go 오브젝티브-C 또는 `gcc-go` 오브젝티브-C++ 프론트엔드를 활성화하지 않습니다.

AL2023 RPM이 빌드되는 기본 컴파일러 플래그에는 최적화 및 강화 플래그가 포함되어 있습니다. GCC로 코드를 직접 빌드하려면 최적화 및 강화 플래그를 포함하는 것이 좋습니다.

Note

`gcc --version`을 호출하면 `gcc (GCC) 11.3.1 20221121 (Red Hat 11.3.1-4)` 같은 버전 문자열이 표시됩니다. Red Hat은 Amazon Linux GCC 패키지의 기반이 되는 [GCC 공급업체 지점](#)을 나타냅니다. 에 표시된 버그 신고 URL에 따르면 모든 버그 신고 및 지원 요청은 Amazon Linux로 보내야 합니다. `gcc --help`
이 공급업체 브랜치의 일부 장기 변경 사항 (예: `__GNUC_RH_RELEASE__` 매크로)에 대한 자세한 내용은 [Fedora 패키지](#) 소스를 참조하십시오.

핵심 툴체인에 대한 자세한 내용은 을 참조하십시오. [코어 툴체인 패키지 glibc, gcc 및 binutils](#)

AL2023 및 다른 Linux 배포판과의 관계에 대한 자세한 내용은 을 참조하십시오. [Fedora와의 관계](#)

AL2와 비교한 AL2023 컴파일러 트리플렛 변경에 대한 자세한 내용은 을 참조하십시오. [컴파일러 트리플렛](#)

AL2023에서 Go 사용

Amazon [GoLinux](#)에서 직접 작성한 코드를 빌드하거나 AL2023 제공 툴체인을 사용하는 것이 좋습니다. AL2와 마찬가지로 AL2023 역시 운영 체제의 수명 주기 내내 Go 툴체인을 업데이트합니다. 이는 당사가 제공하는 툴체인의 CVE에 대한 결과이거나 분기 릴리스에 포함되어 있습니다.

Go비교적 빠르게 변화하는 언어입니다. 로 작성된 기존 응용 프로그램을 새 버전의 Go 툴체인에 맞게 조정해야 하는 상황이 발생할 수 있습니다. Go에 대한 Go 자세한 내용은 [Go1 및 Go 프로그램의 미래를](#) 참조하십시오.

AL2023 출시 기간 동안 새 버전의 Go 툴체인이 포함될 예정이긴 하지만 Go 업스트림 릴리즈와 동일한 수준으로 제공되지는 않을 것입니다. 따라서 언어 및 표준 라이브러리의 최첨단 기능을 사용하여 Go 코드를 빌드하려는 경우 AL2023 제공 Go 툴체인을 사용하는 것이 적합하지 않을 수 있습니다. Go

AL2023 수명 기간 동안에는 이전 패키지 버전이 저장소에서 제거되지 않습니다. 이전 Go 도구 모음이 필요한 경우 새 Go 도구 모음의 버그 및 보안 수정을 생략하고 모든 RPM에서 사용할 수 있는 것과 동일한 메커니즘을 사용하여 저장소에서 이전 버전을 설치할 수 있습니다.

AL2023 기반으로 자체 Go 코드를 빌드하려는 경우 AL2023에 포함된 툴체인을 사용할 수 있습니다. 단, 이 Go 툴체인이 AL2023 수명 기간 동안 계속 발전할 수 있다는 점을 염두에 두세요.

다음과 같이 작성된 AL2023 람다 함수 Go

네이티브 코드로 Go 컴파일할 때 Lambda는 사용자 지정 런타임으로 Go 취급합니다. `provided.al2023` 런타임을 사용하여 AL2023 상의 Go 함수를 Lambda에 배포할 수 있습니다.

자세한 내용은 개발자 안내서의 [Lambda 함수 Go 빌드를](#) 참조하십시오. AWS Lambda

AL2023에서 Java 사용

AL2023 은 기반 워크로드를 지원하기 위해 여러 버전의 [Amazon Corretto](#)를 Java 제공합니다. AL2023 에 포함된 모든 Java 기반 패키지는 다음과 같이 빌드됩니다. Amazon Corretto 17 17

Corretto는 에서 장기간 지원하는 오픈 자바 개발 키트 (OpenJDK) 의 빌드입니다. Amazon Corretto 는 자바 SE 표준을 충족하고 Linux, 및 에서 사용할 수 있는지 확인하기 위해 자바 기술 호환성 키트 (TCK) 를 사용하여 인증을 받았습니다. Windows macOS

Corretto 1.8.0, Corretto 11, Corretto 17에 사용할 수 있는 [Amazon Corretto](#) 패키지가 있습니다.

AL2023에 설치된 Corretto는 Corretto 버전과 동일한 기간 동안 또는 AL2023 기간이 종료되는 시점 중 더 짧은 기간 동안 지원합니다. 자세한 내용은 [아마존 리눅스 패키지 지원 설명문 및 아마존 Corretto FAQ](#)를 참조하십시오.

AL2023에서 Perl 사용

AL2023 [Perl](#) 프로그래밍 언어의 버전 5.32를 제공합니다.

지난 수십 년 동안 Perl 5개 릴리스의 일부로 높은 수준의 언어 호환성을 제공했지만 Perl, Amazon Linux는 AL2023 릴리스 기간 중 Perl 5.32에서 벗어날 것으로 예상되지 않습니다. Amazon Linux는 [패키지 지원 설명에](#) 따라 AL2023 수명 주기 동안 계속해서 보안 패치를 적용할 Perl 예정입니다.

AL2023 Perl 모듈

AL2023 버전에는 다양한 Perl 모듈이 RPM으로 패키징되어 있습니다. RPM으로 사용할 수 있는 Perl 모듈이 많기는 하지만 Amazon Linux는 가능한 Perl 모든 모듈을 패키징하는 것을 목표로 하지는 않습니다. RPM으로 패키징된 모듈은 다른 운영 체제 RPM 패키지에서 사용할 수 있으므로 Amazon Linux 는 순수 기능 업데이트보다 이러한 보안 패치를 우선시합니다.

또한 AL2023 에는 Perl 개발자가 모듈에 대한 관용적 패키지 관리자를 사용할 수 CPAN 있도록 하는 기능이 포함되어 있습니다. Perl

AL2023에서 PHP 사용

AL2023 은 현재 두 가지 버전의 [PHP](#) 프로그래밍 언어를 제공하며, 각 버전은 PHP 업스트림과 동일한 기간 동안 지원됩니다. 자세한 내용은 [패키지 지원 설명](#)을 참조하십시오.

AL2023 버전을 사용하면 PHP 8.2의 새로운 기능을 사용하면서도 PHP 8.1이 필요한 애플리케이션은 계속 지원할 수 있습니다.

PHP 구 버전에서 마이그레이션

업스트림 PHP 커뮤니티는 8.1에서 [PHP8.2로 이전하기 위한 포괄적인 마이그레이션 설명서](#)를 작성했습니다. PHP [PHP 8.0에서 8.1로 마이그레이션](#)에 대한 설명서도 있습니다.

AL2에는 AL2023 PHP 로의 손쉬운 업그레이드 경로를 `amazon-linux-extras` 지원하는 8.0, 8.1 및 8.2가 포함되어 있습니다.

PHP 7.x 버전에서 마이그레이션

Note

[이 PHP 프로젝트는 지원되는 버전의 목록과 일정, 지원되지 않는 브랜치 목록을 유지 관리합니다.](#)

AL2023 출시 당시에는 PHP 커뮤니티에서 7.x 및 5.x 버전을 모두 지원하지 [PHP](#) 않았으며 AL2023 옵션에도 포함되지 않았습니다.

