aws

사용 설명서

Amazon Linux 2

Amazon Linux 2: 사용 설명서

Table of Contents

Amazon Linux 2란 무엇인가요?

Amazon Linux 2(AL2)는 Amazon Web Services()의 Linux 운영 체제입니다AWS. AL2는 Amazon EC2에서 실행되는 애플리케이션에 안정적이고 안전하며 성능이 뛰어난 환경을 제공하도록 설계되었습니다. 또한 시작 구성 도구와 널리 사용되는 여러 AWS라이브러리 및 도구를 AWS포함하여와 효율적으로 통합할 수 있는 패키지도 포함되어 있습니다.는 AL2를 실행하는 모든 인스턴스에 대해 지속적인 보안 및 유지 관리 업데이트를 AWS제공합니다. CentOS에서 개발된 많은 애플리케이션과 유사한 배포는 AL2에서 실행됩니다. AL2는 추가 비용 없이 제공됩니다.

> **Note**
>
> AL2는 더 이상 Amazon Linux의 최신 버전이 아닙니다. AL2023은 AL2의 후속 버전입니다. 자세한 내용은 AL2023 사용 설명서의 AL2 및 AL2023 비교 AL2023 및 AL2023의 패키지 변경 AL2023 사항 목록을 참조하세요.

> **Note**
>
> AL2는 업스트림 Firefox 추가 지원 릴리스(ESR) 버전을 면밀히 따르고 가능한 한 빨리 다음 ESR로 업데이트합니다. 자세한 내용은 Firefox ESR 릴리스 일정 및 Firefox 릴리스 정보를 참조하세요.

Amazon Linux 가용성

AWS는 AL2023, AL2 및 Amazon Linux 1(AL1, 이전 Amazon Linux AMI)을 제공합니다. 다른 Linux 배포에서 Amazon Linux로 마이그레이션하는 경우 AL2023으로 마이그레이션하는 것이 좋습니다.

> **Note**
>
> AL1에 대한 표준 지원은 2020년 12월 31일에 종료되었습니다. AL1 유지 관리 지원 단계는 2023년 12월 31일에 종료되었습니다. AL1 EOL 및 유지 보수 지원에 대한 자세한 내용은 블로그 게시물 Amazon Linux AMI 종료 업데이트를 참조하세요.

Amazon Linux에 대한 자세한 내용은 AL2023, AL2 및 AL1을 참조하세요.

Amazon Linux 컨테이너 이미지는 Amazon Elastic Container Registry User Guide의 Amazon Linux container image를 참조하세요.

AL2에서 더 이상 사용되지 않는 기능

다음 섹션에서는 AL2에서 지원되는 기능과 AL2023에는 없는 기능에 대해 설명합니다. 이는 AL2에는 있지만 AL2023에는 없는 기능 및 패키지와 같은 기능이며 AL2023에는 추가되지 않습니다. AL2에서이 기능이 지원되는 기간은 AL2 설명서를 참조하세요.

`compat-` 패키지

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

Amazon Linux 릴리스(예: AL2)의 모든 `compat-` 패키지는 중단되며 후속 버전(예: AL2023)에는 존재하지 않습니다. 업데이트된 라이브러리 버전에 맞게 소프트웨어를 다시 빌드하는 것이 좋습니다.

AL1에서 더 이상 사용되지 않는 기능이 중단되고 AL2에서 제거됨

이 섹션에서는 AL1에서 사용할 수 있고 AL2에서 더 이상 사용할 수 없는 기능에 대해 설명합니다.

> **Note**
>
> AL1의 유지 관리 지원 단계의 일환으로, 일부 패키지 지원 종료(EOL) 날짜가 AL1 EOL 이전으로 되었습니다. 자세한 내용은 AL1 패키지 지원 설명서 섹션을 참조하세요.

> **Note**
>
> 일부 AL1 기능은 이전 릴리스에서 중단되었습니다. 자세한 내용은 AL1 릴리스 노트를 참조하세요.

주제

- 32비트 x86(i686) AMI
- aws-apitools-* 로 대체됨AWS CLI
- systemd가 AL2에서 upstart를 대체

32비트 x86(i686) AMI

AL1 2014.09 릴리스의 일환으로, Amazon Linux는 32비트 AMI를 생성하는 마지막 릴리스가 될 것이라고 발표했습니다. 따라서 AL1 2015.03 릴리스부터 Amazon Linux는 더 이상 32비트 모드 실행을 지원하지 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 포함되지 않습니다. AL2023으로 마이그레이션하기 전에 64비트 코드로 전환을 완료하는 것이 좋습니다.

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

`aws-apitools-*` 로 대체됨AWS CLI

2013년 9월 AWS CLI가 릴리스되기 전에AWS는 사용자가 Amazon EC2 API를 호출할 수 있도록에서 구현Java된 일련의 명령줄 유틸리티를 사용할 수 있게 되었습니다. 이러한 도구는 2015년에 중단되었으며 명령줄에서 Amazon EC2 APIs와 상호 작용하는 선호되는 방법이 AWS CLI되었습니다. 명령줄 유틸리티 세트에는 다음 `aws-apitools-*` 패키지가 포함됩니다.

- `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 부족하더라도 Amazon Linux는 사용자에게 이전 버전과의 호환성을 제공하기 위해 `aws-apitools-ec2` 같은 명령줄 유틸리티 중 일부를 계속 제공했습니다. AWS CLI는 패키지가 능동적으로 유지 관리되므로 `aws-apitools-*` 패키지보다 더 강력하고 완전한 도구이며 모든 AWSAPIs를 사용할 수 있는 수단을 제공합니다.

`aws-apitools-*` 패키지는 2017년 3월에 더 이상 사용되지 않으며 추가 업데이트를 받지 않습니다. 이러한 패키지의 모든 사용자는 가능한 AWS CLI한 빨리 로 마이그레이션해야 합니다. 이러한 패키지는 AL2023에 없습니다.

또한 AL1은 AL1에서 더 이상 사용되지 않으며 AL2부터 Amazon Linux에 없는 `aws-apitools-iam` 및 `aws-apitools-rds` 패키지를 제공했습니다.

`systemd`가 AL2에서 `upstart`를 대체

AL2는 `systemd` init 시스템을 사용하는 첫 번째 Amazon Linux 릴리스로, AL1에서 `upstart`를 대체했습니다. `upstart` 특정 구성은 AL1에서 최신 버전의 Amazon Linux로 마이그레이션하는 과정에서 변경해야 합니다. AL1에서는 `systemd`를 사용할 수 없으므로 `upstart`에서 `systemd`로 이동하는 것은 AL2 또는 AL2023과 같은 최신 Amazon Linux 메이저 버전으로 이동하는 과정에서만 수행할 수 있습니다.

AL2에서는 기능이 더 이상 사용되지 않고 AL2023에서는 제거됨

이 섹션에서는 AL2에서 사용할 수 있고 AL2023에서는 더 이상 사용할 수 없는 기능에 대해 설명합니다.

주제

- 32비트 x86(i686) 패키지
- aws-apitools-* 로 대체됨 AWS CLI
- 통합 Amazon CloudWatch Logs 에이전트를 위해 awslogs 사용 중지
- bzr 개정 제어 시스템
- cgroup v1
- log4j 핫패치(log4j-cve-2021-44228-hotpatch)
- lsb_release 및 system-lsb-core 패키지
- mcrypt
- OpenJDK 7(java-1.7.0-openjdk)
- Python 2.7
- rsyslog-openssl를 대체합니다. rsyslog-gnutls
- 네트워크 정보 서비스(NIS) / yp
- Amazon VPC create-dhcp-options의 여러 도메인 이름
- Sun RPC의 glibc
- audit 로그의 OpenSSH 키 지문
- ld.gold 링커
- ping6

- ftp 패키지

32비트 x86(i686) 패키지

AL1 2014.09 릴리스의 일환으로, 32비트 AMI를 생성하는 마지막 릴리스가 될 것이라고 발표했습니다. 따라서 AL1 2015.03 릴리스부터 Amazon Linux는 더 이상 32비트 모드 실행을 지원하지 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 포함되지 않습니다. 고객은 64비트 코드로 전환하는 것이 좋습니다.

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

`aws-apitools-*` 로 대체됨 AWS CLI

2013 AWS CLI 년 9월의 릴리스 이전에는에서 구현Java된 일련의 명령줄 유틸리티를 사용할 수 있게 AWS 하여 고객이 Amazon EC2 API를 호출할 수 있게 했습니다. 이러한 도구는 2015년에 더 이상 사용되지 않아 명령줄에서 Amazon EC2 APIs와 상호 작용하는 선호되는 방법이 AWS CLI 되었습니다. 여기에는 다음 `aws-apitools-*` 패키지가 포함됩니다.

- `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 부족하더라도 Amazon Linux는 고객에게 이전 버전과의 호환성을 제공하기 위해 `aws-apitools-ec2` 같은 명령줄 유틸리티 중 일부를 계속 제공했습니다. AWS CLI 는 패키지가 능동적으로 유지 관리되므로 `aws-apitools-*` 패키지보다 더 강력하고 완전한 도구이며 모든 AWS APIs를 사용할 수 있는 수단을 제공합니다.

`aws-apitools-*` 패키지는 2017년 3월에 더 이상 사용되지 않으며 추가 업데이트를 받지 않습니다. 이러한 패키지의 모든 사용자는 가능한 AWS CLI 한 빨리 로 마이그레이션해야 합니다. 이러한 패키지는 AL2023에 없습니다.

통합 Amazon CloudWatch Logs 에이전트를 위해 **awslogs** 사용 중지

awslogs 패키지는 AL2에서 더 이상 사용되지 않으며 AL2023에는 더 이상 존재하지 않습니다. amazon-cloudwatch-agent 패키지에서 사용할 수 있는 통합 CloudWatch Logs 에이전트로 대체됩니다. 자세한 내용은 Amazon CloudWatch Logs 사용자 안내서를 참조하세요.

bzr 개정 제어 시스템

GNU Bazaar(bzr) 개정 제어 시스템은 AL2에서 중단되고 AL2023에는 더 이상 존재하지 않습니다.

bzr의 사용자는 리포지토리를 git로 마이그레이션하는 것이 좋습니다.

cgroup v1

AL2023은 통합 제어 그룹 계층 구조(cgroup v2)로 이동하는 반면, AL2는 cgroup v1을 사용합니다. AL2는 cgroup v2를 지원하지 않으므로 AL2023으로 이전하는 과정에서 이 마이그레이션을 완료해야 합니다.

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

> **Note**
>
> log4j-cve-2021-44228-hotpatch 패키지는 AL2에서는 더 이상 사용되지 않으며 AL2023에서는 제거됩니다.

CVE-2021-44228을 위해 AL1 및 AL2용 Apache Log4j용 핫패치 RPM 패키지 버전을 출시했습니다. Amazon Linux에 핫패치를 추가한다고 발표하면서 "핫패치 설치가 CVE-2021-44228 또는 CVE-2021-45046 완화 기능을 갖춘 log4j 버전 업데이트를 대체하는 것이 아니"라고 언급했습니다.

이 핫패치는 log4j 패치 시간을 벌기 위한 조치였습니다. CVE-2021-44228에 대응한 지 15개월 후에 첫 AL2023 일반 공급(GA)이 이루어졌기 때문에 AL2023에 (활성화 여부와 관계없이) 핫패치를 포함되어 있지 않습니다.

Amazon Linux에서 자체 log4j 버전을 실행한다면 CVE-2021-44228 또는 CVE-2021-45046의 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 설명서를 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 연결된 PHP 확장은 AL2에서 더 이상 사용되지 않으며 AL2023에는 더 이상 존재하지 않습니다.

업스트림 PHP는 2016년 12월에 처음 릴리스된 PHP 7.1의 `mcrypt` 확장을 더 이상 사용하지 않고 2019년 10월에 최종 릴리스되었습니다.

업스트림 `mcrypt` 라이브러리는 2007년에 마지막으로 릴리스되었으며 SourceForge가 2017년에 새 커밋에 필요한 `cvs` 개정 제어에서 마이그레이션하지 않았습니다. 가장 최근 커밋(및 3년 전만 해당)은 2011년부터 유지 관리자가 있는 프로젝트에 대한 언급을 제거한 것입니다.

`mcrypt`의 나머지 사용자는 `mcrypt`가 AL2023에 추가되지 않으므로 코드를 OpenSSL로 포팅하는 것이 좋습니다.

OpenJDK 7(`java-1.7.0-openjdk`)

> **Note**
>
> AL2023은 Java 기반 워크로드를 지원하는 다양한 버전의 Amazon Corretto를 제공합니다. OpenJDK 7 패키지는 AL2에서 더 이상 사용되지 않으며 AL2023에는 더 이상 존재하지 않습니다. AL2023에서 사용할 수 있는 가장 오래된 JDK는 Corretto 8에서 제공합니다.

Amazon Linux Java에 대한 자세한 내용은 Java AL2의 섹션을 참조하세요.

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 Linux Python에 대한 자세한 내용은 Python AL2의 섹션을 참조하세요.

rsyslog-openssl를 대체합니다. **rsyslog-gnutls**

`rsyslog-gnutls` 패키지는 AL2에서는 더 이상 사용되지 않으며 AL2023에는 더 이상 존재하지 않습니다. `rsyslog-openssl` 패키지는 패키지 사용에 대한 드롭인 대체 `rsyslog-gnutls` 패키지여야 합니다.

네트워크 정보 서비스(NIS) / **yp**

원래 노란색 페이지 또는 YP라고 하는 네트워크 정보 서비스(NIS)는 AL2에서 더 이상 사용되지 않으며 AL2023에는 더 이상 존재하지 않습니다. 여기에는 `ypbind`, `ypserv` 및 `yp-tools` 패키지가 포함됩니다. NIS와 통합되는 다른 패키지는 AL2023에서 이 기능이 제거되었습니다.

Amazon VPC **create-dhcp-options**의 여러 도메인 이름

Amazon Linux 2에서는 `domain-name` 파라미터의 여러 도메인 이름을 `create-dhcp-options`에 전달하여 `/etc/resolv.conf`에 `search foo.example.com bar.example.com` 같은 항목이 포함될 수 있습니다. Amazon VPC DHCP 서버는 단일 도메인 이름만 지원하는 DHCP 옵션 15를 사용하여 제공된 도메인 이름 목록을 전송합니다(RFC 2132 섹션 3.17 참조). AL2023은 RFC를 따르는 네트워크 구성에 `systemd-networkd`를 사용하므로 AL2의 이 우발적 기능은 AL2023에 없습니다.

AWS CLI 및 Amazon VPC 설명서에는 다음의 문구가 있습니다. "일부 Linux 운영 체제는 공백으로 구분된 여러 도메인 이름을 허용합니다. 하지만 Windows와 기타 Linux 운영 체제에서는 이 값을 단일 도메인으로 취급하므로 예기치 않은 동작이 발생합니다. DHCP 옵션 세트가 값을 단일 도메인으로 취급하는 운영 체제가 실행되는 인스턴스가 있는 VPC와 연결되어 있는 경우 도메인 이름을 하나만 지정합니다."

AL2023과 같은 이러한 시스템에서 DHCP 옵션 15(하나만 허용)를 사용하여 두 개의 도메인을 지정하면 도메인 이름에 공백 문자가 유효하지 않으므로 공백 문자가 `032`로 인코딩되어 `/etc/resolv.conf`가 `search foo.exmple.com032bar.example.com`을 포함하게 됩니다.

여러 도메인 이름을 지원하려면 DHCP 서버가 DHCP 옵션 119를 사용해야 합니다(RFC 3397, 섹션 2 참조). Amazon VPC DHCP 서버에서 지원되는 경우 Amazon VPC 사용 설명서를 참조하세요.

Sun RPC의 `glibc`

Sun RPC에서의 `glibc` 구현은 AL2에서는 더 이상 사용되지 않으며 AL2023에서는 제거됩니다. Sun RPC 기능이 필요한 경우 `libtirpc` 라이브러리(AL2 및 AL2023에서 사용 가능)를 사용하여 이동하는 것이 좋습니다. 또한 `libtirpc`를 채택하면 애플리케이션이 IPv6을 지원할 수 있습니다.

이 변경 사항은 Fedora의 `glibc`에서 Sun RPC 인터페이스 제거 및 Gentoo의 유사한 변경 등 이러한 기능을 제거하는 `glibc` 업스트림의 광범위한 커뮤니티 채택을 반영합니다.

`audit` 로그의 OpenSSH 키 지문

AL2 수명 주기 후반부에 패치가 OpenSSH 패키지에 추가되어 인증에 사용되는 키 지문을 내보냈습니다. 이 기능은 AL2023에는 없습니다.

`ld.gold` 링커

`ld.gold` 링커는 AL2에서 사용할 수 있으며 AL2023에서 제거됩니다. gold 링커를 명시적으로 참조하는 소프트웨어를 빌드하는 고객은 일반 (`ld.bfd`) 링커로 마이그레이션해야 합니다.

업스트림 GNU Binutils 버전 2.44(2025년 2월 릴리스)의 릴리스 노트에는 다음과 같은 `ld.gold`에 대한 제거 사항이 명시되어 있습니다: "이전 관행과 달리, 이번 릴리스에서 binutils-2.44.tar tarball에는 gold 링커의 소스가 포함되어 있지 않습니다. 이는 이제 골드 링커가 더 이상 사용되지 않으며, 지원자가 앞으로 나아가서 개발 및 유지 관리를 계속할 것을 제안하지 않는 한 결국 제거되기 때문입니다."

ping6

AL2023에서는 일반 `ping` 유틸리티가 기본적으로 IPv6를 지원하므로 별도의 `/bin/ping6`이 더 이상 필요하지 않습니다. AL2023에서 `/usr/sbin/ping6`은 `/usr/bin/ping` 실행 파일에 대한 symlink입니다.

이 변경은 Fedora의 Ping IPv6 변경 등 이러한 기능을 제공하는 더 광범위한 커뮤니티의 최신 `iputils` 버전 채택을 따릅니다.

ftp 패키지

AL2의 ftp 패키지는 AL2023부터 Amazon Linux에서 더 이상 사용할 수 없습니다. 이 결정은 보안, 유지 관리 가능성 및 최신 소프트웨어 개발 관행에 대한 지속적인 노력의 일환으로 이루어졌습니다. AL2023 마이그레이션의 일환으로(또는 마이그레이션 이전에) 레거시 ftp 패키지 사용을 대체 패키지 중 하나로 마이그레이션하는 것이 좋습니다.

배경

레거시 ftp 패키지는 수년 동안 업스트림에서 적극적으로 유지되지 않았습니다. 소스 코드에 대한 마지막 중요 업데이트는 2000년대 초에 이뤄졌으며 원래 소스 리포지토리를 더 이상 사용할 수 없습니다. 일부 Linux 배포판에는 보안 취약성에 대한 패치가 포함되어 있지만 코드베이스는 대부분 유지 관리되지 않습니다.

권장 대안

AL2023은 FTP 기능에 대한 몇 가지 현대적이고 적극적으로 유지 관리되는 대안을 제공합니다.

lftp(AL2 및 AL2023에서 사용 가능)

FTP, HTTP, SFTP 및 기타 프로토콜을 지원하는 정교한 파일 전송 프로그램입니다. 기존 ftp 클라이언트보다 더 많은 기능을 제공하며 적극적으로 유지 관리됩니다.

다음을 사용하여 설치: dnf install lftp

curl(AL2 및 AL2023에서 사용 가능)

URL, FTPS, HTTP, HTTPS 및 기타 여러 프로토콜을 사용하여 데이터를 전송하기 위한 다목적 명령줄 도구입니다.

curl-minimal 패키지를 통해 AL2023에서 기본적으로 사용할 수 있습니다. 보다 광범위한 프로토콜 지원을 위해 선택적으로 dnf swap curl-minimal curl-full을 사용하여 curl-full로 업그레이드할 수 있습니다.

wget(AL2 및 AL2023에서 사용 가능)

웹에서 파일을 다운로드하고 HTTP, HTTPS 및 FTP 프로토콜을 지원하는 비대화형 명령줄 유틸리티입니다.

다음을 사용하여 설치: dnf install wget(일부 AL2023 이미지에는 기본적으로 설치되지 않음)

sftp(AL2 및 AL2023에서 사용 가능)

SSH를 통해 작동하는 보안 파일 전송 프로토콜로, 암호화된 파일 전송을 제공합니다.

기본적으로 OpenSSH 패키지의 일부로 사용할 수 있습니다.

마이그레이션 고려 사항

애플리케이션 또는 스크립트가 레거시 `ftp` 클라이언트에 종속되는 경우 다음 마이그레이션 접근 방식을 고려하세요.

1. 최신 대안을 사용하도록 스크립트 업데이트: 레거시 `ftp` 클라이언트 대신 `lftp`, `curl`, `wget` 또는 `sftp`를 사용하도록 스크립트를 수정합니다.
2. 패키지 종속성 검토: 일부 애플리케이션은 내부적으로 최신 프로토콜을 사용하여 `ftp`로 마이그레이션한 지 오래 되었지만 패키지 메타데이터에 패키지를 종속성으로 나열할 수 있습니다. 이러한 경우 `ftp` 패키지에 `/usr/bin/ftp`가 없더라도 애플리케이션이 AL2023에서 올바르게 작동할 수 있습니다. 명시된 종속성에만 의존하지 않고 애플리케이션의 실제 요구 사항을 검토합니다.
3. 애플리케이션 종속성 업데이트: `ftp` 패키지에 대한 종속성을 여전히 선언하지만 실제로 사용하지 않는 애플리케이션을 유지 관리하는 경우 패키지 메타데이터를 업데이트하여 불필요한 종속성을 제거합니다.

보안 고려 사항

FTP 프로토콜은 인증 자격 증명을 포함한 데이터를 일반 텍스트로 전송합니다. 보안에 민감한 애플리케이션의 경우 권장 대체 도구에서 지원하는 SFTP 또는 HTTPS와 같은 암호화된 대안을 사용하는 것이 좋습니다.

AL2023으로 마이그레이션 준비

AL2를 계속 사용하는 동안 AL2023으로의 전환을 준비할 수 있습니다. AL2

주제

- AL2023의 변경 사항 목록 검토
- cron 작업에서 systemd 타이머로 마이그레이션

AL2023의 변경 사항 목록 검토

AL2023 설명서에는 AL2 이후 구현된 자세한 변경 사항 목록이 포함되어 있습니다. 이 정보는 AL2 및 AL2023 비교 섹션에 있습니다. 또한 AL2023의 패키지 변경 섹션에 소프트웨어 패키지 변경 사항의 포괄적인 목록이 있습니다.

AL2023에는가 포함되지 않습니다`amazon-linux-extras`. 대신 여러 버전이 제공되는 네임스페이스 패키지를 제공합니다. 많은 패키지가 AL2023에서 업데이트되므로 AL2023의 기본 버전은에서 가져오는 버전보다 늦을 수 있습니다`amazon-linux-extras`.

> **Note**
>
> EOL`amazon-linux-extras`이므로를 실행하지 않는 것이 좋습니다.

설명서에서 이러한 섹션을 검토한 후 마이그레이션에 맞게 환경을 조정해야 할 수 있는 AL2023의 변경 사항이 있는지 확인할 수 있습니다. 예를 들어 마지막으로 Python 2.7 스크립트를 Python 3으로 마이그레이션해야 할 수 있습니다.

`cron` 작업에서 `systemd` 타이머로 마이그레이션

기본적으로 `cron`는 AL2023에 설치되지 않습니다. AL2023으로 마이그레이션하기 위해 `cron` 작업을 AL2의 `systemd` 타이머로 마이그레이션할 수 있습니다. `systemd`에는 타이머 실행 시기를 보다 정확하게 제어하고 로깅을 개선하는 등 많은 이점이 있습니다. AL2023

AL2 제한 사항

다음 주제에서는 AL2의 다양한 제한 사항과 최신 버전의 Amazon Linux에서 해결된 경우에 대해 설명합니다.

주제

yum GPG 하위 키로 만든 GPG 서명을 확인할 수 없음

AL2의 `rpm` 패키지 관리자 버전은 GPG 하위 키로 만든 패키지 서명 확인에 대한 지원이 `rpm` 추가되기 전의 버전입니다. AL2와 호환되는 패키지를 생성하는 경우 AL2의 `rpm` 일부인와 호환되는 GPG 서명 키를 사용해야 합니다.

기존 사용자의 이전 버전과의 호환성을 보장하기 위해 AL2의 버전`rpm`은 보안 백포트만 수신합니다.

`rpm` AL2023의 버전에는 GPG 하위 키로 만든 패키지 서명을 확인하기 위한 지원이 포함되어 있습니다.

AL1과 AL2 비교

다음 주제에서는 AL1과 AL2의 주요 차이점을 설명합니다. 또한 수명 및 지원, 패키지 변경에 대한 정보도 포함되어 있습니다.

주제

AL1 지원 및 EOL

AL1은 이제 EOL입니다. AL1은 2020년 12월 31일부터 표준 지원을 종료했으며 2023년 12월 31일까지 유지 관리 지원 단계에 있었습니다.

최신 Amazon Linux 버전으로 업그레이드하는 것이 좋습니다.

AWSGraviton 프로세서 지원

AL2는 Graviton 프로세서에 대한 지원을 도입했습니다. AL2023은 Graviton 프로세서에 추가로 최적화되어 있습니다.

`systemd`는 `upstart`를 `init` 시스템으로 대체합니다.

AL2에서 `upstart`를 `init` 시스템으로 `systemd` 대체합니다.

Python 2.6 및 2.7이 Python 3으로 대체됨

AL1은 2018.03 릴리스에서 Python 2.6을 EOL로 표시했지만 패키지는 여전히 설치할 리포지토리에 있었습니다. AL2는 지원되는 가장 빠른 Python 버전으로 Python 2.7과 함께 제공됩니다.

AL2023은 Python 3으로의 전환을 완료하며 Python 2.x 버전은 리포지토리에 포함되지 않습니다.

AL1 및 AL2 AMIs에 설치된 패키지 비교

패키지	AL1 AMI	AL2 AMI
GeoIP		1.5.0
PyYAML		3.10
acl	2.2.49	2.2.51
acpid	2.0.19	2.0.19
alsa-lib	1.0.22	
amazon-linux-extras		2.0.3
amazon-linux-extras-yum-plugin		2.0.3
amazon-ssm-agent.	3.2.1705.0	3.2.1705.0
at	3.1.10	3.1.13
attr	2.4.46	2.4.46
감사	2.6.5	2.8.1
audit-libs	2.6.5	2.8.1
authconfig	6.2.8	6.2.8
aws-amitools-ec2	1.5.13	
aws-cfn-bootstrap	1.4	2.0
aws-cli	1.18.107	
awscli		1.18.147

패키지	AL1 AMI	AL2 AMI
basesystem	10.0	10.0
bash	4.2.46	4.2.46
bash-completion		2.1
BC	1.06.95	1.06.95
bind-export-libs		9.11.4
bind-libs	9.8.2	9.11.4
bind-libs-lite		9.11.4
bind-license		9.11.4
bind-utils	9.8.2	9.11.4
binutils	2.27	2.29.1
blktrace		1.0.5
boost-date-time		1.53.0
boost-system		1.53.0
boost-thread		1.53.0
bridge-utils		1.5
bzip2	1.0.6	1.0.6
bzip2-libs	1.0.6	1.0.6
ca-certificates	2023년 2월 62일	2023년 2월 62일
checkpolicy	2.1.10	
chkconfig	1.3.49.3	1.7.4

패키지	AL1 AMI	AL2 AMI
chrony		4.2
cloud-disk-utils	0.27	
cloud-init	0.7.6	19.3
cloud-utils-growpart		0.31
copy-jdk-configs	3.3	
coreutils	8.22	8.22
CPIO	2.10	2.12
cracklib	2.8.16	2.9.0
cracklib-dicts	2.8.16	2.9.0
cronie	1.4.4	1.4.11
cronie-anacron	1.4.4	1.4.11
crontabs	1.10	1.11
cryptsetup	1.6.7	1.7.4
cryptsetup-libs	1.6.7	1.7.4
curl	7.61.1	8.3.0
cyrus-sasl	2.1.23	
cyrus-sasl-lib	2.1.23	2.1.26
cyrus-sasl-plain	2.1.23	2.1.26
DASH	0.5.5.1	
db4	4.7.25	

패키지	AL1 AMI	AL2 AMI
db4-utils	4.7.25	
dbus	1.6.12	1.10.24
dbus-libs	1.6.12	1.10.24
dejavu-fonts-common	2.33	
dejavu-sans-fonts	2.33	
dejavu-serif-fonts	2.33	
device-mapper	1.02.135	1.02.170
device-mapper-event	1.02.135	1.02.170
device-mapper-event-libs	1.02.135	1.02.170
device-mapper-libs	1.02.135	1.02.170
device-mapper-persistent-data	0.6.3	0.7.3
dhclient	4.1.1	4.2.5
dhcp-common	4.1.1	4.2.5
dhcp-libs		4.2.5
diffutils	3.3	3.3
dmidecode		3.2
dmraid	1.0.0.rc16	1.0.0.rc16
dmraid-events	1.0.0.rc16	1.0.0.rc16
dosfstools		3.0.20
dracut	004	033

패키지	AL1 AMI	AL2 AMI
dracut-config-ec2		2.0
dracut-config-generic		033
dracut-modules-growroot	0.20	
DUMP	0.4	
dyninst		9.3.1
e2fsprogs	1.43.5	1.42.9
e2fsprogs-libs	1.43.5	1.42.9
ec2-hibinit-agent	1.0.0	1.0.2
ec2-instance-connect		1.1
ec2-instance-connect-selinux		1.1
ec2-net-utils	0.7	1.7.3
ec2-utils	0.7	1.2
ed	1.1	1.9
elfutils-default-yama-scope		0.176
elfutils-libelf	0.168	0.176
elfutils-libs		0.176
Espel Release	6	
ethtool	3.15	4.8
expat	2.1.0	2.1.0
파일	5.37	5.11

패키지	AL1 AMI	AL2 AMI
file-libs	5.37	5.11
filesystem	2.4.30	3.2
findutils	4.4.2	4.5.11
fipscheck	1.3.1	1.4.1
fipscheck-lib	1.3.1	1.4.1
fontconfig	2.8.0	
fontpackages-filesystem	1.41	
freetype	2.3.11	2.8
fuse-libs	2.9.4	2.9.2
gawk	3.1.7	4.0.2
gdbm	1.8.0	1.13
gdisk	0.8.10	0.8.10
generic-logos	17.0.0	18.0.0
get_reference_source	1.2	
gettext		0.19.8.1
gettext-libs		0.19.8.1
giflib	4.1.6	
glib2	2.36.3	2.56.1
glibc	2.17	2.26
glibc-all-langpacks		2.26

패키지	AL1 AMI	AL2 AMI
glibc-common	2.17	2.26
glibc-locale-source		2.26
glibc-minimal-langpack		2.26
gmp	6.0.0	6.0.0
gnupg2	2.0.28	2.0.22
gpgme	1.4.3	1.3.2
gpm-libs	1.20.6	1.20.7
grep	2.20	2.20
groff	1.22.2	
groff-base	1.22.2	1.22.2
grub	0.97	
grub2		2.06
grub2-common		2.06
grub2-efi-x64-ec2		2.06
grub2-pc		2.06
grub2-pc-modules		2.06
grub2-tools		2.06
grub2-tools-minimal		2.06
grubby	7.0.15	8.28
gssproxy		0.7.0

패키지	AL1 AMI	AL2 AMI
gzip	1.5	1.5
hardlink		1.3
hesiod	3.1.0	
hibagent	1.0.0	1.1.0
hmaccalc	0.9.12	
hostname		3.13
hunspell		1.3.2
hunspell-en		0.20121024
hunspell-en-GB		0.20121024
hunspell-en-US		0.20121024
hwdata	0.233	0.252
info	5.1	5.1
initscripts	9.03.58	9.49.47
iproute	4.4.0	5.10.0
iptables	1.4.21	1.8.4
iptables-libs		1.8.4
iputils	20121221	20180629
irqbalance	1.5.0	1.7.0
jansson		2.10
java-1.7.0-openjdk	1.7.0.321	

패키지	AL1 AMI	AL2 AMI
javapackages-tools	0.9.1	
jbigkit-libs		2.0
jpackage-utils	1.7.5	
json-c		0.11
kbd	1.15	1.15.5
kbd-Legacy		1.15.5
kbd-Misc	1.15	1.15.5
kernel	4.14.326	5.10.199
kernel-tools	4.14.326	5.10.199
keyutils	1.5.8	1.5.8
keyutils-libs	1.5.8	1.5.8
kmod	14	25
kmod-libs	14	25
kpartx	0.4.9	0.4.9
kpatch-runtime		0.9.4
krb5-libs	1.15.1	1.15.1
langtable		0.0.31
langtable-data		0.0.31
langtable-python		0.0.31
lcms2	2.6	

패키지	AL1 AMI	AL2 AMI
less	436	458
libICE	1.0.6	
libSM	1.2.1	
libX11	1.6.0	
libX11-Common	1.6.0	
libXau	1.0.6	
libXcomposite	0.4.3	
libXext	1.3.2	
libXfont	1.4.5	
libXi	1.7.2	
libXrender	0.9.8	
libXtst	1.2.2	
libacl	2.2.49	2.2.51
libaio	0.3.109	0.3.109
libassuan	2.0.3	2.1.0
libattr	2.4.46	2.4.46
libbasicobjects		0.1.1
libblkid	2.23.2	2.30.2
libcap	2.16	2.54
libcap-ng	0.7.5	0.7.5

패키지	AL1 AMI	AL2 AMI
libcap54	2.54	
libcgroup	0.40.rc1	
libcollection		0.7.0
libcom_err	1.43.5	1.42.9
libconfig		1.4.9
libcroco		0.6.12
libcrypt		2.26
libcurl	7.61.1	8.3.0
libdaemon		0.14
libdb		5.3.21
libdb-utils		5.3.21
libdrm		2.4.97
libdwarf		20130207
libedit	2.11	3.0
libestr		0.1.9
libevent	2.0.21	2.0.21
libfastjson		0.99.4
libfdisk		2.30.2
libffi	3.0.13	3.0.13
libfontenc	1.0.5	

패키지	AL1 AMI	AL2 AMI
libgcc		7.3.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.5.3
libgomp		7.3.1
libgpg-error	1.11	1.12
libgssglue	0.1	
libicu	50.2	50.2
libidn	1.18	1.28
libidn2	2.3.0	2.3.0
libini_config		1.3.1
libjpeg-turbo	1.2.90	2.0.90
libmetalink		0.1.3
libmnl	1.0.3	1.0.3
libmount	2.23.2	2.30.2
libnetfilter_conntrack	1.0.4	1.0.6
libnfnetlink	1.0.1	1.0.1
libnfsidmap	0.25	0.25
libnghttp2	1.33.0	1.41.0
libnih	1.0.1	
libnl	1.1.4	

패키지	AL1 AMI	AL2 AMI
libnl3		3.2.28
libnl3-cli		3.2.28
libpath_utils		0.2.1
libpcap		1.5.3
libpciaccess		0.14
libpipeline	1.2.3	1.2.3
libpng	1.2.49	1.5.13
libpsl	0.6.2	
libpwquality	1.2.3	1.2.3
libref_array		0.1.5
libseccomp		2.4.1
libselinux	2.1.10	2.5
libselinux-utils	2.1.10	2.5
libsemanage	2.1.6	2.5
libsepol	2.1.7	2.5
libsmartcols	2.23.2	2.30.2
libss	1.43.5	1.42.9
libssh2	1.4.2	1.4.3
libsss_idmap		1.16.5
libsss_nss_idmap		1.16.5

패키지	AL1 AMI	AL2 AMI
libstdc++		7.3.1
libstdc++72	7.2.1	
libstoragemgmt		1.6.1
libstoragemgmt-python		1.6.1
libstoragemgmt-python-clibs		1.6.1
libsysfs	2.1.0	2.1.0
libtasn1	2.3	4.10
libteam		1.27
libtiff		4.0.3
libirpc	0.2.4	0.2.4
libudev	173	
libunistring	0.9.3	0.9.3
libuser	0.60	0.60
libutempter	1.1.5	1.1.6
libuuid	2.23.2	2.30.2
libverto	0.2.5	0.2.5
libverto-libevent		0.2.5
libwebp		0.3.0
libxcb	1.11	
libxml2	2.9.1	2.9.1

패키지	AL1 AMI	AL2 AMI
libxml2-python		2.9.1
libxml2-python27	2.9.1	
libxslt	1.1.28	
libyaml	0.1.6	0.1.4
lm_sensors-libs		3.4.0
log4j-cve-2021-44228-hotpatch	1.3	
logrotate	3.7.8	3.8.6
lsof	4.82	4.87
lua	5.1.4	5.1.4
lvm2	2.02.166	2.02.187
lvm2-libs	2.02.166	2.02.187
lz4		1.7.5
mailcap	2.1.31	
make	3.82	3.82
man-db	2.6.3	2.6.3
man-pages	4.10	3.53
man-pages-overrides		7.5.2
mariadb-libs		5.5.68
mdadm	3.2.6	4.0
microcode_ctl	2.1	2.1

패키지	AL1 AMI	AL2 AMI
mingetty	1.08	
mlocate		0.26
mtr		0.92
nano	2.5.3	2.9.8
NC	1.84	
ncurses	5.7	6.0
ncurses-base	5.7	6.0
ncurses-libs	5.7	6.0
net-tools	1.60	2.0
nettle		2.7.1
newt	0.52.11	0.52.15
newt-python		0.52.15
newt-python27	0.52.11	
nfs-utils	1.3.0	1.3.0
nspr	4.25.0	4.35.0
nss	3.53.1	3.90.0
nss-pem	1.0.3	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

패키지	AL1 AMI	AL2 AMI
nss-tools	3.53.1	3.90.0
nss-util	3.53.1	3.90.0
NTP	4.2.8p15	
ntpdate	4.2.8p15	
ntsysv	1.3.49.3	1.7.4
numactl	2.0.7	
numactl-libs		2.0.9
openldap	2.4.40	2.4.44
openssh	7.4p1	7.4p1
openssh-clients	7.4p1	7.4p1
openssh-server	7.4p1	7.4p1
openssl	1.0.2k	1.0.2k
openssl-libs		1.0.2k
os-prober		1.58
p11-kit	0.18.5	0.23.22
p11-kit-trust	0.18.5	0.23.22
pam	1.1.8	1.1.8
pam_ccreds	10	
pam_krb5	2.3.11	
pam_passwdqc	1.0.5	

패키지	AL1 AMI	AL2 AMI
parted	2.1	3.1
passwd	0.79	0.79
pciutils	3.1.10	3.5.1
pciutils-libs	3.1.10	3.5.1
pcre	8.21	8.32
pcre2		10.23
perl	5.16.3	5.16.3
perl-Carp	1.26	1.26
Perl-Digest	1.17	
perl-Digest-HMAC	1.03	
perl-Digest-MD5	2.52	
perl-Digest-SHA	5.85	
perl-Encode	2.51	2.51
Perl-Exporter	5.68	5.68
perl-File-Path	2.09	2.09
perl-File-Temp	0.23.01	0.23.01
perl-Filter	1.49	1.49
perl-Getopt-Long	2.40	2.40
perl-HTTP-Tiny	0.033	0.033
perl-PathTools	3.40	3.40

패키지	AL1 AMI	AL2 AMI
perl-Pod-Escapes	1.04	1.04
perl-Pod-Perldoc	3.20	3.20
perl-Pod-Simple	3.28	3.28
perl-Pod-Usage	1.63	1.63
perl-Scalar-List-Utils	1.27	1.27
perl-Socket	2.010	2.010
perl-Storable	2.45	2.45
perl-Text-ParseWords	3.29	3.29
perl-Time-HiRes	1.9725	1.9725
perl-Time-Local	1.2300	1.2300
perl-constant	1.27	1.27
perl-libs	5.16.3	5.16.3
perl-macros	5.16.3	5.16.3
perl-parent	0.225	0.225
perl-podlators	2.5.1	2.5.1
perl-threads	1.87	1.87
perl-threads-shared	1.43	1.43
pinentry	0.7.6	0.8.1
pkgconfig	0.27.1	0.27.1
plymouth		0.8.9

패키지	AL1 AMI	AL2 AMI
plymouth-core-libs		0.8.9
plymouth-scripts		0.8.9
pm-utils	1.4.1	1.4.1
policycoreutils	2.1.12	2.5
popt	1.13	1.13
PostFix		2.10.1
procmail	3.22	
procps	3.2.8	
procps-ng		3.3.10
psacct	6.3.2	6.6.1
psmisc	22.20	22.20
pth	2.0.7	2.0.7
pygpgme		0.3
pyliblzma		0.5.3
pystache		0.5.3
python		2.7.18
python-babel		0.9.6
python-backports		1.0
python-backports-ssl_match_hostname		3.5.0.1
python-cffi		1.6.0

패키지	AL1 AMI	AL2 AMI
python-chardet		2.2.1
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
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

패키지	AL1 AMI	AL2 AMI
python-pycurl		7.19.0
python-repoze-lru		0.4
python-requests		2.6.0
python-simplejson		3.2.0
python-urlgrabber		3.10
python-urllib3		1.25.9
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
python2-six		1.11.0

패키지	AL1 AMI	AL2 AMI
python27	2.7.18	
python27-PyYAML	3.10	
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	
python327-colorama	0.4.1	
python27-configobj	4.7.2	
python27-crypto	2.6.1	
python27-daemon	1.5.2	
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	

패키지	AL1 AMI	AL2 AMI
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	
python27-pygpgme	0.3	
python27-pyliblzma	0.5.3	
python27-pystache	0.5.3	
python27-pyxattr	0.5.0	
python27-requests	1.2.3	
python27-rsa	3.4.1	
python27-setuptools	36.2.7	
python27-simplejson	3.6.5	

패키지	AL1 AMI	AL2 AMI
python27-six	1.8.0	
python27-urlgrabber	3.10	
python27-urllib3	1.24.3	
python27-virtualenv	15.1.0	
python3		3.7.16
python3-daemon		2.2.3
python3-docutils		0.14
python3-libs		3.7.16
python3-lockfile		0.11.0
python27-pip		20.2.2
python27-pystache		0.5.4
python3-setuptools		49.1.3
python3-simplejson		3.2.0
pyxattr		0.5.1
qrencode-libs		3.4.1
할당량	4.00	4.01
quota-nls	4.00	4.01
rdate		1.4
readline	6.2	6.2
rmt	0.4	

패키지	AL1 AMI	AL2 AMI
rng-tools	5	6.8
rootfiles	8.1	8.1
rpcbind	0.2.0	0.2.0
RPM	4.11.3	4.11.3
rpm-build-libs	4.11.3	4.11.3
rpm-libs	4.11.3	4.11.3
rpm-plugin-systemd-inhibit		4.11.3
rpm-python27	4.11.3	
rsync	3.0.6	3.1.2
rsyslog	5.8.10	8.24.0
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	
scl-utils		20130529

패키지	AL1 AMI	AL2 AMI
screen	4.0.3	4.1.0
sed	4.2.1	4.2.2
selinux-policy		3.13.1
selinux-policy-targeted		3.13.1
sendmail	8.14.4	
setserial	2.17	2.17
설정	2.8.14	2.8.71
setuptool		1.19.11
sgpio	1.2.0.10	1.2.0.10
shadow-utils	4.1.4.2	4.1.5.1
shared-mime-info	1.1	1.8
slang	2.2.1	2.2.4
sqlite	3.7.17	3.7.17
sssd-client		1.16.5
strace		4.26
sudo	1.8.23	1.8.23
sysctl-defaults	1.0	1.0
sysfsutils	2.1.0	
sysstat		10.1.5
system-release	2018년 3월	2

패키지	AL1 AMI	AL2 AMI
systemd		219
systemd-libs		219
Systed-sysv		219
systemtap-runtime		4.5
sysvinit	2.87	
sysvinit-tools		2.88
tar	1.26	1.26
tcp_wrappers	7.6	7.6
tcp_wrappers-libs	7.6	7.6
tcpdump		4.9.2
tcsh		6.18.01
teamd		1.27
시간	1.7	1.7
tmpwatch	2.9.16	
traceroute	2.0.14	2.0.22
ttmkfdir	3.0.9	
tzdata	2023c	2023c
tzdata-java	2023c	
udev	173	
unzip	6.0	6.0

패키지	AL1 AMI	AL2 AMI
update-motd	1.0.1	1.1.2
upstart	0.6.5	
usermode		1.111
USTR	1.0.4	1.0.4
util-linux	2.23.2	2.30.2
vim-common	9.0.1712	2081년 9월 0일
vim-data	9.0.1712	2081년 9월 0일
vim-enhanced	9.0.1712	2081년 9월 0일
vim-filesystem	9.0.1712	2081년 9월 0일
vim-minimal	9.0.1712	2081년 9월 0일
virt-what		1.18
wget	1.18	1.14
which	2.19	2.20
words	3.0	3.0
xfsdump		3.1.8
xfsprogs		5.0.0
xorg-x11-font-utils	7.2	
xorg-x11-fonts-Type1	7.2	
xxd	9.0.1712	2081년 9월 0일
XZ	5.2.2	5.2.2

패키지	AL1 AMI	AL2 AMI
xz-libs	5.2.2	5.2.2
yajl		2.0.4
yum	3.4.3	3.4.3
yum-langpacks		0.4.2
yum-metadata-parser	1.1.4	1.1.4
yum-plugin-priorities	1.1.31	1.1.31
yum-plugin-upgrade-helper	1.1.31	
Yum-utils	1.1.31	1.1.31
zip	3.0	3.0
ZLIB	1.2.8	1.2.7

AL1 및 AL2 기본 컨테이너 이미지에 설치된 패키지 비교

패키지	AL1 컨테이너	AL2 컨테이너
amazon-linux-extras		2.0.3
basesystem	10.0	10.0
bash	4.2.46	4.2.46
bzip2-libs	1.0.6	1.0.6
ca-certificates	2023년 2월 62일	2023년 2월 62일
chkconfig	1.3.49.3	1.7.4
coreutils	8.22	8.22

패키지	AL1 컨테이너	AL2 컨테이너
CPIO		2.12
curl	7.61.1	8.3.0
cyrus-sasl-lib	2.1.23	2.1.26
db4	4.7.25	
db4-utils	4.7.25	
diffutils		3.3
elfutils-libelf	0.168	0.176
expat	2.1.0	2.1.0
file-libs	5.37	5.11
filesystem	2.4.30	3.2
findutils		4.5.11
gawk	3.1.7	4.0.2
gdbm	1.8.0	1.13
glib2	2.36.3	2.56.1
glibc	2.17	2.26
glibc-common	2.17	2.26
glibc-langpack-en		2.26
glibc-minimal-langpack		2.26
gmp	6.0.0	6.0.0
gnupg2	2.0.28	2.0.22

패키지	AL1 컨테이너	AL2 컨테이너
gpgme	1.4.3	1.3.2
grep	2.20	2.20
gzip	1.5	
info	5.1	5.1
keyutils-libs	1.5.8	1.5.8
krb5-libs	1.15.1	1.15.1
libacl	2.2.49	2.2.51
libassuan	2.0.3	2.1.0
libattr	2.4.46	2.4.46
libblkid		2.30.2
libcap	2.16	2.54
libcom_err	1.43.5	1.42.9
libcrypt		2.26
libcurl	7.61.1	8.3.0
libdb		5.3.21
libdb-utils		5.3.21
libffi	3.0.13	3.0.13
libgcc		7.3.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.5.3

패키지	AL1 컨테이너	AL2 컨테이너
libgpg-error	1.11	1.12
libicu	50.2	
libidn2	2.3.0	2.3.0
libmetalink		0.1.3
libmount		2.30.2
libnghttp2	1.33.0	1.41.0
libpsl	0.6.2	
libselinux	2.1.10	2.5
libsepol	2.1.7	2.5
libssh2	1.4.2	1.4.3
libstdc++		7.3.1
libstdc++72	7.2.1	
libtasn1	2.3	4.10
libunistring	0.9.3	0.9.3
libuuid		2.30.2
libverto	0.2.5	0.2.5
libxml2	2.9.1	2.9.1
libxml2-python27	2.9.1	
lua	5.1.4	5.1.4
make	3.82	

패키지	AL1 컨테이너	AL2 컨테이너
ncurses	5.7	6.0
ncurses-base	5.7	6.0
ncurses-libs	5.7	6.0
nspr	4.25.0	4.35.0
nss	3.53.1	3.90.0
nss-pem	1.0.3	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	3.90.0
nss-util	3.53.1	3.90.0
openldap	2.4.40	2.4.44
openssl	1.0.2k	
openssl-libs		1.0.2k
p11-kit	0.18.5	0.23.22
p11-kit-trust	0.18.5	0.23.22
pcre	8.21	8.32
pinentry	0.7.6	0.8.1
pkgconfig	0.27.1	
popt	1.13	1.13

패키지	AL1 컨테이너	AL2 컨테이너
pth	2.0.7	2.0.7
pygpgme		0.3
pyliblzma		0.5.3
python		2.7.18
python-iniparse		0.4
python-libs		2.7.18
python-pycurl		7.19.0
python-urlgrabber		3.10
python2-rpm		4.11.3
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-pygpgme	0.3	
python27-pyliblzma	0.5.3	
python27-pyxattr	0.5.0	
python27-urlgrabber	3.10	
pyxattr		0.5.1

패키지	AL1 컨테이너	AL2 컨테이너
readline	6.2	6.2
RPM	4.11.3	4.11.3
rpm-build-libs	4.11.3	4.11.3
rpm-libs	4.11.3	4.11.3
rpm-python27	4.11.3	
sed	4.2.1	4.2.2
설정	2.8.14	2.8.71
shared-mime-info	1.1	1.8
sqlite	3.7.17	3.7.17
sysctl-defaults	1.0	
system-release	2018년 3월	2
tar	1.26	
tzdata	2023c	2023c
vim-data		2081년 9월 0일
vim-minimal		2081년 9월 0일
xz-libs	5.2.2	5.2.2
yum	3.4.3	3.4.3
yum-metadata-parser	1.1.4	1.1.4
yum-plugin-ovl	1.1.31	1.1.31
yum-plugin-priorities	1.1.31	1.1.31

패키지	AL1 컨테이너	AL2 컨테이너
Yum-utils	1.1.31	
ZLIB	1.2.8	1.2.7

Amazon EC2의 AL2 Amazon EC2

> **Note**
>
> AL2는 더 이상 Amazon Linux의 최신 버전이 아닙니다. AL2023은 AL2의 후속 버전입니다. 자세한 내용은 AL2023 사용 설명서의 AL2와 AL2023 비교 AL2023 및 AL2023의 패키지 변경 AL2023 사항 목록을 참조하세요.

주제

AL2 AMI를 사용하여 Amazon EC2 인스턴스 시작 AL2

AL2 AMI를 사용하여 Amazon EC2 인스턴스를 시작할 수 있습니다. AL2 자세한 내용은 1단계: 인스턴스 시작을 참조하세요.

Systems Manager를 사용하여 최신 AL2 AMI 찾기

Amazon EC2는 인스턴스를 시작할 때 사용할 수 AWS 있는에서 유지 관리하는 AWS Systems Manager 퍼블릭 AMIs에 대한 퍼블릭 파라미터를 제공합니다. 예를 들어 EC2-provided 파라미터/`aws/service/ami-amazon-linux-latest/amzn2-ami-kernel-default-hvm-x86_64-gp2`는 모든 리전에서 사용할 수 있으며 항상 지정된 리전에서 AL2 AMI의 최신 버전을 가리킵니다.

를 사용하여 최신 AL2023 AMI를 찾으려면 AL2023 시작하기를 AWS Systems Manager참조하세요. AL2023

Amazon EC2 AMI 퍼블릭 파라미터는 다음 경로에서 사용할 수 있습니다.

`/aws/service/ami-amazon-linux-latest`

다음 AWS CLI 명령을 실행하여 현재 AWS 리전의 모든 Amazon Linux AMIs 목록을 볼 수 있습니다.

```
aws ssm get-parameters-by-path --path /aws/service/ami-amazon-linux-latest --query
 "Parameters[].Name"
```

퍼블릭 파라미터를 사용하여 인스턴스를 시작하려면

다음 예제에서는 EC2-provided 퍼블릭 파라미터를 사용하여 최신 AL2 AMI를 사용하여 `m5.xlarge` 인스턴스를 시작합니다.

명령에서 파라미터를 지정하려면 `resolve:ssm:public-parameter` 구문을 사용합니다. 여기서 `resolve:ssm`은 표준 접두사이고 `public-parameter`는 퍼블릭 파라미터의 경로와 이름입니다.

이 예에는 `--count` 및 `--security-group` 파라미터가 포함되어 있지 않습니다. `--count`의 기본값은 1입니다. 기본 VPC와 기본 보안 그룹이 있는 경우 이들이 사용됩니다.

```
aws ec2 run-instances
    --image-id resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-kernel-
default-hvm-x86_64-gp2
    --instance-type m5.xlarge
    --key-name MyKeyPair
```

자세한 내용은 AWS Systems Manager 사용 설명서의 퍼블릭 파라미터 사용을 참조하세요.

Amazon Linux 2 AMI 이름 이해

Amazon Linux 2 AMI 이름은 다음 이름 지정 체계를 사용합니다.

`amzn2-ami-[minimal-][kernel-{5.10,default,4.14}]-hvm-{x86_64,aarch64}-{ebs,gp2}`

- 최소 AMIs 이미지 크기를 줄이기 위해 최소화된 사전 설치된 패키지 세트가 함께 제공됩니다.
- 커널-VERSION은 각 AMI에 사전 설치된 커널 버전을 결정합니다.
  - `kernel-5.10`는 Linux 커널 버전 5.10을 선택합니다. AL2에 권장되는 커널 버전입니다.
  - `kernel-default`는 AL2에 권장되는 기본 커널을 선택합니다. 커널 5.10의 별칭입니다.

  - `kernel-4.14`는 Linux 커널 버전 4.14를 선택합니다. 이는 이전 AMI 릴리스와의 호환성을 위해서만 제공됩니다. 새 인스턴스 시작에는이 버전을 사용하지 마십시오. 이 AMI가 지원되지 않을 것으로 예상합니다.
  - 특정 커널을 참조하지 않고 특별한 AMI 이름 집합이 존재합니다. 이러한 AMIs는 커널 4.14의 별칭입니다. 이러한 AMIs는 이전 AMI 릴리스와의 호환성을 위해서만 제공됩니다. 새 인스턴스 시작에는이 AMI 이름을 사용하지 마십시오. 이러한 AMIs 것으로 예상합니다.
- x86_64/aarch64는 AMI를 실행할 CPU 플랫폼을 결정합니다. Intel 및 AMD 기반 EC2 인스턴스의 경우 x86_64를 선택합니다. EC2 Graviton 인스턴스에 대해 aarch64를 선택합니다.
- ebs/gp2는 각 AMI를 제공하는 데 사용되는 EBS 볼륨 유형을 결정합니다. 참조는 EBS 볼륨 유형을 참조하세요. 항상 gp2를 선택합니다.

Amazon EC2 인스턴스에 연결

SSH 및 AWS Systems Manager Session Manager EC2 Instance Connect를 포함하여 Amazon Linux 인스턴스에 연결하는 방법에는 여러 가지가 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 Linux 인스턴스에 연결을 참조하세요.

SSH 사용자 및 sudo

Amazon Linux는 기본적으로 원격 `root` 보안 쉘(SSH)을 허용하지 않습니다. 또한 암호 인증은 무차별 대입 공격을 방지하기 위해 비활성화됩니다. Amazon Linux 인스턴스에 대해 SSH 로그인을 사용하려면 시작 시 인스턴스에 키 페어를 제공해야 합니다. 또한 SSH 액세스를 허용하도록 인스턴스를 시작하는 데 사용되는 보안 그룹을 설정해야 합니다. 기본적으로 SSH를 사용하여 원격으로 로그인할 수 있는 유일한 계정은 입니다`ec2-user`. 이 계정에는 sudo 권한도 있습니다. 원격 `root` 로그인을 활성화하는 경우 키 페어와 보조 사용자를 사용하는 것보다 안전하지 않다는 점에 유의하세요.

AL2 AMI 부팅 모드

AL2 AMIs에는 부팅 모드 파라미터가 설정되어 있지 않습니다. AL2 AMIs에서 시작된 인스턴스는 인스턴스 유형의 기본 부팅 모드 값을 따릅니다. 자세한 내용은 Amazon EC2 사용 설명서의 부팅 모드를 참조하세요.

패키지 리포지토리

이 정보는 AL2에 적용됩니다. AL2023에 대한 자세한 내용은 Amazon Linux 2023 사용 설명서의 AL2023에서 패키지 및 운영 체제 업데이트 관리를 참조하세요.

AL2 및 AL1은 각 Amazon EC2 AWS 리전에서 호스팅되는 온라인 패키지 리포지토리와 함께 사용하도록 설계되었습니다. 리포지토리는 yum 업데이트 도구를 사용하여 액세스되며 모든 리전에서 사용할 수 있습니다. 각 리전에서 리포지토리를 호스팅하면 데이터 전송 요금 없이 데이터를 신속히 배포할 수 있습니다.

> **Important**
>
> AL1의 마지막 버전은 2023년 12월 31일에 EOL에 도달했으며 2024년 1월 1일부터 보안 업데이트 또는 버그 수정을 받지 않습니다. 자세한 내용은 Amazon Linux AMI 지원 종료를 참조하세요.

인스턴스에 대한 데이터 또는 사용자 지정을 보존할 필요가 없는 경우 현재 AL2 AMI를 사용하여 새 인스턴스를 시작할 수 있습니다. 인스턴스에 대한 데이터 또는 사용자 지정을 보존해야 하는 경우 Amazon Linux 패키지 리포지토리를 통해 해당 인스턴스를 유지할 수 있습니다. 이러한 리포지토리에는 업데이트된 모든 패키지가 포함되어 있습니다. 실행 중인 인스턴스에 이러한 업데이트를 적용하도록 선택할 수 있습니다. 이전 버전의 AMI 및 업데이트 패키지는 새 버전이 릴리스되더라도 계속 사용할 수 있습니다.

> **Note**
>
> Amazon EC2 인스턴스에서 인터넷 액세스 없이 패키지를 업데이트하고 설치하려면 AL1, AL2 또는 AL2023을 실행하는 Amazon EC2AL2 인스턴스에서 인터넷 액세스 없이 yum을 업데이트하거나 패키지를 설치하려면 어떻게 해야 하나요?를 참조하세요.

패키지를 설치하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo yum install package
```

필요한 애플리케이션이 Amazon Linux에 없는 경우 Amazon Linux 인스턴스에 해당 애플리케이션을 설치하면 됩니다. Amazon Linux는 yum 패키지 관리에 RPMs 및를 사용하며, 이는 새 애플리케이션을 설치하는 가장 직접적인 방법일 수 있습니다. 중앙 Amazon Linux 리포지토리에는 사용할 수 있는 애플리케이션이 많으므로 이곳에서 애플리케이션을 사용할 수 있는지부터 확인해야 합니다. 여기에서 이러한 애플리케이션을 Amazon Linux 인스턴스에 추가할 수 있습니다.

애플리케이션을 실행 중인 Amazon Linux 인스턴스에 업로드하려면 scp 또는 sftp를 사용하고 인스턴스에 로그인하여 애플리케이션을 구성합니다. 또한 기본 제공된 cloud-init 패키지의

PACKAGE_SETUP 작업을 사용하여 인스턴스 시작 중에 애플리케이션을 업로드할 수도 있습니다. 자세한 내용은 AL2에서 cloud-init 사용 섹션을 참조하세요.

보안 업데이트

보안 업데이트는 패키지 리포지토리를 사용하여 제공됩니다. 보안 업데이트와 업데이트된 AMI 보안 알림은 모두 Amazon Linux 보안 센터에 게시됩니다. AWS 보안 정책에 대한 자세한 내용을 찾아보거나 보안 문제를 보고하려면 AWS 클라우드 보안을 참조하세요.

AL1 및 AL2는 시작 시 중요하거나 중요한 보안 업데이트를 다운로드하고 설치하도록 구성됩니다. 커널 업데이트는 이 구성에 포함되지 않습니다.

AL2023에서는이 구성이 AL1 및 AL2에 비해 변경되었습니다. AL2023의 보안 업데이트에 대한 자세한 내용은 Amazon Linux 2023 사용 설명서의 보안 업데이트 및 기능을 참조하세요.

시작 후에는 사용 사례를 위해 필수 업데이트를 설치하는 것이 좋습니다. 예를 들어 시작 시 모든 업데이트(보안 업데이트뿐만 아니라)를 적용하거나 각 업데이트를 평가하고 시스템에 적용되는 업데이트만 적용할 수 있습니다. 다음과 같은 cloud-init 설정 `repo_upgrade`를 사용하여 이를 제어합니다. 다음 cloud-init 구성 코드 조각은 인스턴스 초기화에 전달하는 사용자 데이터 텍스트의 설정을 변경하는 방법을 보여 줍니다.

```
#cloud-config
repo_upgrade: security
```

`repo_upgrade`에 가능한 값은 다음과 같습니다.

`critical`

대기 중인 심각한 보안 업데이트를 적용합니다.

`important`

대기 중인 심각하거나 중요한 보안 업데이트를 적용합니다.

`medium`

대기 중인 심각하거나 중요하거나 심각도가 보통인 보안 업데이트를 적용합니다.

`low`

심각도가 낮은 보안 업데이트를 포함하여 대기 중인 보안 업데이트를 모두 적용합니다.

`security`

Amazon에서 보안 업데이트로 표시하는 대기 중인 중요하거나 심각한 업데이트를 적용합니다.

`bugfix`

Amazon에서 버그 수정 사항으로 표시하는 업데이트를 적용합니다. 버그 수정 사항은 대규모 업데이트 세트이며 보안 업데이트와 사소한 기타 버그에 대한 수정 사항을 포함합니다.

`all`

분류와 관계 없이 해당되는 모든 업데이트를 적용합니다.

`none`

시작 시 인스턴스에 어떠한 업데이트도 적용하지 마세요.

> **Note**
>
> Amazon Linux는 업데이트를 로 표시하지 않습니다`bugfix`. Amazon Linux에서 비보안 관련 업데이트를 적용하려면를 사용합니다`repo_upgrade: all`.

`repo_upgrade`에 대한 기본 설정은 security입니다. 즉, 사용자 데이터에 다른 값을 지정하지 않은 경우 기본적으로 Amazon Linux는 해당 시점에 설치된 모든 패키지에 대해 시작 시 보안 업그레이드를 수행합니다. 또한 Amazon Linux는 로그인 시 `/etc/motd` 파일을 사용하여 사용 가능한 업데이트 수를 나열하는 방법으로 설치된 패키지에 대한 모든 업데이트를 사용자에게 알립니다. 이러한 업데이트를 설치하려면 인스턴스에서 sudo yum upgrade를 실행해야 합니다.

리포지토리 구성

AL1 및 AL2의 경우 AMIs는 보안 업데이트를 제외하고 AMI가 생성된 시점에 사용 가능한 패키지의 스냅샷입니다. 원래 AMI에 없지만 런타임에 설치된 모든 패키지는 사용 가능한 최신 버전이 됩니다. AL2에 사용할 수 있는 최신 패키지를 가져오려면를 실행합니다`yum update -y`.

> **문제 해결 도움말**
>
> 예를 들어 nano 인스턴스 유형(예: `t3.nano`)에서 yum update 실행 중 `cannot allocate memory` 오류가 발생하는 경우 업데이트를 활성화하기 위해 스왑 공간을 할당해야 할 수 있습니다.

AL2023의 경우 리포지토리 구성이 AL1 및 AL2에 비해 변경되었습니다. AL2023 리포지토리에 대한 자세한 내용은 운영 체제 및 패키지 업데이트 관리를 참조하세요.

AL2023까지의 버전에서는 Amazon Linux의 한 마이너 버전에서 다음 버전으로의 연속 업데이트 흐름, 즉 롤링 릴리스를 제공하도록 구성되었습니다. 가장 좋은 방법은 이전 AMI를 시작하고 업데이트를 적용하는 대신 AMI를 사용 가능한 최신 AMIs로 업데이트하는 것입니다.

AL1에서 AL2로 또는 AL2에서 AL2023으로 등 주요 Amazon Linux 버전 간에는 현재 위치 업그레이드 AL2가 지원되지 않습니다. 자세한 내용은 Amazon Linux 가용성 단원을 참조하십시오.

AL2에서 cloud-init 사용

cloud-init 패키지는 Canonical에서 구축한 오픈 소스 애플리케이션이며 Amazon EC2와 같은 클라우드 컴퓨팅 환경에서 Linux 이미지 부트스트랩을 수행하는 데 사용됩니다. Amazon Linux에는 cloud-init의 사용자 지정 버전이 포함됩니다. 이렇게 하면 부팅 시 인스턴스에 발생해야 하는 작업을 지정할 수 있습니다. 인스턴스를 시작할 때 사용자 데이터 필드를 통해 원하는 작업을 cloud-init로 전달할 수 있습니다. 즉, 여러 사용 사례에 공통 AMI를 사용하고 이러한 AMI를 시작 시 동적으로 구성할 수 있습니다. 또한 Amazon Linux는 cloud-init를 사용하여 ec2-user 계정의 초기 구성을 수행합니다.

자세한 내용은 cloud-init 설명서를 참조하세요.

Amazon Linux는 `/etc/cloud/cloud.cfg.d` 및 `/etc/cloud/cloud.cfg`에 있는 cloud-init 작업을 사용합니다. `/etc/cloud/cloud.cfg.d`에서 자체 cloud-init 작업 파일을 만들 수 있습니다. 이 디렉터리의 모든 파일은 cloud-init으로 읽습니다. 어휘 순으로 읽히며, 나중의 파일이 이전 파일의 값을 덮어씁니다.

cloud-init 패키지는 부팅 시 인스턴스에 대해 다음과 같은 일반적인 구성 작업을 수행합니다.

- 기본 로캘 설정.
- 호스트 이름 설정.
- 사용자 데이터 구문 분석 및 처리.
- 호스트 프라이빗 SSH 키 생성.
- 손쉬운 로그인 및 관리를 위해 사용자의 퍼블릭 SSH 키를 `.ssh/authorized_keys`에 추가.
- 패키지 관리를 위한 리포지토리 준비.
- 사용자 데이터에 정의된 패키지 작업 처리.
- 사용자 데이터에 있는 사용자 스크립트를 실행합니다.
- 해당하는 경우 인스턴스 스토어 볼륨을 탑재합니다.

- 기본적으로 ephemeral0 인스턴스 스토어 볼륨은 /media/ephemeral0이 있고 유효한 파일 시스템이 포함된 경우 여기에 마운트됩니다. 그렇지 않은 경우 마운트되지 않습니다.
- 기본적으로 인스턴스와 연결된 모든 스왑 볼륨이 탑재됩니다(m1.small 및 c1.medium 인스턴스 유형만 해당).
- 다음 cloud-init 명령을 사용하여 기본 인스턴스 스토어 볼륨 탑재를 재정의할 수 있습니다.

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

  탑재를 더 세부적으로 제어하려면 cloud-init 설명서의 탑재를 참조하세요.
- TRIM을 지원하는 인스턴스 스토어 볼륨은 인스턴스를 시작할 때 포맷되지 않으므로, 볼륨을 파티셔닝하고 포맷한 후 탑재해야 합니다. 자세한 내용은 인스턴스 스토어 볼륨 TRIM 지원을 참조하세요. disk_setup 모듈을 사용하여 부팅 시 인스턴스 스토어 볼륨을 파티셔닝하고 포맷할 수 있습니다. 자세한 내용은 cloud-init 문서의 Disk Setup을 참조하세요.

지원되는 사용자 데이터 형식

cloud-init 패키지는 다양한 형식의 사용자 데이터 처리를 지원합니다.

- Gzip
  - 사용자 데이터가 gzip으로 압축된 경우 cloud-init는 데이터의 압축을 풀고 적절하게 처리합니다.
- MIME 멀티파트
  - MIME 멀티파트 파일을 사용하여 두 가지 이상의 데이터 유형을 지정할 수 있습니다. 예를 들어 사용자 데이터 스크립트와 클라우드 구성 유형을 모두 지정할 수 있습니다. 멀티파트 파일의 각 부분은 지원되는 형식 중 하나일 경우 cloud-init에 의해 처리할 수 있습니다.
- Base64 디코딩
  - 사용자 데이터가 base64로 인코딩된 경우 cloud-init는 디코딩된 데이터를 지원되는 유형 중 하나로 이해할 수 있는지 여부를 결정합니다. 디코딩된 데이터를 인식하는 경우 데이터를 디코딩하여 그에 맞게 처리합니다. 그렇지 않을 경우 base64 데이터를 원상태로 반환합니다.
- 사용자 데이터 스크립트
  - #! 또는 Content-Type: text/x-shellscript로 시작합니다.
  - 스크립트는 첫 부팅 주기에 /etc/init.d/cloud-init-user-scripts에 의해 실행됩니다. 이 동작은 부팅 프로세스 후반(초기 구성 작업이 수행된 후)에 발생합니다.

- Include 파일
  - `#include` 또는 `Content-Type: text/x-include-url`로 시작합니다.
  - 이것은 include 파일의 내용입니다. 이 파일에는 줄당 URL 하나씩, URL 목록이 포함되어 있습니다. 각각의 URL을 읽어오며 해당 내용이 이 동일한 규칙 세트를 통과합니다. URL에서 읽어온 콘텐츠는 gzip으로 압축된 형태이거나 MIME-multi-part 또는 일반 텍스트 형태일 수 있습니다.
- 클라우드 구성 데이터
  - `#cloud-config` 또는 `Content-Type: text/cloud-config`로 시작합니다.
  - 이 콘텐츠는 클라우드 구성 데이터입니다.
- Upstart 작업(AL2에서는 지원되지 않음)
  - `#upstart-job` 또는 `Content-Type: text/upstart-job`로 시작합니다.
  - 이 콘텐츠는의 파일에 저장되며`/etc/init`, 업스타트는 다른 업스타트 작업과 마찬가지로 콘텐츠를 사용합니다.
- 클라우드 부트후크
  - `#cloud-boothook` 또는 `Content-Type: text/cloud-boothook`로 시작합니다.
  - 이것은 bookhook 데이터의 내용입니다. `/var/lib/cloud`에 있는 파일에 저장된 후 즉시 실행됩니다.
  - 이것은 맨 처음으로 사용 가능한 hook입니다. 한 번만 실행되도록 제공된 메커니즘이 없습니다. boothook는 이 부분을 자체적으로 처리해야 합니다. 환경 변수 `INSTANCE_ID`에 인스턴스 ID가 함께 제공됩니다. 이 변수를 사용하여 boothook 데이터의 인스턴스당 1회 세트를 제공하세요.

AL2 인스턴스 구성

AL2 인스턴스를 성공적으로 시작하고 로그인한 후 인스턴스를 변경할 수 있습니다. 특정 애플리케이션의 요구 사항에 맞춰 다양한 방법으로 인스턴스를 구성할 수 있습니다. 다음은 관련 내용을 익히는 데 도움이 되는 몇 가지 일반적인 작업입니다.

목차

- AL2용 ec2-net-utils를 사용하여 네트워크 인터페이스 구성

일반적인 구성 시나리오

Amazon Linux의 기본 배포에는 기본적인 서버 작업에 필요한 소프트웨어 패키지 및 유틸리티가 포함되어 있습니다. 이외에도 다양한 소프트웨어 리포지토리의 더 많은 소프트웨어 패키지를 사용할 수 있고, 훨씬 더 많은 패키지를 소스 코드로 개발할 수 있습니다. 이러한 위치의 소프트웨어를 설치 및 개발하는 방법에 대한 자세한 내용은 AL2 인스턴스에서 소프트웨어 관리 섹션을 참조하세요.

Amazon Linux 인스턴스는 `ec2-user`로 미리 구성되어 제공되지만 수퍼유저 권한이 없는 다른 사용자를 추가할 수도 있습니다. 사용자 추가 및 제거에 대한 자세한 내용은 Amazon EC2 사용 설명서의 Linux 인스턴스에서 사용자 관리를 참조하세요.

도메인 이름이 등록된 자체 네트워크를 보유한 경우 인스턴스의 호스트 이름을 변경하여 해당 도메인에 속한 것으로 표시할 수 있습니다. 호스트 이름 설정은 그대로 두고 시스템 프롬프트를 더욱 의미 있는 이름으로 변경할 수도 있습니다. 자세한 내용은 AL2 인스턴스의 호스트 이름 변경 섹션을 참조하세요. 인스턴스에서 동적 DNS 서비스 공급자를 사용하도록 구성할 수 있습니다. 자세한 내용은 AL2 인스턴스에서 동적 DNS 설정 섹션을 참조하세요.

Amazon EC2에서 인스턴스를 시작할 때 사용자 데이터를 인스턴스에 전달하여 일반적인 구성 작업을 수행하는 데 사용하도록 할 수 있고, 인스턴스가 시작된 후에 스크립트를 실행할 수도 있습니다. Amazon EC2에 cloud-init 명령 및 shell 스크립트라는 두 가지 유형의 사용자 데이터를 전달할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 시작 시 Linux 인스턴스에서 명령 실행을 참조하세요.

AL2 인스턴스에서 소프트웨어 관리

Amazon Linux의 기본 배포에는 기본적인 서버 작업에 필요한 소프트웨어 패키지 및 유틸리티가 포함되어 있습니다.

이 정보는 AL2에 적용됩니다. AL2023에 대한 자세한 내용은 Amazon Linux 2023 사용 설명서의 AL2023에서 패키지 및 운영 체제 업데이트 관리를 참조하세요.

소프트웨어를 최신 상태로 유지하는 것이 중요합니다. Linux 배포의 다양한 패키지가 버그 수정, 기능 추가 및 보안 취약점 해결을 위해 자주 업데이트됩니다. 자세한 내용은 AL2 인스턴스에서 인스턴스 소프트웨어 업데이트 단원을 참조하십시오.

기본적으로 AL2 인스턴스는 다음 리포지토리가 활성화된 상태로 시작됩니다.

- `amzn2-core`

- `amzn2extra-docker`

에서 업데이트한 이러한 리포지토리에서 사용할 수 있는 패키지가 많지만 다른 리포지토리에 포함된 패키지를 설치하려는 AWS패키지가 있을 수 있습니다. 자세한 내용은 AL2 인스턴스에 리포지토리 추가 단원을 참조하십시오. 활성화된 리포지토리에서 패키지를 찾고 설치하는 방법은 AL2 인스턴스에서 소프트웨어 패키지 찾기 및 설치의 내용을 참조하세요.

리포지토리에 저장된 소프트웨어 패키지만 사용할 수 있는 것은 아닙니다. 일부 소프트웨어의 경우 인스턴스에서 소스 코드를 컴파일해야 합니다. 자세한 내용은 AL2 인스턴스에서 소프트웨어 컴파일 준비 단원을 참조하십시오.

AL2 인스턴스는 yum 패키지 관리자를 사용하여 소프트웨어를 관리합니다. yum 패키지 관리자는 소프트웨어를 설치, 제거 및 업데이트하고 각 패키지의 모든 종속성을 관리할 수 있습니다.

내용

AL2 인스턴스에서 인스턴스 소프트웨어 업데이트

소프트웨어를 최신 상태로 유지하는 것이 중요합니다. Linux 배포의 패키지가 버그 수정, 기능 추가 및 보안 취약점 해결을 위해 자주 업데이트됩니다. 처음으로 Amazon Linux 인스턴스를 시작하여 연결하면 보안을 위해 소프트웨어 패키지를 업데이트하라는 메시지가 표시될 수 있습니다. 이 섹션에서는 전체 시스템 또는 단일 패키지를 업데이트하는 방법을 보여 줍니다.

이 정보는 AL2에 적용됩니다. AL2023에 대한 자세한 내용은 Amazon Linux 2023 사용 설명서의 AL2023에서 패키지 및 운영 체제 업데이트 관리를 참조하세요.

AL2의 변경 사항 및 업데이트에 대한 자세한 내용은 AL2 릴리스 정보를 참조하세요.

AL2023 관련 변경 내용 및 업데이트에 대한 자세한 내용은 AL2023 릴리스 정보를 참조하세요.

> **Important**
>
> Amazon Linux 2 AMI를 사용하는 EC2 인스턴스를 IPv6 전용 서브넷으로 시작한 경우 인스턴스에 연결하고 `sudo amazon-linux-https disable`을 실행해야 합니다. 이렇게 하면 http

patch 서비스를 사용하여 AL2 인스턴스를 IPv6을 통해 S3의 yum 리포지토리에 연결할 수 있습니다.

AL2 인스턴스의 모든 패키지를 업데이트하려면

1. (선택 사항) shell 창에서 screen 세션을 시작합니다. 경우에 따라 네트워크 장애로 인해 인스턴스에 대한 SSH 연결이 끊어질 수 있습니다. 오래 걸리는 소프트웨어 업데이트 중에 연결이 끊어진 경우 인스턴스가 복구 가능한 혼동 상태로 유지될 수 있습니다. 연결이 끊어진 경우에도 screen 세션을 통해 업데이트가 계속 실행되며, 이후에 아무런 문제 없이 세션에 다시 연결할 수 있습니다.

   a. screen 명령을 실행하여 세션을 시작합니다.

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

   b. 세션의 연결이 끊어진 경우 인스턴스에 다시 로그인하고 사용 가능한 화면을 나열합니다.

   ```
   [ec2-user ~]$ screen -ls
   There is a screen on:
    17793.pts-0.ip-12-34-56-78 (Detached)
   1 Socket in /var/run/screen/S-ec2-user.
   ```

   c. 이전 명령에서 확인한 프로세스 ID와 screen -r 명령을 사용하여 화면에 다시 연결합니다.

   ```
   [ec2-user ~]$ screen -r 17793
   ```

   d. screen 사용을 마쳤으면 exit 명령을 사용하여 세션을 닫습니다.

   ```
   [ec2-user ~]$ exit
   [screen is terminating]
   ```

2. yum update 명령을 실행합니다. `--security` 플래그를 추가하여 보안 업데이트만 적용할 수도 있습니다.

   ```
   [ec2-user ~]$ sudo yum update
   ```

3. 나열된 패키지를 검토하고, **y**를 입력하고, Enter 키를 눌러 업데이트를 수락합니다. 시스템의 모든 패키지를 업데이트하는 데 몇 분이 걸릴 수 있습니다. yum 출력은 실행 중인 업데이트의 상태를 보여줍니다.

4. (선택 사항) 인스턴스를 재부팅하여 업데이트의 최신 패키지와 라이브러리를 사용하고 있는지 확인합니다. 커널 업데이트는 재부팅이 발생할 때까지 로드되지 않습니다. glibc 라이브러리를 업데이트한 이후에도 항상 재부팅해야 합니다. 서비스를 제어하는 패키지를 업데이트할 경우 서비스를 다시 시작하여 업데이트를 선택하면 되지만, 시스템을 재부팅하면 이전의 모든 패키지 및 라이브러리 업데이트가 완료됩니다.

AL2 인스턴스에서 단일 패키지를 업데이트하려면

이 절차를 사용하여 전체 시스템이 아닌 단일 패키지와 해당 종속 패키지를 업데이트할 수 있습니다.

1. 업데이트할 패키지의 이름이 포함된 yum update 명령을 실행합니다.