업스트림 PHP 커뮤니티에서는 7.4에서 8.0으로 [전환하기 PHP 위한 포괄적인 마이그레이션 문서](#)를 작성했습니다. PHP [PHP8.1](#) 및 [PHP 8.2](#)로의 마이그레이션에 대한 이전 섹션에서 참조한 설명서와 함께 사용하면 기반 애플리케이션을 최신 버전으로 마이그레이션할 수 있습니다. [PHP](#)

Note

AL2에는 PHP 7.1, 7.2, 7.3 및 7.4인치가 포함됩니다. `amazon-linux-extras` 참고로, 이러한 모든 엑스트라는 추가 보안 업데이트를 보장하지 않으며 `end-of-life` 앞으로 제공되지 않습니다.

AL2023 PHP 모듈

AL2023에는 PHP Core에 포함된 많은 PHP 모듈이 포함되어 있습니다. AL2023 패키지는 [PHP 확장 커뮤니티 라이브러리 \(PECL\)](#)의 모든 패키지를 포함하는 것을 목표로 하지 않습니다.

AL2023에서 Python 사용

AL2023 Python 2.7이 제거되었으며 필요한 모든 구성 Python 요소가 이제 Python 3과 함께 작동하도록 작성되었습니다.

AL2023 버전에서는 고객 코드와의 호환성을 유지하기 `/usr/bin/python3` 위해 Python 3개를 사용할 수 있으며, AL2023와 함께 제공되는 Python 코드도 사용할 수 있습니다. 이 코드는 AL2023 수명 기간 동안 Python 3.9로 유지됩니다.

`/usr/bin/python3`가리키는 파이썬 버전은 시스템 Python으로 간주되며 AL2023 버전의 경우 Python 3.9입니다.

최신 버전 (예: Python 3.11)은 AL2023 패키지로 제공되며 업스트림 버전의 수명 기간 동안 지원됩니다. Python 파이썬 3.11이 얼마나 오랫동안 지원되는지에 대한 정보는 [파이썬 3.11](#)을 참조하십시오.

AL2023에 Python 여러 버전을 동시에 설치할 수 있습니다. 항상 Python 3.9이지만 `/usr/bin/python3`, Python의 각 버전은 네임스페이스가 지정되며 버전 번호로 찾을 수 있습니다. 예를 들어 `python3.11`이 설치되어 있으면 `/usr/bin/python3.11`은 `/usr/bin/python3.9`과 함께 존재하며 `/usr/bin/python3`은 `/usr/bin/python3.9`의 심볼릭 링크로 존재합니다.

Note

`/usr/bin/python3` 심볼릭 링크가 가리키는 내용을 변경하지 마십시오. 변경하면 AL2023 핵심 기능이 손상될 수 있습니다.

AL2023 Python 모듈

AL2023 버전에는 다양한 Python 모듈이 RPM으로 패키징되어 있습니다. 일반적으로 Python 모듈의 RPM은 Python 시스템 버전만을 대상으로 빌드되었습니다.

AL2023에서 Rust 사용

Amazon [RustLinux](#)에서 코드를 직접 작성하거나 AL2023 제공 툴체인을 사용하는 것이 좋습니다.

AL2와 마찬가지로 AL2023 역시 운영 체제의 수명 주기 내내 Rust 툴체인을 업데이트합니다. 이는 당사가 제공하는 툴체인의 CVE에 대한 결과이거나 분기 릴리스에 포함되어 있습니다.

[Rust](#)는 비교적 빠르게 업데이트되는 언어로서, 약 6주마다 새 릴리스가 출시됩니다. 이러한 릴리스에 새로운 언어 또는 표준 라이브러리 기능이 추가될 수 있습니다. AL2023 출시 기간 동안 새 버전의 Rust 툴체인이 통합되기는 하지만 업스트림 릴리즈에서는 이를 따라가지 못할 것입니다. Rust 따라서 AL2023 언어의 최첨단 기능을 사용하여 Rust 코드를 작성하려는 경우 에서 제공하는 Rust 툴체인을 사용하는 것이 적합하지 않을 수 있습니다. Rust

AL2023 서비스 기간 동안 패키지 구 버전은 리포지토리에서 제거되지 않습니다. 이전 Rust 툴체인이 필요한 경우 새 Rust 툴체인의 버그 및 보안 수정을 생략하고 모든 RPM에서 사용할 수 있는 것과 동일한 메커니즘을 사용하여 리포지토리에서 이전 버전을 설치할 수 있습니다.

AL2023 기반으로 자체 Rust 코드를 빌드하려는 경우 AL2023 에 포함된 툴체인을 사용할 수 있습니다. 단, 이 Rust 툴체인은 AL2023 수명 기간 동안 계속 발전할 수 있다는 점을 알고 있어야 합니다.

다음과 같이 작성된 AL2023 람다 함수 Rust

네이티브 코드로 Rust 컴파일되므로 Lambda는 사용자 지정 런타임으로 Rust 취급합니다. `provided.al2023` 런타임을 사용하여 AL2023 상의 Rust 함수를 Lambda에 배포할 수 있습니다.

자세한 내용은 개발자 안내서의 [Lambda 함수 Rust 빌드](#)를 참조하십시오. AWS Lambda

아마존 리눅스 2의 보안 및 규정 준수

Important

AWS 클라우드 서비스 또는 오픈 소스 프로젝트와 관련된 취약성이나 보안 문제가 있는 경우 [취약성 보고 페이지](#)를 사용하여 AWS 보안 팀에 문의하십시오.

의 클라우드 AWS 보안이 최우선 과제입니다. AWS 고객은 가장 보안에 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 기업과 기업 간의 AWS 공동 책임입니다. [공동 책임 모델](#)은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호하는 역할을 합니다. AWS 또한 안전하게 사용할 수 있는 서비스를 제공합니다. AL2에 적용되는 규정 준수 프로그램에 대해 자세히 알아보려면 규정 준수 [프로그램별 범위 내 AWS 서비스 규정 준수](#) 참조하십시오.
- 클라우드 내 보안 - 귀하의 책임은 귀하가 사용하는 AWS 서비스로 결정됩니다. 또한 귀하는 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

주제

- [AL2023 관련 아마존 리눅스 보안 권고](#)
- [AL2023 SELinux 모드 설정](#)
- [AL2023 버전에서 FIPS 모드 활성화](#)
- [AL2023 커널 하드닝](#)
- [AL2023 기반 UEFI 보안 부팅](#)

AL2023 관련 아마존 리눅스 보안 권고

Amazon Linux의 보안을 유지하기 위해 최선을 다하고 있지만 때로는 해결이 필요한 보안 문제가 있을 수 있습니다. 수정 사항이 나오면 권고가 발행됩니다. 권고 사항을 게시하는 주요 장소는 Amazon Linux 보안 센터 (ALAS) 입니다. 자세한 정보는 [Amazon Linux 보안 센터](#)를 참조하세요.

⚠ Important

[AWS 클라우드 서비스 또는 오픈 소스 프로젝트와 관련된 취약성이나 보안 문제가 있는 경우 취약성 보고 페이지를 사용하여 AWS 보안 팀에 문의하세요.](#)

Amazon Linux 팀은 AL2023 관련 문제 및 관련 업데이트에 대한 정보를 여러 위치에 게시합니다. 보안 도구가 이러한 기본 소스에서 정보를 가져와 결과를 제공합니다. 따라서 Amazon Linux에서 게시하는 기본 소스와 직접 상호 작용하는 것이 아니라 Amazon [Inspector](#)와 같이 선호하는 도구에서 제공하는 인터페이스와 상호 작용할 수 있습니다.