```
[ec2-user ~]$ sudo yum update openssl
```

2. 나열된 패키지 정보를 검토하고, **y**을 입력하고, Enter 키를 눌러 업데이트를 수락합니다. 해결되어야 하는 패키지 종속성이 있는 경우 둘 이상의 패키지가 나열될 수 있습니다. yum 출력은 실행 중인 업데이트의 상태를 보여줍니다.
3. (선택 사항) 인스턴스를 재부팅하여 업데이트의 최신 패키지 및 라이브러리를 사용하고 있는지 확인합니다. 커널 업데이트는 재부팅이 발생할 때까지 로드되지 않습니다. glibc 라이브러리를 업데이트한 이후에도 항상 재부팅해야 합니다. 서비스를 제어하는 패키지를 업데이트할 경우 서비스를 다시 시작하여 업데이트를 선택하면 되지만, 시스템을 재부팅하면 이전의 모든 패키지 및 라이브러리 업데이트가 완료됩니다.

AL2 인스턴스에 리포지토리 추가

이 정보는 AL2에 적용됩니다. AL2023에 대한 자세한 내용은 Amazon Linux 2023 사용 설명서의 AL2023에서 버전이 지정된 리포지토리를 통한 결정적 업그레이드를 참조하세요.

기본적으로 AL2 인스턴스는 다음 리포지토리가 활성화된 상태로 시작됩니다.

- amzn2-core
- amzn2extra-docker

Amazon Web Services에서 업데이트하는 이러한 리포지토리의 다양한 패키지 이외에도 다른 리포지토리에 포함된 패키지를 설치할 수 있습니다.

yum이 아닌 다른 리포지토리의 패키지를 설치하려면 /etc/yum.conf 디렉터리의 *repository*.repo 파일 또는 자체 /etc/yum.repos.d 파일에 리포지토리 정보를 추가해야 합

니다. 이 작업을 직접 수행할 수도 있지만, 대부분의 yum 리포지토리는 리포지토리 URL을 통해 자체 *repository*.repo 파일을 제공합니다.

yum 리포지토리가 이미 설치되어 있는지 확인하려면

다음 명령을 사용하여 설치되어 있는 yum 리포지토리를 조회합니다.

```
[ec2-user ~]$ yum repolist all
```

명령 결과에 설치된 리포지토리가 출력되고 각 상태가 보고됩니다. 사용 가능한 리포지토리에는 해당 리포지토리에 포함된 패키지 수가 표시됩니다.

/etc/yum.repos.d에 yum 리포지토리를 추가하려면 다음을 수행합니다.

1. .repo 파일의 위치를 찾습니다. 위치는 추가할 리포지토리에 따라 다를 수 있습니다. 이 예시에서 .repo 파일은 https://www.*example*.com/*repository*.repo에 있습니다.
2. yum-config-manager 명령을 사용하여 리포지토리를 추가합니다.

```
[ec2-user ~]$ sudo yum-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 활성화해야 합니다.

/etc/yum.repos.d에 yum 리포지토리를 활성화하려면 다음을 수행합니다.

yum-config-manager 플래그와 함께 `--enable` *repository* 명령을 사용합니다. 다음 명령은 Fedora 프로젝트의 EPEL(Extra Packages for Enterprise Linux) 리포지토리를 활성화합니다. 이 리포지토리는 기본적으로 Amazon Linux AMI 인스턴스의 /etc/yum.repos.d에 있지만 활성화되지 않은 상태입니다.

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

자세한 내용과이 패키지의 최신 버전을 다운로드하려면 https://fedoraproject.org/wiki/EPEL 참조하십시오.

AL2 인스턴스에서 소프트웨어 패키지 찾기 및 설치

패키지 관리 도구를 사용하여 소프트웨어 패키지를 찾고 설치할 수 있습니다. Amazon Linux 2에서 기본 소프트웨어 패키지 관리 도구는 입니다YUM. AL2023에서 기본 소프트웨어 패키지 관리 도구는 DNF입니다. 자세한 내용은 Amazon Linux 2023 사용 설명서의 패키지 관리 도구를 참조하세요.

AL2 인스턴스에서 소프트웨어 패키지 찾기

yum search 명령을 사용하여 구성된 리포지토리에서 사용 가능한 패키지 설명을 검색할 수 있습니다. 이 기능은 설치할 패키지의 이름을 정확히 알지 못할 때 특히 유용합니다. 명령에 검색어를 덧붙이기만 하면 됩니다. 여러 단어를 검색하려는 경우 검색어를 따옴표로 묶습니다.

```
[ec2-user ~]$ yum search "find"
```

다음은 예제 출력입니다.

```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
============================== N/S matched: find ===============================
findutils.x86_64 : The GNU versions of find utilities (find and xargs)
gedit-plugin-findinfiles.x86_64 : gedit findinfiles plugin
ocaml-findlib-devel.x86_64 : Development files for ocaml-findlib
perl-File-Find-Rule.noarch : Perl module implementing an alternative interface to
 File::Find
robotfindskitten.x86_64 : A game/zen simulation. You are robot. Your job is to find
 kitten.
mlocate.x86_64 : An utility for finding files by name
ocaml-findlib.x86_64 : Objective CAML package manager and build helper
perl-Devel-Cycle.noarch : Find memory cycles in objects
perl-Devel-EnforceEncapsulation.noarch : Find access violations to blessed objects
perl-File-Find-Rule-Perl.noarch : Common rules for searching for Perl things
perl-File-HomeDir.noarch : Find your home and other directories on any platform
perl-IPC-Cmd.noarch : Finding and running system commands made easy
perl-Perl-MinimumVersion.noarch : Find a minimum required version of perl for Perl code
texlive-xesearch.noarch : A string finder for XeTeX
valgrind.x86_64 : Tool for finding memory management bugs in programs
valgrind.i686 : Tool for finding memory management bugs in programs
```

여러 단어를 따옴표로 묶은 검색어를 사용하면 검색어와 정확히 일치하는 결과만 반환됩니다. 원하는 패키지가 검색되지 않은 경우 키워드를 하나만 사용하여 검색한 후 결과를 살펴봅니다. 뜻이 같은 키워드를 사용하여 검색 범위를 넓힐 수도 있습니다.

AL2용 패키지에 대한 자세한 내용은 다음을 참조하세요.

- AL2 Extras 라이브러리
- 패키지 리포지토리

AL2 인스턴스에 소프트웨어 패키지 설치

AL2에서 yum 패키지 관리 도구는 활성화된 모든 리포지토리에서 서로 다른 소프트웨어 패키지를 검색하고 소프트웨어 설치 프로세스의 모든 종속성을 처리합니다. AL2023에 소프트웨어 패키지를 설치하는 방법에 대한 자세한 내용은 Amazon Linux 2023 사용 설명서의 패키지 및 운영 체제 업데이트 관리를 참조하세요.

리포지토리에서 패키지를 설치하려면

yum install ***package*** 명령을 사용하고 *package*를 설치할 소프트웨어의 이름으로 바꿉니다. 예를 들어 links 텍스트 기반 웹 브라우저를 설치하려면 다음 명령을 입력합니다.

```
[ec2-user ~]$ sudo yum install links
```

다운로드한 RPM 패키지 파일을 설치하려면

yum install을 사용하여 인터넷에서 다운로드한 RPM 패키지 파일을 설치할 수도 있습니다. 이렇게 하려면 설치 명령에 리포지토리 패키지 이름 대신 RPM 파일의 경로 이름을 덧붙이면 됩니다.

```
[ec2-user ~]$ sudo yum install my-package.rpm
```

설치된 패키지를 나열하려면

인스턴스에 설치된 패키지 목록을 확인하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ yum list installed
```

AL2 인스턴스에서 소프트웨어 컴파일 준비

인터넷에 있는 오픈 소스 소프트웨어 중에는 아직 컴파일되지 않은 상태로 패키지 리포지토리에서 다운로드 가능한 것도 있습니다. 또한 이후에 소스 코드로 직접 컴파일해야 하는 소프트웨어 패키지를 검색할 수도 있습니다. 시스템이 AL2 및 Amazon Linux에서 소프트웨어를 컴파일할 수 있으려면 , make gcc및와 같은 여러 개발 도구를 설치해야 합니다autoconf.

소프트웨어 컴파일은 모든 Amazon EC2 인스턴스에 필요한 작업은 아니기 때문에 이러한 도구는 기본적으로 설치되지 않고 "Development Tools"라는 패키지 그룹으로 제공됩니다. yum groupinstall 명령으로 인스턴스에 이 그룹을 손쉽게 추가할 수 있습니다.

```
[ec2-user ~]$ sudo yum groupinstall "Development Tools"
```

https://github.com/ 및 http://sourceforge.net/ 등의 웹 사이트에서 소프트웨어 소스 코드 패키지를 tarball이라는 압축된 아카이브 파일로 다운로드할 수 있는 경우가 많습니다. 이러한 tarball의 파일 확장명은 일반적으로 `.tar.gz`입니다. tar 명령으로 이러한 아카이브의 압축을 풀 수 있습니다.

```
[ec2-user ~]$ tar -xzf software.tar.gz
```

소스 코드 패키지의 압축을 풀고 아카이빙을 해제한 후에는 소스 코드 디렉터리의 `README` 또는 `INSTALL` 파일을 참조하여 자세한 소스 코드 컴파일 및 설치 방법을 확인해야 합니다.

Amazon Linux 패키지의 소스 코드를 검색하려면

Amazon Web Services에서는 유지 관리되는 패키지의 소스 코드를 제공합니다. yumdownloader --source 명령을 사용하여 설치된 패키지의 소스 코드를 다운로드할 수 있습니다.

yumdownloader --source ***package*** 명령을 실행하여 *package*에 대한 소스 코드를 다운로드합니다. 예를 들어 `htop` 패키지의 소스 코드를 다운로드하려면 다음 명령을 입력합니다.

```
[ec2-user ~]$ yumdownloader --source htop

Loaded plugins: priorities, update-motd, upgrade-helper
Enabling amzn-updates-source repository
Enabling amzn-main-source repository
amzn-main-source
                        | 1.9 kB  00:00:00
amzn-updates-source
                        | 1.9 kB  00:00:00
(1/2): amzn-updates-source/latest/primary_db
                        |  52 kB  00:00:00
(2/2): amzn-main-source/latest/primary_db
                        | 734 kB  00:00:00
htop-1.0.1-2.3.amzn1.src.rpm
```

소스 RPM의 위치는 명령을 실행했던 디렉터리에 있습니다.

Amazon EC2 AL2 인스턴스에 대한 프로세서 상태 제어

C 상태는 유휴 상태일 때 코어가 진입하는 절전 수준을 제어합니다. C 상태는 C0(코어가 완전 활성 상태에서 명령을 실행하는 가장 얕은 단계) ~ C6(코어의 전원이 꺼지는 가장 깊은 유휴 단계)의 숫자로 표시됩니다.

P 상태는 코어의 성능(CPU 주파수)을 제어합니다. P 상태는 P0(코어가 인텔 Turbo Boost Technology를 사용하여 최대 주파수로 증가하는 최고 성능 설정)에서 시작하여 P1(최대 기준 주파수의 P 상태) ~ P15(최저 주파수)의 숫자로 표시됩니다.

프로세서의 성능 일관성을 향상하고 지연 시간을 줄이거나 특정 워크로드에 대해 인스턴스를 조정하기 위해 C 상태 또는 P 상태 설정을 변경할 수 있습니다. 기본 C 상태 및 P 상태는 대부분의 최고 성능을 제공하도록 설정되어 있고 대부분의 워크로드에 적합합니다. 그러나 애플리케이션에서 단일 또는 이중 코어의 높은 주파수에서 지연 시간을 줄이는 것이 비용상 이익이 되거나 Turbo Boost 버스트 주파수에 비해 낮은 주파수에서 일관된 성능을 제공하는 것이 이익이 되는 경우 이러한 인스턴스에서 사용 가능한 C 상태 또는 P 상태 설정을 시험해보는 것을 고려하세요.

운영 체제가 프로세서 C 상태 및 P 상태를 제어할 수 있는 기능을 제공하는 Amazon EC2 인스턴스 유형에 대한 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2 인스턴스에 대한 프로세서 상태 제어를 참조하세요. Amazon EC2

다음 섹션은 다른 프로세서 상태 구성 및 구성에 따른 영향을 확인하는 방법에 대해 설명합니다. 이러한 절차는 Amazon Linux용으로 작성되어 적용되지만 Linux 커널 버전이 3.9 이상인 다른 Linux 배포판에서도 작동할 수 있습니다.

> **Note**
>
> 이 페이지의 예에서는 다음을 사용합니다.
>
> - 프로세서 주파수 및 C-상태 정보를 표시하는 turbostat 유틸리티입니다. turbostat 유틸리티는 기본적으로 Amazon Linux에서 사용할 수 있습니다.
> - 워크로드를 시뮬레이션하는 stress 명령. stress를 설치하려면 먼저 sudo amazon-linux-extras install epel을 실행하여 EPEL 리포지토리를 활성화한 다음 sudo yum install -y stress를 실행합니다.
>
> 출력에 C 상태 정보가 표시되지 않을 경우 명령에 --debug 옵션을 포함시킵니다(sudo turbostat --debug stress ***<options>***).

목차

최고 Turbo Boost 주파수에서 최상의 성능

이는 Amazon Linux AMI의 기본 프로세서 상태 제어 구성이고 대부분의 워크로드에 권장됩니다. 이 구성은 변동성이 낮은 최고 성능을 제공합니다. 비활성 코어가 더 깊은 절전 상태로 진입하도록 함으로써 필요한 가용 온도를 제공하여 단일 또는 듀얼 코어 프로세서가 최대 Turbo Boost 성능을 실현할 수 있습니다.

다음 예제는 적극적으로 작업을 수행하는 코어 2개가 있는 `c4.8xlarge` 인스턴스가 최대 프로세서 Turbo Boost 주파수에 도달한 것을 보여줍니다.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [30680] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [30680] successful run completed in 10s
pk cor CPU    %c0  GHz  TSC SMI    %c1    %c3    %c6    %c7   %pc2   %pc3   %pc6   %pc7
  Pkg_W RAM_W PKG_% RAM_%
            5.54 3.44 2.90   0   9.18   0.00  85.28   0.00   0.00   0.00   0.00   0.00
  94.04 32.70 54.18  0.00
 0   0   0   0.12 3.26 2.90   0   3.61   0.00  96.27   0.00   0.00   0.00   0.00   0.00
  48.12 18.88 26.02  0.00
 0   0  18   0.12 3.26 2.90   0   3.61
 0   1   1   0.12 3.26 2.90   0   4.11   0.00  95.77   0.00
 0   1  19   0.13 3.27 2.90   0   4.11
 0   2   2   0.13 3.28 2.90   0   4.45   0.00  95.42   0.00
 0   2  20   0.11 3.27 2.90   0   4.47
 0   3   3   0.05 3.42 2.90   0  99.91   0.00   0.05   0.00
 0   3  21  97.84 3.45 2.90   0   2.11
...
 1   1  10   0.06 3.33 2.90   0  99.88   0.01   0.06   0.00
 1   1  28  97.61 3.44 2.90   0   2.32
...
10.002556 sec
```

이 예제에서는 다른 코어가 C6 절전 상태에 진입하여 전력을 절감하고 작업 코어에 전력과 가용 온도를 제공하기 때문에 vCPU 21 및 28은 최대 Turbo Boost 주파수로 실행될 수 있습니다. vCPU 3 및 10(각각은 vCPU 21 및 28과 프로세서 코어를 공유)은 C1 상태에서 명령을 대기합니다.

다음 예에서 18개 코어 모두는 적극적으로 작업을 수행하여 최대 Turbo Boost의 가용 온도가 없지만 3.2GHz의 "전체 코어 Turbo Boost" 속도에서 모두 실행됩니다.

```
[ec2-user ~]$ sudo turbostat stress -c 36 -t 10
stress: info: [30685] dispatching hogs: 36 cpu, 0 io, 0 vm, 0 hdd
```

```
stress: info: [30685] successful run completed in 10s
pk cor CPU    %c0  GHz  TSC SMI    %c1    %c3    %c6    %c7   %pc2   %pc3   %pc6   %pc7
  Pkg_W RAM_W PKG_% RAM_%
           99.27 3.20 2.90   0   0.26   0.00   0.47   0.00   0.00   0.00   0.00   0.00
 228.59 31.33 199.26  0.00
 0   0   0  99.08 3.20 2.90   0   0.27   0.01   0.64   0.00   0.00   0.00   0.00   0.00
 114.69 18.55 99.32  0.00
 0   0  18  98.74 3.20 2.90   0   0.62
 0   1   1  99.14 3.20 2.90   0   0.09   0.00   0.76   0.00
 0   1  19  98.75 3.20 2.90   0   0.49
 0   2   2  99.07 3.20 2.90   0   0.10   0.02   0.81   0.00
 0   2  20  98.73 3.20 2.90   0   0.44
 0   3   3  99.02 3.20 2.90   0   0.24   0.00   0.74   0.00
 0   3  21  99.13 3.20 2.90   0   0.13
 0   4   4  99.26 3.20 2.90   0   0.09   0.00   0.65   0.00
 0   4  22  98.68 3.20 2.90   0   0.67
 0   5   5  99.19 3.20 2.90   0   0.08   0.00   0.73   0.00
 0   5  23  98.58 3.20 2.90   0   0.69
 0   6   6  99.01 3.20 2.90   0   0.11   0.00   0.89   0.00
 0   6  24  98.72 3.20 2.90   0   0.39
...
```

C 상태 심화 제한을 통한 고성능 및 저 지연 시간

C 상태는 비활성 상태일 때 코어가 진입하는 절전 수준을 제어합니다. C 상태를 제어하여 시스템의 지연 시간과 성능 조합을 미세 조정할 수 있습니다. 코어가 절전 상태에 진입하기 위해서는 시간이 소요되고 비록 한 코어가 절전 중이면 다른 코어는 더 많은 가용 온도로 더 높은 주파수로 동작할 수 있지만 절전 중인 코어가 다시 정상 상태로 돌아와 작업을 수행하는 데는 시간이 소요됩니다. 예를 들어, 네트워크 패킷 인터럽트를 처리하는 코어가 절전 상태인 경우 인터럽트 상태를 해결하는 것이 지연될 수 있습니다. 그 경우 C 상태가 심화되지 않도록 시스템을 구성하여 프로세서 반응 지연 시간을 줄일 수 있지만 그 대가로 Turbo Boost를 위해 다른 코어에서 사용할 수 있는 가용성이 줄어듭니다.

절전 상태가 심화되지 않도록 설정하는 일반적인 방법에서는 Redis 데이터베이스 애플리케이션이 사용되고 이 경우 최대한 빠른 쿼리 응답 시간이 제공되도록 시스템 메모리에 데이터베이스가 저장됩니다.

AL2에서 더 깊은 절전 상태를 제한하려면

1. 원하는 편집기를 사용하여 `/etc/default/grub` 파일을 엽니다.

   ```
   [ec2-user ~]$ sudo vim /etc/default/grub
   ```

2. GRUB_CMDLINE_LINUX_DEFAULT 라인을 수정하고 intel_idle.max_cstate=1 및 processor.max_cstate=1 옵션을 추가하여 C1을 유휴 코어의 최대 유휴 C 상태로 설정합니다.

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0
 biosdevname=0 nvme_core.io_timeout=4294967295 intel_idle.max_cstate=1
 processor.max_cstate=1"
GRUB_TIMEOUT=0
```

   intel_idle.max_cstate=1 옵션은 인텔 기반 인스턴스에 대한 C 상태 제한을 구성하고 processor.max_cstate=1 옵션은 AMD 기반 인스턴스에 대한 C 상태 제한을 구성합니다. 구성에 두 옵션을 모두 추가하는 것이 안전합니다. 이를 통해 단일 구성으로 인텔과 AMD 플랫폼에서 원하는 동작을 설정할 수 있습니다.

3. 파일을 저장하고 편집기를 종료합니다.
4. 다음 명령을 실행하여 부팅 구성을 재구성합니다.

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

5. 인스턴스를 재부팅하여 새 커널 옵션을 활성화합니다.

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

Amazon Linux AMI에서 절전 상태 심화를 제한하려면

1. 원하는 편집기를 사용하여 /boot/grub/grub.conf 파일을 엽니다.

```
[ec2-user ~]$ sudo vim /boot/grub/grub.conf
```

2. 처음 항목의 kernel 라인을 수정하고 intel_idle.max_cstate=1 및 processor.max_cstate=1 옵션을 추가하여 C1을 유휴 코어의 최대 유휴 C 상태로 설정합니다.

```
# created by imagebuilder
default=0
timeout=1
hiddenmenu

title Amazon Linux 2014.09 (3.14.26-24.46.amzn1.x86_64)
```

```
root (hd0,0)
kernel /boot/vmlinuz-3.14.26-24.46.amzn1.x86_64 root=LABEL=/ console=ttyS0
 intel_idle.max_cstate=1  processor.max_cstate=1
initrd /boot/initramfs-3.14.26-24.46.amzn1.x86_64.img
```

`intel_idle.max_cstate=1` 옵션은 인텔 기반 인스턴스에 대한 C 상태 제한을 구성하고 `processor.max_cstate=1` 옵션은 AMD 기반 인스턴스에 대한 C 상태 제한을 구성합니다. 구성에 두 옵션을 모두 추가하는 것이 안전합니다. 이를 통해 단일 구성으로 인텔과 AMD 플랫폼에서 원하는 동작을 설정할 수 있습니다.

3. 파일을 저장하고 편집기를 종료합니다.
4. 인스턴스를 재부팅하여 새 커널 옵션을 활성화합니다.

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

다음 예제는 "전체 코어 Turbo Boost" 코어 주파수에서 적극적으로 작업을 수행하는 코어 2개가 있는 `c4.8xlarge` 인스턴스를 보여줍니다.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [5322] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [5322] successful run completed in 10s
pk cor CPU    %c0  GHz  TSC SMI    %c1    %c3    %c6    %c7   %pc2   %pc3   %pc6   %pc7
  Pkg_W RAM_W PKG_% RAM_%
            5.56 3.20 2.90   0  94.44   0.00   0.00   0.00   0.00   0.00   0.00   0.00
 131.90 31.11 199.47  0.00
 0   0   0   0.03 2.08 2.90   0  99.97   0.00   0.00   0.00   0.00   0.00   0.00   0.00
  67.23 17.11 99.76  0.00
 0   0  18   0.01 1.93 2.90   0  99.99
 0   1   1   0.02 1.96 2.90   0  99.98   0.00   0.00   0.00
 0   1  19  99.70 3.20 2.90   0   0.30
...
 1   1  10   0.02 1.97 2.90   0  99.98   0.00   0.00   0.00
 1   1  28  99.67 3.20 2.90   0   0.33
 1   2  11   0.04 2.63 2.90   0  99.96   0.00   0.00   0.00
 1   2  29   0.02 2.11 2.90   0  99.98
...
```

이 예에서 vCPUs 19 및 28 코어는 3.2GHz에서 동작하고 다른 코어는 C1 C 상태에서 명령을 대기합니다. 비록 작업 중인 코어는 최대 Turbo Boost 주파수에 도달할 수 없지만 비활성 코어는 가장 깊은 C6 C 상태에 있을 때보다 훨씬 빠르게 새 요청에 응답할 수 있습니다.

변동성이 가장 낮은 기준 성능

P 상태를 조정하여 프로세서 주파수의 변동성을 줄일 수 있습니다. P 상태는 코어의 성능(CPU 주파수)을 제어합니다. 대부분의 워크로드는 P0에서 더 좋은 성능을 발휘하지만 그 경우 Turbo Boost가 필요합니다. 그러나 Turbo Boost 주파수가 사용되는 경우 발생할 수 있는 성능 버스트보다 일관적인 성능을 갖도록 시스템을 미세 조정하는 것이 필요할 때가 있습니다.

인텔 Advanced Vector Extensions(AVX 또는 AVX2) 워크로드는 낮은 주파수에서 좋은 성능을 보이고 AVX 명령은 더 많은 전력을 사용할 수 있습니다. Turbo Boost를 비활성화하여 낮은 주파수에서 프로세서를 실행하면 사용 전력을 줄이고 스피드를 좀 더 일관성 있게 유지할 수 있습니다. 인스턴스 구성 최적화 및 AVX 워크로드에 대한 자세한 내용은 Intel 웹 사이트를 참조하세요.

CPU 유휴 드라이버는 P 상태를 제어합니다. 최신 CPU 세대에는 다음과 같이 커널 수준에 해당하는 업데이트된 CPU 유휴 드라이버가 필요합니다.

- Linux 커널 버전 6.1 이상 - Intel Granite Rapids 지원(예: R8i)
- Linux 커널 버전 5.10 이상 - AMD 밀라노(예: M6a) 지원
- Linux 커널 버전 5.6 이상 - Intel Icelake 지원(예: M6i)

실행 중인 시스템의 커널이 CPU를 인식하는지 확인하려면 다음 명령을 실행합니다.

```
if [ -d /sys/devices/system/cpu/cpu0/cpuidle ]; then echo "C-state control enabled";
 else echo "Kernel cpuidle driver does not recognize this CPU generation"; fi
```

이 명령의 출력으로 지원이 부족하다는 것을 나타내면 커널을 업그레이드하는 것이 좋습니다.

이 섹션은 절전 상태가 심화되는 것을 제한하고 Turbo Boost(P1 P 상태 요청)를 비활성화하여 이러한 워크로드 유형에 짧은 지연 시간과 낮은 프로세서 속도 변동성을 제공하는 방법에 대해 설명합니다.

더 깊은 절전 상태를 제한하고 AL2에서 Turbo Boost를 비활성화하려면

1. 원하는 편집기를 사용하여 `/etc/default/grub` 파일을 엽니다.

   ```
   [ec2-user ~]$ sudo vim /etc/default/grub
   ```

2. `GRUB_CMDLINE_LINUX_DEFAULT` 라인을 수정하고 `intel_idle.max_cstate=1` 및 `processor.max_cstate=1` 옵션을 추가하여 C1을 유휴 코어의 최대 유휴 C 상태로 설정합니다.

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0
 biosdevname=0 nvme_core.io_timeout=4294967295 intel_idle.max_cstate=1
 processor.max_cstate=1"
GRUB_TIMEOUT=0
```

`intel_idle.max_cstate=1` 옵션은 인텔 기반 인스턴스에 대한 C 상태 제한을 구성하고 `processor.max_cstate=1` 옵션은 AMD 기반 인스턴스에 대한 C 상태 제한을 구성합니다. 구성에 두 옵션을 모두 추가하는 것이 안전합니다. 이를 통해 단일 구성으로 인텔과 AMD 플랫폼에서 원하는 동작을 설정할 수 있습니다.

3. 파일을 저장하고 편집기를 종료합니다.
4. 다음 명령을 실행하여 부팅 구성을 재구성합니다.

```
[ec2-user ~]$ grub2-mkconfig -o /boot/grub2/grub.cfg
```

5. 인스턴스를 재부팅하여 새 커널 옵션을 활성화합니다.

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

6. P1 P 상태가 제공하는 낮은 프로세서 속도 변동성이 필요한 경우 다음 명령을 사용하여 Turbo Boost를 비활성화합니다.

```
[ec2-user ~]$ sudo sh -c "echo 1 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

7. 워크로드가 종료되면 다음 명령으로 Turbo Boost를 다시 활성화할 수 있습니다.

```
[ec2-user ~]$ sudo sh -c "echo 0 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

Amazon Linux AMI에서 절전 상태 심화를 제한하고 Turbo Boost를 비활성화하려면

1. 원하는 편집기를 사용하여 `/boot/grub/grub.conf` 파일을 엽니다.

```
[ec2-user ~]$ sudo vim /boot/grub/grub.conf
```

2. 처음 항목의 `kernel` 라인을 수정하고 `intel_idle.max_cstate=1` 및 `processor.max_cstate=1` 옵션을 추가하여 C1을 유휴 코어의 최대 유휴 C 상태로 설정합니다.