아마존 리눅스 보안 센터 공지

Amazon Linux 공지는 권고 사항에 적합하지 않은 항목에 대해 제공됩니다. 이 섹션에는 ALAS 자체에 대한 발표와 권고에 맞지 않는 정보가 포함되어 있습니다. 자세한 내용은 [Amazon Linux 보안 센터 \(ALAS\) 공지 사항](#)을 참조하십시오.

예를 들어, [2021-001 - Apache Log4j를 위한 Amazon Linux 핫패치 발표](#)는 권고가 아니라 발표에 적합합니다. 이번 발표에서 Amazon Linux는 고객이 Amazon Linux에 포함되지 않은 소프트웨어의 보안 문제를 완화하는 데 도움이 되는 패키지를 추가했습니다.

[아마존 리눅스 보안 센터 CVE 익스플로러](#)도 ALAS 발표를 통해 발표되었습니다. 자세한 내용은 CVE용 [새 웹 사이트](#)를 참조하십시오.

Amazon Linux 보안 센터 자주 묻는 질문

ALAS에 대한 몇 가지 자주 묻는 질문과 Amazon Linux가 CVE를 평가하는 방법에 대한 답변은 [Amazon Linux 보안 센터 \(ALAS\) 자주 묻는 질문 \(FAQ\)](#) 을 참조하십시오.

AL2023 SELinux 모드 설정

기본적으로 보안 강화 리눅스 (SELinux) 는 AL2023 모드로 설정되어 enabled 있으며 모드로 설정되어 있습니다. permissive 허용 모드에서는 권한 거부가 기록되지만 강제되지는 않습니다. SELinux 는 커널의 주요 하위 시스템에 강력하고 유연한 필수 액세스 제어 (MAC) 아키텍처를 제공하는 커널 기능 및 유틸리티를 포함하고 있습니다.

SELinux는 기밀 유지 및 무결성 요구 사항에 따라 정보를 분리하도록 하는 강화된 메커니즘을 제공합니다. 이렇게 정보를 분리하면 애플리케이션 보안 메커니즘의 변조 및 우회 위협이 줄어듭니다. 또한 악의적이거나 결함이 있는 애플리케이션으로 인해 발생할 수 있는 피해도 제한합니다.

SELinux에는 일상적인 보안 목표를 충족하도록 설계된 샘플 보안 정책 구성 파일 세트가 포함되어 있습니다.

SELinux 기능에 대한 자세한 내용은 [SELinux 노트](#) 및 [정책 언어](#)를 참조하세요.

주제

- [AL2023 기본 SELinux 상태 및 모드](#)
- [enforcing 모드로 변경](#)
- [AL2023 용 SELinux를 비활성화하는 옵션](#)

AL2023 기본 SELinux 상태 및 모드

AL2023의 경우 SELinux는 기본적으로 모드로 설정되어 `enabled` 있으며 모드로 설정되어 있습니다. `permissive` `permissive` 모드에서는 권한 거부 기록이 기록되지만 강제하지는 않습니다.

`getenforce` 또는 `sestatus` 명령은 현재 SELinux 상태, 정책 및 모드를 알려줍니다.

기본 상태가 `enabled` 및 `permissive`로 설정된 경우 `getenforce` 명령이 `permissive`로 변경됩니다.

이 `sestatus` 명령은 다음 예와 같이 SELinux 상태와 현재 SELinux 정책을 반환합니다.

```
$ sestatus
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:       /etc/selinux
Loaded policy name:           targeted
Current mode:                  permissive
Mode from config file:         permissive
Policy MLS status:            enabled
Policy deny_unknown status:    allowed
Memory protection checking:    actual (secure)
Max kernel policy version:     33
```

SELinux를 `permissive` 모드에서 실행하면 사용자가 파일에 잘못된 레이블을 지정할 수 있습니다. `disabled` 상태에서 SELinux를 실행하면 파일에 레이블이 지정되지 않습니다. 잘못된 파일이나 레이블이 없는 파일 모두 `enforcing` 모드로 변경되면 문제를 일으킬 수 있습니다.

SELinux는 이 문제를 방지하기 위해 파일의 레이블을 자동으로 다시 지정합니다. SELinux는 상태를 `enabled`로 변경할 때 자동으로 레이블을 다시 지정하여 레이블 문제를 방지합니다.

enforcing 모드로 변경

SELinux enforcing 모드로 실행하는 경우 SELinux enforcing 유틸리티가 구성된 정책입니다. SELinux 정책 규칙에 따라 액세스를 허용하거나 거부하여 일부 응용 프로그램의 기능을 제어합니다.

현재 SELinux 모드를 찾으려면 명령을 실행합니다. `getenforce`

```
getenforce
Permissive
```

구성 파일을 수정하여 **enforcing** 모드를 활성화합니다.

모드를 로 enforcing 변경하려면 다음 단계를 사용하십시오.

1. `/etc/selinux/config` 파일을 편집하여 enforcing 모드로 변경합니다. SELINUX 설정은 다음 예와 같아야 합니다.

```
SELINUX=enforcing
```

2. 시스템을 다시 시작하여 enforcing 모드 변경을 완료하세요.

```
$ sudo reboot
```

다음 부팅 시 시스템의 모든 파일 및 디렉터리에 SELinux 레이블을 다시 지정합니다. SELinux 또한 생성되었을 때 생성된 파일 및 디렉터리에 대한 SELinux 컨텍스트도 추가합니다. SELinux disabled

enforcing 모드로 변경한 후 SELinux 정책 규칙이 잘못되거나 누락되어 일부 작업이 SELinux 거부될 수 있습니다. 다음 명령을 사용하여 SELinux 거부하는 작업을 볼 수 있습니다.

```
$ sudo ausearch -m AVC,USER_AVC,SELINUX_ERR,USER_SELINUX_ERR -ts recent
```

cloud-init를 사용하여 **enforcing**을 활성화합니다.

또는 인스턴스를 실행할 때 다음 cloud-config을 사용자 데이터로 전송하여 enforcing 모드를 사용할 수 있습니다.

```
#cloud-config
selinux:
```

```
mode: enforcing
```

이렇게 설정하면 인스턴스가 재부팅됩니다. 안정성을 높이려면 인스턴스를 재부팅하는 것이 좋습니다. 하지만 다음 `cloud-config`를 입력하면 재부팅을 건너뛸 수 있습니다.

```
#cloud-config
selinux:
  mode: enforcing
  selinux_no_reboot: 1
```

AL2023 용 SELinux를 비활성화하는 옵션

사용하지 않도록 설정하면 SELinux 정책이 로드되거나 적용되지 않으며 AVC (Access Vector Cache) 메시지가 기록되지 않습니다. 실행으로 얻을 수 있는 모든 이점을 잃게 됩니다. SELinux

SELinux 비활성화하는 대신 `permissive` 모드를 사용하는 것이 좋습니다. SELinux 완전히 비활성화하는 경우보다 `permissive` 모드에서 실행하는 데 드는 비용만 조금 더 듭니다. `permissive` 모드에서 모드로 전환할 때는 `enforcing` 비활성화한 후 다시 `enforcing` 모드로 전환하는 것보다 구성 조정이 훨씬 적게 필요합니다. SELinux 파일에 레이블을 지정할 수 있으며, 시스템에 활성 정책이 거부된 작업을 추적하고 기록할 수 있습니다.

모드로 변경 SELinux `permissive`

SELinux `permissive` 모드에서 실행하면 SELinux 정책이 적용되지 않습니다. `permissive` 모드에서는 AVC 메시지를 SELinux 기록하지만 작업을 거부하지는 않습니다. 이러한 AVC 메시지를 문제 해결, 디버깅 및 정책 개선에 사용할 수 있습니다. SELinux

허용 모드로 SELinux 변경하려면 다음 단계를 사용하십시오.

1. `/etc/selinux/config` 파일을 편집하여 `permissive` 모드로 변경합니다. SELINUX값은 다음 예와 같아야 합니다.

```
SELINUX=permissive
```

2. 시스템을 다시 시작하여 `permissive` 모드 변경을 완료하세요.

```
sudo reboot
```

SELinux 비활성화

사용하지 않도록 설정하면 SELinux 정책이 로드되거나 적용되지 않으며 AVC 메시지가 기록되지 않습니다. 달리기 of 모든 이점을 잃게 됩니다. SELinux

SELinux 비활성화하려면 다음 단계를 사용하세요.

1. grubby 패키지가 설치되었는지 확인하십시오.

```
rpm -q grubby
grubby-version
```

2. 커널 명령줄에 selinux=0을 추가하여 부트로더를 구성하세요.

```
sudo grubby --update-kernel ALL --args selinux=0
```

3. 시스템을 재시작합니다.

```
sudo reboot
```

4. getenforce 명령을 실행하여 제대로 SELinux 설치되었는지 확인합니다 Disabled.

```
$ getenforce
Disabled
```

에 대한 SELinux 자세한 내용은 [SELinux노트북](#) 및 [SELinux구성을](#) 참조하십시오.

AL2023 버전에서 FIPS 모드 활성화

이 섹션에서는 AL2023에서 연방 정보 처리 표준(FIPS)을 활성화하는 방법을 안내합니다. FIPS에 대한 자세한 내용은 다음을 참조하세요.

- [연방 정보 처리 표준\(FIPS\)](#)
- [연방 정보 처리 표준 규정 준수 FAQs](#)

Note

이 섹션에서는 AL2023 FIPS 모드를 활성화하는 방법을 설명하며, AL2023 암호화 모듈의 인증 상태는 다루지 않습니다.

필수 조건

- 인터넷에 액세스하여 필수 패키지를 다운로드할 수 있는 기존 AL2023(AL2023.2 이상) Amazon EC2 인스턴스. AL2023 Amazon EC2 인스턴스를 실행하는 방법에 대한 자세한 내용은 [아마존 EC2 콘솔을 사용하여 AL2023 실행](#)을 참조하세요.
- SSH 또는 AWS Systems Manager를 사용하여 Amazon EC2 인스턴스를 연결합니다. 자세한 정보는 [AL2023 인스턴스에 연결](#)을 참조하세요.

Important

ED25519 SSH 사용자 키는 FIPS 모드에서 지원되지 않습니다. ED25519 SSH 키 페어를 사용하여 Amazon EC2 인스턴스를 실행한 경우 다른 알고리즘(예: RSA)을 사용하여 새 키를 생성해야 합니다. 그렇지 않으면 FIPS 모드를 활성화한 후 인스턴스 액세스 권한을 잃을 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 [키 페어 생성](#)을 참조하십시오.

FIPS 모드 활성화

1. SSH 또는 AWS Systems Manager로 인스턴스에 연결합니다.
2. 시스템이 최신 버전인지 확인합니다. 자세한 정보는 [AL2023 패키지 및 운영 체제 업데이트 관리](#)를 참조하세요.
3. `crypto-policies` 유틸리티가 설치되어 있는지 확인하십시오. `up-to-date`

```
sudo dnf -y install crypto-policies crypto-policies-scripts
```

4. 다음 명령을 실행하여 FIPS 모드를 활성화합니다.

```
sudo fips-mode-setup --enable
```

5. 다음 명령을 사용하여 인스턴스를 재부팅하세요.

```
sudo reboot
```

6. FIPS 모드가 활성화되었는지 확인하려면 인스턴스에 다시 연결하고 다음 명령을 실행합니다.

```
sudo fips-mode-setup --check
```

다음 예시 출력은 FIPS 모드가 활성화되었음을 나타냅니다.

```
FIPS mode is enabled.
Initramfs fips module is enabled.
The current crypto policy (FIPS) is based on the FIPS policy.
```

AL2023 커널 하드닝

AL2023 버전의 6.1 Linux 커널은 몇 가지 강화 옵션 및 기능으로 구성 및 구축되었습니다.

커널 강화 옵션 (독립 아키텍처)

CONFIG 옵션	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_ACPI_CUSTOM_METHOD	n	n
CONFIG_BINFORMT_MISC	m	m
CONFIG_BUG	y	y
CONFIG_BUG_ON_DATA_CORRUPTION	y	y
CONFIG_CFI_CLANG	N/A	해당 사항 없음
CONFIG_CFI_PERMISSIVE	해당 사항 없음	해당 사항 없음
CONFIG_COMPAT	y	y

CONFIG 옵션	AL2023/6.1/aarch64	AL2023/6.1/x86_64
<u>CONFIG_COMPAT_BRK</u>	n	n
<u>CONFIG_COMPAT_VDSO</u>	해당 사항 없음	n
<u>CONFIG_DEBUG_CREDENTIALS</u>	n	n
<u>CONFIG_DEBUG_LIST</u>	y	y
<u>CONFIG_DEBUG_NOTIFIERS</u>	n	n
<u>CONFIG_DEBUG_SG</u>	n	n
<u>CONFIG_DEBUG_VIRTUAL</u>	n	n
<u>CONFIG_DEBUG_WX</u>	n	n
<u>CONFIG_DEFAULT_MMAP_MIN_ADDR</u>	65536	65536
<u>CONFIG_DEVMEM</u>	해당 사항 없음	해당 사항 없음
<u>CONFIG_DEVMEM</u>	n	n
<u>CONFIG_EFI_DISABLE_PCI_DMA</u>	n	n
<u>CONFIG_FORTIFY_SOURCE</u>	y	y
<u>CONFIG_HARDENED_USERCOPY</u>	y	y
<u>CONFIG_HARDENED_USERCOPY_FALLBACK</u>	해당 사항 없음	해당 사항 없음

CONFIG 옵션	AL2023/6.1/aarch64	AL2023/6.1/x86_64
<u>CONFIG_HARDENED_USERCOPY_PAGESPAN</u>	해당 사항 없음	해당 사항 없음
<u>CONFIG_HIBERNATION</u>	y	y
<u>CONFIG_HW_RANDOM_TPM</u>	해당 사항 없음	해당 사항 없음
<u>CONFIG_INET_DIAG</u>	m	m
<u>CONFIG_INIT_ON_ALLOC_DEFAULT_ON</u>	n	n
<u>CONFIG_INIT_ON_FREE_DEFAULT_ON</u>	n	n
<u>CONFIG_INIT_STACK_ALL_ZERO</u>	해당 사항 없음	해당 사항 없음
<u>CONFIG_IOMMU_DEFAULT_DMA_STRICT</u>	n	n
<u>CONFIG_IOMMU_SUPPORT</u>	y	y
<u>CONFIG_IO_STRICT_DEVMEM</u>	해당 사항 없음	해당 사항 없음
<u>CONFIG_KEXEC</u>	y	y
<u>CONFIG_KFENCE</u>	n	n
<u>CONFIG_LDISC_AUTOLOAD</u>	n	n
<u>CONFIG_LEGACY_PTYS</u>	n	n
<u>CONFIG_LOCK_DOWN_KERNEL_FORCE_CONFIDENTIALITY</u>	n	n