```
# created by imagebuilder
default=0
timeout=1
hiddenmenu

title Amazon Linux 2014.09 (3.14.26-24.46.amzn1.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-3.14.26-24.46.amzn1.x86_64 root=LABEL=/ console=ttyS0
 intel_idle.max_cstate=1 processor.max_cstate=1
initrd /boot/initramfs-3.14.26-24.46.amzn1.x86_64.img
```

`intel_idle.max_cstate=1` 옵션은 인텔 기반 인스턴스에 대한 C 상태 제한을 구성하고 `processor.max_cstate=1` 옵션은 AMD 기반 인스턴스에 대한 C 상태 제한을 구성합니다. 구성에 두 옵션을 모두 추가하는 것이 안전합니다. 이를 통해 단일 구성으로 인텔과 AMD 플랫폼에서 원하는 동작을 설정할 수 있습니다.

3. 파일을 저장하고 편집기를 종료합니다.
4. 인스턴스를 재부팅하여 새 커널 옵션을 활성화합니다.

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

5. P1 P 상태가 제공하는 낮은 프로세서 속도 변동성이 필요한 경우 다음 명령을 사용하여 Turbo Boost를 비활성화합니다.

```
[ec2-user ~]$ sudo sh -c "echo 1 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

6. 워크로드가 종료되면 다음 명령으로 Turbo Boost를 다시 활성화할 수 있습니다.

```
[ec2-user ~]$ sudo sh -c "echo 0 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

다음 예제는 Turbo Boost 없이 기준 코어 주파수에서 적극적으로 작업을 수행하는 vCPU 2개가 있는 `c4.8xlarge` 인스턴스를 보여줍니다.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [5389] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [5389] successful run completed in 10s
pk cor CPU    %c0  GHz  TSC SMI    %c1    %c3    %c6    %c7   %pc2   %pc3   %pc6   %pc7
  Pkg_W RAM_W PKG_% RAM_%
```

```
             5.59 2.90 2.90    0  94.41    0.00    0.00    0.00    0.00    0.00    0.00    0.00
 128.48 33.54 200.00  0.00
 0    0    0    0.04 2.90 2.90    0  99.96    0.00    0.00    0.00    0.00    0.00    0.00    0.00
  65.33 19.02 100.00  0.00
 0    0   18    0.04 2.90 2.90    0  99.96
 0    1    1    0.05 2.90 2.90    0  99.95    0.00    0.00    0.00
 0    1   19    0.04 2.90 2.90    0  99.96
 0    2    2    0.04 2.90 2.90    0  99.96    0.00    0.00    0.00
 0    2   20    0.04 2.90 2.90    0  99.96
 0    3    3    0.05 2.90 2.90    0  99.95    0.00    0.00    0.00
 0    3   21   99.95 2.90 2.90    0   0.05
...
 1    1   28   99.92 2.90 2.90    0   0.08
 1    2   11    0.06 2.90 2.90    0  99.94    0.00    0.00    0.00
 1    2   29    0.05 2.90 2.90    0  99.95
```

vCPUs 21 및 28용 코어는 2.9GHz의 기준 프로세서 속도에서 적극적으로 작업을 수행하고 모든 비활성 코어도 또한 C1 C 상태에서 기준 속도로 동작하여 명령을 수락할 수 있습니다.

AL2용 I/O 스케줄러

I/O 스케줄러는 I/O 요청을 정렬 및 병합하고 처리 순서를 결정하는 Linux 운영 체제의 일부입니다.

I/O 스케줄러는 탐색 시간이 비싸고 같은 위치에 있는 요청을 병합하는 것이 최적인 마그네틱 하드 드라이브와 같은 디바이스에 특히 유용합니다. I/O 스케줄러는 솔리드 스테이트 디바이스와 가상 환경에서 효과가 적습니다. 솔리드 스테이트 디바이스의 경우 순차 액세스와 무작위 액세스가 다르지 않고 가상 환경의 경우 호스트가 자체 스케줄링 계층을 제공하기 때문입니다.

이 주제에서는 Amazon Linux I/O 스케줄러에 대해 설명합니다. 다른 Linux 배포판에서 사용하는 I/O 스케줄러에 대한 자세한 내용은 해당 설명서를 참조하세요.

주제

- 지원되는 스케줄러
- 기본 스케줄러
- 스케줄러 변경

지원되는 스케줄러

Amazon Linux는 다음과 같은 I/O 스케줄러를 지원합니다.

- deadline - 기한 I/O 스케줄러는 I/O 요청을 정렬하고 가장 효율적인 순서로 처리합니다. 각 I/O 요청의 시작 시간을 보장합니다. 또한 너무 오랫동안 보류 중인 I/O 요청에 더 높은 우선순위를 부여합니다.
- cfq - 완전히 공정한 대기열(CFQ) I/O 스케줄러는 프로세스 간에 I/O 리소스를 공정하게 할당하려고 합니다. I/O 요청을 정렬하여 프로세스별 대기열에 삽입합니다.
- noop - 작업 없음(noop) I/O 스케줄러는 모든 I/O 요청을 FIFO 대기열에 삽입한 다음 단일 요청으로 병합합니다. 이 스케줄러는 요청 정렬을 수행하지 않습니다.

기본 스케줄러

작업 없음(noop)은 Amazon Linux의 기본 I/O 스케줄러입니다. 이 스케줄러는 다음과 같은 이유로 사용됩니다.

- 대부분의 인스턴스 유형은 기본 호스트가 인스턴스에 대한 예약을 수행하는 가상 디바이스를 사용합니다.
- 솔리드 스테이트 디바이스는 I/O 스케줄러의 이점이 효과가 적은 많은 인스턴스 유형에 사용됩니다.
- 침범성이 가장 적은 I/O 스케줄러이며 필요한 경우 사용자 지정할 수 있습니다.

스케줄러 변경

I/O 스케줄러를 변경하면 스케줄러가 주어진 시간에 완료되는 I/O 요청의 수를 늘리는지 아니면 줄이는지에 따라 성능이 향상되거나 저하될 수 있습니다. 이는 주로 워크로드, 사용 중인 인스턴스 유형의 생성 및 액세스 중인 디바이스 유형에 따라 달라집니다. 사용 중인 I/O 스케줄러를 변경하는 경우 iotop과 같은 도구를 사용하여 I/O 성능을 측정하고 변경 사항이 사용 사례에 유용한지 여부를 확인하는 것이 좋습니다.

예를 들어 nvme0n1과 같은 명령을 사용하여 디바이스에 대한 I/O 스케줄러를 볼 수 있습니다. 다음 명령에서 nvme0n1을 인스턴스의 /sys/block에 나열된 디바이스로 대체합니다.

```
$  cat /sys/block/nvme0n1/queue/scheduler
```

디바이스에 대한 I/O 스케줄러를 설정하려면 다음 명령을 사용합니다.

```
$  echo cfq|deadline|noop > /sys/block/nvme0n1/queue/scheduler
```

예를 들어, *xvda* 디바이스에 대해 noop에서 cfq로 I/O 스케줄러를 설정하려면 다음 명령을 사용합니다.

```
$  echo cfq > /sys/block/xvda/queue/scheduler
```

AL2 인스턴스의 호스트 이름 변경

프라이빗 VPC에서 인스턴스를 시작하는 경우 Amazon EC2에서 게스트 OS 호스트 이름을 할당합니다. Amazon EC2에서 할당하는 호스트 이름의 유형은 서브넷 설정에 따라 다릅니다. EC2 호스트 이름에 대한 자세한 내용은 Amazon EC2 사용 설명서의 Amazon EC2 인스턴스 호스트 이름 유형을 참조하세요. Amazon EC2

IPv4 주소를 갖는 IP 기반 이름 지정을 사용하여 구성되는 일반적인 Amazon EC2 프라이빗 DNS 이름은 `ip-12-34-56-78.us-west-2.compute.internal`과 같이 보이며, 여기서 이름은 내부 도메인, 서비스(이 경우 `compute`), 리전 및 프라이빗 IPv4 주소 형태로 구성됩니다. 인스턴스에 로그인하면 shell 프롬프트에 이 호스트 이름의 일부(예: `ip-12-34-56-78`)가 표시됩니다. 탄력적 IP 주소를 사용하지 않는 경우 Amazon EC2 인스턴스를 중지하고 다시 시작할 때마다 퍼블릭 IPv4 주소, 퍼블릭 DNS 이름, 시스템 호스트 이름 및 shell 프롬프트가 바뀝니다.

> **Important**
>
> 이 정보는 Amazon Linux에 적용됩니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하세요.

시스템 호스트 이름 변경

인스턴스의 IP 주소에 퍼블릭 DNS 이름을 등록한 경우(예: `webserver.mydomain.com`) 인스턴스가 자신이 해당 도메인에 속함을 인식하도록 시스템 호스트 이름을 설정할 수 있습니다. 이렇게 하면에서 제공하는 호스트 이름 대신이 이름의 첫 번째 부분이 표시되도록 셸 프롬프트도 변경됩니다 AWS (예: `ip-12-34-56-78`). 퍼블릭 DNS 이름을 등록하지 않은 경우에도 호스트 이름을 변경할 수 있지만 절차가 약간 다릅니다.

호스트 이름 업데이트를 유지하려면 `preserve_hostname` cloud-init 설정이 `true`로 설정되어 있는지 확인해야 합니다. 다음 명령을 실행하여 이 설정을 편집하거나 추가할 수 있습니다.

```
sudo vi /etc/cloud/cloud.cfg
```

`preserve_hostname` 설정이 나열되어 있지 않으면 파일 끝에 다음 텍스트 줄을 추가합니다.

```
preserve_hostname: true
```

시스템 호스트 이름을 퍼블릭 DNS 이름으로 변경하려면 다음을 수행합니다.

이미 퍼블릭 DNS 이름을 등록한 경우 이 절차를 따릅니다.

1. • AL2의 경우: hostnamectl 명령을 사용하여 정규화된 도메인 이름(예:)을 반영하도록 호스트 이름을 설정합니다**webserver.mydomain.com**.

```
[ec2-user ~]$ sudo hostnamectl set-hostname webserver.mydomain.com
```

   • Amazon Linux AMI: 인스턴스에서 선호하는 텍스트 편집기로 /etc/sysconfig/network 구성 파일을 열고 HOSTNAME 항목을 변경하여 정규화된 도메인 이름을 반영합니다(예: **webserver.mydomain.com**).

```
HOSTNAME=webserver.mydomain.com
```

2. 인스턴스를 재부팅하여 새 호스트 이름을 적용합니다.

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

   또는 Amazon EC2 콘솔을 사용하여 재부팅할 수 있습니다(인스턴스 페이지에서 인스턴스를 선택하고 인스턴스 상태, 인스턴스 재부팅을 차례로 선택).

3. 인스턴스에 로그인하고 호스트 이름이 업데이트되었는지 확인합니다. 프롬프트에 새 호스트 이름이 첫 번째 "."까지 표시되어야 하고, hostname 명령이 정규화된 도메인 이름을 표시해야 합니다.

```
[ec2-user@webserver ~]$ hostname
webserver.mydomain.com
```

퍼블릭 DNS 이름 없이 시스템 호스트 이름을 변경하려면 다음을 수행합니다.

1. • AL2의 경우: hostnamectl 명령을 사용하여 원하는 시스템 호스트 이름(예:)을 반영하도록 호스트 이름을 설정합니다**webserver**.

```
[ec2-user ~]$ sudo hostnamectl set-hostname webserver.localdomain
```

   • Amazon Linux AMI: 인스턴스에서 선호하는 텍스트 편집기로 /etc/sysconfig/network 구성 파일을 열고 HOSTNAME 항목을 변경하여 원하는 호스트 이름을 반영합니다(예: **webserver**).

```
HOSTNAME=webserver.localdomain
```

2. 선호하는 텍스트 편집기로 /etc/hosts 파일을 열고 **127.0.0.1**로 시작되는 항목을 아래 예제와 일치하도록 변경합니다. 원하는 호스트 이름을 대신 입력하면 됩니다.

```
127.0.0.1 webserver.localdomain webserver localhost4 localhost4.localdomain4
```

3. 인스턴스를 재부팅하여 새 호스트 이름을 적용합니다.

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

또는 Amazon EC2 콘솔을 사용하여 재부팅할 수 있습니다(인스턴스 페이지에서 인스턴스를 선택하고 인스턴스 상태, 인스턴스 재부팅을 차례로 선택).

4. 인스턴스에 로그인하고 호스트 이름이 업데이트되었는지 확인합니다. 프롬프트에 새 호스트 이름이 첫 번째 "."까지 표시되어야 하고, hostname 명령이 정규화된 도메인 이름을 표시해야 합니다.

```
[ec2-user@webserver ~]$ hostname
webserver.localdomain
```

사용자 데이터를 지정하여 인스턴스를 구성하는 등 더 프로그래밍 방식의 솔루션을 구현할 수도 있습니다. 인스턴스가 Auto Scaling 그룹의 일부인 경우 수명 주기 후크를 사용하여 사용자 데이터를 정의할 수 있습니다. 자세한 내용은 시작 시 Linux 인스턴스에서 명령 실행 및 AWS CloudFormation 사용 설명서의 인스턴스 시작을 위한 수명 주기 후크를 참조하세요.

호스트 이름에 영향을 주지 않고 shell 프롬프트 변경

인스턴스의 호스트 이름을 수정하지 않고에서 제공하는 프라이빗 이름(예: **webserver**)보다 더 유용한 시스템 이름 AWS (예:)을 표시하려면 쉘 프롬프트 구성 파일을 편집하여 호스트 이름 대신 시스템 별명을 표시할 ip-12-34-56-78수 있습니다.

shell 프롬프트를 호스트 별칭으로 변경하려면 다음을 수행합니다.

1. /etc/profile.d에 NICKNAME이라는 환경 변수를 shell 프롬프트로 사용할 값으로 설정하는 파일을 생성합니다. 예를 들어 시스템 별칭을 **webserver**라고 설정하려면 다음 명령을 실행합니다.

```
[ec2-user ~]$ sudo sh -c 'echo "export NICKNAME=webserver" > /etc/profile.d/
prompt.sh'
```

2. 즐겨 찾는 텍스트 편집기(예: /etc/bashrc 또는 /etc/bash.bashrc)에서 vim(Red Hat) 또는 nano(Debian/Ubuntu) 파일을 엽니다. sudo 및 /etc/bashrc는 /etc/bash.bashrc가 소유하므로 root와 함께 편집기 명령을 사용해야 합니다.
3. 파일을 편집하여 호스트 이름 대신 별칭을 표시하도록 shell 프롬프트 변수(PS1)를 변경합니다. /etc/bashrc 또는 /etc/bash.bashrc에서 shell 프롬프트를 설정하는 다음 줄을 찾습니다. 아래에서는 참조를 위해 위아래 몇 줄을 함께 표시했으며, ["$PS1"로 시작되는 줄을 찾으면 됩니다.

```
# Turn on checkwinsize
shopt -s checkwinsize
[ "$PS1" = "\\s-\\v\\$ " ] && PS1="[\u@\h \W]\\$ "
# You might want to have e.g. tty in prompt (e.g. more virtual machines)
# and console windows
```

   그 줄에서 \h(hostname에 대한 기호)를 NICKNAME 변수로 변경하세요.

```
# Turn on checkwinsize
shopt -s checkwinsize
[ "$PS1" = "\\s-\\v\\$ " ] && PS1="[\u@$NICKNAME \W]\\$ "
# You might want to have e.g. tty in prompt (e.g. more virtual machines)
# and console windows
```

4. (선택 사항) shell 창의 제목을 새 별칭으로 설정하려면 다음 단계를 완료합니다.

   a. /etc/sysconfig/bash-prompt-xterm이라는 이름의 파일을 만듭니다.

   ```
   [ec2-user ~]$ sudo touch /etc/sysconfig/bash-prompt-xterm
   ```

   b. 다음 명령으로 파일을 실행 가능하도록 만듭니다.

   ```
   [ec2-user ~]$ sudo chmod +x /etc/sysconfig/bash-prompt-xterm
   ```

   c. 선호하는 텍스트 편집기(예: /etc/sysconfig/bash-prompt-xterm 또는 vim)에서 nano 파일을 엽니다. sudo는 /etc/sysconfig/bash-prompt-xterm가 소유하므로 root와 함께 에디터 명령을 사용해야 합니다.

   d. 파일에 다음 줄을 추가합니다.

   ```
   echo -ne "\033]0;${USER}@${NICKNAME}:${PWD/#$HOME/~}\007"
   ```

5. 로그아웃하고 다시 로그인하여 새 별칭 값을 적용합니다.

다른 Linux 배포판에서 호스트 이름 변경

이 페이지의 절차는 Amazon Linux에서 사용하기 위한 것입니다. 다른 Linux 배포판에 대한 자세한 내용은 해당 설명서와 다음 항목을 참조하세요.

- RHEL 7 또는 Centos 7을 실행하는 프라이빗 Amazon EC2 인스턴스로 정적 호스트 이름을 할당하려면 어떻게 해야 합니까?

AL2 인스턴스에서 동적 DNS 설정

EC2 인스턴스를 시작하면 인터넷에서 인스턴스에 접속하는 데 사용할 수 있는 퍼블릭 IP 주소와 퍼블릭 도메인 이름 시스템(DNS) 이름이 지정됩니다. Amazon Web Services 도메인에는 수없이 많은 호스트가 있으므로 퍼블릭 이름이 상당히 길어야 각 이름의 고유성을 유지할 수 있습니다. 일반적인 Amazon EC2 퍼블릭 DNS 이름은 다음과 같습니다. `ec2-12-34-56-78.us-west-2.compute.amazonaws.com`여기서 이름은 Amazon Web Services 도메인, 서비스(이 경우 `compute`), AWS 리전및 퍼블릭 IP 주소의 형태로 구성됩니다.

동적 DNS 서비스는 도메인 영역 내에서 기억하기 쉽고 호스트의 사용 사례에 더욱 적합한 맞춤형 DNS 호스트 이름을 제공합니다. 경우에 따라 이러한 서비스를 무료로 이용할 수도 있습니다. Amazon EC2에 동적 DNS 공급자를 사용하고 인스턴스가 시작될 때마다 퍼블릭 DNS 이름에 연결된 IP 주소를 업데이트하도록 인스턴스를 구성할 수 있습니다. 매우 다양한 공급자 중에서 선택할 수 있으며, 적합한 공급자를 선택하는 구체적인 방법 및 이름을 등록하는 방법은 본 안내서의 범위를 벗어납니다.

Amazon EC2에 동적 DNS를 사용하려면 다음을 수행합니다.

1. 동적 DNS 서비스 공급자의 서비스에 가입하고 퍼블릭 DNS 이름을 등록합니다. 이 절차에서는 noip.com/free의 무료 서비스를 예제로 사용합니다.
2. 동적 DNS 업데이트 클라이언트를 구성합니다. 동적 DNS 서비스 공급자의 서비스에 가입하고 퍼블릭 DNS 이름을 등록했으면 DNS 이름에 인스턴스의 IP 주소를 가리킵니다. 공급자에 따라 (noip.com 포함) 공급자 웹 사이트의 계정 페이지에서 수동으로 입력하거나 소프트웨어 업데이트 클라이언트를 지원합니다. 업데이트 클라이언트가 EC2 인스턴스에서 실행되고 있다면 종료 및 재시작 후와 같이 IP 주소가 바뀔 때마다 동적 DNS 레코드가 업데이트됩니다. 이 예제에서는 noip.com에서 제공하는 서비스와 연동되는 noip2 클라이언트를 설치합니다.

    a. EPEL(Extra Packages for Enterprise Linux) 리포지토리를 활성화하여 `noip2` 클라이언트에 액세스할 수 있습니다.

> **Note**
>
> AL2 인스턴스에는 기본적으로 EPEL 리포지토리에 대한 GPG 키 및 리포지토리 정보가 설치되어 있습니다. 자세한 내용과이 패키지의 최신 버전을 다운로드하려면 https://fedoraproject.org/wiki/EPEL 참조하십시오.

```
[ec2-user ~]$ sudo amazon-linux-extras install epel -y
```

b. noip 패키지를 설치합니다.

```
[ec2-user ~]$ sudo yum install -y noip
```

c. 구성 파일을 생성합니다. 요청에 따라 로그인 및 암호 정보를 입력하고 후속 질문에 답하여 클라이언트를 구성합니다.

```
[ec2-user ~]$ sudo noip2 -C
```

3. noip 서비스를 활성화합니다.

```
[ec2-user ~]$ sudo systemctl enable noip.service
```

4. noip 서비스를 시작합니다.

```
[ec2-user ~]$ sudo systemctl start noip.service
```

이 명령은 클라이언트를 시작합니다. 클라이언트는 앞서 생성한 구성 파일(/etc/no-ip2.conf)을 읽고 사용자가 선택한 퍼블릭 DNS 이름의 IP 주소를 업데이트합니다.

5. 업데이트 클라이언트가 동적 DNS 이름의 IP 주소를 올바르게 설정했는지 확인합니다. 몇 분 동안 DNS 레코드가 업데이트되기를 기다린 후, 이 절차에서 구성한 퍼블릭 DNS 이름을 사용하여 SSH를 통해 인스턴스에 연결해 봅니다.

AL2용 ec2-net-utils를 사용하여 네트워크 인터페이스 구성

Amazon Linux 2 AMIs에는 ec2-net-utils라고 AWS하는이 설치한 추가 스크립트가 포함될 수 있습니다. 이러한 스크립트는 네트워크 인터페이스의 구성을 선택적으로 구성합니다. 이러한 스크립트는 AL2에서만 사용할 수 있습니다.

> **Note**
>
> Amazon Linux 2023의 경우 `amazon-ec2-net-utils` 패키지는 `/run/systemd/network` 디렉터리에서 인터페이스별 구성을 생성합니다. 자세한 정보는 Amazon Linux 2023 사용 설명서의 네트워킹 서비스를 참조하세요.

패키지가 아직 설치되지 않은 경우 다음 명령을 사용하여 AL2에 패키지를 설치하거나, 패키지가 설치되어 있고 추가 업데이트를 사용할 수 있는 경우 패키지를 업데이트합니다.

```
$ yum install ec2-net-utils
```

다음 구성 요소는 ec2-net-utils의 일부입니다.

udev 규칙(`/etc/udev/rules.d`)

실행 중인 인스턴스에 연결, 분리 또는 다시 연결될 때 네트워크 인터페이스를 식별하며 핫플러그 스크립트(`53-ec2-network-interfaces.rules`)가 실행되도록 합니다. MAC 주소를 드라이브 이름(`75-persistent-net-generator.rules`, 여기서 `70-persistent-net.rules`를 생성)에 매핑합니다.

핫플러그 스크립트

DHCP에서 사용하기에 적합한 인터페이스 구성 파일을 생성합니다(`/etc/sysconfig/network-scripts/ifcfg-ethN`). 또한 라우팅 구성 파일도 생성합니다(`/etc/sysconfig/network-scripts/route-ethN`).

DHCP 스크립트

네트워크 인터페이스에서 새 DHCP 임대를 수신할 때마다 이 스크립트는 인스턴스 메타데이터에 탄력적 IP 주소를 쿼리합니다. 각 탄력적 IP 주소마다 라우팅 정책 데이터베이스에 규칙을 추가하여 해당 주소의 아웃바운드 트래픽에 올바른 네트워크 인터페이스가 사용되도록 합니다. 또한 각 프라이빗 IP 주소를 네트워크 인터페이스에 부 주소로 추가합니다.

ec2ifup ethN (`/usr/sbin/`)

표준 ifup의 기능을 확장합니다. 이 스크립트는 구성 파일 `ifcfg-ethN` 및 `route-ethN`을 다시 쓴 후 ifup을 실행합니다.

ec2ifdown ethN (`/usr/sbin/`)

표준 ifdown의 기능을 확장합니다. 이 스크립트는 라우팅 정책 데이터베이스에서 네트워크 인터페이스 관련 규칙을 모두 제거한 후 ifdown을 실행합니다.

ec2ifscan (/usr/sbin/)

구성되지 않은 네트워크 인터페이스가 있는지 확인하고 이러한 인터페이스를 구성합니다.

이 스크립트는 ec2-net-utils의 초기 릴리스에서는 사용할 수 없습니다.

ec2-net-utils에서 생성된 구성 파일을 나열하려면 다음 명령을 사용합니다.

```
$ ls -l /etc/sysconfig/network-scripts/*-eth?
```

자동화를 비활성화하려는 경우 EC2SYNC=no를 해당 ifcfg-ethN 파일에 추가할 수 있습니다. 예를 들어, 다음 명령을 사용하여 eth1 인터페이스에 대한 자동화를 사용하지 않도록 설정합니다.

```
$ sed -i -e 's/^EC2SYNC=yes/EC2SYNC=no/' /etc/sysconfig/network-scripts/ifcfg-eth1
```

자동화를 완전히 사용하지 않으려면 다음 명령을 사용하여 패키지를 제거할 수 있습니다.

```
$ yum remove ec2-net-utils
```

사용자 제공 커널

Amazon EC2 인스턴스에 사용자 지정 커널이 필요할 경우 가장 적합한 AMI를 사용하여 시작한 후 해당 인스턴스에서 사용자 지정 커널을 컴파일하고, 부트로더를 업데이트하여 새 커널을 지정합니다. 이 프로세스는 AMI에서 사용하는 가상화 유형에 따라 다릅니다. 자세한 내용은 Amazon EC2 사용 설명서의 Linux AMI 가상화 유형을 참조하세요.

내용

- HVM AMIs(GRUB)
- AMIs 반가상화(PV-GRUB)

HVM AMIs(GRUB)

HVM 인스턴스 볼륨은 실제 물리적 디스크인 것처럼 취급됩니다. 부팅 프로세스는 디스크 파티션이 설정되고 부트로더가 있는 베어 메탈(bare metal) 운영 체제의 부팅 프로세스와 비슷하며, 현재 지원되는 모든 Linux 배포판을 사용할 수 있습니다. 가장 일반적인 부트로더는 GRUB 또는 GRUB2입니다.

기본적으로 GRUB는 추가적인 부팅 지연을 발생시키지 않기 위해 인스턴스 콘솔에 출력을 전송하지 않습니다. 자세한 내용은 Amazon EC2 사용 설명서의 인스턴스 콘솔 출력을 참조하세요. 사용자 지정 커널을 설치하는 경우 GRUB 출력을 사용하도록 설정해야 합니다.

대체 커널을 지정할 필요는 없지만, 새 커널을 테스트할 때 대체 커널을 사용하는 것이 권장됩니다. GRUB은 새 커널에 장애가 발생한 경우 다른 커널로 이를 대체하여 사용할 수 있습니다. 대체 커널이 있으면 새 커널이 없는 경우에도 인스턴스가 부팅될 수 있습니다.

Amazon Linux용 레거시 GRUB에서는 `/boot/grub/menu.lst`를 사용하며, AL2용 GRUB2는를 사용합니다`/etc/default/grub`. 부트로더에서 기본 커널을 업데이트하는 방법에 대한 자세한 내용은 해당 Linux 배포판의 설명서를 참조하세요.

AMIs 반가상화(PV-GRUB)

반가상화(PV) 가상화를 사용하는 AMIs 부팅 프로세스 중에 PV-GRUB이라는 시스템을 사용합니다. PV-GRUB는 GNU GRUB 0.97의 패치 버전을 실행하는 반가상화 부트로더입니다. 인스턴스를 실행하면 PV-GRUB가 부팅 과정을 실행하고 이미지의 `menu.lst` 파일에 지정된 커널을 체인로드합니다.

PV-GRUB는 표준 `grub.conf` 또는 `menu.lst` 명령을 이해할 수 있으며 따라서 모든 최신 지원 Linux 배포판과 함께 사용할 수 있습니다. Ubuntu 10.04 LTS, Oracle Enterprise Linux, CentOS 5.x 등 이전 배포판은 특별한 "ec2" 또는 "xen" 커널 패키지를 필요로 하지만, 새 배포판은 필요한 드라이버를 기본 커널 패키지에 포함하고 있습니다.

대부분의 PV(반가상화) AMI는 PV-GRUB AKI를 기본적으로 사용하므로(Amazon EC2 Launch Wizard Start 메뉴에서 제공되는 모든 PV Linux AMI 포함), 사용할 다른 커널이 사용자의 배포판과 호환되는 경우라면 인스턴스에서 해당 커널을 사용하기 위해 별도의 조치를 취할 필요는 없습니다. 인스턴스에서 사용자 지정 커널을 실행하는 최상의 방법은 원하는 것과 가장 근접한 AMI로 시작하여, 인스턴스에서 사용자 지정 커널을 컴파일하고, `menu.lst` 파일을 수정하는 것입니다.

AMI의 커널 이미지가 PV-GRUB AKI인지 확인할 수 있습니다. 다음 describe-images 명령(해당 커널 이미지 ID로 대체)을 실행하여 `Name` 필드가 `pv-grub`로 시작하는지 확인합니다.

```
aws ec2 describe-images --filters Name=image-id,Values=aki-880531cd
```

목차

- PV-GRUB 업데이트

PV-GRUB의 제한 사항

PV-GRUB에는 다음과 같은 제한 사항이 있습니다.

- 64비트 버전의 PV-GRUB을 사용해서 32비트 커널을 실행할 수는 없으며, 32비트 버전의 PV-GRUB을 사용해서 64비트 커널을 실행할 수도 없습니다.
- PV-GRUB AKI를 사용할 때 ARI(Amazon 램디스크 이미지)를 지정할 수 없습니다.
- AWS 는 PV-GRUB이 EXT2, EXT3, EXT4, JFS, XFS 및 ReiserFS 파일 시스템 형식으로 작동하는지 테스트하고 확인했습니다. 그 밖의 파일 시스템 포맷은 PV-GRUB에서 작동하지 않을 수 있습니다.
- PV-GRUB은 gzip, bzip2, lzo, xz 압축 포맷을 사용해서 압축된 커널을 부팅시킬 수 있습니다.
- Cluster AMI는 완전한 HVM(하드웨어 가상 머신)을 사용하기 때문에 PV-GRUB을 지원하지 않으며 이를 필요로 하지도 않습니다. PV(반가상화) 인스턴스는 PV-GRUB을 사용해서 부팅하지만, HVM 인스턴스 볼륨은 실제 디스크처럼 취급되며 그 부팅 과정은 파티션 처리된 디스크와 부트로더가 있는 베어 메탈(bare metal) 운영 체제의 부팅 과정과 유사합니다.
- PV-GRUB 버전 1.03 및 그 이하 버전은 GPT 파티셔닝을 지원하지 않으며 MBR 파티셔닝만 지원합니다.
- Amazon Elastic Block Store(Amazon EBS) 볼륨으로 LVM(Logical Volume Manager)를 사용할 계획인 경우, LVM 외부의 개별적인 부트 파티션을 필요로 합니다. 상기 요건이 갖추어지면 LVM으로 논리적 볼륨을 생성할 수 있게 됩니다.

반가상화 AMIs대해 GRUB 구성

PV-GRUB을 부팅하려면 이미지 내에 GRUB `menu.lst` 파일이 존재해야 합니다. 이 파일의 가장 일반적인 위치는 `/boot/grub/menu.lst`입니다.

다음은 PV-GRUB AKI로 AMI를 부팅하는 것에 대한 `menu.lst` 구성 파일의 예입니다. 이 예에서는 Amazon Linux 2018.03(이 AMI에 대한 원래 커널) 커널과 Vanilla Linux 4.16.4(https://www.kernel.org/에서 제공되는 Vanilla Linux 커널의 최신 버전) 커널 중에서 선택할 수 있습니다. Vanilla 항목은 해당 AMI에 대한 원래 항목에서 복제된 것이며, `kernel` 및 `initrd` 경로는 새 위치로 업데이트됩니다. `default 0` 파라미터는 부트로더가 발견한 첫 번째 항목(이 경우는 Vanilla 항목)을 참조하게 하고, `fallback 1` 파라미터는 첫 항목 부팅에 문제가 있는 경우 부트로더가 두 번째 항목을 참조하게 합니다.

```
default 0
```

```
fallback 1
timeout 0
hiddenmenu

title Vanilla Linux 4.16.4
root (hd0)
kernel /boot/vmlinuz-4.16.4 root=LABEL=/ console=hvc0
initrd /boot/initrd.img-4.16.4

title Amazon Linux 2018.03 (4.14.26-46.32.amzn1.x86_64)
root (hd0)
kernel /boot/vmlinuz-4.14.26-46.32.amzn1.x86_64 root=LABEL=/ console=hvc0
initrd /boot/initramfs-4.14.26-46.32.amzn1.x86_64.img
```

menu.lst 파일에서 대체 커널을 지정할 필요는 없지만, 새 커널을 테스트할 때 대체 커널을 사용하는 것이 권장됩니다. PV-GRUB은 새 커널에 장애가 발생한 경우 다른 커널로 이를 대체하여 사용할 수 있습니다. 대체 커널이 있을 경우 인스턴스는 새 커널을 발견할 수 없는 경우에도 부팅할 수 있습니다.

PV-GRUB은 menu.lst를 찾기 위해 다음 위치를 검사합니다(발견한 경우 그 이하 경로는 검색 안 함).

- (hd0)/boot/grub
- (hd0,0)/boot/grub
- (hd0,0)/grub
- (hd0,1)/boot/grub
- (hd0,1)/grub
- (hd0,2)/boot/grub
- (hd0,2)/grub
- (hd0,3)/boot/grub
- (hd0,3)/grub

PV-GRUB 1.03 이하 버전은 이 목록에서 첫 2개의 위치만 검색합니다.

Amazon PV-GRUB 커널 이미지 ID

PV-GRUB AKI는 아시아 태평양(오사카) 리전을 제외한 모든 Amazon EC2 리전에서 제공됩니다. 32비트 및 64비트의 두 아키텍처 유형에 대한 AKI가 존재합니다. 가장 최신의 AMI는 기본적으로 PV-GRUB AKI를 사용합니다.

모든 PV-GRUB 버전이 모든 인스턴스 유형과 호환되는 것은 아니기 때문에, 언제나 PV-GRUB AKI의 최신 버전을 사용하는 것이 권장됩니다. 다음 describe-images 명령을 사용하여 현재 리전에 대한 PV-GRUB AKI 목록을 가져옵니다.

```
aws ec2 describe-images --owners amazon --filters Name=name,Values=pv-grub-*.gz
```

PV-GRUB는 ap-southeast-2 리전에서만 제공되는 AKI입니다. 사용자가 해당 리전에 복사할 AMI가 해당 리전에서 사용 가능한 PV-GRUB의 버전을 사용하는지를 확인해야 합니다.

다음은 각 리전에 대한 현재 AKI ID입니다. hd0 AKI를 사용해서 새 AMI를 등록할 수 있습니다.

> **Note**
>
> hd00 AKI가 이전에 제공되었던 리전의 경우 이전 버전과의 호환성을 위해 hd00 AKI가 계속해서 제공되고 있습니다.

ap-northeast-1, Asia Pacific (Tokyo)

이미지 ID	이미지 이름
aki-f975a998	pv-grub-hd0_1.05-i386.gz
aki-7077ab11	pv-grub-hd0_1.05-x86_64.gz

ap-southeast-1, Asia Pacific (Singapore) Region

이미지 ID	이미지 이름
aki-17a40074	pv-grub-hd0_1.05-i386.gz
aki-73a50110	pv-grub-hd0_1.05-x86_64.gz

ap-southeast-2, Asia Pacific (Sydney)

이미지 ID	이미지 이름
aki-ba5665d9	pv-grub-hd0_1.05-i386.gz

이미지 ID	이미지 이름
aki-66506305	pv-grub-hd0_1.05-x86_64.gz

eu-central-1, Europe (Frankfurt)

이미지 ID	이미지 이름
aki-1419e57b	pv-grub-hd0_1.05-i386.gz
aki-931fe3fc	pv-grub-hd0_1.05-x86_64.gz

eu-west-1, Europe (Ireland)

이미지 ID	이미지 이름
aki-1c9fd86f	pv-grub-hd0_1.05-i386.gz
aki-dc9ed9af	pv-grub-hd0_1.05-x86_64.gz

sa-east-1, South America (São Paulo)

이미지 ID	이미지 이름
aki-7cd34110	pv-grub-hd0_1.05-i386.gz
aki-912fbcfd	pv-grub-hd0_1.05-x86_64.gz

us-east-1, US East (N. Virginia)

이미지 ID	이미지 이름
aki-04206613	pv-grub-hd0_1.05-i386.gz
aki-5c21674b	pv-grub-hd0_1.05-x86_64.gz

us-gov-west-1, AWS GovCloud(미국 서부)

이미지 ID	이미지 이름
aki-5ee9573f	pv-grub-hd0_1.05-i386.gz
aki-9ee55bff	pv-grub-hd0_1.05-x86_64.gz

us-west-1, US West (N. California)

이미지 ID	이미지 이름
aki-43cf8123	pv-grub-hd0_1.05-i386.gz
aki-59cc8239	pv-grub-hd0_1.05-x86_64.gz

us-west-2, US West (Oregon)

이미지 ID	이미지 이름
aki-7a69931a	pv-grub-hd0_1.05-i386.gz
aki-70cb0e10	pv-grub-hd0_1.05-x86_64.gz

PV-GRUB 업데이트

모든 PV-GRUB 버전이 모든 인스턴스 유형과 호환되는 것은 아니기 때문에, 언제나 PV-GRUB AKI의 최신 버전을 사용하는 것이 권장됩니다. 또한 PV-GRUB의 이전 버전이 모든 리전에서 사용 가능한 것은 아니므로, 이전 버전을 사용하는 AMI를 해당 버전을 지원하지 않는 리전으로 복사한 경우는 커널 이미지를 업데이트할 때까지 AMI에서 실행된 인스턴스를 부팅시킬 수 없습니다. 다음 절차를 사용해 PV-GRUB의 인스턴스 버전을 확인하고 필요한 경우 업데이트를 하세요.

PV-GRUB 버전 확인 방법

1. 인스턴스에 대한 커널 ID를 찾습니다.

```
aws ec2 describe-instance-attribute --instance-id instance_id --attribute kernel --
region region
```

```
{
    "InstanceId": "instance_id",
    "KernelId": "aki-70cb0e10"
}
```

이 인스턴스에 대한 커널 ID는 `aki-70cb0e10`입니다.

2. 해당 커널 ID의 버전 정보를 확인합니다.

```
aws ec2 describe-images --image-ids aki-70cb0e10 --region region

{
    "Images": [
        {
            "VirtualizationType": "paravirtual",
            "Name": "pv-grub-hd0_1.05-x86_64.gz",
            ...
            "Description": "PV-GRUB release 1.05, 64-bit"
        }
    ]
}
```

여기서 커널 이미지는 PV-GRUB 1.05입니다. 사용자의 PV-GRUB 버전이 최신 버전이 아닌 경우 (Amazon PV-GRUB 커널 이미지 ID에서 확인 가능), 다음 절차를 사용하여 이를 업데이트해야 합니다.

PV-GRUB 버전 업데이트 방법

인스턴스가 PV-GRUB의 이전 버전을 사용하는 경우, 이를 최신 버전으로 업데이트해야 합니다.

1. Amazon PV-GRUB 커널 이미지 ID에서 리전 및 프로세스 아키텍처에 대한 최신 PV-GRUB AKI를 확인합니다.
2. 인스턴스를 중단합니다. 사용하고 있는 커널 이미지를 수정하려면 인스턴스를 중단할 필요가 있습니다.

```
aws ec2 stop-instances --instance-ids instance_id --region region
```

3. 인스턴스에 대해 사용되는 커널 이미지를 수정합니다.

```
aws ec2 modify-instance-attribute --instance-id instance_id --kernel kernel_id --
region region
```

4. 인스턴스를 재시작합니다.

```
aws ec2 start-instances --instance-ids instance_id --region region
```

AL2 AMI 릴리스 알림

최신 Amazon Linux AMI가 배포될 때 알림을 받으려면 Amazon SNS를 사용하여 구독할 수 있습니다.

AL2023 알림 구독에 대한 자세한 내용은 Amazon Linux 2023 사용 설명서의 새 업데이트에 대한 알림 수신을 참조하세요.

> **Note**
>
> AL1에 대한 표준 지원은 2020년 12월 31일에 종료되었습니다. AL1 유지 관리 지원 단계는 2023년 12월 31일에 종료되었습니다. AL1 EOL 및 유지 관리 지원에 대한 자세한 내용은 블로그 게시물 Update on Amazon Linux AMI end-of-life를 참조하세요.

Amazon Linux 알림을 구독하려면

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. 필요한 경우 탐색 모음에서 리전을 미국 동부(버지니아 북부)로 변경합니다. 구독하려는 SNS 알림이 생성된 리전을 선택해야 합니다.
3. 탐색 창에서 구독과 구독 생성을 선택합니다.
4. 구독 생성 대화 상자에서 다음과 같이 수행합니다.
   a. [AL2] 주제 ARN에서 다음 Amazon 리소스 이름(ARN)을 복사하여 붙여 넣습니다. **arn:aws:sns:us-east-1:137112412989:amazon-linux-2-ami-updates**.
   b. [Amazon Linux] 주제 ARN의 경우, 다음 Amazon 리소스 이름(ARN)을 복사합니다. **arn:aws:sns:us-east-1:137112412989:amazon-linux-ami-updates**.
   c. 프로토콜에서 이메일을 선택합니다.
   d. 엔드포인트에 알림 받을 이메일 주소를 입력합니다.
   e. Create subscription을 선택합니다.

5. "AWS 알림 - 구독 확인"이라는 제목의 확인 이메일을 받게 됩니다. 이메일을 열고 구독 확인을 선택하여 구독을 완료합니다.

AMI가 릴리스될 때마다, 해당 주제의 구독자에게 알림이 발송됩니다. 이 알림을 수신하지 않으려면 다음 절차를 수행하여 구독 해제합니다.

Amazon Linux 알림을 구독 해제하려면

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. 필요한 경우 탐색 모음에서 리전을 미국 동부(버지니아 북부)로 변경합니다. SNS 알림이 생성된 리전을 사용해야 합니다.
3. 탐색 창에서 구독을 선택하고 해당 구독을 선택한 후 작업, 구독 삭제를 선택합니다.
4. 확인 메시지가 나타나면 삭제를 선택합니다.

Amazon Linux AMI SNS 메시지 형식

SNS 메시지의 스키마는 다음과 같습니다.

```
{
    "description": "Validates output from AMI Release SNS message",
    "type": "object",
    "properties": {
        "v1": {
            "type": "object",
            "properties": {
                "ReleaseVersion": {
                    "description": "Major release (ex. 2018.03)",
                    "type": "string"
                },
                "ImageVersion": {
                    "description": "Full release (ex. 2018.03.0.20180412)",
                    "type": "string"
                },
                "ReleaseNotes": {
                    "description": "Human-readable string with extra information",
                    "type": "string"
                },
                "Regions": {
                    "type": "object",
                    "description": "Each key will be a region name (ex. us-east-1)",
```

```
                    "additionalProperties": {
                        "type": "array",
                        "items": {
                            "type": "object",
                            "properties": {
                                "Name": {
                                    "description": "AMI Name (ex. amzn-ami-
hvm-2018.03.0.20180412-x86_64-gp2)",
                                    "type": "string"
                                },
                                "ImageId": {
                                    "description": "AMI Name (ex.ami-467ca739)",
                                    "type": "string"
                                }
                            },
                            "required": [
                                "Name",
                                "ImageId"
                            ]
                        }
                    }
                }
            },
            "required": [
                "ReleaseVersion",
                "ImageVersion",
                "ReleaseNotes",
                "Regions"
            ]
        }
    },
    "required": [
        "v1"
    ]
}
```

AL2 MATE 데스크톱 연결 구성

MATE 데스크톱 환경은 다음 설명과 함께 AMI에 사전 설치 및 사전 구성됩니다.

".NET Core *x.x*, Mono *x.xx*, PowerShell *x.x*, and MATE DE pre-installed to run your .NET applications on Amazon Linux 2 with Long Term Support (LTS)."

환경은 명령줄을 최소한으로 사용하여 AL2 인스턴스를 관리하기 위한 직관적인 그래픽 사용자 인터페이스를 제공합니다. 인터페이스에는 아이콘, 창, 도구 모음, 폴더, 배경 화면 및 바탕 화면 위젯과 같은 그래픽 표현이 사용됩니다. 기본 제공 GUI 기반 도구를 사용하여 일반적인 작업을 수행할 수 있습니다. 예를 들어 소프트웨어 추가 및 제거, 업데이트 적용, 파일 구성, 프로그램 시작 및 시스템 상태 모니터링을 위한 도구가 있습니다.

Important

`xrdp`는 AMI에 번들로 제공되는 원격 데스크톱 소프트웨어입니다. 기본적으로 `xrdp`는 자체 서명된 TLS 인증서를 사용하여 원격 데스크톱 세션을 암호화합니다. AWS 도 `xrdp` 유지 관리자도 프로덕션 환경에서 자체 서명된 인증서를 사용하지 않는 것이 좋습니다. 대신 적절한 CA(인증 기관)에서 인증서를 가져와 인스턴스에 설치합니다. TLS 구성에 대한 자세한 내용은 `xrdp` wiki의 TLS 보안 계층을 참조하세요.

Note

xrdp 대신 가상 네트워크 컴퓨팅(VNC) 서비스를 사용하려면 AL2Knowledge Center를 실행하는 Amazon EC2 인스턴스에 GUI를 설치하려면 어떻게 해야 합니까 AWS ? 문서를 참조하세요.

사전 조건

이 주제에 표시된 명령을 실행하려면 AWS Command Line Interface (AWS CLI) 또는를 설치하고 AWS 프로필을 AWS Tools for Windows PowerShell구성해야 합니다.

옵션

1. 설치 AWS CLI - 자세한 내용은 AWS Command Line Interface 사용 설명서의 설치 AWS CLI 및 구성 기본 사항을 참조하세요.
2. Tools for Windows PowerShell 설치 - 자세한 내용은 AWS Tools for PowerShell 사용 설명서에서 AWS Tools for Windows PowerShell설치 및 공유 자격 증명을 참조하세요.

> **Tip**
>
> 를 완전히 설치하는 대신 AWS CLI에서 직접 시작하는 브라우저 기반 사전 인증된 AWS CloudShell 쉘에를 사용할 수 있습니다 AWS Management Console. 지원되는 AWS 리전를 확인하여 작업 중인 리전에서 사용할 수 있는지 확인합니다.

RDP 연결 구성

다음 단계에 따라 로컬 시스템에서 MATE 데스크톱 환경을 실행하는 AL2 인스턴스로 RDP(원격 데스크톱 프로토콜) 연결을 설정합니다.

1. AMI 이름에 MATE가 포함된 AL2용 AMI의 ID를 가져오려면 로컬 명령줄 도구에서 describe-images 명령을 사용할 수 있습니다. 명령줄 도구를 설치하지 않은 경우 AWS CloudShell 세션에서 직접 다음 쿼리를 수행할 수 있습니다. CloudShell에서 셸 세션을 시작하는 방법에 대한 정보는 AWS CloudShell로 시작하기를 참조하세요. Amazon EC2 콘솔에서 인스턴스를 시작하고 AMI 검색 창에 MATE를 입력하여 MATE 포함 AMI를 찾을 수 있습니다. MATE가 사전 설치된 AL2 퀵 스타트가 검색 결과에 표시됩니다.

```
aws ec2 describe-images --filters "Name=name,Values=amzn2*MATE*" --query
 "Images[*].[ImageId,Name,Description]"
[
    [
        "ami-0123example0abc12",
        "amzn2-x86_64-MATEDE_DOTNET-2020.12.04",
        ".NET Core 5.0, Mono 6.12, PowerShell 7.1, and MATE DE pre-installed to run
 your .NET applications on Amazon Linux 2 with Long Term Support (LTS)."
    ],
    [
        "ami-0456example0def34",
        "amzn2-x86_64-MATEDE_DOTNET-2020.04.14",
        "Amazon Linux 2 with .Net Core, PowerShell, Mono, and MATE Desktop
 Environment"
    ]
]
```

   사용하기에 적합한 AMI를 선택합니다.

2. 이전 단계에서 찾은 AMI로 EC2 인스턴스를 시작합니다. 포트 3389로의 인바운드 TCP 트래픽을 허용하도록 보안 그룹을 구성합니다. 보안 그룹 구성에 대한 자세한 내용은 VPC의 보안 그룹을 참조하세요. 이 구성을 사용하면 RDP 클라이언트를 사용하여 인스턴스에 연결할 수 있습니다.
3. SSH를 이용해 인스턴스에 연결합니다.
4. 인스턴스의 소프트웨어 및 커널을 업데이트하세요.

```
[ec2-user ~]$ sudo yum update
```

업데이트가 완료되면 인스턴스를 재부팅하여 업데이트에서 최신 패키지 및 라이브러리를 사용 중인지를 확인합니다. 커널 업데이트를 로드하려면 재부팅해야 합니다.

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

5. 인스턴스에 다시 연결하고 Linux 인스턴스에서 다음 명령을 실행하여 ec2-user의 암호를 설정하세요.

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

6. 인증서와 키를 설치합니다.

인증서와 키가 이미 있는 경우 다음과 같이 /etc/xrdp/ 디렉터리에 복사합니다.

- 인증서 - /etc/xrdp/cert.pem
- 키 - /etc/xrdp/key.pem

인증서와 키가 없는 경우 다음 명령을 사용하여 /etc/xrdp 디렉터리에 생성합니다.

```
$ sudo openssl req -x509 -sha384 -newkey rsa:3072 -nodes -keyout /etc/xrdp/key.pem
 -out /etc/xrdp/cert.pem -days 365
```

> **Note**
>
> 이 명령은 365일간 유효한 인증서를 생성합니다.

7. 인스턴스에 연결할 컴퓨터에서 RDP 클라이언트를 엽니다(예: Microsoft Windows를 실행하는 컴퓨터에서 원격 데스크톱 연결). 사용자 이름으로 ec2-user를 입력하고 이전 단계에서 설정한 암호를 입력합니다.

Amazon EC2 인스턴스에서 **xrdp**를 비활성화하려면

Linux 인스턴스에서 다음의 명령 중 하나를 실행하여 언제든 xrdp를 비활성화할 수 있습니다. 다음의 명령은 X11 서버로 MATE를 사용하는 능력에 영향을 주지 않습니다.

```
[ec2-user ~]$ sudo systemctl disable xrdp
```

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

Amazon EC2 인스턴스에서 **xrdp**를 활성화하려면

MATE 데스크톱 환경을 실행하는 AL2 인스턴스에 연결할 수 xrdp 있도록 다시 활성화하려면 Linux 인스턴스에서 다음 명령 중 하나를 실행합니다.

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

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

AL2 자습서

다음 자습서에서는 AL2를 실행하는 Amazon EC2 인스턴스를 사용하여 일반적인 작업을 수행하는 방법을 보여줍니다. 동영상 자습서는 AWS 교육 동영상 및 실습 섹션을 참조하세요.

AL2023 지침은 Amazon Linux 2023 사용 설명서의 자습서를 참조하세요.

자습서

- 자습서: AL2에 LAMP 서버 설치
- 자습서: AL2에서 SSL/TLS 구성
- 자습서: AL2에서 WordPress 블로그 호스팅

자습서: AL2에 LAMP 서버 설치

다음 절차는 AL2 인스턴스(LAMP 웹 서버 또는 LAMP 스택이라고도 함)에 PHP 및 MariaDB(MySQL의 커뮤니티 개발 포크)를 지원하는 Apache 웹 서버를 설치하는 데 도움이 됩니다. 이 서버를 사용해서 고정 웹사이트를 호스팅하거나 데이터베이스에서 정보를 읽고 쓰는 동적 PHP 애플리케이션을 배포할 수 있습니다.

> **Important**
>
> Ubuntu 또는 Red Hat Enterprise, Linux와 같은 다른 배포에서 LAMP 웹 서버를 설정하려는 경우 이 자습서를 이용할 수 없습니다. AL2023의 경우 AL2023에 LAMP 서버 설치를 참조하세요. Ubuntu의 경우 Ubuntu 커뮤니티 설명서 ApacheMySQLPHP를 참조하세요. 다른 배포는 관련 설명서를 참조하세요.

옵션: 자동화를 사용하여 이 자습서 완료

다음 작업 대신 AWS Systems Manager 자동화를 사용하여이 자습서를 완료하려면 AWS Docs-InstallALAMPServer-AL2 자동화 문서를 실행합니다.

작업

1단계: LAMP 서버 준비

사전 조건

- 이 자습서에서는 인터넷에서 연결할 수 있는 퍼블릭 DNS 이름으로 AL2를 사용하여 새 인스턴스를 이미 시작했다고 가정합니다. 자세한 내용은 Amazon EC2 사용 설명서의 인스턴스 시작을 참조하세요. SSH(포트 22), HTTP(포트 80), HTTPS(포트 443) 연결을 허용할 수 있도록 보안 그룹을 구성해야 합니다. 이러한 사전 조건에 대한 자세한 내용은 Amazon EC2 사용 설명서의 보안 그룹 규칙을 참조하세요.
- 다음 절차에서는 현재 AL2에서 사용할 수 있는 최신 PHP 버전을 설치합니다`php8.2`. 이 자습서에서 명시한 애플리케이션이 아닌 PHP 애플리케이션을 사용하려는 경우 `php8.2`와의 호환성을 확인해야 합니다.

LAMP 서버를 준비하려면

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

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

   ```
   [ec2-user ~]$ sudo yum update -y
   ```
3. `mariadb10.5` Amazon Linux Extras 리포지토리를 설치하여 최신 버전의 MariaDB 패키지를 가져옵니다.

   ```
   [ec2-user ~]$ sudo amazon-linux-extras install mariadb10.5
   ```

   `sudo: amazon-linux-extras: command not found` 오류가 발생하면 인스턴스가 Amazon Linux 2 AMI로 실행되지 않은 것입니다(Amazon Linux AMI를 사용하고 있는 것일 수 있음). 다음 명령을 사용하여 Amazon Linux 버전을 볼 수 있습니다.

   ```
   cat /etc/system-release
   ```
4. `php8.2` Amazon Linux Extras 리포지토리를 설치하여 AL2용 PHP 패키지의 최신 버전을 가져옵니다.

   ```
   [ec2-user ~]$ sudo amazon-linux-extras install php8.2
   ```
5. 이제 인스턴스가 최신 상태이므로 Apache 웹 서버, MariaDB 및 PHP 소프트웨어 패키지를 설치할 수 있습니다. yum install 명령을 사용하여 여러 소프트웨어 패키지와 모든 관련 종속 프로그램을 동시에 설치합니다.

   ```
   [ec2-user ~]$ sudo yum install -y httpd
   ```

   다음 명령을 사용하여 이러한 패키지의 현재 버전을 볼 수 있습니다.

   ```
   yum info package_name
   ```
6. Apache 웹 서버를 시작합니다.

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

7. 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
```

8. 인스턴스에 대해 인바운드 HTTP(포트 80) 연결을 허용하는 보안 규칙이 없는 경우 추가합니다. 기본적으로 초기화 중에 인스턴스에 대해 launch-wizard-*N* 보안 그룹이 설정됩니다. 이 그룹에는 SSH 연결을 허용하는 규칙이 한 개 들어 있습니다.

   a. https://console.aws.amazon.com/ec2/에서 Amazon EC2 콘솔을 엽니다.

   b. 인스턴스를 선택하고 해당 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**
>
> `0.0.0.0/0`을 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.

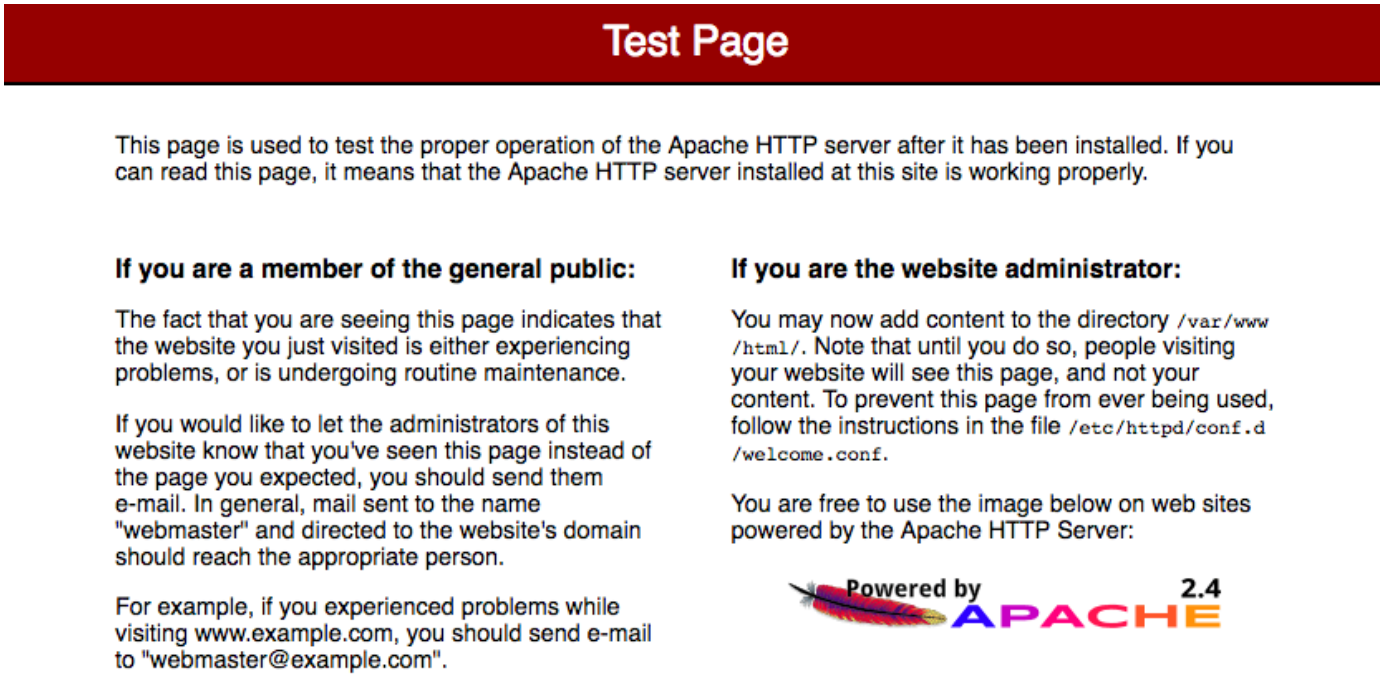
   d. 보안 그룹에 대한 링크를 선택합니다. 보안 그룹에 규칙 추가의 절차에 따라 다음 값을 사용하여 새 인바운드 보안 규칙을 추가합니다.

   - 유형: HTTP
   - 프로토콜: TCP
   - 포트 범위: 80
   - 소스: 사용자 지정

9. 웹 서버를 테스트합니다. 웹 브라우저에서 인스턴스의 공용 DNS 주소(또는 공용 IP 주소)를 입력합니다. `/var/www/html`에 콘텐츠가 없으면 Apache 테스트 페이지가 표시됩니다. Amazon EC2 콘솔을 사용하여 인스턴스의 퍼블릭 DNS를 확인할 수 있습니다. 퍼블릭 DNS 열을 확인합니다. 이 열이 숨겨진 경우 열 표시/숨기기 아이콘(기어 모양 아이콘)을 선택하고 퍼블릭 DNS를 선택합니다.

   인스턴스의 보안 그룹에 포트 80에서 HTTP 트래픽을 허용하는 규칙이 포함되어 있는지 확인합니다. 자세한 내용은 보안 그룹 규칙 추가 섹션을 참조하세요.

   > **Important**
   >
   > Amazon Linux을 사용하지 않는 경우, 이러한 연결을 허용하도록 인스턴스의 방화벽을 구성할 필요가 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-user`를

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/www와 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 자습서: AL2에서 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

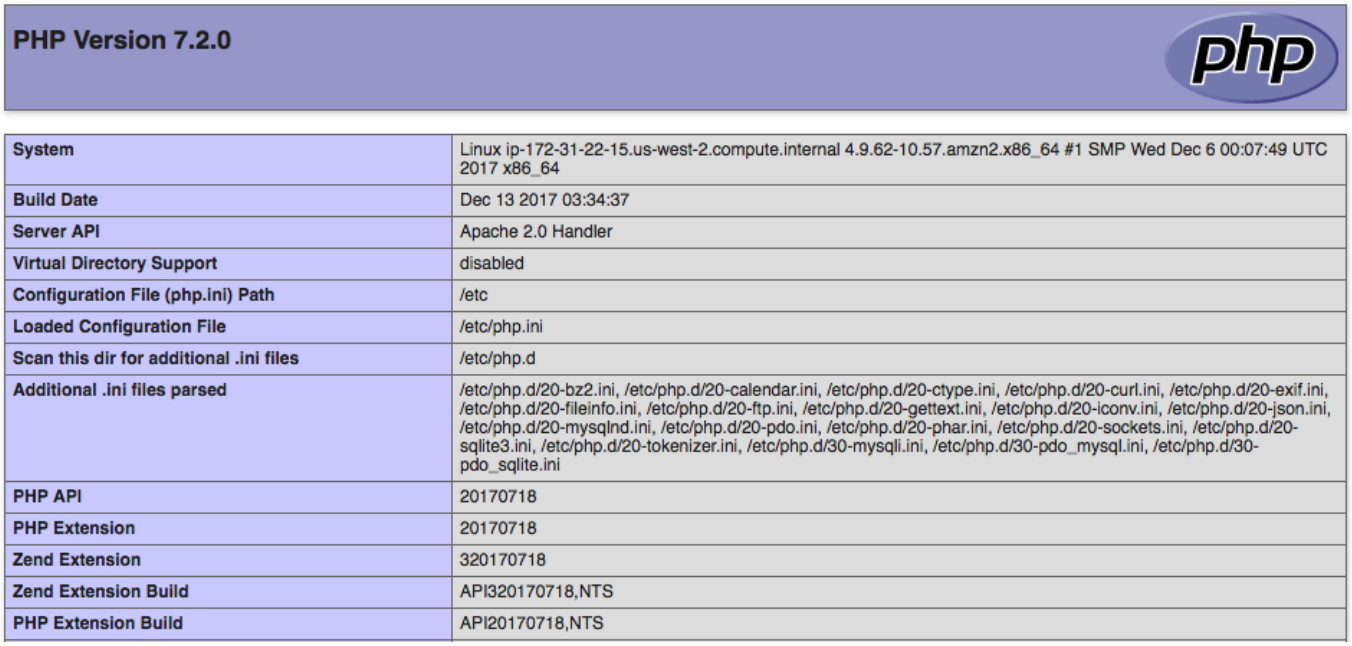
1. Apache 문서 루트에서 PHP 파일을 생성합니다.

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

   이 명령을 실행하는 동안 "Permission denied" 오류가 발생하면, 로그아웃하고 다시 로그인한 다음, 파일 권한 설정에서 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 7.2.0

System	Linux ip-172-31-22-15.us-west-2.compute.internal 4.9.62-10.57.amzn2.x86_64 #1 SMP Wed Dec 6 00:07:49 UTC 2017 x86_64
Build Date	Dec 13 2017 03:34:37
Server API	Apache 2.0 Handler
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/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-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-json.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS

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 yum list installed httpd mariadb-server php-mysqlnd
```

출력에서 필요한 패키지가 하나라도 나열되지 않으면, sudo yum install ***package*** 명령을 사용하여 패키지를 설치합니다. `php7.2` 및 `lamp-mariadb10.2-php7.2` extra가 amazon-linux-extras 명령의 출력에서 활성화되는지도 확인합니다.

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 설치는 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_installation를 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) phpMyAdmin 설치

phpMyAdmin은 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 사항을 보려면 phpMyAdmin 설치 보안을 참조하세요. EC2 인스턴스에서의 웹 서버 보안에 대한 일반적인 정보는 자습서: AL2에서 SSL/TLS 구성 섹션을 참조하세요.

phpMyAdmin을 설치하려면

1. 필요한 종속 항목을 설치합니다.

```
[ec2-user ~]$ sudo yum 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/downloads에서 phpMyAdmin 최신 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. *phpMyAdmin-latest-all-languages.tar.gz* tarball을 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 브라우저에서 phpMyAdmin 설치의 URL을 입력합니다. 아래의 예와 같이 이 URL은 인스턴스의 퍼블릭 DNS 주소(또는 퍼블릭 IP 주소)에 슬래시(/)와 설치 디렉터리의 이름이 추가된 형태입니다. 다음 예를 참조하세요.

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

   사용자는 phpMyAdmin 로그인 페이지를 볼 수 있어야 합니다:

10. 앞서 만든 `root` 사용자 이름 및 MySQL 루트 암호로 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. 이 파일을 `config.inc.php`가 포함된 phpMyAdmin 디렉토리에 `index.php`로 저장하세요.

    c. 추가 설정에 대해서는 phpMyAdmin 설치 지침에서 설정 스크립트 사용 섹션의 파일 생성 후 지침을 참조하세요.

    phpMyAdmin에 대한 자세한 내용은 phpMyAdmin 사용 설명서를 참조하세요.

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 엑세스 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 준비하려면에 설명된 단계를 반복합니다.
- 방화벽이 올바르게 구성되었습니까?

  인스턴스의 보안 그룹에 포트 80에서 HTTP 트래픽을 허용하는 규칙이 포함되어 있는지 확인합니다. 자세한 내용은 보안 그룹 규칙 추가 섹션을 참조하세요.

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 자습서: AL2에서 SSL/TLS 구성 섹션을 참조하세요.
- 방화벽이 올바르게 구성되었습니까?

  인스턴스의 보안 그룹에 포트 443에서 HTTP 트래픽을 허용하는 규칙이 포함되어 있는지 확인합니다. 자세한 내용은 보안 그룹에 규칙 추가를 참조하세요.

관련 주제

파일을 인스턴스에 전송하거나 웹 서버에 WordPress 블로그를 설치하는 것에 대한 자세한 다용은 다음 문서를 참고하세요.

- 를 사용하여 Linux 인스턴스로 파일을 전송합니다WinSCP.
- SCP 클라이언트를 사용하여 Linux 인스턴스로 파일을 전송합니다.
- 자습서: AL2에서 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/
- chmod 명령: https://en.wikipedia.org/wiki/Chmod
- chown 명령: https://en.wikipedia.org/wiki/Chown

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 마이그레이션을 참조하세요.

자습서: AL2에서 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 AL2 및 Apache 웹 서버가 있는 EC2 인스턴스에서 SSL/TLS에 대한 지원을 수동으로 추가하는 방법을 설명합니다. 이 자습서에서는 로드 밸런서를 사용하고 있지 않다고 가정합니다. Elastic Load Balancing를 사용하는 경우 AWS Certificate Manager의 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. AL2는 기본적으로 모든 버전의 SSL에 대한 서버 측 지원을 비활성화합니다. 보안 표준 본문에서는 TLS 1.0이 안전하지 않다고 간주합니다. TLS 1.0 및 TLS 1.1은 2021년 3월에 공식적으로 사용 중단되었습니다. 이 자습서에서는 TLS 1.2 활성화만을 기반으로 하여 지침을 제공합니다. TLS 1.3은 2018년에 완료되었으며, 기본 TLS 라이브러리(이 자습서의 OpenSSL)가 지원 및 사용 설정되는 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 8446을 참조하세요.

이 자습서는 현대 웹 암호화를 단순히 TLS로 언급합니다.

> **Important**
>
> 이 절차는 AL2와 함께 사용하기 위한 것입니다. 또한 사용자가 새 Amazon EC2 인스턴스로 시작한다고 가정합니다. 다른 배포를 실행하는 EC2 인스턴스 또는 이전 버전의 AL2를 실행하는 인스턴스를 설정하려는 경우이 자습서의 일부 절차가 작동하지 않을 수 있습니다. Ubuntu의 경우 커뮤니티 설명서 Ubuntu에서 SSL 열기를 참조하십시오. Red Hat Enterprise Linux의 경우 Apache HTTP 웹 서버 설정을 참조하세요. 다른 배포는 관련 설명서를 참조하세요.

> **Note**
>
> 또는 AWS Nitro Enclaves를 사용하여 Amazon EC2 인스턴스에서 실행되는 웹 애플리케이션 및 서버에 퍼블릭 및 프라이빗 SSL/TLS 인증서를 사용할 수 있는 엔클레이브 애플리케이션인 AWS Nitro 엔클레이브에 AWS Certificate Manager (ACM)을 사용할 수 있습니다. 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 AWS Nitro Enclaves란 무엇입니까?를 참조하세요. AWS Certificate Manager Nitro Enclaves 사용 설명서AWS 의 Nitro Enclaves용 및 .

내용

사전 조건

이 자습서를 시작하기 전에 다음 단계를 완료합니다.

- Amazon EBS 지원 AL2 인스턴스를 시작합니다. 자세한 내용은 Amazon EC2 사용 설명서의 인스턴스 시작을 참조하세요.

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

  자세한 내용은 Amazon EC2 사용 설명서의 보안 그룹 규칙을 참조하세요.
- Apache 웹 서버를 설치합니다. step-by-step자습서: AL2에 LAMP 웹 서버 설치를 참조하세요. 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 53을 사용할 수도 있습니다.

1단계: 서버에서 TLS 활성화

옵션: 자동화를 사용하여 이 자습서 완료

다음 작업 대신 AWS Systems Manager 자동화를 사용하여이 자습서를 완료하려면 자동화 문서를 실행합니다.

이 절차에서는 자체 서명된 디지털 인증서를 사용하여 AL2에서 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.

   ```
   [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
```

2. 모든 소프트웨어 패키지가 최신 상태로 업데이트되어 있는지 확인하기 위해, 인스턴스에서 퀵 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 yum update -y
```

3. 이제 인스턴스가 최신 상태이므로 다음과 같은 Apache module mod_ssl을 설치하여 TLS 지원을 추가합니다.

```
[ec2-user ~]$ sudo yum install -y mod_ssl
```

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

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

  mod_ssl의 구성 파일입니다. 여기에는 Apache에 암호화 키 및 인증서의 위치, 허용하는 TLS 프로토콜 버전, 허용하는 암호화 암호를 알려주는 명령이 포함되어 있습니다.

- `/etc/pki/tls/certs/make-dummy-cert`

  서버 호스트에 대한 프라이빗 키와 자체 서명된 X.509 인증서 생성용 스크립트입니다. 이 인증서는 Apache가 TLS를 사용하도록 올바르게 설치되었는지 테스트하는 데 유용합니다. 제공하는 자격 증명이 없기 때문에 프로덕션에서 사용되어서는 안됩니다. 프로덕션 환경에서 사용되는 경우 웹 브라우저에서 경고를 트리거합니다.

4. 스크립트를 실행하여 자체 서명된 테스트용 더미 인증서와 키를 생성합니다.

```
[ec2-user ~]$ cd /etc/pki/tls/certs
sudo ./make-dummy-cert localhost.crt
```

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

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-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQD2KKx/8Zk94m1q
3gQMZF9ZN66Ls19+3tHAgQ5Fpo9KJDhzLjOOCI8u1PTcGmAah5kEitCEc0wzmNeo
BCl0wYR6G0rGaKtK9Dn7CuIjvubtUysVyQoMVPQ97ldeakHWeRMiEJFXg6kZZ0vr
GvwnKoMh3DlK44D9dX7IDua2PlYx5+eroA+1Lqf32ZSaAO0bBIMIYTHigwbHMZoT
...
56tE7THvH7vOEf4/iUOsIrEzaMaJ0mqkmY1A70qQGQKBgBF3H1qNRNHuyMcPODFs
27hDzPDinrquSEvoZIggkDMlh2irTiipJ/GhkvTpoQlv0fK/VXw8vSgeaBuhwJvS
LXU9HvYq0U6O4FgD3nAyB9hI0BE13r1HjUvbjT7moH+RhnNz6eqqdscCS09VtRAo
4QQvAqOa8UheYeoXLdWcHaLP
-----END PRIVATE KEY-----

-----BEGIN CERTIFICATE-----
MIIEazCCA1OgAwIBAgICWxQwDQYJKoZIhvcNAQELBQAwgbExCzAJBgNVBAYTAi0t
MRIwEAYDVQQIDAlTb21lU3RhdGUxETAPBgNVBAcMCFNvbWVDaXR5MRkwFwYDVQQK
DBBTb21lT3JnYW5pemF0aW9uMR8wHQYDVQQLDBZTb21lT3JnYW5pemF0aW9uYWxV
bml0MRkwFwYDVQQDDBBpcC0xNzItMzEtMjAtMjM2MSQwIgYJKoZIhvcNAQkBFhVy
...
z5rRUE/XzxRLBZOoWZpNWTXJkQ3uFYH6s/sBwtHpKKZMzOvDedREjNKAvk4ws6F0
CuIjvubtUysVyQoMVPQ97ldeakHWeRMiEJFXg6kZZ0vrGvwnKoMh3DlK44D9dlU3
WanXWehT6FiSZvB4sTEXXJN2jdw8g+sHGnZ8zCOsclknYhHrCVD2vnBlZJKSZvak
3ZazhBxtQSukFMOnWPP2a0DMMFGYUHOd0BQE8sBJxg==
-----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.