CONFIG 옵션	AL2023/6.1/aarch64	AL2023/6.1/x86_64
<u>CONFIG_MODULES</u>	y	y
<u>CONFIG_MODULE_SIG</u>	y	y
<u>CONFIG_MODULE_SIG_</u> <u>ALL</u>	y	y
<u>CONFIG_MODULE_SIG_</u> <u>FORCE</u>	n	n
<u>CONFIG_MODULE_SIG_</u> <u>HASH</u>	sha512	sha512
<u>CONFIG_MODULE_SIG_</u> <u>KEY</u>	certs/signing_key. pem	certs/signing_key. pem
<u>CONFIG_MODULE_SIG_</u> <u>SHA512</u>	y	y
<u>CONFIG_PAGE_POISON</u> <u>ING</u>	n	n
<u>CONFIG_PAGE_POISON</u> <u>ING_NO_SANITY</u>	해당 사항 없음	해당 사항 없음
<u>CONFIG_PAGE_POISON</u> <u>ING_ZERO</u>	해당 사항 없음	해당 사항 없음
<u>CONFIG_PANIC_ON_OOPS</u>	y	y
<u>CONFIG_PANIC_TIMEOUT</u>	0	0
<u>CONFIG_PROC_KCORE</u>	y	y
<u>CONFIG_RANDOMIZE_K</u> <u>STACK_OFFSET_DEFAU</u> <u>LT</u>	n	n

CONFIG 옵션	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_RANDOM_TRUST_BOOTLOADER	y	y
CONFIG_RANDOM_TRUST_CPU	y	y
CONFIG_REFCOUNT_FULL	해당 사항 없음	해당 사항 없음
CONFIG_SCHED_CORE	해당 사항 없음	y
CONFIG_SCHED_STACK_END_CHECK	y	y
CONFIG_SECCOMP	y	y
CONFIG_SECCOMP_FILTER	y	y
CONFIG_SECURITY	y	y
CONFIG_SECURITY_DMESG_RESTRICT	y	y
CONFIG_SECURITY_LANDLOCK	n	n
CONFIG_SECURITY_LOCKDOWN_LSM	y	y
CONFIG_SECURITY_LOCKDOWN_LSM_EARLY	y	y
CONFIG_SECURITY_SELINUX_BOOTPARAM	y	y
CONFIG_SECURITY_SELINUX_DEVELOP	y	y

CONFIG 옵션	AL2023/6.1/aarch64	AL2023/6.1/x86_64
<u>CONFIG_SECURITY_SELINUX_DISABLE</u>	n	n
<u>CONFIG_SECURITY_WRITABLE_HOOKS</u>	해당 사항 없음	N/A
<u>CONFIG_SECURITY_YAMA</u>	y	y
<u>CONFIG_SHUFFLE_PAGE_ALLOCATOR</u>	y	y
<u>CONFIG_SLAB_FREELIST_HARDENED</u>	y	y
<u>CONFIG_SLAB_FREELIST_RANDOM</u>	y	y
<u>CONFIG_SLUB_DEBUG</u>	y	y
<u>CONFIG_STACKPROTECTOR</u>	y	y
<u>CONFIG_STACKPROTECTOR_STRONG</u>	y	y
<u>CONFIG_STATIC_USERMODEHELPER</u>	n	n
<u>CONFIG_STRICT_DEVMEM</u>	n	n
<u>CONFIG_STRICT_KERNEL_RWX</u>	y	y
<u>CONFIG_STRICT_MODULE_RWX</u>	y	y
<u>CONFIG_SYN_COOKIES</u>	y	y

CONFIG 옵션	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_VMAP_STACK	y	y
CONFIG_WERROR	n	n
CONFIG_ZERO_CALL_U SED_REGS	n	n

런타임 시 ACPI 메서드를 삽입/교체할 수 있도록 허용 (CONFIG_ACPI_CUSTOM_METHOD)

root 사용자가 임의의 커널 메모리에 쓸 수 있으므로 Amazon Linux에서 이 옵션은 비활성화됩니다.

이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

기타 바이너리 포맷 (**binfmt_misc**)

이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다. 이는 AL2023 선택 기능이며 커널 모듈로 빌드됩니다.

BUG() 지원

이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

커널 메모리 구조의 유효성을 검사할 때 커널에서 데이터 손상이 발생하면 **BUG()**가 생깁니다.

Linux 커널 기능으로 데이터 구조의 내부 일관성을 검사할 수 있으며 데이터 손상이 감지되면 **BUG()**가 발생합니다.

이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

COMPAT_BRK

이 옵션을 비활성화(Amazon Linux에서 커널 구성)하면 `randomize_va_space` sysctl 설정은 2로 기본 설정되며 mmap 베이스, 스택 및 VDSO 페이지 무작위화를 기반으로 힙 무작위화도 활성화됩니다.

이 옵션은 1996년 이전의 `libc.so.5` 바이너리와 호환하기 위해 커널에 존재합니다.

이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

COMPAT_VDSO

이 구성 옵션은 aarch64가 아닌 x86-64에 해당합니다. n로 설정하면 Amazon Linux 커널에서 32비트 가상 동적 공유 객체(VDSO)를 예측 가능한 주소에 표시할 수 없습니다. 이 옵션을 n로 설정하면 glibc 오류가 발생한 것으로 알려진 가장 최신 버전은 2004년에 출시된 glibc 2.3.3입니다.

이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

CONFIG_DEBUG 게이트 강화

CONFIG_DEBUG가 제어하는 Linux 커널 구성 옵션은 일반적으로 디버깅 문제를 위해 빌드된 커널에서 사용하려고 만들었으며 성능은 우선 순위가 아닙니다. AL2023 기능을 CONFIG_DEBUG_LIST 사용하면 강화 옵션을 사용할 수 있습니다.

IOMMU를 구성하기 전에 EFI 스텝에서 PCI 디바이스용 DMA를 비활성화하세요.

이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다.

커널과 사용자 공간 간 메모리 복제를 위한 강화

커널이 사용자 공간으로 메모리를 복사하거나 사용자 공간 메모리를 복사해야 하는 경우 이 옵션을 사용하면 일부 검사 기능을 통해 일부 클래스의 힙 오버플로 문제를 방지할 수 있습니다.

커널 4.16~5.15에는 커널 개발자가 WARN()를 통해 누락된 허용 목록 항목을 찾을 수 있도록 하는 CONFIG_HARDENED_USERCOPY_FALLBACK 옵션이 있었습니다. AL2023 버전은 6.1 커널을 제공하기 때문에 이 옵션은 더 이상 AL2023 버전과 관련이 없습니다.

이 CONFIG_HARDENED_USERCOPY_PAGESPAN 옵션은 주로 개발자를 위한 디버깅 옵션으로 커널에 존재했으며 AL2023 6.1 커널에는 더 이상 적용되지 않습니다.

이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

하이버네이트 지원

이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다. 이 옵션을 활성화해야 [온디맨드 인스턴스 하이버네이트](#)와 [중단된 스팟 인스턴스 하이버네이트](#) 기능을 지원할 수 있습니다.

난수 생성

AL2023 커널은 EC2 내에서 적절한 엔트로피를 사용할 수 있도록 구성되어 있습니다.

CONFIG_INET_DIAG

이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다. 이는 AL2023 선택 기능이며 커널 모듈로 빌드됩니다.

할당 및 할당 해제 시 모든 커널 페이지 및 슬래브 할당자 메모리를 0으로 설정합니다.

이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다. 기본 기능으로 설정하면 성능에 영향을 미칠 수 있으므로 AL2023에서는 이러한 옵션이 비활성화됩니다. 커널 명령줄에 `init_on_alloc=1`을 추가하면 `CONFIG_INIT_ON_ALLOC_DEFAULT_ON` 동작을 활성화할 수 있고, 추가하여 `init_on_free=1`를 추가하면 `CONFIG_INIT_ON_FREE_DEFAULT_ON` 동작을 활성화할 수 있습니다.

모든 스택 변수를 0(`CONFIG_INIT_STACK_ALL_ZERO`)으로 초기화합니다.

이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다. 이 옵션은 GCC 12 이상에서 작동하지만 AL2023에 GCC 11을 함께 제공합니다.

커널 모듈 서명

AL2023 는 커널 모듈의 서명을 서명하고 서명을 검증합니다. 모듈에 유효한 서명이 있어야 하는 이 `CONFIG_MODULE_SIG_FORCE` 옵션은 타사 모듈을 구축하는 사용자의 호환성을 유지하기 위해 활성화되지 않습니다. 모든 커널 모듈에 서명이 있는지 확인하고 싶은 사용자는 이것이 적용된 [Lockdown Linux 보안 모듈\(LSM\)](#) 을 구성할 수 있습니다.

kexec

이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다. 이 옵션은 `kdump` 기능을 사용할 수 있도록 활성화되어 있습니다.

IOMMU 지원

AL2023 덕분에 IOMMU를 지원할 수 있습니다. 이 `CONFIG_IOMMU_DEFAULT_DMA_STRICT` 옵션은 기본적으로 활성화되어 있지 않지만 커널 명령줄에 `iommu.passthrough=0 iommu.strict=1`를 추가하면 이 기능을 설치할 수 있습니다.

kfence

이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다.

레거시 pty 지원

AL2023 는 최신 PTY 인터페이스 () devpts 를 사용합니다.

이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

Lockdown Linux 보안 모듈(LSM)

AL2023 는 lockdown LSM을 빌드하며, LSM은 보안 부팅을 사용할 때 커널을 자동으로 잠급니다.

CONFIG_LOCK_DOWN_KERNEL_FORCE_CONFIDENTIALITY 옵션이 활성화되어 있지 않습니다. 이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다. 보안 부팅을 사용하지 않는 경우 Lockdown LSM을 활성화하고 원하는 대로 구성할 수 있습니다.

페이지 포이징

이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다. 마찬가지로 [할당 및 할당 해제 시 모든 커널 페이지 및 슬래브 할당자 메모리를 0으로 설정합니다.](#), AL2023 커널에서도 성능에 영향을 미칠 수 있으므로 이 기능은 비활성화되어 있습니다.

스택 프로텍터

AL2023 커널은 스택 프로텍터 기능을 옵션으로 GCC 활성화하여 빌드되었습니다. `-fstack-protector-strong`

이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

seccomp BPF API

seccomp 강화 기능으로 systemd와 컨테이너 런타임과 같은 소프트웨어에서 사용자 공간 애플리케이션을 강화할 수 있습니다.

이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

panic() 타임아웃

AL2023 커널은 이 값을 로 설정하여 구성되어 있습니다. 즉, 패닉이 발생한 후 커널이 재부팅되지 않습니다. 이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다. 이는 `sysctl, /proc/sys/kernel/panic` 및 커널 명령줄로 구성할 수 있습니다.

보안 모델

AL2023 는 기본적으로 허용 모드에서 SELinux를 활성화합니다. 자세한 정보는 [AL2023 SELinux 모드 설정](#)을 참조하세요.

[Lockdown Linux 보안 모듈\(LSM\)](#) 및 `yama` 모듈도 활성화되어 있습니다.

/proc/kcore

이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다.

시스템 호출 입력 시 커널 스택 오프셋 랜덤

이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다. 커널 명령줄을 `randomize_kstack_offset=on`로 설정하면 이 기능을 활성화할 수 있습니다.

참조 카운팅 검사 (CONFIG_REFCOUNT_FULL)

이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다. 이 옵션은 성능에 영향을 미칠 수 있으므로 현재 활성화되어 있지 않습니다.

스케줄러의 SMT 코어 인식 (CONFIG_SCHED_CORE)

사용자 공간 애플리케이션에서 사용할 수 있는 CONFIG_SCHED_CORE AL2023 커널이 내장되어 있습니다. `prctl(PR_SCHED_CORE)` 이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

schedule()(CONFIG_SCHED_STACK_END_CHECK) 호출 시 스택 손상 확인

AL2023 커널은 CONFIG_SCHED_STACK_END_CHECK 활성화된 상태로 빌드되었습니다. 이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

메모리 할당 강화

AL2023 커널을 사용하

면 `CONFIG_SHUFFLE_PAGE_ALLOCATOR`, `CONFIG_SLAB_FREELIST_HARDENED`, 옵션을 사용하여 커널 메모리 할당자를 강화할 수 있습니다. `CONFIG_SLAB_FREELIST_RANDOM` 이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

SLUB 디버깅 지원

이 옵션을 사용하면 커널 `CONFIG_SLUB_DEBUG` 명령줄에서 활성화할 수 있는 할당자에 대한 선택적 디버깅 기능을 사용할 수 있으므로 AL2023 커널이 활성화됩니다. 이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

`CONFIG_STATIC_USERMODEHELPER`

이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다. `CONFIG_STATIC_USERMODEHELPER`를 사용하려면 배포판의 특별 지원이 필요한데 현재 Amazon Linux에는 없습니다.

읽기 전용 커널 텍스트 및 rodata (`CONFIG_STRICT_KERNEL_RWX`와 `CONFIG_STRICT_MODULE_RWX`)

AL2023 커널은 커널 및 커널 모듈 텍스트와 rodata 메모리를 읽기 전용으로 표시하고 텍스트가 아닌 메모리는 실행 불가능으로 표시하도록 구성되어 있습니다. 이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

TCP syncookie 지원 (`CONFIG_SYN_COOKIES`)

AL2023 커널은 TCP 동기화 쿠키를 지원하도록 구축되었습니다. 이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

가드 페이지가 있는 가상 매핑 스택 (`CONFIG_VMAP_STACK`)

로 구축된 AL2023 커널은 가드 페이지와 함께 `CONFIG_VMAP_STACK` 커널 스택을 가상으로 매핑할 수 있도록 만들어졌습니다. 이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

컴파일러 경고를 오류(`CONFIG_WERROR`)로 빌드

이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다.

함수를 종료(**CONFIG_ZERO_CALL_USED_REGS**)할 때 레지스터 제로화

이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다.

사용자 공간 할당에 필요한 최소 주소

이 강화 옵션으로 커널 NULL 포인터 버그의 영향을 줄일 수 있습니다. 이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

clang 특정 강화 옵션

AL2023 커널은 GCC 대신 로 clang 빌드되었으므로 CONFIG_CFI_CLANG 강화 옵션을 활성화할 수 없으며 이 역시 적용할 수 없습니다. CONFIG_CFI_PERMISSIVE 이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다.

x86-64 전용 커널 강화 옵션

CONFIG 옵션	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_AMD_IOMMU	N/A	y
CONFIG_AMD_IOMMU_V2	해당 사항 없음	y
CONFIG_IA32_EMULAT ION	해당 사항 없음	y
CONFIG_INTEL_IOMMU	해당 사항 없음	y
CONFIG_INTEL_IOMMU _DEFAULT_ON	해당 사항 없음	n
CONFIG_INTEL_IOMMU _SVM	해당 사항 없음	n
CONFIG_LEGACY_VSYS CALL_NONE	해당 사항 없음	n
CONFIG_MODIFY_LDT_ SYSCALL	해당 사항 없음	n

CONFIG 옵션	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_PAGE_TABLE_ISOLATION	해당 사항 없음	y
CONFIG_RANDOMIZE_MEMORY	해당 사항 없음	y
CONFIG_X86_64	해당 사항 없음	y
CONFIG_X86_MSR	해당 사항 없음	y
CONFIG_X86_VSYSCALL_EMULATION	해당 사항 없음	y
CONFIG_X86_X32	해당 사항 없음	해당 사항 없음
CONFIG_X86_X32_ABI	N/A	n

x86-64 지원

물리적 주소 확장(PAE) 및 실행 안 함(NX) 비트 지원이 x86-64 기본 지원입니다. 이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

AMD와 Intel IOMMU를 지원합니다.