5. 자체 서명된 더미 인증서에도 키가 포함되어 있으므로 자주 사용하는 텍스트 편집기(예: vim 또는 nano)를 루트 사용자로 사용하여 `/etc/httpd/conf.d/ssl.conf` 파일을 열고 다음 줄에 주석을 답니다. 다음 단계를 완료하기 전에 이 행을 주석으로 처리하지 않는 경우 Apache 서비스가 시작되지 않습니다.

```
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
```

6. Apache를 다시 시작합니다.

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

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

7. 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 수성하여 인증서 사용

자체 서명된 TLS X.509 호스트 인증서는 CA가 서명한 인증서와 암호적으로 동일합니다. 그 차이는 수학적인 것이 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 Encrycrypt 인증서 사용에 대한 자세한 내용은 Certbot 받기를 참조하세요.

상업용 서비스를 제공할 계획이라면 AWS Certificate Manager가 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. AL2에서 OpenSSL에 의해 생성된 기본 모듈러스 크기는 2048비트이며 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 건너뜁니다.
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. NIST에 따르면 이 키의 암호화 강도는 2048비트 RSA 키보다 약간 더 높습니다.

> **Note**
>
> 모든 CA에서 타원 곡선 기반 키에 대해 RSA 키와 동등한 수준의 지원을 제공하지는 않습니다.

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.

3. 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.

이름	설명	예제
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

이름	설명	예제
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 내부에서 파일 소유권, 그룹 및 권한 설정이 매우 제한적인 AL2 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 내부에서 다음 명령을 사용하여 파일 소유권, 그룹 및 권한 설정이 매우 제한적인 AL2 기본값(소유자=루트, 그룹=루트, 소유자 전용 읽기/쓰기)과 일치하는지 확인합니다.

```
[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.conf를 편집합니다.

   a. Apache의 SSLCertificateFile 명령에 CA가 서명한 호스트 인증서의 경로와 파일 이름을 입력합니다.

   ```
   SSLCertificateFile /etc/pki/tls/certs/custom.crt
   ```

   b. 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 SSLCACertificateFile 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.conf를 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 하드닝

TLS이 작동되고 일반에 공개된 후 이의 실제 보안 수준을 테스트해야 합니다. 보안 설정을 무료로 완벽하게 분석해 주는 Qualys SSL Labs와 같은 온라인 서비스를 사용하면 이를 손쉽게 수행할 수 있습니다. 그 결과에 따라 수용할 프로토콜, 원하는 암호 및 제외할 암호를 관리하여 기본 보안 구성을 하드닝할 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 AL2의 기본 Apache 구성과 동일한 설정과 기본 Certbot 인증서가 있는 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.

✗ 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

취약점이 있는 것으로 알려져 있습니다. 타당한 레거시 브라우저 지원 사유가 없다면 비활성화해야 합니다.

✗ 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.

✗ 전방향 보안은 부분적으로 지원됩니다. 전방향 보안은 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 Generator의 출력에 기반합니다. 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/ssl.conf에 복사합니다.

> **Note**
>
> 여기에서는 가독성을 위해 여러 줄로 표시했지만, 이 명령은 /etc/httpd/conf.d/ssl.conf에 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/ssl.conf`에 저장하고 Apache를 재시작합니다.

Qualys SSL Labs에서 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 AL2 인스턴스를 up-to-date 유지하고, OpenSSL의 보안 공지를 주시하고, 기술 보도에서 새로운 보안 악용에 대한 보고를 주의 깊게 확인하세요.

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.key`라는 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 yum 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 AL2를 실행하고 있지 않음을 의미합니다. 이 자습서는 공식 AL2 AMI에서 새로 생성된 인스턴스만 지원합니다.

자습서: AL2에서 WordPress 블로그 호스팅

다음 절차는 AL2 인스턴스에 WordPress 블로그를 설치, 구성 및 보호하는 데 도움이 됩니다. 본 자습서는 기존 호스팅 서비스에서는 일반적이지 않은 WordPress 블로그를 호스팅하는 웹 서버를 사용자가 완전히 제어할 수 있다는 점에서 Amazon EC2 사용에 있어 좋은 입문 기회를 제공합니다.

사용자는 서버에 대한 소프트웨어 패키지를 업데이트하고 보안 패치를 유지관리할 책임이 있습니다. 웹 서버 구성과 직접 상호 작용할 필요가 없는 보다 자동화된 WordPress 설치를 위해 CloudFormation 서비스는 빠르게 시작할 수 있는 WordPress 템플릿을 제공합니다. 자세한 내용은 AWS CloudFormation 사용 설명서에서 시작하기를 참조하세요. 데이터베이스가 분리된 고가용성 솔루션이 필요하다면 AWS Elastic Beanstalk 개발자 안내서에서 고가용성 WordPress 웹 사이트 배포를 참조하세요.

> **Important**
>
> 이 절차는 AL2와 함께 사용하기 위한 것입니다. 기타 배포에 대한 자세한 내용은 해당 설명서를 참조하세요. 본 자습서에 있는 단계의 상당수가 Ubuntu 인스턴스에서 작동하지 않습니다. Ubuntu 인스턴스에 WordPress를 설치하는 방법은 Ubuntu 설명서에서 WordPress 섹션을 참조하세요. CodeDeploy를 사용하여 Amazon Linux, macOS 또는 Unix 시스템에서 태스크를 수행할 수도 있습니다.

주제

사전 조건

이 자습서에서는의 모든 단계에 따라 PHP 및 데이터베이스(MySQL 또는 MariaDB)가 지원되는 기능 웹 서버가 있는 AL2 인스턴스를 시작했다고 가정합니다자습서: AL2에 LAMP 서버 설치. 또한 본 자습서는 보안 그룹이 HTTP 및 HTTPS 트래픽을 허용하도록 구성하는 단계와 파일 권한이 웹 서버에 맞게 적절하게 설정되어 있는지 확인하는 여러 단계를 포함하고 있습니다. 보안 그룹 규칙 추가에 대한 자세한 내용은 보안 그룹에 규칙 추가 섹션을 참조하세요.

탄력적 IP 주소(EIP)는 WordPress 블로그를 호스팅하는 데 사용 중인 인스턴스와 연결하는 것이 가장 바람직합니다. 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 주소를 참조하세요.

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 등록을 참조하세요.

WordPress 설치

옵션: 자동화를 사용하여 이 자습서 완료

다음 작업 대신 AWS Systems Manager 자동화를 사용하여이 자습서를 완료하려면 자동화 문서를 실행합니다.

인스턴스에 연결한 후 WordPress 설치 패키지를 다운로드합니다.

WordPress 설치 패키지의 다운로드 및 압축해제 방법

1. wget 명령을 사용하여 최신 WordPress 설치 패키지를 다운로드 합니다. 다음 명령을 사용할 경우 언제나 최신 릴리스를 다운로드합니다.

```
[ec2-user ~]$ wget https://wordpress.org/latest.tar.gz
```

2. 설치 패키지의 압축 및 아카이빙을 해제합니다. 설치 폴더는 wordpress라는 폴더로 압축 해제됩니다.

```
[ec2-user ~]$ tar -xzf latest.tar.gz
```

WordPress 설치에 대한 데이터베이스 사용자 및 데이터베이스를 생성하려면

WordPress 설치 시 블로그 게시물, 사용자 의견 등의 정보를 데이터베이스에 저장해야 합니다. 다음 절차를 통해 블로그의 데이터베이스와 이 데이터베이스에 대해 정보 읽기 및 저장 권한이 있는 사용자를 생성할 수 있습니다.

1. 데이터베이스 서버를 시작합니다.
   -
```
[ec2-user ~]$ sudo systemctl start mariadb
```
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 MariaDB 서버의 보안을 유지하려면 (AL2)를 참조하세요.

```
[ec2-user ~]$ mysql -u root -p
```
3. MySQL 데이터베이스에 대한 사용자 및 암호를 생성합니다. WordPress 설치는 MySQL 데이터베이스를 통신하기 위해 상기 값을 사용합니다.

   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.

   고유한 사용자 이름과 암호로 해당 부분을 대체하여 다음 명령을 입력합니다.

```
CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY 'your_strong_password';
```
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`;
```

5. 데이터베이스에 대한 전체 권한을 이전에 생성한 WordPress 사용자에게 부여합니다.

```
GRANT ALL PRIVILEGES ON `wordpress-db`.* TO "wordpress-user"@"localhost";
```

6. 데이터베이스 권한을 새로고침(flush)해서 모든 변경사항이 적용되도록 합니다.

```
FLUSH PRIVILEGES;
```

7. `mysql` 클라이언트를 종료합니다.

```
exit
```

wp-config.php 파일 생성 및 편집 방법

WordPress 설치 폴더는 `wp-config-sample.php`라는 샘플 구성 파일을 포함하고 있습니다. 본 절차에서는 이 파일을 복사하고 특정 구성에 맞도록 편집합니다.

1. `wp-config-sample.php` 파일을 `wp-config.php`라는 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
```

2. `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
```

a. DB_NAME을(를) 정의하는 줄을 찾고 database_name_here을(를) Step 4의 WordPress 설치에 대한 데이터베이스 사용자 및 데이터베이스를 생성하려면에서 생성한 데이터베이스 이름으로 변경합니다.

```
define('DB_NAME', 'wordpress-db');
```

b. DB_USER을(를) 정의하는 줄을 찾고 username_here을(를) Step 3의 WordPress 설치에 대한 데이터베이스 사용자 및 데이터베이스를 생성하려면에서 생성한 데이터베이스 사용자로 변경합니다.

```
define('DB_USER', 'wordpress-user');
```

c. DB_PASSWORD을(를) 정의하는 줄을 찾고 password_here을(를) Step 3의 WordPress 설치에 대한 데이터베이스 사용자 및 데이터베이스를 생성하려면에서 생성한 보안성이 강력한 암호로 변경합니다.

```
define('DB_PASSWORD', 'your_strong_password');
```

d. Authentication Unique Keys and Salts라는 섹션을 검색합니다. 이 KEY 및 SALT 값은 WordPress 사용자가 로컬 컴퓨터에 저장하는 브라우저 쿠키에 암호 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 내용을 보려면 http://codex.wordpress.org/Editing_wp-config.php#Security_Keys로 이동하세요.

> **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 ?
6OP$eJT@;+(ndLg');
```

```
define('LOGGED_IN_KEY',    'ju}qwre3V*+8f_zOWf?{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-
Es7Q1O-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.

WordPress 파일을 Apache 문서 루트 아래에 설치하려면

- 설치 폴더 압축을 해제하고 MySQL 데이터베이스 및 맞춤형 WordPress 구성 파일을 사용자 설정했으므로, 이제 설치 파일을 웹 서버 문서 루트에 복사하여 설치를 완료하는 설치 스크립트를 실행할 수 있습니다. 이 파일의 위치는 WordPress 블로그를 웹 서버의 실제 루트(예: *my.public.dns.amazonaws.com*)에서 사용하도록 할지 아니면 루트 아래의 하위 디렉터리나 폴더(예: *my.public.dns.amazonaws.com/blog*)에서 사용하도록 할지에 따라 다릅니다.
  - WordPress를 문서 루트에서 실행하려면 WordPress 설치 디렉터리의 파일(디렉터리 자체는 제외)을 다음과 같이 복사합니다.

    ```
    [ec2-user ~]$ cp -r wordpress/* /var/www/html/
    ```

  - WordPress를 문서 루트의 대체 디렉터리에서 실행하려면 먼저 해당 디렉터리를 생성한 후 파일을 그 디렉터리로 복사합니다. 이 예에서는 다음과 같이 WordPress가 `blog` 디렉터리에서 실행됩니다.

    ```
    [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. Wordpress 설치를 Apache 문서 루트 아래로 이동한 후에는 WordPress 설치 스크립트가 보호되지 않는 상태이기 때문에 Apache 웹 서버가 실행 중

일 때 블로그에 침입자가 액세스할 가능성이 있습니다. Apache 웹 서버를 중지하려면 sudo systemctl stop httpd 명령을 입력합니다. 다음 절차로 즉시 이동하는 경우는 Apache 웹 서버를 중단시킬 필요가 없습니다.

WordPress에서 퍼머링크(permalinks)를 사용하는 방법

WordPress가 올바로 작동하려면 Apache `.htaccess` 파일을 사용해야 하지만 Amazon Linux에서는 기본적으로 이 파일을 사용할 수 없습니다. 따라서 아래 방법에 따라 Apache 문서 루트에서 모든 재정의를 허용해야 합니다.

1. 자주 사용하는 텍스트 편집기(`httpd.conf` 또는 nano)로 vim 파일을 엽니다. 원하는 텍스트 편집기가 없는 경우 초보자에게는 nano가 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.

AL2에 PHP 그래픽 그리기 라이브러리를 설치하려면

PHP용 GD 라이브러리를 사용하면 이미지를 수정할 수 있습니다. 블로그의 헤더 이미지를 잘라야 하는 경우 이 라이브러리를 설치합니다. 설치하는 phpMyAdmin 버전에는 이 라이브러리의 특정 최소 버전(예: 버전 7.2)이 필요할 수 있습니다.

다음 명령을 사용하여 AL2에 PHP 그래픽 그리기 라이브러리를 설치합니다. 예를 들어, LAMP 스택 설치의 일부로 amazon-linux-extras에서 php7.2를 설치한 경우 이 명령은 PHP 그래픽 그리기 라이브러리의 버전 7.2를 설치합니다.

```
[ec2-user ~]$ sudo yum install php-gd
```

설치된 버전을 확인하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo yum list installed php-gd
```

다음은 예제 출력입니다.

```
php-gd.x86_64                  7.2.30-1.amzn2           @amzn2extra-php7.2
```

Apache 웹 서버에 대한 파일 권한 수정 방법

WordPress의 제공 기능 중 일부(예: 관리 화면을 통한 미디어 업로드 등)는 Apache 문서 루트에 대한 쓰기 권한을 필요로 합니다. 아직 적용하지 않은 경우 다음 그룹 멤버십 및 권한을 적용합니다(에 자세히 설명되어 있음자습서: AL2에 LAMP 서버 설치).

1. /var/www의 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**
   >
   > WordPress를 FTP 서버로도 사용하려는 경우 여기서 더 많은 권한 그룹 설정이 필요합니다. 이 작업을 수행하려면 권장된 WordPress 단계 및 보안 설정을 검토하세요.

5. Apache 웹 서버를 재시작해서 새 그룹 및 권한을 가져옵니다.

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

AL2를 사용하여 WordPress 설치 스크립트 실행

이제 WordPress를 설치할 준비가 되었습니다. 사용하는 명령은 운영 체제에 따라 다릅니다. 이 절차의 명령은 AL2와 함께 사용됩니다.

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 URL을 입력합니다(인스턴스에 대한 퍼블릭 DNS 주소 또는 blog 폴더 다음의 주소). 이제 WordPress 설치 스크립트가 나타납니다. WordPress 설치에 필요한 정보를 제공합니다. WordPress 설치(Install WordPress)를 선택해서 설치를 완료합니다. 자세한 내용은 WordPress 웹 사이트의 5단계: 설치 스크립트 실행을 참조하세요.

다음 단계

WordPress 블로그를 테스트한 후 구성을 업데이트하세요.

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 참조하세요.

블로그 구성

다른 테마와 플러그인을 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 생성

하여 설치 중 오류가 발생하더라도 블로그를 복구할 수 있도록 대비하는 것이 좋습니다. 자세한 내용은 자체 AMI 생성을 참조하세요.

용량 증가

운영하는 WordPress 블로그가 유명해지고 그에 따라 보다 많은 컴퓨팅 파워 또는 스토리지가 필요하게 될 경우 다음 단계를 고려하세요.

- 인스턴스에서 스토리지 공간을 확장합니다. 자세한 내용은 Amazon EBS 사용 설명서의 Amazon EBS Elastic Volumes를 참조하세요.
- MySQL 데이터베이스를 Amazon RDS로 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 Accelerator는 사용자의 클라이언트 디바이스와 AWS에서 실행되는 WordPress 애플리케이션 간의 인터넷 트래픽 성능을 개선하여 지연 시간을 줄일 수 있도록 돕습니다. Global Accelerator는 AWS 글로벌 네트워크를 사용하여 클라이언트와 가장 가까운 AWS 리전의 정상 애플리케이션 엔드포인트로 트래픽을 전달합니다.

WordPress에 대해 자세히 알아보기

WordPress에 대한 자세한 내용은 http://codex.wordpress.org/에서 WordPress Codex 도움 문서를 참조하세요.

설치 문제 해결에 대한 자세한 내용은 일반적인 설치 문제를 참조하세요.

WordPress 블로그의 보안을 강화하는 방법에 대한 자세한 내용은 WordPress 강화를 참조하세요.

WordPress 블로그up-to-date 유지하는 방법에 대한 자세한 내용은 WordPress 업데이트를 참조하세요.

도움말! 내 퍼블릭 DNS 이름이 변경되어 블로그를 사용할 수 없습니다.

WordPress 설치 위치는 EC2 인스턴스의 퍼블릭 DNS 주소를 사용해 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 사이트 URL 변경에 요약되어 있습니다.

WordPress 설치에이 문제가 발생한 경우 WordPress용 wp-cli 명령줄 인터페이스를 사용하는 아래 절차를 사용하여 블로그를 복구할 수 있습니다.

wp-cli를 사용하여 WordPress 사이트 URL을 바꾸는 방법

1. SSH를 통해 EC2 인스턴스에 연결합니다.
2. 인스턴스의 이전 사이트 URL과 새로운 사이트 URL을 기록합니다. 이전 사이트 URL은 WordPress 설치 시 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. 아래와 같은 명령으로 이전 사이트 URL을 찾아 WordPress 설치 위치로 바꿉니다. EC2 인스턴스의 이전 사이트 URL과 새로운 사이트 URL, 그리고 WordPress 설치 경로(일반적으로 `/var/www/html` 또는 `/var/www/html/blog`)를 치환합니다.

   ```
   [ec2-user ~]$ php wp-cli.phar search-replace 'old_site_url' 'new_site_url' --path=/
   path/to/wordpress/installation --skip-columns=guid
   ```

5. 웹 브라우저에서 WordPress 블로그의 새로운 사이트 URL을 입력하여 사이트에 올바로 접속되는지 다시 확인합니다. 그렇지 않은 경우 사이트 URL 변경 및 일반적인 설치 문제에서 자세한 내용을 참조하세요.

Amazon EC2 외부에서 Amazon Linux 2 사용

AL2 컨테이너 이미지는 호환되는 컨테이너 런타임 환경에서 실행할 수 있습니다.

AL2는 Amazon EC2에서 직접 실행되는 외부의 가상화된 게스트로 실행할 수도 있습니다.

> **Note**
>
> AL2 이미지의 구성은 AL2023과 다릅니다.
> AL2023으로 마이그레이션할 때는 Amazon EC2 외부에서 Amazon Linux 2023 사용을 검토하고 AL2023과 호환되도록 구성을 조정해야 합니다.

온프레미스에서 가상 머신으로 AL2 실행

온프레미스 개발 및 테스트에 AL2 가상 머신(VM) 이미지를 사용합니다. 지원되는 각 가상화 플랫폼에 대해 서로 다른 AL2 VM 이미지를 제공합니다. Amazon Linux 2 virtual machine images(Amazon Linux 2 가상 머신 이미지) 페이지에서 지원되는 플랫폼 목록을 볼 수 있습니다.

지원되는 가상화 플랫폼 중 하나에서 AL2 가상 머신 이미지를 사용하려면 다음을 수행합니다.

- 1단계: seed.iso 부팅 이미지 준비
- 2단계: AL2 VM 이미지 다운로드
- 3단계: 새 VM 부팅 및 연결

1단계: `seed.iso` 부팅 이미지 준비

`seed.iso` 부팅 이미지에는 네트워크 구성, 호스트 이름, 사용자 데이터와 같이 새 VM을 부팅하는 데 필요한 초기 구성 정보가 포함되어 있습니다.

> **Note**
>
> `seed.iso` 부팅 이미지에는 VM을 부팅하는 데 필요한 구성 정보만 포함합니다. AL2 운영 체제 파일은 포함되지 않습니다.

`seed.iso` 부팅 이미지를 생성하려면 다음과 같은 구성 파일 두 개가 필요합니다.

- meta-data - 이 파일에는 VM에 대한 호스트 이름과 정적 네트워크 설정이 포함됩니다.
- user-data - 이 파일에서는 사용자 계정을 구성하고 해당 계정의 암호, 키 페어, 액세스 메커니즘을 지정합니다. 기본적으로 AL2 VM 이미지는 ec2-user 사용자 계정을 생성합니다. user-data 구성 파일을 사용하여 기본 사용자 계정의 암호를 설정합니다.

seed.iso 부팅 디스크를 생성하려면

1. seedconfig라는 새 폴더를 만들고 이 폴더로 이동합니다.
2. meta-data 구성 파일을 생성합니다.

   a. meta-data라는 이름의 새로운 파일을 만듭니다.

   b. 원하는 편집기를 사용하여 meta-data 파일을 열고 다음을 추가합니다.

   ```
   local-hostname: vm_hostname
   # eth0 is the default network interface enabled in the image. You can configure
    static network settings with an entry like the following.
   network-interfaces: |
     auto eth0
     iface eth0 inet static
     address 192.168.1.10
     network 192.168.1.0
     netmask 255.255.255.0
     broadcast 192.168.1.255
     gateway 192.168.1.254
   ```

   *vm_hostname*을 선택한 VM 호스트 이름으로 바꾸고 필요에 따라 네트워크 설정을 구성합니다.

   c. meta-data 구성 파일을 저장하고 닫습니다.

   VM 호스트 이름(meta-data)을 지정하고, 기본 네트워크 인터페이스(amazonlinux.onprem)를 구성하며, 필요한 네트워크 디바이스의 정적 IP 주소를 지정하는 eth0 구성 파일의 예는 샘플 Seed.iso 파일을 참조하세요.
3. user-data 구성 파일을 생성합니다.

   a. user-data라는 이름의 새로운 파일을 만듭니다.

   b. 원하는 편집기를 사용하여 user-data 파일을 열고 다음을 추가합니다.

```
#cloud-config
#vim:syntax=yaml
users:
# A user by the name `ec2-user` is created in the image by default.
  - default
chpasswd:
  list: |
    ec2-user:plain_text_password
# In the above line, do not add any spaces after 'ec2-user:'.
```

*plain_text_password*를 기본 ec2-user 사용자 계정을 대해 선택한 암호로 바꿉니다.

c. (선택 사항) 기본적으로 cloud-init은 VM이 부팅될 때마다 네트워크 설정을 적용합니다. 부팅될 때마다 cloud-init이 네트워크 설정을 적용하지 않고, 첫 번째 부팅 중에 적용된 네트워크 설정을 유지하려면 다음을 추가합니다.

```
# NOTE: Cloud-init applies network settings on every boot by default. To retain
 network settings
# from first boot, add the following 'write_files' section:
write_files:
  - path: /etc/cloud/cloud.cfg.d/80_disable_network_after_firstboot.cfg
    content: |
      # Disable network configuration after first boot
      network:
        config: disabled
```

d. user-data 구성 파일을 저장하고 닫습니다.

추가 사용자 계정을 생성하고 이 계정의 액세스 메커니즘, 암호 및 키 페어를 지정할 수도 있습니다. 지원되는 명령에 대한 자세한 내용은 모듈 참조에서 확인하세요. 추가 사용자를 세 명 생성하고 기본 user-data 사용자 계정의 사용자 지정 암호를 지정하는 ec2-user 파일의 예는 샘플 Seed.iso 파일을 참조하세요.

4. seed.iso 및 meta-data 구성 파일을 사용하여 user-data 부팅 이미지를 생성합니다.

Linux의 경우 genisoimage와 같은 도구를 사용합니다. seedconfig 폴더로 이동하여 다음 명령을 실행합니다.

```
$ genisoimage -output seed.iso -volid cidata -joliet -rock user-data meta-data
```

macOS의 경우 hdiutil과 같은 도구를 사용합니다. `seedconfig` 폴더에서 한 수준 위로 이동하여 다음 명령을 실행합니다.

```
$ hdiutil makehybrid -o seed.iso -hfs -joliet -iso -default-volume-name cidata
 seedconfig/
```

2단계: AL2 VM 이미지 다운로드

지원되는 각 가상화 플랫폼에 대해 서로 다른 AL2 VM 이미지를 제공합니다. 지원되는 플랫폼 목록을 보고 Amazon Linux 2 virtual machine images(Amazon Linux 2 가상 머신 이미지) 페이지에서 선택한 플랫폼에 대한 올바른 VM 이미지를 다운로드할 수 있습니다.

3단계: 새 VM 부팅 및 연결

부팅하고 새 VM에 연결하려면 `seed.iso` 부팅 이미지(1단계에서 생성됨)와 AL2 VM 이미지(2단계에서 다운로드됨)가 있어야 합니다. 선택한 VM 플랫폼에 따라 단계가 달라집니다.

VMware vSphere

VMware용 VM 이미지는 OVF 형식으로 제공됩니다.

VMware vSphere를 사용하여 VM을 부팅하려면

1. `seed.iso` 파일에 대한 새 데이터스토어를 생성하거나 이 파일을 기존 데이터 스토어에 추가합니다.
2. OVF 템플릿을 배포하되 아직 VM을 시작하지 않습니다.
3. Navigator(탐색기 패널에서 새 가상 머신을 마우스 오른쪽 버튼으로 클릭하고 설정 편집을 선택합니다.
4. Virtual Hardware(가상 하드웨어) 탭에서 New device(새 디바이스)에 대해 CD/DVD Drive(CD/DVD 드라이브)를 선택한 다음 추가를 선택합니다.
5. New CD/DVD Drive(새 CD/DVD 드라이브)에서 Datastore ISO File(데이터 스토어 ISO 파일)을 선택합니다. `seed.iso` 파일을 추가한 데이터 스토어를 선택하고 `seed.iso` 파일을 찾아 선택한 다음 확인을 선택합니다.
6. 새 DC/DVD 드라이브에서 연결을 선택한 다음 확인을 선택합니다.

데이터 스토어를 VM과 연결한 후에는 VM을 부팅할 수 있어야 합니다.

KVM

KVM을 사용하여 VM을 부팅하려면

1. Create new VM(새 VM 만들기) 마법사를 엽니다.
2. 1단계에서 Import existing disk image(기존 디스크 이미지 가져오기)를 선택합니다.
3. 2단계에서 VM 이미지를 찾아 선택합니다. OS 유형(OS type) 및 버전(Version)에서 Linux 및 Red Hat Enterprise Linux 7.0을 각각 선택합니다.
4. 3단계에서 사용할 RAM 용량과 CPU 수를 지정합니다.
5. 4단계에서 새 VM의 이름을 입력하고 Customize configuration before install(설치 전에 구성 사용자 지정)을 선택한 다음 마침을 선택합니다.
6. VM의 구성 창에서 Add Hardware(하드웨어 추가)를 선택합니다.
7. Add New Virtual Hardware(새 가상 하드웨어 추가) 창에서 스토리지를 선택합니다.
8. 스토리지 구성에서 Select or create custom storage(사용자 지정 스토리지 선택 또는 생성)을 선택합니다. 디바이스 유형에서 CDROM device(CDROM 디바이스)를 선택합니다. 관리, Browse Local(로컬 찾아보기)를 선택한 다음 `seed.iso` 파일을 찾아 선택합니다. [마침]을 클릭합니다.
9. Begin Installation(설치 시작)을 선택합니다.

Oracle VirtualBox

Oracle VirtualBox를 사용하여 VM을 부팅하려면

1. Oracle VirtualBox를 열고 새로 만들기(New)를 선택합니다.
2. 이름(Name)에 가상 머신의 알기 쉬운 이름을 입력하고 유형(Type) 및 버전(Version)에서 각각 Linux와 Red Hat(64비트)(Red Hat (64-bit))을 선택합니다. [Continue]를 선택합니다.
3. 메모리 크기(Memory size)에 가상 머신에 할당할 메모리 양을 지정한 다음 계속(Continue)을 선택합니다.
4. 하드 디스크(Hard disk)에서 기존 가상 하드 디스크 파일 사용(Use an existing virtual hard disk file)을 선택하고 VM 이미지를 연 다음 생성(Create)을 선택합니다.
5. VM을 시작하기 전에 가상 머신의 가상 광학 드라이브에 `seed.iso` 파일을 로드해야 합니다.

    a. 새 VM을 선택하고 설정을 선택한 다음 스토리지를 선택합니다.

    b. 스토리지 디바이스(Storage Devices) 목록의 컨트롤러: IDE(Controller: IDE) 아래에서 빈(Empty) 광학 드라이브를 선택합니다.

c. 광학 드라이브의 속성 섹션에서 찾아보기 버튼을 선택하고 Choose Virtual Optical Disk File(가상 광학 디스크 파일 선택)을 선택한 다음 `seed.iso` 파일을 선택합니다. 확인(OK)을 선택하여 변경 사항을 적용하고 설정을 닫습니다.

가상 광학 드라이브에 `seed.iso` 파일을 추가한 후에는 VM을 부팅할 수 있습니다.

Microsoft Hyper-V

Microsoft Hyper-V용 VM 이미지는 zip 파일로 압축됩니다. 이 zip 파일의 내용을 추출해야 합니다.

Microsoft Hyper-V를 사용하여 VM을 부팅하려면

1. 새 가상 컴퓨터 마법사(New Virtual Machine Wizard)를 엽니다.
2. 세대를 선택하라는 메시지가 표시되면 Generation 1(1세대)을 선택합니다.
3. 네트워크 어댑터를 구성하라는 메시지가 표시되면 연결에 외부를 선택합니다.
4. 가상 하드 디스크를 연결하라는 메시지가 표시되면 Use an existing virtual hard disk(기존 가상 하드 디스크 사용)를 선택하고 찾아보기를 선택한 다음 VM 이미지를 찾아 선택합니다. 마침을 선택하여 VM을 생성합니다.
5. 새 VM을 마우스 오른쪽 버튼으로 클릭하고 설정을 선택합니다. 설정 창의 IDE Controller 1(IDE 컨트롤러 1)에서 DVD Drive(DVD 드라이브)를 선택합니다.
6. DVD 드라이브에서 이미지 파일을 선택한 다음 `seed.iso` 파일을 찾아 선택합니다.
7. 변경 사항을 적용하고 VM을 시작합니다.

VM이 부팅된 후에 `user-data` 구성 파일에 정의된 사용자 계정 중 하나를 사용해 로그인합니다. 처음 로그인한 후 VM에서 `seed.iso` 부팅 이미지를 연결 해제할 수 있습니다.

Amazon Linux 인스턴스 및 버전 식별

OS 이미지 또는 인스턴스의 Linux 배포와 배포 버전을 확인할 수 있어야 할 수 있습니다. Amazon Linux는 다른 Linux 배포판과 별도로 Amazon Linux를 식별하고 이미지의 Amazon Linux 릴리스를 식별하는 메커니즘을 제공합니다.

이 섹션에서는 사용할 수 있는 다양한 방법, 제한 사항을 다루고 몇 가지 사용 예제를 살펴봅니다.

주제

`os-release` 표준 사용

Amazon Linux는 Linux 배포판 식별을 위해 `os-release` 표준을 준수합니다. 이 파일은 운영 체제 식별 및 버전 정보에 대해 기계가 판독할 수 있는 정보를 제공합니다.

> **Note**
>
> 표준은 `/etc/os-release`가 먼저 구문 분석되고 그 뒤에 `/usr/lib/os-release`가 오도록 지시합니다. 파일 이름 및 경로에 대한 표준을 따르도록 주의해야 합니다.

주제

주요 식별 차이점

`os-release`는 `/etc/os-release`에서 찾을 수 있으며, 없는 경우 `/usr/lib/os-release`에서 찾을 수 있습니다. 자세한 내용은 `os-release` 표준 섹션을 참조하세요.

인스턴스가 Amazon Linux를 실행하는지 확인하는 가장 신뢰할 수 있는 방법은os-release의 ID 필드를 확인하는 것입니다.

버전을 구분하는 가장 신뢰할 수 있는 방법은 os-release에서 VERSION_ID 필드를 확인하는 것입니다.

- Amazon Linux AMI: VERSION_ID는 날짜 기반 버전 포함(예: 2018.03)
- AL2: VERSION_ID="2"
- AL2023: VERSION_ID="2023"

> **Note**
>
> VERSION_ID는 프로그래밍 방식으로 사용하기 위한 기계가 판독할 수 있는 필드인 반면, PRETTY_NAME은 사용자에게 표시하도록 설계되었습니다. 필드 유형에 대한 자세한 내용은 the section called “필드 유형” 섹션을 참조하세요.

필드 유형: 기계가 판독할 수 있음 및 사람이 읽을 수 있음

/etc/os-release 파일(또는 /etc/os-release가 존재하지 않는 경우 /usr/lib/os-release)에는 프로그래밍 방식으로 사용하기 위한 기계가 판독할 수 있는 필드와 사용자에게 표시하기 위해 사람이 읽을 수 있는 필드라는 두 가지 유형의 필드가 포함되어 있습니다.

기계가 판독할 수 있는 필드

이러한 필드는 표준화된 형식을 사용하며 스크립트, 패키지 관리자 및 기타 자동화된 도구로 처리하기 위한 것입니다. 소문자, 숫자 및 제한된 구두점(마침표, 밑줄 및 하이픈)만 포함합니다.

- ID - 운영 체제 식별자입니다. Amazon Linux는 모든 버전에서 amzn을 사용하여 Debian(debian), Ubuntu(ubuntu) 또는 Fedora(fedora)와 같은 다른 배포판과 구별합니다.
- VERSION_ID - 프로그래밍 방식 사용을 위한 운영 체제 버전(예: 2023)
- ID_LIKE - 관련 배포판의 공백으로 구분된 목록(예: fedora)
- VERSION_CODENAME - 스크립트의 릴리스 코드 이름(예: karoo)
- VARIANT_ID - 프로그래밍 방식의 결정을 위한 변형 식별자
- BUILD_ID - 시스템 이미지의 빌드 식별자
- IMAGE_ID - 컨테이너화된 환경의 이미지 식별자

- PLATFORM_ID – 플랫폼 식별자(예: platform:al2023)

사람이 읽을 수 있는 필드

이러한 필드는 사용자에게 표시하기 위한 것이며 공백, 대소문자 혼합 및 설명 텍스트를 포함할 수 있습니다. 사용자 인터페이스에 운영 체제 정보를 제공할 때 사용해야 합니다.

- NAME - 표시할 운영 체제 이름(예: Amazon Linux)
- PRETTY_NAME - 표시할 버전이 있는 전체 운영 체제 이름(예: Amazon Linux 2023.8.20250721)
- VERSION - 사용자 프레젠테이션에 적합한 버전 정보
- VARIANT - 표시할 변형 또는 에디션 이름(예: Server Edition)

기타 정보 필드

이러한 필드는 운영 체제에 대한 추가 메타데이터를 제공합니다.

- HOME_URL - 프로젝트 홈페이지 URL
- DOCUMENTATION_URL - 설명서 URL
- SUPPORT_URL – 지원 정보 URL
- BUG_REPORT_URL - 버그 보고 URL
- VENDOR_NAME - 공급업체 이름
- VENDOR_URL - 공급업체 URL
- SUPPORT_END - YYYY-MM-DD 형식의 지원 종료일
- CPE_NAME - 공통 플랫폼 열거 식별자
- ANSI_COLOR - 터미널 표시용 ANSI 색상 코드

Amazon Linux를 프로그래밍 방식으로 식별해야 하는 스크립트 또는 애플리케이션을 작성할 때는 ID 및 VERSION_ID 같이 기계가 판독할 수 있는 필드를 사용합니다. 운영 체제 정보를 사용자에게 표시할 때는 PRETTY_NAME 같이 사람이 읽을 수 있는 필드를 사용합니다.

/etc/os-release 예제

/etc/os-release 파일 콘텐츠는 Amazon Linux 버전마다 다릅니다.

AL2023