AL2023 커널은 AMD와 IOMMU를 인텔을 지원하여 빌드됩니다. 이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

이 CONFIG_INTEL_IOMMU_DEFAULT_ON 옵션은 설정되지 않았지만 커널 명령줄로 intel_iommu=on를 작성하면 활성화됩니다. 이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다.

이 CONFIG_INTEL_IOMMU_SVM 옵션은 현재 AL2023 버전에서 활성화되어 있지 않습니다. 이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다.

32비트 사용자 공간 지원

Important

32비트 x86 사용자 공간 지원은 중단되었으며 향후 Amazon Linux 주요 버전에서 32비트 사용자 공간 바이너리를 실행할 수 없습니다.

Note

AL2023 에는 더 이상 32비트 패키지가 포함되지 않지만 커널은 32비트 사용자 공간 실행을 계속 지원합니다. 자세한 정보는 [32비트 x86 \(i686\) 패키지](#)을 참조하세요.

32비트 사용자 공간 응용 프로그램 실행을 지원하기 위해 AL2023 에서는 이 옵션을 활성화하지 않고, 및 CONFIG_X86_VSYSCALL_EMULATION 옵션을 활성화합니다 CONFIG_IA32_EMULATION. CONFIG_COMPAT CONFIG_X86_VSYSCALL_EMULATION 이 옵션은 [커널 자체 보호 프로젝트 \(KSPP\) 권장 설정](#) 중 하나지만 KSPP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다.

64비트 프로세서 x32 네이티브 32비트 ABI는 활성화되지 않았습니다 (CONFIG_X86_X32 및 CONFIG_X86_X32_ABI). 이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

x86 모델별 레지스터(MSR) 지원

이 CONFIG_X86_MSR 옵션은 turbostat 지원을 위해 활성화되어 있습니다. 이 옵션은 [커널 자체 보호 프로젝트 \(KSPP\) 권장 설정](#) 중 하나지만 KSPP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다.

modify_ldt 시스콜

AL2023 사용자는 사용자 프로그램이 syscall을 사용하여 x86 로컬 디스크립터 테이블 (LDT) 을 수정하는 것을 허용하지 않습니다. modify_ldt 이 호출을 사용하려면 16비트 또는 세그먼트 코드를 실행해야 하며 이 호출이 없으면 dosemu 소프트웨어, WINE에서 실행하는 프로그램, 실행하거나 아주 오래된 스테딩 라이브러리가 중단됩니다. 이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

사용자 모드에서 커널 매핑을 제거합니다.

AL2023 커널 주소의 대부분이 사용자 공간에 매핑되지 않도록 커널을 구성합니다. 이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

커널 메모리 섹션 무작위화

AL2023 커널 메모리 섹션의 기본 가상 주소를 무작위로 지정하도록 커널을 구성합니다. 이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

aarch64 전용 커널 강화 옵션

CONFIG 옵션	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_ARM64_BTI	y	N/A
CONFIG_ARM64_BTI_KERNEL	해당 사항 없음	해당 사항 없음
CONFIG_ARM64_PTR_AUTH	y	해당 사항 없음
CONFIG_ARM64_PTR_AUTH_KERNEL	y	해당 사항 없음
CONFIG_ARM64_SW_TTBR0_PAN	y	해당 사항 없음
CONFIG_UNMAP_KERNEL_AT_EL0	y	N/A

브랜치 타겟 식별

AL2023 커널을 사용하면 분기 대상 식별 () 을 지원할 수 있습니다. [CONFIG_ARM64_BTI](#) 이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

이 [CONFIG_ARM64_BTI_KERNEL](#) 옵션은 GCC로 빌드되어 AL2023에서 사용할 수 없으며, 이 옵션으로 커널 빌드하기는 [gcc 버그로 인해 현재 업스트림 커널에서 비활성화되어 있습니다](#). 이 옵션은 [커널 자체 보호 프로젝트 \(KSP\) 권장 설정](#) 중 하나지만 KSP 권장 설정으로 AL2023 구성 옵션을 설정하지 않습니다.

포인터 인증 (CONFIG_ARM64_PTR_AUTH)

AL2023 커널은 포인터 인증 확장 (ARMv8.3 확장의 일부) 을 지원하도록 구축되었으며, 이는 반환 지향 프로그래밍 (ROP) 기법을 완화하는 데 사용할 수 있습니다. [Graviton](#) 포인터 인증에 필요한 하드웨어 지원은 Graviton 3에 추가되었습니다.

이 CONFIG_ARM64_PTR_AUTH 옵션은 활성화되어 있으며 사용자 공간 포인터 인증을 지원합니다. 이 CONFIG_ARM64_PTR_AUTH_KERNEL 옵션도 활성화되었으므로 AL2023 커널은 자체적으로 반송 주소 보호 기능을 사용할 수 있습니다.

이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

권한 있는 액세스 에뮬레이션 (TTBR0_EL1 전환 사용하지 않음)

사용자 액세스 루틴으로만 일시적으로만 TTBR0_EL1이 유효한 값으로 설정되므로 이 옵션으로 커널이 사용자 공간 메모리에 직접 액세스하는 것을 방지할 수 있습니다.

이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

사용자 공간에서 실행 중인 경우 커널 매핑을 해제합니다.

AL2023 커널은 userspace () CONFIG_UNMAP_KERNEL_AT_EL0 에서 실행할 때 커널 매핑을 해제하도록 구성되어 있습니다. 이 옵션은 [커널 자체 보호 프로젝트 권장 설정](#) 중 하나입니다.

AL2023 기반 UEFI 보안 부팅

AL2023 릴리즈 2023.1부터 UEFI 보안 부팅을 지원합니다. UEFI와 UEFI 보안 부팅을 모두 지원하는 Amazon EC2 인스턴스로 AL2023을 사용하세요. 자세한 내용은 Amazon EC2 사용 설명서의 [인스턴스 시작](#)을 참조하십시오.

UEFI 보안 부팅이 활성화된 AL2023 인스턴스는 Linux 커널과 모듈을 포함하여 서명된 커널 수준 코드만 Amazon 허용하므로 인스턴스가 서명한 커널 수준 코드만 실행하도록 할 수 있습니다. AWS

Amazon EC2 인스턴스 및 UEFI 보안 부팅에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [UEFI](#) 보안 부팅을 참조하십시오.

필수 조건

- AL2023 릴리즈 2023.1 이상에 설치된 AMI가 필요합니다.

- 인스턴스 유형은 UEFI 보안 부팅을 지원해야 합니다. 자세한 내용은 Amazon EC2 사용 설명서의 [인스턴스 시작](#)을 참조하십시오.

AL2023 기반 UEFI 보안 부팅 활성화

기본 AL2023 AMI는 당사 키 서명이 있는 커널과 부트로더와 통합되어 있습니다. 기존 인스턴스를 등록하거나 스냅샷 이미지 등록으로 UEFI 보안 부팅이 활성화된 AMI를 생성하여 UEFI 보안 부팅을 사용할 수 있습니다. UEFI 보안 부팅은 기본 AL2023 AMI에서 사용할 수 없습니다.