```
[ec2-user ~]$ cat /etc/os-release
```

```
NAME="Amazon Linux"
VERSION="2023"
ID="amzn"
ID_LIKE="fedora"
VERSION_ID="2023"
PLATFORM_ID="platform:al2023"
PRETTY_NAME="Amazon Linux 2023.8.20250721"
ANSI_COLOR="0;33"
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2023"
HOME_URL="https://aws.amazon.com/linux/amazon-linux-2023/"
DOCUMENTATION_URL="https://docs.aws.amazon.com/linux/"
SUPPORT_URL="https://aws.amazon.com/premiumsupport/"
BUG_REPORT_URL="https://github.com/amazonlinux/amazon-linux-2023"
VENDOR_NAME="AWS"
VENDOR_URL="https://aws.amazon.com/"
SUPPORT_END="2029-06-30"
```

AL2

```
[ec2-user ~]$ cat /etc/os-release
```

```
NAME="Amazon Linux"
VERSION="2"
ID="amzn"
ID_LIKE="centos rhel fedora"
VERSION_ID="2"
PRETTY_NAME="Amazon Linux 2"
ANSI_COLOR="0;33"
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2"
HOME_URL="https://amazonlinux.com/"
SUPPORT_END="2026-06-30"
```

Amazon Linux AMI

```
[ec2-user ~]$ cat /etc/os-release
```

```
NAME="Amazon Linux AMI"
```

```
VERSION="2018.03"
ID="amzn"
ID_LIKE="rhel fedora"
VERSION_ID="2018.03"
PRETTY_NAME="Amazon Linux AMI 2018.03"
ANSI_COLOR="0;33"
CPE_NAME="cpe:/o:amazon:linux:2018.03:ga"
HOME_URL="http://aws.amazon.com/amazon-linux-ami/"
```

다른 배포판과 비교

Amazon Linux가 더 광범위한 Linux 에코시스템에 어떻게 적합한지 이해하려면 `/etc/os-release` 형식을 다른 주요 배포판과 비교하세요.

Fedora

```
[ec2-user ~]$ cat /etc/os-release
```

```
NAME="Fedora Linux"
VERSION="42 (Container Image)"
RELEASE_TYPE=stable
ID=fedora
VERSION_ID=42
VERSION_CODENAME=""
PLATFORM_ID="platform:f42"
PRETTY_NAME="Fedora Linux 42 (Container Image)"
ANSI_COLOR="0;38;2;60;110;180"
LOGO=fedora-logo-icon
CPE_NAME="cpe:/o:fedoraproject:fedora:42"
DEFAULT_HOSTNAME="fedora"
HOME_URL="https://fedoraproject.org/"
DOCUMENTATION_URL="https://docs.fedoraproject.org/en-US/fedora/f42/system-
administrators-guide/"
SUPPORT_URL="https://ask.fedoraproject.org/"
BUG_REPORT_URL="https://bugzilla.redhat.com/"
REDHAT_BUGZILLA_PRODUCT="Fedora"
REDHAT_BUGZILLA_PRODUCT_VERSION=42
REDHAT_SUPPORT_PRODUCT="Fedora"
REDHAT_SUPPORT_PRODUCT_VERSION=42
SUPPORT_END=2026-05-13
VARIANT="Container Image"
```

```
VARIANT_ID=container
```

Debian

```
[ec2-user ~]$ cat /etc/os-release
```

```
PRETTY_NAME="Debian GNU/Linux 12 (bookworm)"
NAME="Debian GNU/Linux"
VERSION_ID="12"
VERSION="12 (bookworm)"
VERSION_CODENAME=bookworm
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
```

Ubuntu

```
[ec2-user ~]$ cat /etc/os-release
```

```
PRETTY_NAME="Ubuntu 24.04.2 LTS"
NAME="Ubuntu"
VERSION_ID="24.04"
VERSION="24.04.2 LTS (Noble Numbat)"
VERSION_CODENAME=noble
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=noble
LOGO=ubuntu-logo
```

기계가 판독할 수 있는 필드가 배포판 전반에서 일관된 식별을 제공하는 방법을 확인합니다.

- `ID` - 운영 체제 고유 식별: Amazon Linux의 경우 `amzn`, Fedora의 경우 `fedora`, Debian의 경우 `debian`, Ubuntu의 경우 `ubuntu`
- `ID_LIKE` - 배포 관계 표시: Amazon Linux는 `fedora`(AL2023) 또는 `centos rhel fedora`(AL2)를 사용하고 Ubuntu는 Debian의 전통을 나타내는 `debian`를 표시

- VERSION_ID - 기계 구문 분석 가능한 버전 정보 제공: AL2023의 경우 2023, Fedora의 경우 42, Debian의 경우 12, Ubuntu의 경우 24.04

반면 사람이 읽을 수 있는 필드는 사용자에게 표시되도록 설계되었습니다.

- NAME – 사용자 친화적인 OS 이름: Amazon Linux, Fedora Linux, Debian GNU/Linux, Ubuntu
- PRETTY_NAME – 버전이 다음에 해당하는 전체 표시 이름: Amazon Linux 2023.8.20250721, Fedora Linux 42 (Container Image), Debian GNU/Linux 12 (bookworm), Ubuntu 24.04.2 LTS
- VERSION - 코드 이름 또는 릴리스 유형과 같은 추가 컨텍스트가 있는 사람이 읽을 수 있는 버전

교차 플랫폼 스크립트를 작성할 때는 항상 로직 및 결정에 기계가 판독할 수 있는 필드(ID, VERSION_ID, ID_LIKE)를 사용하고 사용자에게 정보를 표시할 때만 사람이 읽을 수 있는 필드(PRETTY_NAME, NAME)를 사용합니다.

Amazon Linux 특정

Amazon Linux와 해당 버전을 식별하는 데 사용할 수 있는 Amazon Linux와 관련된 일부 파일이 있습니다. 새 코드는 배포판 간 호환을 위해 /etc/os-release 표준을 사용해야 합니다. Amazon Linux 특정 파일은 사용하지 않는 것이 좋습니다.

주제

- /etc/system-release 파일
- 이미지 식별 파일
- Amazon Linux 특정 파일의 예제

/etc/system-release 파일

Amazon Linux에는 설치된 현재 릴리스를 지정하는 /etc/system-release 파일이 포함되어 있습니다. 이 파일은 패키지 관리자를 사용하여 업데이트되며 Amazon Linux는 system-release 패키지의 일부입니다. Fedora와 같은 일부 다른 배포판에도 이 파일이 있지만 Ubuntu와 같은 Debian 기반 배포판에는 없습니다.

> **Note**
>
> `/etc/system-release` 파일에는 사람이 읽을 수 있는 문자열이 포함되어 있으므로 OS 또는 릴리스를 식별하는 데 프로그래밍 방식으로 사용해서는 안 됩니다. 대신 `/etc/os-release`의 기계가 판독할 수 있는 필드(또는 `/etc/os-release`가 존재하지 않는 경우 `/usr/lib/os-release`)를 사용합니다.

Amazon Linux에는 `/etc/system-release`의 기계가 판독할 수 있는 버전도 포함되어 있으며 `/etc/system-release-cpe` 파일의 CPE(Common Platform Enumeration) 사양을 따릅니다.

이미지 식별 파일

각 Amazon Linux 이미지에는 Amazon Linux 팀에서 생성한 원본 이미지에 대한 추가 정보를 제공하는 고유한 `/etc/image-id` 파일이 포함되어 있습니다. 이 파일은 Amazon Linux에만 해당되며 Debian, Ubuntu 또는 Fedora와 같은 다른 Linux 배포판에서는 찾을 수 없습니다. 이 파일에는 이미지에 대한 다음 정보가 포함되어 있습니다.

- `image_name`, `image_version`, `image_arch` - 이미지 생성에 사용한 빌드 레시피의 값입니다.
- `image_stamp` - 이미지 생성 중에 생성된 고유한 임의 16진수 값입니다.
- `image_date` - YYYYMMDDhhmmss 형식으로 표시되는 이미지 생성 시간(UTC)입니다.
- `recipe_name`, `recipe_id` - 이미지 생성에 사용한 빌드 레시피의 이름과 ID입니다.

Amazon Linux 특정 파일의 예제

다음 섹션에서는 Amazon Linux의 각 메이저 버전에 대한 Amazon Linux별 식별 파일의 예제를 제공합니다.

> **Note**
>
> 실제 코드에서 `/etc/os-release` 파일이 없는 경우 `/usr/lib/os-release`를 사용해야 합니다.

AL2023

다음 예제에서는 AL2023의 식별 파일을 보여줍니다.

AL2023용 /etc/image-id의 예제:

```
[ec2-user ~]$ cat /etc/image-id
```

```
image_name="al2023-container"
image_version="2023"
image_arch="x86_64"
image_file="al2023-container-2023.8.20250721.2-x86_64"
image_stamp="822b-1a9e"
image_date="20250719211531"
recipe_name="al2023 container"
recipe_id="89b25f7b-be82-2215-a8eb-6e63-0830-94ea-658d41c4"
```

AL2023용 /etc/system-release의 예제:

```
[ec2-user ~]$ cat /etc/system-release
```

```
Amazon Linux release 2023.8.20250721 (Amazon Linux)
```

AL2

다음 예제에서는 AL2의 식별 파일을 보여줍니다.

AL2용 /etc/image-id의 예제:

```
[ec2-user ~]$ cat /etc/image-id
```

```
image_name="amzn2-container-raw"
image_version="2"
image_arch="x86_64"
image_file="amzn2-container-raw-2.0.20250721.2-x86_64"
image_stamp="4126-16ad"
image_date="20250721225801"
recipe_name="amzn2 container"
recipe_id="948422df-a4e6-5fc8-ba89-ef2e-0e1f-e1bb-16f84087"
```

AL2용 /etc/system-release의 예제:

```
[ec2-user ~]$ cat /etc/system-release
```

```
Amazon Linux release 2 (Karoo)
```

Amazon Linux AMI

다음 예제에서는 Amazon Linux AMI의 식별 파일을 보여줍니다.

Amazon Linux AMI용 `/etc/image-id`의 예제:

```
[ec2-user ~]$ cat /etc/image-id
```

```
image_name="amzn-container-minimal"
image_version="2018.03"
image_arch="x86_64"
image_file="amzn-container-minimal-2018.03.0.20231218.0-x86_64"
image_stamp="407d-5ef3"
image_date="20231218203210"
recipe_name="amzn container"
recipe_id="b1e7635e-14e3-dd57-b1ab-7351-edd0-d9e0-ca6852ea"
```

Amazon Linux AMI용 `/etc/system-release`의 예제:

```
[ec2-user ~]$ cat /etc/system-release
```

```
Amazon Linux AMI release 2018.03
```

OS 감지를 위한 코드 예제

다음 예제에서는 `/etc/os-release`(또는 `/etc/os-release`가 존재하지 않는 경우 `/usr/lib/os-release`) 파일을 사용하여 운영 체제 및 버전을 프로그래밍 방식으로 감지하는 방법을 보여줍니다. 이 예제에서는 Amazon Linux와 다른 배포를 구분하는 방법과 `ID_LIKE` 필드를 사용하여 배포판 제품군을 결정하는 방법을 보여줍니다.

아래 스크립트는 여러 프로그래밍 언어로 구현되며 각 구현은 동일한 출력을 생성합니다.

Shell

```
#!/bin/bash

# Function to get a specific field from os-release file
```

```
get_os_release_field() {
    local field="$1"
    local os_release_file

    # Find the os-release file
    if [ -f /etc/os-release ]; then
        os_release_file='/etc/os-release'
    elif [ -f /usr/lib/os-release ]; then
        os_release_file='/usr/lib/os-release'
    else
        echo "Error: os-release file not found" >&2
        return 1
    fi

    # Source the file in a subshell and return the requested field.
    #
    # A subshell means that variables from os-release are only available
    # within the subshell, and the main script environment remains clean.
    (
        . "$os_release_file"
        eval "echo \"\$$field\""
    )
}

is_amazon_linux() {
    [ "$(get_os_release_field ID)" = "amzn" ]
}

is_fedora() {
    [ "$(get_os_release_field ID)" = "fedora" ]
}

is_ubuntu() {
    [ "$(get_os_release_field ID)" = "ubuntu" ]
}

is_debian() {
    [ "$(get_os_release_field ID)" = "debian" ]
}

# Function to check if this is like Fedora (includes Amazon Linux, CentOS, RHEL,
 etc.)
is_like_fedora() {
    local id="$(get_os_release_field ID)"
```

```
    local id_like="$(get_os_release_field ID_LIKE)"
    [ "$id" = "fedora" ] || [[ "$id_like" == *"fedora"* ]]
}

# Function to check if this is like Debian (includes Ubuntu and derivatives)
is_like_debian() {
    local id="$(get_os_release_field ID)"
    local id_like="$(get_os_release_field ID_LIKE)"
    [ "$id" = "debian" ] || [[ "$id_like" == *"debian"* ]]
}

# Get the main fields we'll use multiple times
ID="$(get_os_release_field ID)"
VERSION_ID="$(get_os_release_field VERSION_ID)"
PRETTY_NAME="$(get_os_release_field PRETTY_NAME)"
ID_LIKE="$(get_os_release_field ID_LIKE)"

echo "Operating System Detection Results:"
echo "==================================="
echo "Is Amazon Linux: $(is_amazon_linux && echo YES || echo NO)"
echo "Is Fedora: $(is_fedora && echo YES || echo NO)"
echo "Is Ubuntu: $(is_ubuntu && echo YES || echo NO)"
echo "Is Debian: $(is_debian && echo YES || echo NO)"
echo "Is like Fedora: $(is_like_fedora && echo YES || echo NO)"
echo "Is like Debian: $(is_like_debian && echo YES || echo NO)"
echo
echo "Detailed OS Information:"
echo "========================"
echo "ID: $ID"
echo "VERSION_ID: $VERSION_ID"
echo "PRETTY_NAME: $PRETTY_NAME"
[ -n "$ID_LIKE" ] && echo "ID_LIKE: $ID_LIKE"

# Amazon Linux specific information
if is_amazon_linux; then
    echo ""
    echo "Amazon Linux Version Details:"
    echo "============================="
    case "$VERSION_ID" in
        2018.03)
            echo "Amazon Linux AMI (version 1)"
            ;;
        2)
            echo "Amazon Linux 2"
```

```
            ;;
        2023)
            echo "Amazon Linux 2023"
            ;;
        *)
            echo "Unknown Amazon Linux version: $VERSION_ID"
            ;;
    esac

    # Check for Amazon Linux specific files
    [ -f /etc/image-id ] && echo "Amazon Linux image-id file present"
fi
```

Python 3.7-3.9

```
#!/usr/bin/env python3

import os
import sys

def parse_os_release():
    """Parse the os-release file and return a dictionary of key-value pairs."""
    os_release_data = {}

    # Try /etc/os-release first, then /usr/lib/os-release
    for path in ['/etc/os-release', '/usr/lib/os-release']:
        if os.path.exists(path):
            try:
                with open(path, 'r') as f:
                    for line in f:
                        line = line.strip()
                        if line and not line.startswith('#') and '=' in line:
                            key, value = line.split('=', 1)
                            # Remove quotes if present
                            value = value.strip('"\'')
                            os_release_data[key] = value
                return os_release_data
            except IOError:
                continue

    print("Error: os-release file not found")
    sys.exit(1)
```

```
def is_amazon_linux(os_data):
    """Check if this is Amazon Linux."""
    return os_data.get('ID') == 'amzn'

def is_fedora(os_data):
    """Check if this is Fedora."""
    return os_data.get('ID') == 'fedora'

def is_ubuntu(os_data):
    """Check if this is Ubuntu."""
    return os_data.get('ID') == 'ubuntu'

def is_debian(os_data):
    """Check if this is Debian."""
    return os_data.get('ID') == 'debian'

def is_like_fedora(os_data):
    """Check if this is like Fedora (includes Amazon Linux, CentOS, RHEL, etc.)."""
    if os_data.get('ID') == 'fedora':
        return True
    id_like = os_data.get('ID_LIKE', '')
    return 'fedora' in id_like

def is_like_debian(os_data):
    """Check if this is like Debian (includes Ubuntu and derivatives)."""
    if os_data.get('ID') == 'debian':
        return True
    id_like = os_data.get('ID_LIKE', '')
    return 'debian' in id_like

def main():
    # Parse os-release file
    os_data = parse_os_release()

    # Display results
    print("Operating System Detection Results:")
    print("==================================")
    print(f"Is Amazon Linux: {'YES' if is_amazon_linux(os_data) else 'NO'}")
    print(f"Is Fedora: {'YES' if is_fedora(os_data) else 'NO'}")
    print(f"Is Ubuntu: {'YES' if is_ubuntu(os_data) else 'NO'}")
    print(f"Is Debian: {'YES' if is_debian(os_data) else 'NO'}")
    print(f"Is like Fedora: {'YES' if is_like_fedora(os_data) else 'NO'}")
    print(f"Is like Debian: {'YES' if is_like_debian(os_data) else 'NO'}")
```

```
    # Additional information
    print()
    print("Detailed OS Information:")
    print("======================")
    print(f"ID: {os_data.get('ID', '')}")
    print(f"VERSION_ID: {os_data.get('VERSION_ID', '')}")
    print(f"PRETTY_NAME: {os_data.get('PRETTY_NAME', '')}")
    if os_data.get('ID_LIKE'):
        print(f"ID_LIKE: {os_data.get('ID_LIKE')}")

    # Amazon Linux specific information
    if is_amazon_linux(os_data):
        print()
        print("Amazon Linux Version Details:")
        print("============================")
        version_id = os_data.get('VERSION_ID', '')
        if version_id == '2018.03':
            print("Amazon Linux AMI (version 1)")
        elif version_id == '2':
            print("Amazon Linux 2")
        elif version_id == '2023':
            print("Amazon Linux 2023")
        else:
            print(f"Unknown Amazon Linux version: {version_id}")

        # Check for Amazon Linux specific files
        if os.path.exists('/etc/image-id'):
            print("Amazon Linux image-id file present")

if __name__ == '__main__':
    main()
```

Python 3.10+

```
#!/usr/bin/env python3

import os
import sys
import platform

def is_amazon_linux(os_data):
    """Check if this is Amazon Linux."""
    return os_data.get('ID') == 'amzn'
```

```
def is_fedora(os_data):
    """Check if this is Fedora."""
    return os_data.get('ID') == 'fedora'

def is_ubuntu(os_data):
    """Check if this is Ubuntu."""
    return os_data.get('ID') == 'ubuntu'

def is_debian(os_data):
    """Check if this is Debian."""
    return os_data.get('ID') == 'debian'

def is_like_fedora(os_data):
    """Check if this is like Fedora (includes Amazon Linux, CentOS, RHEL, etc.)."""
    if os_data.get('ID') == 'fedora':
        return True
    id_like = os_data.get('ID_LIKE', '')
    return 'fedora' in id_like

def is_like_debian(os_data):
    """Check if this is like Debian (includes Ubuntu and derivatives)."""
    if os_data.get('ID') == 'debian':
        return True
    id_like = os_data.get('ID_LIKE', '')
    return 'debian' in id_like

def main():
    # Parse os-release file using the standard library function (Python 3.10+)
    try:
        os_data = platform.freedesktop_os_release()
    except OSError:
        print("Error: os-release file not found")
        sys.exit(1)

    # Display results
    print("Operating System Detection Results:")
    print("==================================")
    print(f"Is Amazon Linux: {'YES' if is_amazon_linux(os_data) else 'NO'}")
    print(f"Is Fedora: {'YES' if is_fedora(os_data) else 'NO'}")
    print(f"Is Ubuntu: {'YES' if is_ubuntu(os_data) else 'NO'}")
    print(f"Is Debian: {'YES' if is_debian(os_data) else 'NO'}")
    print(f"Is like Fedora: {'YES' if is_like_fedora(os_data) else 'NO'}")
    print(f"Is like Debian: {'YES' if is_like_debian(os_data) else 'NO'}")
```

```
    # Additional information
    print()
    print("Detailed OS Information:")
    print("======================")
    print(f"ID: {os_data.get('ID', '')}")
    print(f"VERSION_ID: {os_data.get('VERSION_ID', '')}")
    print(f"PRETTY_NAME: {os_data.get('PRETTY_NAME', '')}")
    if os_data.get('ID_LIKE'):
        print(f"ID_LIKE: {os_data.get('ID_LIKE')}")

    # Amazon Linux specific information
    if is_amazon_linux(os_data):
        print()
        print("Amazon Linux Version Details:")
        print("============================")
        version_id = os_data.get('VERSION_ID', '')
        if version_id == '2018.03':
            print("Amazon Linux AMI (version 1)")
        elif version_id == '2':
            print("Amazon Linux 2")
        elif version_id == '2023':
            print("Amazon Linux 2023")
        else:
            print(f"Unknown Amazon Linux version: {version_id}")

        # Check for Amazon Linux specific files
        if os.path.exists('/etc/image-id'):
            print("Amazon Linux image-id file present")

if __name__ == '__main__':
    main()
```

Perl

```
#!/usr/bin/env perl

use strict;
use warnings;

# Function to parse the os-release file and return a hash of key-value pairs
sub parse_os_release {
    my %os_release_data;
```

```
    # Try /etc/os-release first, then /usr/lib/os-release
    my @paths = ('/etc/os-release', '/usr/lib/os-release');

    for my $path (@paths) {
        if (-f $path) {
            if (open(my $fh, '<', $path)) {
                while (my $line = <$fh>) {
                    chomp $line;
                    next if $line =~ /^\s*$/ || $line =~ /^\s*#/;

                    if ($line =~ /^([^=]+)=(.*)$/) {
                        my ($key, $value) = ($1, $2);
                        # Remove quotes if present
                        $value =~ s/^["']|["']$//g;
                        $os_release_data{$key} = $value;
                    }
                }
                close($fh);
                return %os_release_data;
            }
        }
    }

    die "Error: os-release file not found\n";
}

# Function to check if this is Amazon Linux
sub is_amazon_linux {
    my %os_data = @_;
    return ($os_data{ID} // '') eq 'amzn';
}

# Function to check if this is Fedora
sub is_fedora {
    my %os_data = @_;
    return ($os_data{ID} // '') eq 'fedora';
}

# Function to check if this is Ubuntu
sub is_ubuntu {
    my %os_data = @_;
    return ($os_data{ID} // '') eq 'ubuntu';
}
```

```
# Function to check if this is Debian
sub is_debian {
    my %os_data = @_;
    return ($os_data{ID} // '') eq 'debian';
}

# Function to check if this is like Fedora (includes Amazon Linux, CentOS, RHEL,
 etc.)
sub is_like_fedora {
    my %os_data = @_;
    return 1 if ($os_data{ID} // '') eq 'fedora';
    my $id_like = $os_data{ID_LIKE} // '';
    return $id_like =~ /fedora/;
}

# Function to check if this is like Debian (includes Ubuntu and derivatives)
sub is_like_debian {
    my %os_data = @_;
    return 1 if ($os_data{ID} // '') eq 'debian';
    my $id_like = $os_data{ID_LIKE} // '';
    return $id_like =~ /debian/;
}

# Main execution
my %os_data = parse_os_release();

# Display results
print "Operating System Detection Results:\n";
print "==================================\n";
print "Is Amazon Linux: " . (is_amazon_linux(%os_data) ? "YES" : "NO") . "\n";
print "Is Fedora: " . (is_fedora(%os_data) ? "YES" : "NO") . "\n";
print "Is Ubuntu: " . (is_ubuntu(%os_data) ? "YES" : "NO") . "\n";
print "Is Debian: " . (is_debian(%os_data) ? "YES" : "NO") . "\n";
print "Is like Fedora: " . (is_like_fedora(%os_data) ? "YES" : "NO") . "\n";
print "Is like Debian: " . (is_like_debian(%os_data) ? "YES" : "NO") . "\n";
print "\n";

# Additional information
print "Detailed OS Information:\n";
print "========================\n";
print "ID: " . ($os_data{ID} // '') . "\n";
print "VERSION_ID: " . ($os_data{VERSION_ID} // '') . "\n";
print "PRETTY_NAME: " . ($os_data{PRETTY_NAME} // '') . "\n";
```

```
print "ID_LIKE: " . ($os_data{ID_LIKE} // '') . "\n" if $os_data{ID_LIKE};

# Amazon Linux specific information
if (is_amazon_linux(%os_data)) {
    print "\n";
    print "Amazon Linux Version Details:\n";
    print "============================\n";
    my $version_id = $os_data{VERSION_ID} // '';

    if ($version_id eq '2018.03') {
        print "Amazon Linux AMI (version 1)\n";
    } elsif ($version_id eq '2') {
        print "Amazon Linux 2\n";
    } elsif ($version_id eq '2023') {
        print "Amazon Linux 2023\n";
    } else {
        print "Unknown Amazon Linux version: $version_id\n";
    }

    # Check for Amazon Linux specific files
    if (-f '/etc/image-id') {
        print "Amazon Linux image-id file present\n";
    }
}
```

다른 시스템에서 실행하면 스크립트는 다음 출력을 생성합니다.

AL2023

```
Operating System Detection Results:
===================================
Is Amazon Linux: YES
Is Fedora: NO
Is Ubuntu: NO
Is Debian: NO
Is like Fedora: YES
Is like Debian: NO

Detailed OS Information:
========================
ID: amzn
VERSION_ID: 2023
```

```
PRETTY_NAME: Amazon Linux 2023.8.20250721
ID_LIKE: fedora

Amazon Linux Version Details:
==============================
Amazon Linux 2023
Amazon Linux image-id file present
```

AL2

```
Operating System Detection Results:
===================================
Is Amazon Linux: YES
Is Fedora: NO
Is Ubuntu: NO
Is Debian: NO
Is like Fedora: YES
Is like Debian: NO

Detailed OS Information:
========================
ID: amzn
VERSION_ID: 2
PRETTY_NAME: Amazon Linux 2
ID_LIKE: centos rhel fedora

Amazon Linux Version Details:
==============================
Amazon Linux 2
Amazon Linux image-id file present
```

Amazon Linux AMI

```
Operating System Detection Results:
===================================
Is Amazon Linux: YES
Is Fedora: NO
Is Ubuntu: NO
Is Debian: NO
Is like Fedora: YES
Is like Debian: NO

Detailed OS Information:
```

```
=======================
ID: amzn
VERSION_ID: 2018.03
PRETTY_NAME: Amazon Linux AMI 2018.03
ID_LIKE: rhel fedora

Amazon Linux Version Details:
============================
Amazon Linux AMI (version 1)
Amazon Linux image-id file present
```

Ubuntu

```
Operating System Detection Results:
==================================
Is Amazon Linux: NO
Is Fedora: NO
Is Ubuntu: YES
Is Debian: NO
Is like Fedora: NO
Is like Debian: YES

Detailed OS Information:
=======================
ID: ubuntu
VERSION_ID: 24.04
PRETTY_NAME: Ubuntu 24.04.2 LTS
ID_LIKE: debian
```

Debian

```
Operating System Detection Results:
==================================
Is Amazon Linux: NO
Is Fedora: NO
Is Ubuntu: NO
Is Debian: YES
Is like Fedora: NO
Is like Debian: YES

Detailed OS Information:
=======================
ID: debian
```

```
VERSION_ID: 12
PRETTY_NAME: Debian GNU/Linux 12 (bookworm)
```

Fedora

```
Operating System Detection Results:
==================================
Is Amazon Linux: NO
Is Fedora: YES
Is Ubuntu: NO
Is Debian: NO
Is like Fedora: YES
Is like Debian: NO