인스턴스 유형이 UEFI를 지원하는 경우 이러한 AMI로 시작된 인스턴스가 UEFI 펌웨어를 사용하도록 AL2023 AMI 부팅 모드는 `uefi-preferred`로 설정되어 있습니다. 인스턴스 유형이 UEFI를 지원하지 않는 경우 인스턴스는 레거시 BIOS 펌웨어에서 실행합니다. 인스턴스가 레거시 BIOS 모드에서 시작되면 UEFI 보안 부팅이 되지 않습니다.

Amazon EC2 인스턴스의 AMI 부팅 모드에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [부팅 모드](#)를 참조하십시오.

주제

- [기존 인스턴스 등록](#)
- [스냅샷 이미지 등록](#)
- [폐기 업데이트](#)
- [UEFI 보안 부팅이 AL2023 환경에서 작동하는 방식](#)
- [자체 키 등록](#)

기존 인스턴스 등록

기존 인스턴스를 등록하려면 펌웨어가 부트로더를 검증하고 부트로더가 다음 부팅 시 커널을 검증하는 키 세트를 특정 UEFI 펌웨어 변수로 입력하세요.

1. Amazon Linux는 등록 프로세스를 간소화하는 도구를 제공합니다. 다음 명령을 실행하여 필요한 키와 인증서 세트를 인스턴스에 프로비저닝합니다.

```
sudo amazon-linux-sb enroll
```

2. 다음 명령을 실행하여 인스턴스를 재부팅합니다. 인스턴스가 재부팅되면 UEFI 보안 부팅이 활성화됩니다.

sudo reboot**Note**

Amazon Linux AMI는 현재 Nitro 신뢰 플랫폼 모듈(NitroTPM)을 지원하지 않습니다. UEFI 보안 부팅뿐만 아니라 NitroTPM이 필요한 경우 다음 섹션의 정보를 사용하세요.

스냅샷 이미지 등록

Amazon EC2 `register-image` API를 사용하여 Amazon EBS 루트 볼륨의 스냅샷에서 AMI를 등록할 때 UEFI 변수 저장소의 상태가 포함된 바이너리 볼륨으로 AMI를 프로비저닝할 수 있습니다. AL2023 UefiData로 UEFI 보안 부팅이 활성화되므로 이전 섹션의 단계를 실행하지 않아도 됩니다.

바이너리 볼륨 생성 및 사용에 대한 자세한 내용은 Amazon EC2 사용 설명서의 [옵션 B: 미리 채워진 변수 저장소가 포함된 바이너리 볼륨 생성](#)을 참조하십시오.

AL2023은 Amazon EC2 인스턴스에서 직접 사용할 수 있는 사전 빌드된 바이너리 볼륨을 제공합니다. 바이너리 볼륨은 실행 중인 인스턴스의 `/usr/share/amazon-linux-sb-keys/uefi.vars`에 있습니다. AL2023 AMI 2023.1 릴리스부터 기본으로 설치된 `amazon-linux-sb-keys` RPM 패키지에 이 볼륨이 있습니다.

Note

최신 버전의 키와 폐기를 사용하려면 AMI를 생성할 때 사용한 것과 동일한 AL2023 릴리스의 볼륨을 사용하세요.

이미지를 등록할 때는 `uefi`로 설정된 [RegisterImage](#) API의 `BootMode` 파라미터를 사용하는 것이 좋습니다. 이렇게 하면 `TpmSupport` 파라미터를 `v2.0`로 설정하여 NitroTPM을 활성화할 수 있습니다. UEFI를 지원하지 않는 인스턴스 유형으로 전환할 때 UEFI 보안 부팅이 활성화되고 실수로 비활성화되지 않도록 `BootMode` 설정을 `uefi`로 하세요.

NitroTPM에 대한 자세한 내용은 Amazon EC2 사용 [설명서의 NitroTPM](#)을 참조하십시오.

폐기 업데이트

Amazon Linux에서 업데이트된 키 서명이 포함된 최신 부트로더 버전 grub2 또는 Linux 커널을 배포해야 합니다. 이 경우 구 버전 부트로더에서 악용될 수 있는 버그가 UEFI 보안 부팅 검증 프로세스를 우회할 가능성을 방지하기 위해 구 버전 키를 폐기해야 합니다.

grub2 또는 kernel 패키지를 업데이트하면 실행 중인 인스턴스의 UEFI 변수 저장소 취소 목록을 자동으로 업데이트합니다. 즉, UEFI 보안 부팅을 활성화하면 패키지 보안 업데이트를 설치한 후에는 이전 패키지 버전을 더 이상 실행할 수 없습니다.

UEFI 보안 부팅이 AL2023 환경에서 작동하는 방식

다른 Linux 배포판과 달리 Amazon Linux에는 1단계 부트로더 역할을 하는 shim (shim) 추가 구성 요소가 없습니다. shim은 일반적으로 Microsoft 키 서명이 있습니다. 예를 들어 shim이 포함된 Linux 배포판에서 shim은 자체 코드를 사용하여 Linux 커널을 검증하는 grub2 부트로더를 로드합니다. 또한 shim은 mokutil 도구로 UEFI 변수 저장소에서 머신 소유 키(MOK) 데이터베이스에 자체 키 및 폐기 세트를 유지 관리합니다.

Amazon Linux에는 shim이 없습니다. AMI 소유자가 UEFI 변수를 관리하기 때문에 이 중간 단계는 필요하지 않으며 시작 및 부팅에 부정적인 영향을 미칠 수 있습니다. 또한 원하지 않는 바이너리가 실행될 가능성을 줄이기 위해 기본적으로 공급업체 키에 대한 신뢰를 포함시키지 않았습니다. 하지만 고객은 원한다면 바이너리를 포함시킬 수 있습니다.

Amazon Linux를 사용하면 UEFI로 grub2 부트로더를 로드하고 검증할 수 있습니다. Linux 커널을 로드한 후 UEFI를 사용하여 Linux 커널을 검증하도록 grub2 부트로더를 수정했습니다. 따라서 일반 UEFI db 변수(인증된 키 데이터베이스)에 저장된 동일한 인증서를 사용하여 Linux 커널을 검증하고 부트로더 및 다른 UEFI 바이너리와 동일한 dbx 변수(폐기 데이터베이스)를 테스트합니다. db 데이터베이스 및 dbx 데이터베이스에 대한 액세스를 제어하는 자체 PK 및 KEK 키로 shim과 같은 중개자 없이 필요에 따라 서명된 업데이트 및 폐기를 수행할 수 있습니다.

UEFI 보안 부팅에 대한 자세한 내용은 Amazon EC2 [사용 설명서의 UEFI 보안 부팅 작동 방식을](#) 참조하십시오.

자체 키 등록

이전 섹션에서 설명했듯이 Amazon EC2에서 UEFI 보안 부팅을 사용하기 위해 shimAmazon Linux에 가 없어도 됩니다. 다른 Linux 배포판에 대한 설명서에서 AL2023에는 없는 mokutil로 머신 소유자 키(MOK) 데이터베이스를 관리하는 방법에 대한 문서가 있습니다. shim 및 MOK 환경에서는

Amazon EC2에서 UEFI 보안 부팅할 때 사용할 수 없는 UEFI 펌웨어 키 등록의 일부 제한을 우회합니다. Amazon EC2에는 UEFI 변수 저장소의 키를 쉽게 조작할 수 있는 메커니즘이 있습니다.

키를 직접 등록하려면 기존 인스턴스 내에서 변수 저장소를 조작하거나([인스턴스 내에서 변수 저장소에 키 추가](#)(Add keys to the variable store from within the instance) 참조) 미리 채워진 바이너리 블록을 생성하면 됩니다([미리 채워진 변수 저장소가 포함된 바이너리 블록 생성](#)(Create a binary blob containing a pre-filled variable store) 참조).

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.