Detailed OS Information:
=======================
ID: fedora
VERSION_ID: 42
PRETTY_NAME: Fedora Linux 42 (Container Image)
```

AWSAL2의 통합

AWS명령줄 도구

AWS Command Line Interface(AWS CLI)는 명령줄 셸에서 명령을 AWS 서비스사용하여와 상호 작용할 수 있는 일관된 인터페이스를 제공하는 오픈 소스 도구입니다. 자세한 내용은 AWS Command Line Interface사용 설명서의 란 무엇입니까AWS Command Line Interface?를 참조하세요.

AL2 및 AL1에는의 버전 1이 AWS CLI사전 설치되어 있습니다. Amazon Linux의 현재 릴리스인 AL2023에는의 버전 2가 AWS CLI사전 설치되어 있습니다. AL2023AWS CLI에서를 사용하는 방법에 대한 자세한 내용은 Amazon Linux 2023 사용 설명서의 AL2023 시작하기를 참조하세요.

프로그래밍 런타임 시작하기

AL2는 특정 언어 런타임의 다양한 버전을 제공합니다. 여러 버전을 동시에 지원하는 PHP와 같은 업스트림 프로젝트로 작업합니다. 이름 버전이 지정된 패키지를 설치하고 관리하는 방법에 대한 정보를 찾으려면 yum 명령을 사용하여 이러한 패키지를 검색하고 설치합니다. 자세한 내용은 패키지 리포지토리 단원을 참조하십시오.

다음 주제에서는 AL2에서 각 언어 런타임이 작동하는 방식을 설명합니다.

주제

- CAL2Fortran의 C++, 및
- AL2로 이동
- Java AL2의
- Perl AL2의
- PHP AL2의
- Python AL2의
- AL2의 Rust

CAL2Fortran의 C++, 및

AL2에는 GNU 컴파일러 컬렉션(GCC)과의 Clang 프런트엔드가 모두 포함됩니다LLVM.

의 메이저 버전은 AL2 수명 기간 동안 일정하게 GCC 유지됩니다. 버그 및 보안 수정 사항은 AL2에 GCC 있는의 메이저 버전으로 백포트될 수 있습니다.

기본적으로 AL2에는 거의 모든 패키지를 빌드GCC하는 버전 7.3이 포함되어 있습니다. `gcc10` 패키지는 제한된 범위에서 GCC 10을 사용할 수 있지만 패키지를 빌드하는 데 GCC 10을 사용하지 않는 것이 좋습니다.

AL2 RPMs을 빌드하는 기본 컴파일러 플래그에는 일부 최적화 및 강화 플래그가 포함됩니다. 를 사용하여 자체 코드를 빌드하는 경우 일부 최적화 및 강화 플래그를 포함하는 것이 좋습니다GCC.

AL2023의 기본 컴파일러 및 최적화 플래그는 AL2에 있는 기능을 개선합니다.

AL2로 이동

AL2와 함께 제공되는 도구 체인을 사용하여 Amazon LinuxGo에서에 작성된 자체 코드를 빌드할 수 있습니다.

Go 도구 체인은 AL2 수명 주기 동안 업데이트됩니다. 이는 배송하는 도구 체인의 CVE에 대한 응답이거나 다른 패키지의 CVE를 해결하기 위한 사전 조건일 수 있습니다.

Go는 비교적 빠르게 움직이는 프로그래밍 언어입니다. Go에 작성된 기존 애플리케이션이 새 버전의 Go 도구 체인에 적응해야 하는 상황이 있을 수 있습니다. Go에 대한 자세한 내용은 Go 1 및 Go 프로그램의 미래 섹션을 참조하세요.

AL2는 수명 주기 동안 새로운 버전의 Go 도구 체인을 통합하지만 업스트림 Go 릴리스에서는 잠기지 않습니다. 따라서 Go 언어 및 표준 라이브러리의 최첨단 기능을 사용하여 Go 코드를 빌드하려는 경우 AL2에 제공된 Go 도구 체인을 사용하는 것이 적합하지 않을 수 있습니다.

AL2 수명 동안 이전 패키지 버전은 리포지토리에서 제거되지 않습니다. 이전 Go 도구 체인이 필요한 경우 최신 Go 도구 체인의 버그 및 보안 수정을 포기하고 RPM에 사용할 수 있는 것과 동일한 메커니즘을 사용하여 리포지토리에서 이전 버전을 설치하도록 선택할 수 있습니다.

AL2에서 자체 Go 코드를 빌드하려면이 Go 도구 체인이 AL2 수명 주기 동안 진행될 수 있다는 사실을 알고 AL2에 포함된 도구 체인을 사용할 수 있습니다.

Java AL2의

AL2는 Java 기반 워크로드를 지원하는 여러 버전의 Amazon Corretto와 일부 OpenJDK 버전을 제공합니다. AL2023으로 마이그레이션하기 위해 Amazon Corretto로 마이그레이션하는 것이 좋습니다.

Corretto는 Amazon의 장기 지원을 받는 공개 Java 개발 키트(OpenJDK)입니다. Corretto는 Java SE 표준을 충족하고 Linux, Windows및에서 사용할 수 있도록 Java 기술 호환성 키트(TCK)를 사용하여 인증을 받았습니다macOS.

Amazon Corretto 패키지는 각 Corretto 1.8.0, Corretto 11 및 Corretto 17에 사용할 수 있습니다.

AL2의 각 Corretto 버전은 Corretto 버전과 동일한 기간 동안 또는 AL2 수명 종료 중 먼저 발생하는 시점까지 지원됩니다. 자세한 내용은 Amazon Corretto FAQs.

Perl AL2의

AL2는 Perl 프로그래밍 언어 버전 5.16을 제공합니다.

Perl AL2의 모듈

다양한 Perl 모듈이 AL2에서 RPMs으로 패키징됩니다. RPMs으로 사용할 수 있는 Perl 모듈이 많지만 Amazon Linux는 가능한 모든 Perl 모듈을 패키징하려고 하지 않습니다. RPMs으로 패키징된 모듈은 다른 운영 체제 RPM 패키지에 의존할 수 있으므로 Amazon Linux는 순수 기능 업데이트보다 보안 패치를 우선시합니다.

AL2에는 Perl 개발자가 Perl 모듈에 대해 관용 패키지 관리자를 사용할 수 CPAN 있도록 도 포함되어 있습니다.

PHP AL2의

AL2는 현재의 일부로 PHP 프로그래밍 언어의 완전 지원 버전 2개를 제공합니다AL2 Extras 라이브러리. 각 PHP 버전은의 더 이상 사용되지 않는 날짜에 나열된 PHP 업스트림과 동일한 기간 동안 지원됩니다Amazon Linux 2 추가 항목 목록.

AL2 Extras를 사용하여 인스턴스에 애플리케이션 및 소프트웨어 업데이트를 설치하는 방법에 대한 자세한 내용은 섹션을 참조하세요AL2 Extras 라이브러리.

AL2023으로의 마이그레이션을 지원하기 위해 PHP AL2 및 AL2023에서 8.1 및 8.2를 모두 사용할 수 있습니다.

> **Note**
>
> AL2의 `amazon-linux-extras`에는 PHP 7.1, 7.2, 7.3 및 7.4가 포함되어 있습니다. 이러한 모든 추가 항목은 EOL이며 추가 보안 업데이트를 받을 수 없습니다.
> AL2에서 각 버전의 PHP가 더 이상 사용되지 않는 시기를 알아보려면 섹션을 참조하세요Amazon Linux 2 추가 항목 목록.

이전 PHP 8.x 버전에서 마이그레이션

업스트림 PHP 커뮤니티는 8.1에서 PHP 8.2로 전환하기 위한 포괄적인 마이그레이션 설명서를 마련 PHP했습니다. 8.0에서 PHP 8.1로 마이그레이션하기 위한 설명서도 있습니다.

AL2에는 PHP AL2023으로의 효율적인 업그레이드 경로를 `amazon-linux-extras` 지원하는 8.0, 8.1 및 8.2가 포함되어 있습니다. AL2023 AL2에서 각 버전의 PHP가 더 이상 사용되지 않는 시기를 알아보려면 섹션을 참조하세요Amazon Linux 2 추가 항목 목록.

PHP 7.x 버전에서 마이그레이션

업스트림 PHP 커뮤니티에서 PHP 8.0에서 PHP 7.4로 이전하기 위한 마이그레이션 종합 설명서를 만들었습니다. 8.1 및 PHP 8.PHP2로 마이그레이션하는 방법에 대한 이전 섹션에서 참조한 설명서와 함께 PHP 기반 애플리케이션을 최신 로 마이그레이션하는 데 필요한 모든 단계가 있습니다PHP.

PHP 프로젝트는 지원되지 않는 브랜치 목록과 함께 지원되는 버전의 목록과 일정을 유지합니다.

> **Note**
>
> AL2023이 릴리스되었을 때의 모든 7.x 및 5.x 버전PHP은 PHP 커뮤니티에서 지원되지 않았으며 AL2023의 옵션으로 포함되지 않았습니다.

Python AL2의

AL2는 Python AL2 코어 패키지에 대한 장기 지원 약정의 일환으로 2026년 6월까지 AL2.7에 대한 지원 및 보안 패치를 제공합니다. 이 지원은 2020년 Python 1월 2.7 EOL의 업스트림 Python 커뮤니티 선언을 넘어 확장됩니다.

> **Note**
>
> AL2023은 Python 2.7을 완전히 제거했습니다. 이제가 필요한 모든 구성 요소가 Python 3과 함께 작동하도록 작성Python되었습니다.

AL2는 Python 2.7에 대한 엄격한 종속성이 있는 yum 패키지 관리자를 사용합니다. AL2023에서는 dnf 패키지 관리자가 Python 3으로 마이그레이션되었으며 더 이상 Python 2.7이 필요하지 않습니다. AL2023이 Python 3으로 완전히 이동했습니다. Python 3으로의 마이그레이션을 완료하는 것이 좋습니다.

AL2의 Rust

AL2와 함께 제공되는 도구 체인을 사용하여 AL2Rust에 작성된 자체 코드를 빌드할 수 있습니다.

Rust 도구 체인은 AL2 수명 주기 동안 업데이트됩니다. 이는 배송하는 도구 체인의 CVE에 대한 응답이거나 다른 패키지의 CVE 업데이트에 대한 사전 조건일 수 있습니다.

Rust는 비교적 빠르게 움직이는 언어로, 약 6주 주기로 새 릴리스가 출시됩니다. 새 릴리스에는 새 언어 또는 표준 라이브러리 기능이 추가될 수 있습니다. AL2는 수명 주기 동안 새로운 버전의 Rust 도구 체인을 통합하지만 업스트림 Rust 릴리스에서는 잠기지 않습니다. 따라서 Rust 언어의 최첨단 기능을 사용하여 Rust 코드를 빌드하려는 경우 AL2에 제공된 Rust 도구 체인을 사용하는 것이 적합하지 않을 수 있습니다.

AL2 수명 동안 이전 패키지 버전은 리포지토리에서 제거되지 않습니다. 이전 Rust 도구 체인이 필요한 경우 최신 Rust 도구 체인의 버그 및 보안 수정을 포기하고 RPM에 사용할 수 있는 것과 동일한 프로세스를 사용하여 리포지토리에서 이전 버전을 설치하도록 선택할 수 있습니다.

AL2에서 자체 Rust 코드를 빌드하려면이 Rust 도구 체인이 AL2 수명 주기 동안 앞으로 나아갈 수 있다는 지식과 함께 AL2에 포함된 도구 체인을 사용합니다.

AL2 커널

AL2는 원래 4.14 커널과 함께 제공되었으며 버전 5.10은 현재 기본값입니다. 여전히 4.14 커널을 사용하는 경우 5.10 커널로 마이그레이션하는 것이 좋습니다.

커널 라이브 패치는 AL2에서 지원됩니다.

주제

- AL2 지원 커널
- AL2의 커널 라이브 패치

AL2 지원 커널

지원되는 커널 버전

현재 AL2 AMIs는 커널 버전 4.14 및 5.10에서 사용할 수 있으며 버전 5.10이 기본값입니다. 커널 5.10과 함께 AL2 AMI를 사용하는 것이 좋습니다.

AL2023 AMI는 커널 버전 6.1에서 사용할 수 있습니다. 자세한 내용은 Amazon Linux AL2023 사용 설명서의 AL2에서 변경된 AL2023 커널을 참조하세요.

지원 기간

AL2에서 사용할 수 있는 5.10 커널은 AL2 AMI가 표준 지원이 종료될 때까지 지원됩니다.

라이브 패치 지원

AL2 커널 버전	커널 라이브 패치 지원
4.14	예
5.10	예
5.15	아니요

AL2의 커널 라이브 패치

> **Important**
>
> Amazon Linux는 2025-10-31에 AL2 커널 4.14에 대한 라이브 패치를 종료합니다. 커널 5.10을 AL2의 기본 커널로 사용하거나(AL2 지원 커널 참조) 커널 6.1 및 6.12를 사용하여 AL2023으로 전환하는 것이 좋습니다.
> Amazon Linux는 2026-06-30에 AL2 수명이 끝날 때까지 AL2 커널 5.10에 대한 라이브 패치를 제공합니다.

AL2용 커널 라이브 패치를 사용하면 애플리케이션을 재부팅하거나 중단하지 않고도 실행 중인 Linux 커널에 특정 보안 취약성 및 중요한 버그 패치를 적용할 수 있습니다. 이를 통해 서비스 및 애플리케이션 가용성을 개선하는 동시에 시스템을 재부팅할 수 있을 때까지 이러한 수정 사항을 적용할 수 있습니다.

AL2023용 커널 라이브 패치에 대한 자세한 내용은 Amazon Linux 2023 사용 설명서의 AL2023의 커널 라이브 패치를 참조하세요.

AWS 는 AL2에 대해 두 가지 유형의 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 따라는 CVE가 할당되기 전에 업데이트를 제공할 수 AWS 있습니다. 이러한 경우 버그 수정 패치를 배포합니다.
- 버그 수정 - CVE와 관련되지 않은 중요한 버그 및 안정성 문제에 대한 수정 사항이 포함되어 있습니다.

AWS 는 릴리스 후 최대 3개월 동안 AL2 커널 버전에 대한 커널 라이브 패치를 제공합니다. 3개월 후 커널 라이브 패치를 계속 받으려면 최신 커널 버전으로 업데이트해야 합니다.

AL2 커널 라이브 패치는 기존 AL2 리포지토리에서 서명된 RPM 패키지로 제공됩니다. 패치는 기존 yum 워크플로를 사용하여 개별 인스턴스에 설치하거나 AWS Systems Manager를 사용하여 관리형 인스턴스 그룹에 설치할 수 있습니다.

AL2의 커널 라이브 패치는 추가 비용 없이 제공됩니다.

주제

지원되는 구성 및 필수 조건

커널 라이브 패치는 AL2를 실행하는 Amazon EC2 인스턴스 및 온프레미스 가상 머신에서 지원됩니다. AL2

AL2에서 커널 라이브 패치를 사용하려면 다음을 사용해야 합니다.

- `x86_64` 아키텍처의 커널 버전 `4.14` 또는 `5.10`
- `ARM64` 아키텍처의 커널 버전 `5.10`

정책 요구 사항

Amazon Linux 리포지토리에서 패키지를 다운로드하려면 Amazon EC2가 서비스 소유 Amazon S3 버킷에 액세스해야 합니다. 환경에서 Amazon S3용 Amazon Virtual Private Cloud(VPC) 엔드포인트를 사용하는 경우 VPC 엔드포인트 정책이 해당 퍼블릭 버킷에 대한 액세스를 허용하는지 확인해야 합니다.

이 표에서는 커널 라이브 패치를 위해 EC2가 액세스해야 할 수 있는 각 Amazon S3 버킷에 대해 설명합니다.

S3 버킷 ARN	설명
arn:aws:s3:::packages.*region*.amazonaws.com/*	Amazon Linux AMI 패키지가 포함된 Amazon S3 버킷
arn:aws:s3:::repo.*region*.amazonaws.com/*	Amazon Linux AMI 리포지토리가 포함된 Amazon S3 버킷
arn:aws:s3:::amazonlinux.*region*.amazonaws.com/*	AL2 리포지토리가 포함된 Amazon S3 버킷
arn:aws:s3:::amazonlinux-2-repos-*region*/*	AL2 리포지토리가 포함된 Amazon S3 버킷

다음 정책은 조직에 속한 아이덴티티와 리소스에 대한 액세스를 제한하고 커널 라이브 패치에 필요한 Amazon S3 버킷에 대한 액세스를 제공하는 방법을 보여줍니다. *region*, *principal-org-id* 및 *resource-org-id*를 해당 조직의 값으로 바꿉니다.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRequestsByOrgsIdentitiesToOrgsResources",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "principal-org-id",
          "aws:ResourceOrgID": "resource-org-id"
        }
      }
    },
    {
      "Sid": "AllowAccessToAmazonLinuxAMIRepositories",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::packages.region.amazonaws.com/*",
        "arn:aws:s3:::repo.region.amazonaws.com/*",
        "arn:aws:s3:::amazonlinux.region.amazonaws.com/*",
        "arn:aws:s3:::amazonlinux-2-repos-region/*"
      ]
    }
  ]
}
```

커널 라이브 패치 작업

인스턴스 자체의 명령줄을 사용하여 개별 인스턴스에서 커널 라이브 패치를 활성화하고 사용하거나 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 설명서의 AL2 인스턴스에서 커널 라이브 패치 사용을 참조하세요.

주제

커널 라이브 패치 활성화

커널 라이브 패치는 AL2에서 기본적으로 비활성화되어 있습니다. 라이브 패치를 사용하려면 커널 라이브 패치용 yum 플러그인을 설치하고 라이브 패치 기능을 활성화해야 합니다.

사전 조건

커널 라이브 패치는 `binutils`가 필요합니다. `binutils`가 설치하지 않은 경우 다음 명령을 사용하여 설치합니다.

```
$ sudo yum install binutils
```

커널 라이브 패치를 활성화하려면

1. 커널 라이브 패치는 다음 AL2 커널 버전에서 사용할 수 있습니다.

   - x86_64 아키텍처의 커널 버전 `4.14` 또는 `5.10`

- ARM64 아키텍처의 커널 버전 5.10

커널 버전을 확인하려면 다음 명령을 실행합니다.

```
$ sudo yum list kernel
```

2. 지원되는 커널 버전을 이미 사용하는 경우 이 단계를 건너뜁니다. 지원되는 커널 버전을 사용하지 않는 경우 다음 명령을 실행하여 커널을 최신 버전으로 업데이트하고 인스턴스를 재부팅합니다.

```
$ sudo yum install -y kernel
```

```
$ sudo reboot
```

3. 커널 라이브 패치용 yum 플러그인을 설치합니다.

```
$ sudo yum install -y yum-plugin-kernel-livepatch
```

4. 커널 라이브 패치용 yum 플러그인을 활성화합니다.

```
$ sudo yum kernel-livepatch enable -y
```

또한 이 명령은 구성된 리포지토리에서 최신 버전의 커널 라이브 패치 RPM을 설치합니다.

5. 커널 라이브 패치용 yum 플러그인이 성공적으로 설치되었는지 확인하려면 다음 명령을 실행합니다.

```
$ rpm -qa | grep kernel-livepatch
```

커널 라이브 패치를 활성화하면 빈 커널 라이브 패치 RPM이 자동으로 적용됩니다. 커널 라이브 패치가 성공적으로 활성화된 경우 초기 빈 커널 라이브 패치 RPM을 포함하는 목록이 나타납니다. 다음은 예제 출력입니다.

```
yum-plugin-kernel-livepatch-1.0-0.11.amzn2.noarch
kernel-livepatch-5.10.102-99.473-1.0-0.amzn2.x86_64
```

6. kpatch 패키지를 설치합니다.

```
$ sudo yum install -y kpatch-runtime
```

7. kpatch 서비스가 이전에 설치된 경우 업데이트합니다.

```
$ sudo yum update kpatch-runtime
```

8. 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
```

9. AL2 Extras Library에서 커널 라이브 패치 주제를 활성화합니다. 이 주제에는 커널 라이브 패치가 포함되어 있습니다.

```
$ sudo amazon-linux-extras enable livepatch
```

사용 가능한 커널 라이브 패치 보기

Amazon Linux 보안 알림이 Amazon Linux 보안 센터에 게시됩니다. 커널 라이브 패치에 대한 알림을 포함하는 AL2 보안 알림에 대한 자세한 내용은 Amazon Linux 보안 센터를 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.

```
$ yum updateinfo list
```

다음은 출력의 예입니다.

```
Loaded plugins: extras_suggestions, kernel-livepatch, langpacks, priorities, update-
motd
ALAS2LIVEPATCH-2020-002 important/Sec. kernel-
livepatch-5.10.102-99.473-1.0-3.amzn2.x86_64
ALAS2LIVEPATCH-2020-005 medium/Sec. kernel-livepatch-5.10.102-99.473-1.0-4.amzn2.x86_64
updateinfo list done
```

CVE와 관련된 사용 가능한 모든 커널 라이브 패치를 나열하려면

다음 명령을 사용합니다.

```
$ yum updateinfo list cves
```

다음은 출력의 예입니다.

```
Loaded plugins: extras_suggestions, kernel-livepatch, langpacks, priorities, update-
motdamzn2-core/2/x86_64 | 2.4 kB 00:00:00
CVE-2019-15918 important/Sec. kernel-livepatch-5.10.102-99.473-1.0-3.amzn2.x86_64
CVE-2019-20096 important/Sec. kernel-livepatch-5.10.102-99.473-1.0-3.amzn2.x86_64
CVE-2020-8648 medium/Sec. kernel-livepatch-5.10.102-99.473-1.0-4.amzn2.x86_64
updateinfo list done
```

커널 라이브 패치 적용

yum 패키지 관리자를 사용하여 정기적인 업데이트를 적용하는 것과 같은 방식으로 커널 라이브 패치를 적용합니다. 커널 라이브 패치용 yum 플러그인은 적용할 수 있는 커널 라이브 패치를 관리합니다.

> **Tip**
>
> 커널 라이브 패치를 사용하여 커널을 정기적으로 업데이트하여 시스템을 재부팅할 수 있을 때까지 중요한 보안 수정 사항과 긴급 보안 수정 사항을 수신하도록 하는 것이 좋습니다. 또한 라이브 패치로 배포할 수 없는 네이티브 커널 패키지에 대한 추가 수정 사항을 사용할 수 있는지 확인하고 이러한 경우 커널 업데이트로 업데이트하고 재부팅하세요.

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 보기에 설명된 명령 중 하나를 사용하여 커널 라이브 패치 버전을 설치합니다.
2. AL2 커널에 커널 라이브 패치를 적용합니다.

   ```
   $ sudo yum install kernel-livepatch-kernel_version.x86_64
   ```

   예를 들어 다음 명령은 AL2 커널 버전에 커널 라이브 패치를 적용합니다`5.10.102-99.473`.

```
$ sudo yum install kernel-livepatch-5.10.102-99.473-1.0-4.amzn2.x86_64
```

정기 보안 업데이트와 함께 사용 가능한 커널 라이브 패치를 적용하려면

다음 명령을 사용합니다.

```
$ sudo yum update --security
```

버그 수정을 포함하려면 `--security` 옵션을 생략하세요.

> **Important**
>
> - 커널 버전은 커널 라이브 패치를 적용한 후에 업데이트되지 않으며, 인스턴스를 재부팅한 후에만 새 버전으로 업데이트됩니다.
> - AL2 커널은 3개월 동안 커널 라이브 패치를 수신합니다. 3개월이 경과한 후에는 해당 커널 버전에 대한 새로운 커널 라이브 패치가 릴리스되지 않습니다. 3개월 후에도 계속 커널 라이브 패치를 받으려면 인스턴스를 재부팅하여 새 커널 버전으로 전환해야 합니다. 그러면 다음 3개월 동안 커널 라이브 패치가 계속 수신됩니다. 커널 버전에 대한 지원 기간을 확인하려면 `yum kernel-livepatch supported`를 실행합니다.

적용된 커널 라이브 패치 보기

설치된 커널 라이브 패치를 보려면

다음 명령을 사용합니다.

```
$ kpatch list
```

이 명령은 로드되고 설치된 보안 업데이트 커널 라이브 패치 목록을 반환합니다. 다음은 예시 출력입니다.

```
Loaded patch modules:
livepatch_cifs_lease_buffer_len [enabled]
livepatch_CVE_2019_20096 [enabled]
livepatch_CVE_2020_8648 [enabled]
```

```
Installed patch modules:
livepatch_cifs_lease_buffer_len (5.10.102-99.473.amzn2.x86_64)
livepatch_CVE_2019_20096 (5.10.102-99.473.amzn2.x86_64)
livepatch_CVE_2020_8648 (5.10.102-99.473.amzn2.x86_64)
```

> **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.

커널 라이브 패치를 비활성화하려면

1. 적용된 커널 라이브 패치의 RPM 패키지를 제거합니다.

   ```
   $ sudo yum kernel-livepatch disable
   ```

2. 커널 라이브 패치용 yum 플러그인을 제거합니다.

   ```
   $ sudo yum remove yum-plugin-kernel-livepatch
   ```

3. 인스턴스를 재부팅합니다.

   ```
   $ sudo reboot
   ```

제한 사항

커널 라이브 패치에는 다음과 같은 제한 사항이 있습니다.

- 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.
- > **Note**
  >
  > 기술적 제한으로 인해 일부 문제는 라이브 패치로 해결할 수 없습니다. 따라서 이러한 수정 사항은 커널 라이브 패치 패키지로 배송되지 않고 네이티브 커널 패키지 업데이트로만 배송

됩니다. 기본 커널 패키지 업데이트를 설치하고 시스템을 재부팅하여 평소와 같이 패치를 활성화할 수 있습니다.

자주 묻는 질문(FAQ)

AL2용 커널 라이브 패치에 대해 자주 묻는 질문은 Amazon Linux 2 커널 라이브 패치 FAQ를 참조하세요.

AL2 Extras 라이브러리

> **Warning**
>
> epel Extra는 타사 EPEL7리포지토리를 활성화합니다. 2024년 6월 30일부터 타사 EPEL7 리포지토리는 더 이상 유지 관리되지 않습니다.
> 이 타사 리포지토리는 향후 업데이트되지 않습니다. 즉, EPEL 리포지토리의 패키지에 대한 보안 수정 사항이 없습니다.
> 일부 EPEL 패키지에 대한 옵션은 EPEL Amazon Linux 2023 사용 설명서의 섹션을 참조하세요.

AL2에서는 Extras Library를 사용하여 인스턴스에 애플리케이션 및 소프트웨어 업데이트를 설치할 수 있습니다. 이러한 소프트웨어 업데이트를 주제라고 합니다. 특정 버전의 주제를 설치하거나 버전 정보를 생략하여 최신 버전을 사용할 수 있습니다. 추가 항목은 운영 체제의 안정성과 사용 가능한 소프트웨어의 최신성 간에 손상을 줄여야 하는 문제를 완화하는 데 도움이 됩니다.

추가 항목 주제의 내용은 장기 지원 및 이진 호환성에 대한 Amazon Linux 정책에서 제외됩니다. Extras 주제는 큐레이션된 패키지 목록에 대한 액세스를 제공합니다. 패키지 버전은 자주 업데이트되거나 AL2와 동일한 시간 동안 지원되지 않을 수 있습니다.

> **Note**
>
> AL2가 EOL에 도달하기 전에 개별 추가 항목 주제가 더 이상 사용되지 않을 수 있습니다.

사용 가능한 주제를 나열하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ amazon-linux-extras list
```

주제를 활성화하고 최신 버전의 패키지를 설치하여 최신 상태를 유지하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo amazon-linux-extras install topic
```

안정성을 보장하기 위해 주제를 활성화하고 패키지의 특정 버전을 설치하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo amazon-linux-extras install topic=version topic=version
```

주제에서 설치된 패키지를 제거하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo yum remove $(yum list installed | grep amzn2extra-topic | awk
 '{ print $1 }')
```

> **Note**
>
> 이 명령은 Extra의 종속 항목으로 설치된 패키지를 제거하지 않습니다.

주제를 비활성화하고 yum 패키지 관리자가 패키지에 액세스할 수 없도록 하려면 다음 명령을 사용합니다.

```
[ec2-user ~]$ sudo amazon-linux-extras disable topic
```

> **Important**
>
> 이 명령은 고급 사용자를 위한 것입니다. 이 명령을 잘못 사용하면 패키지 호환성 충돌이 발생할 수 있습니다.

Amazon Linux 2 추가 항목 목록

추가 이름	더 이상 사용되지 않는 날짜
BCC	
GraphicsMagick1.3	
R3.4	
R4	
asible2	2023-09-30
aws-nitro-enclaves-cli	

추가 이름	더 이상 사용되지 않는 날짜
awscli1	
collectd	
collectd-python3	
corretto8	
dnsmasq	
dnsmasq2.85	2025-05-01
Docker	
ecs	
emacs	2018-11-14
Epel	2024-06-30
Firecracker	2022-11-08
Firefox	
gimp	2018-11-14
golang1.11	2023-08-01
golang1.19	2023-09-30
golang1.9	2018-12-14
하프록시2	
httpd_modules	
java-openjdk11	2024-09-30
커널-5.10	

추가 이름	더 이상 사용되지 않는 날짜
커널-5.15	
커널-5.4	
커널-ng	2022-08-08
lamp-mariadb10.2-php7.2	2020-11-30
libreoffice	
livepatch	
광택	
lustre2.10	
lynis	
mariadb10.5	2025-06-24
mate-desktop1.x	
memcached1.5	
mock	
mock2	
mono	
nano	2018-11-14
nginx1	
nginx1.12	2019-09-20
nginx1.22.1	
php7.1	2020-01-15

추가 이름	더 이상 사용되지 않는 날짜
php7.2	2020-11-30
php7.3	2021-12-06
php7.4	2022-11-03
php8.0	2023-11-26
php8.1	2025-12-31
php8.2	
postgresql10	2023-09-30
postgresql11	2023-11-09
postgresql12	2024-11-14
postgresql13	2025-11-13
postgresql14	
postgresql9.6	2022-08-09
python3	2018-08-22
python3.8	2024-10-14
redis4.0	2021-05-25
redis6	2026-01-31
ruby2.4	2020-08-27
ruby2.6	2023-03-31
ruby3.0	2024-03-31
rust1	2025-05-01

추가 이름	더 이상 사용되지 않는 날짜
selinux-ng	
오징어4	2023-09-30
테스트	
tomcat8.5	2024-03-31
tomcat9	
unbound1.13	2025-05-01
unbound1.17	
vim	2018-11-14

AL2 예약 사용자 및 그룹

AL2는 이미지 프로비저닝 및 특정 패키지 설치 중에 특정 사용자 및 그룹을 사전 할당합니다. 충돌을 방지하기 위해 사용자, 그룹 및 관련 UID와 GID가 여기에 나열됩니다.

주제

Amazon Linux 2 예약 사용자 목록

UID로 나열됨

사용자 이름	UID
Root	0
bin	1
daemon	2
adm	3
lp	4
동기화	5
종료	6
중지	7
메일	8
uucp	10
연산자	11
게임	12

사용자 이름	UID
ftp	14
oprofile	16
pkiuser	17
squid	23
named	25
postgres	26
mysql	27
nscd	28
nscd	28
rpcuser	29
rpc	32
amandabackup	33
NTP	38
메일맨	41
gdm	42
mailnull	47
apache	48
smmsp	51
tomcat	53
ldap	55

사용자 이름	UID
tss	59
nslcd	65
Pegasus	66
avahi	70
tcpdump	72
sshd	74
radvd	75
사이러스	76
arpwatch	77
fax	78
dbus	81
PostFix	89
쿼가	92
반경	95
반경	95
hsqldb	96
dovecot	97
ident	98
nobody	99
Qemu	107

사용자 이름	UID
usbmuxd	113
stap-server	155
avahi-autoipd	170
pulse	171
rtkit	172
dhcpd	177
sanlock	179
haproxy	188
하클러스터	189
systemd-journal-gateway	191
systemd-network	192
systemd-resolve	193
uuidd	357
tang	358
stapdev	359
stapsys	360
stapusr	361
systemd-journal-upload	362
systemd-journal-remote	363
saned	364

사용자 이름	UID
pesign	365
pcpqa	366
pcp	367
memcached	368
ipsilon	369
ipaapi	370
kdcproxy	371
ods	372
sssd	373
글러스터	374
fedfs	375
dovenull	376
coroqnetd	377
클레비스	378
clamscan	379
clamilt	380
clamupdate	381
colord	382
geoclue	383
aws-kinesis-agent-user	384

사용자 이름	UID
cwagent	385
unbound	386
polkitd	387
saslauth	388
dirsrv	389
chrony	996
ec2-instance-connect	997
rngd	998
libstoragemgmt	999
ec2-user	1000
nfsnobody	65534

이름별로 나열됨

사용자 이름	UID
adm	3
amandabackup	33
apache	48
arpwatch	77
avahi	70
avahi-autoipd	170

사용자 이름	UID
aws-kinesis-agent-user	384
bin	1
chrony	996
clamilt	380
clamscan	379
clamupdate	381
클레비스	378
colord	382
coroqnetd	377
cwagent	385
사이러스	76
daemon	2
dbus	81
dhcpd	177
dirsrv	389
dovecot	97
dovenull	376
ec2-instance-connect	997
ec2-user	1000
fax	78

사용자 이름	UID
fedfs	375
ftp	14
게임	12
gdm	42
geoclue	383
글러스터	374
하클러스터	189
중지	7
haproxy	188
hsqldb	96
ident	98
ipaapi	370
ipsilon	369
kdcproxy	371
ldap	55
libstoragemgmt	999
lp	4
메일	8
메일맨	41
mailnull	47

사용자 이름	UID
memcached	368
mysql	27
named	25
nfsnobody	65534
nobody	99
nscd	28
nscd	28
nslcd	65
NTP	38
ods	372
연산자	11
oprofile	16
pcp	367
pcpqa	366
Pegasus	66
pesign	365
pkiuser	17
polkitd	387
PostFix	89
postgres	26

사용자 이름	UID
pulse	171
Qemu	107
쿼가	92
반경	95
반경	95
radvd	75
rngd	998
Root	0
rpc	32
rpcuser	29
rtkit	172
saned	364
sanlock	179
saslauth	388
종료	6
smmsp	51
squid	23
sshd	74
sssd	373
stap-server	155

사용자 이름	UID
stapdev	359
stapsys	360
stapusr	361
동기화	5
systemd-journal-gateway	191
systemd-journal-remote	363
systemd-journal-upload	362
systemd-network	192
systemd-resolve	193
tang	358
tcpdump	72
tomcat	53
tss	59
unbound	386
usbmuxd	113
uucp	10
uuidd	357

Amazon Linux 2 예약 그룹 목록

GID로 나열됨

그룹 이름	GID
Root	0
bin	1
daemon	2
sys	3
adm	4
tty	5
디스크	6
디스크	6
lp	7
mem	8
kmem	9
wheel	10
cdrom	11
메일	12
uucp	14
man	15
oprofile	16
pkiuser	17
dialout	18
floppy	19

그룹 이름	GID
게임	20
slocate	21
utmp	22
squid	23
named	25
postgres	26
mysql	27
nscd	28
nscd	28
rpcuser	29
rpc	32
테이프	33
테이프	33
utempter	35
kvm	36
NTP	38
비디오	39
딥	40
메일맨	41
gdm	42

그룹 이름	GID
mailnull	47
apache	48
ftp	50
smmsp	51
tomcat	53
잠금	54
ldap	55
tss	59
오디오	63
Pegasus	65
avahi	70
tcpdump	72
sshd	74
radvd	75
saslauth	76
saslauth	76
arpwatch	77
fax	78
dbus	81
screen	84

그룹 이름	GID
쿼가브	85
wbpriv	88
wbpriv	88
PostFix	89
postdrop	90
쿼가	92
반경	95
반경	95
hsqldb	96
dovecot	97
ID	98
nobody	99
사용자	100
Qemu	107
usbmuxd	113
stap-server	155
stapusr	156
stapusr	156
stapsys	157
stapdev	158

그룹 이름	GID
avahi-autoipd	170
pulse	171
rtkit	172
dhcpd	177
sanlock	179
haproxy	188
haclient	189
systemd-journal	190
systemd-journal	190
systemd-journal-gateway	191
systemd-network	192
systemd-resolve	193
usbmon	351
wireshark	352
uuidd	353
tang	354
systemd-journal-upload	355
sfcb	356
systemd-journal-remote	356
saned	357

그룹 이름	GID
pesign	358
pcpqa	359
pcp	360
memcached	361
virtlogin	362
ipsilon	363
pkcs11	364
ipaapi	365
kdcproxy	366
ods	367
sssd	368
libvirt	369
글러스터	370
fedfs	371
dovenull	372
Docker	373
coroqnetd	374
클레비스	375
clamscan	376
clamilt	377

그룹 이름	GID
virusgroup	378
virusgroup	378
virusgroup	378
clamupdate	379
colord	380
geoclue	381
printadmin	382
aws-kinesis-agent-user	383
cwagent	384
pulse-rt	385
pulse-access	386
unbound	387
polkitd	388
dirsrv	389
cgred	993
chrony	994
ec2-instance-connect	995
rngd	996
libstoragemgmt	997
ssh_keys	998

그룹 이름	GID
입력	999
ec2-user	1000
nfsnobody	65534

이름별로 나열됨

그룹 이름	GID
adm	4
apache	48
arpwatch	77
오디오	63
avahi	70
avahi-autoipd	170
aws-kinesis-agent-user	383
bin	1
cdrom	11
cgred	993
chrony	994
clamilt	377
clamscan	376
clamupdate	379

그룹 이름	GID
클레비스	375
colord	380
coroqnetd	374
cwagent	384
daemon	2
dbus	81
dhcpd	177
dialout	18
딥	40
dirsrv	389
디스크	6
디스크	6
Docker	373
dovecot	97
dovenull	372
ec2-instance-connect	995
ec2-user	1000
fax	78
fedfs	371
floppy	19

그룹 이름	GID
ftp	50
게임	20
gdm	42
geoclue	381
글러스터	370
haclient	189
haproxy	188
hsqldb	96
ident	98
입력	999
ipaapi	365
ipsilon	363
kdcproxy	366
kmem	9
kvm	36
ldap	55
libstoragemgmt	997
libvirt	369
잠금	54
lp	7

그룹 이름	GID
메일	12
메일맨	41
mailnull	47
man	15
mem	8
memcached	361
mysql	27
named	25
nfsnobody	65534
nobody	99
nscd	28
nscd	28
NTP	38
ods	367
oprofile	16
pcp	360
pcpqa	359
Pegasus	65
pesign	358
pkcs11	364

그룹 이름	GID
pkiuser	17
polkitd	388
postdrop	90
PostFix	89
postgres	26
printadmin	382
pulse	171
pulse-access	386
pulse-rt	385
Qemu	107
쿼가	92
쿼가브	85
반경	95
반경	95
radvd	75
rngd	996
Root	0
rpc	32
rpcuser	29
rtkit	172

그룹 이름	GID
saned	357
sanlock	179
saslauth	76
saslauth	76
screen	84
sfcb	356
slocate	21
smmsp	51
squid	23
ssh_keys	998
sshd	74
sssd	368
stap-server	155
stapdev	158
stapsys	157
stapusr	156
stapusr	156
sys	3
systemd-journal	190
systemd-journal	190

그룹 이름	GID
systemd-journal-gateway	191
systemd-journal-remote	356
systemd-journal-upload	355
systemd-network	192
systemd-resolve	193
tang	354
테이프	33
테이프	33
tcpdump	72
tomcat	53
tss	59
tty	5
unbound	387
usbmon	351
usbmuxd	113
사용자	100
utempter	35
utmp	22
uucp	14
uuidd	353

그룹 이름	GID
비디오	39
virtlogin	362
virusgroup	378
virusgroup	378
virusgroup	378
wbpriv	88
wbpriv	88
wheel	10
wireshark	352

AL2 소스 패키지

Amazon Linux에 제공된 도구를 사용하여 참조 목적으로 인스턴스에 설치한 패키지의 원본을 볼 수 있습니다. Amazon Linux에 포함된 모든 패키지와 온라인 패키지 리포지토리에 대해 원본 패키지를 사용할 수 있습니다. 설치할 소스 패키지의 패키지 이름을 결정하고 yumdownloader --source 명령을 사용하여 실행 중인 인스턴스 내에서 소스를 확인합니다. 예제:

```
[ec2-user ~]$ yumdownloader --source bash
```

소스 RPM은 압축을 풀고 참조용으로 표준 RPM 도구를 사용하여 소스 트리를 볼 수 있습니다. 디버깅을 완료한 후에는 패키지를 사용할 수 있습니다.

AL2의 보안 및 규정 준수

의 클라우드 보안AWS이 최우선 순위입니다. AWS고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS는 AWS클라우드에서 AWS서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS규정 준수 프로그램 일환으로 보안의 효과를 정기적으로 테스트하고 확인합니다. AL2023에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램 AWS제공 범위 내 서비스규정 준수 프로그램 .
- 클라우드 내 보안 – 귀하의 책임은 귀하가 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

AL2에서 FIPS 모드 활성화

이 섹션에서는 AL2에서 FIPS(연방 정보 처리 표준)를 활성화하는 방법을 설명합니다. FIPS에 대한 자세한 내용은 다음을 참조하세요.

- 연방 정보 처리 표준(FIPS)
- 연방 정보 처리 표준 규정 준수 FAQs

사전 조건

- 필요한 패키지를 다운로드하기 위해 인터넷에 액세스할 수 있는 기존 AL2 Amazon EC2 인스턴스입니다. AL2 Amazon EC2 인스턴스 시작에 대한 자세한 내용은 섹션을 참조하세요Amazon EC2의 AL2 Amazon EC2.
- SSH 또는 AWS Systems Manager를 사용하여 Amazon EC2 인스턴스에 연결해야 합니다.

> **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 생성

해야 합니다. 그렇지 않으면 FIPS 모드를 활성화한 후 인스턴스 액세스 권한을 잃을 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 키 쌍 생성 섹션을 참조하세요.

FIPS 모드 활성

1. SSH 또는를 사용하여 AL2 인스턴스에 연결합니다AWS Systems Manager.
2. 시스템이 최신 버전인지 확인합니다. 자세한 내용은 패키지 리포지토리 단원을 참조하십시오.
3. 다음 명령을 실행하여 `dracut-fips` 모듈을 설치하고 활성화합니다.

   ```
   sudo yum -y install dracut-fips
   sudo dracut -f
   ```

4. 다음 명령을 사용하여 Linux 커널 명령줄에서 FIPS 모드를 활성화합니다. 이렇게 하면 AL2 FAQ에 나열된 모듈에 대해 시스템 전체에서 FIPS 모드가 활성화됩니다.

   ```
   sudo /sbin/grubby --update-kernel=ALL --args="fips=1"
   ```

5. AL2 인스턴스를 재부팅합니다.

   ```
   sudo reboot
   ```

6. FIPS 모드가 활성화되었는지 확인하려면 인스턴스에 다시 연결하고 다음 명령을 실행합니다.

   ```
   sysctl crypto.fips_enabled
   ```

   다음 결과가 표시됩니다.

   ```
   crypto.fips_enabled = 1
   ```

   다음 명령을 실행하여 OpenSSH가 FIPS 모드인지 확인할 수도 있습니다.

   ```
   ssh localhost 2>&1 | grep FIPS
   ```

   다음 결과가 표시됩니다.

   ```
   FIPS mode initialized
   ```

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.