aws

Benutzerhandbuch

Amazon Linux 2023

Amazon Linux 2023: Benutzerhandbuch

Table of Contents

Was ist Amazon Linux 2023? ... 1
Release-Taktfrequenz ... 1
Haupt- und Nebenversionen ... 3
Aufnahme neuer Versionen ... 4
Langfristige Support-Richtlinie ... 4
Benennung und Versionsverwaltung ... 4
Leistungs- und Betriebsoptimierungen ... 6
Beziehung zu Fedora ... 7
Anpassen von cloud-init ... 7
Sicherheits-Updates und Features ... 9
Verwaltete Updates ... 9
Sicherheit in der Cloud ... 10
SELinux-Modi ... 10
Compliance-Programm ... 10
SSH-Server-Standard ... 10
Hauptfunktionen von OpenSSL 3 ... 10
Netzwerkdienst ... 11
Core-Toolchain-Paketeglibc, gcc und binutils ... 11
Paketmanagement-Tool ... 12
Standard-SSH-Serverkonfiguration ... 13
Veraltete Funktionalität ... 15
`compat-`-Pakete ... 15
Veraltete Funktionalität in AL1 eingestellt, in AL2 entfernt ... 15
32-Bit-x86-AMIs (i686) ... 16
`aws-apitools-*`ersetzt durch AWS CLI ... 16
`systemd`ersetzt in AL2 `upstart` ... 17
Funktionalität in AL2 veraltet und in AL2023 entfernt ... 17
32-Bit-x86-Pakete (i686) ... 18
`aws-apitools-*`ersetzt durch AWS CLI ... 18
`bzr`Revisionskontrollsystem ... 19
cgroup v1 ... 19
log4jHotpatch () `log4j-cve-2021-44228-hotpatch` ... 19
`lsb_release` und das `system-lsb-core`-Paket ... 20
`mcrypt` ... 20

OpenJDK (7) `java-1.7.0-openjdk` ... 21
Python 2.7 ... 21
`rsyslog-openssl`ersetzt `rsyslog-gnutls` ... 21
Netzwerkinformationsdienst (NIS)/`yp` ... 21
In AL2023 veraltet ... 22
32-Bit-x86-Laufzeitunterstützung (i686) ... 22
Berkeley-Datenbank () `libdb` ... 22
`cron` ... 23
IMDSv1 ... 23
`pcre` Version 1 ... 23
System V init (`sysvinit`) ... 24
Vergleich von AL2 und AL2023 ... 25
Hinzugefügte, aktualisierte und entfernte Pakete ... 26
Support für die einzelnen Versionen ... 26
Änderungen bei der Benennung und Versionierung ... 26
Optimierungen ... 26
Python 2.7 wurden durch Python 3 ersetzt ... 27
Sicherheits-Updates ... 27
SELinux ... 27
OpenSSL 3 ... 28
IMDSv2 ... 28
Entfernen des log4j-Hotpatch (`log4j-cve-2021-44228-hotpatch`) ... 29
Deterministische Upgrades für Stabilität ... 29
Aus mehreren Upstream-Quellen ... 30
AMI-Root-Dateisystem und standardmäßiger Amazon-EBS-Volume-Typ ... 30
Netzwerksystem-Dienst ... 30
Vereinheitlichte Kontrollgruppenhierarchie (cgroup v2) ... 30
Aufgabenplanung ... 31
Pakete für glibc, gcc und binutils ... 31
Paketmanager ... 32
Protokollierungssystem ... 32
Paketänderungen für `curl` und `libcurl` ... 32
GNU Privacy Guard (GNUPG) ... 32
Amazon Corretto als Standard-JVM ... 33
AWS CLI v2 ... 33
UEFI Preferred ... 33

Änderungen der Standardkonfiguration des SSH-Servers ... 33
Extra Packages for Enterprise Linux (EPEL) ... 34
Verwenden von cloud-init ... 34
Grafische Desktop-Unterstützung ... 35
Compiler-Triplet ... 35
32-Bit x86-(i686)-Pakete ... 35
`lsb_release` und das `system-lsb-core`-Paket ... 35
Kernel-Änderungen in AL2023 gegenüber AL2 ... 36
Sicherheitsorientierte Änderungen an der Kernel-Konfiguration ... 36
Weitere Änderungen in der Kernelkonfiguration ... 40
Unterstützung für das Kernel-Dateisystem ... 42
Vergleich von Amazon Linux 2 und AL2023 AMI ... 47
Vergleich von Amazon Linux 2 und AL2023 Minimal AMI ... 80
Vergleich von Amazon Linux 2 und AL2023 Container ... 100
Vergleich von AL1 und AL2023 ... 109
Support für die einzelnen Versionen ... 109
`systemd` ersetzt `upstart` als `init`-System ... 110
Python 2.6 und 2.7 wurden durch Python 3 ersetzt ... 110
OpenJDK 8 als ältestes JDK ... 110
Kernel-Änderungen in AL2023 gegenüber AL1 ... 110
Kernel-Live-Patching ... 110
Unterstützung für das Kernel-Dateisystem ... 110
Sicherheitsorientierte Änderungen an der Kernel-Konfiguration ... 112
Weitere Änderungen in der Kernelkonfiguration ... 114
Vergleich zwischen AL1 und AL2023 AMI ... 115
Vergleich zwischen AL1 und AL2023 Minimal AMI ... 149
Vergleich der Container AL1 und AL2023 ... 169
Systemanforderungen ... 178
CPU-Anforderungen für die Ausführung von AL2023 ... 178
ARM-CPU-Anforderungen für AL2023 ... 178
x86-64-CPU-Anforderungen für AL2023 ... 179
Speicheranforderungen (RAM) für die Ausführung von AL2023 ... 180
Verwenden von AL2023 auf AWS ... 181
Erste Schritte mit AWS ... 181
Melde dich an für ein AWS-Konto ... 181
Erstellen Sie einen Benutzer mit Administratorzugriff ... 182

Erteilen programmgesteuerten Zugriffs ... 183
AL2023 auf Amazon EC2 ... 185
AL2023 mit der Amazon EC2 EC2-Konsole starten ... 186
Starten von AL2023 mit dem SSM-Parameter und AWS CLI ... 187
Starten des neuesten AL2023 AMI mit AWS CloudFormation ... 188
AL2023 mit einer bestimmten AMI-ID starten ... 190
AL2023 AMI: Veraltete Version und Lebenszyklus ... 190
Verbindung zu AL203-Instances herstellen ... 191
Vergleich von AL2023 Standard (Standardversion) und Minimal-AMIs ... 191
AL2023 in Containern ... 219
AL2023-Basiscontainer-Image ... 219
AL2023 Minimales Container-Image ... 222
Erstellung einfacher AL203-Container-Images ... 224
Vergleich der AL2023–Container-Image-Paketliste ... 228
AL2023-Minimal-AMI Vergleich mit Container-Images ... 233
AL2023 auf Elastic Beanstalk ... 250
AL2023 CloudShell ... 251
AL2023 für Amazon ECS-Container-Hosts ... 251
Amazon ECS-relevante Änderungen seit AL2 ... 252
Benutzerdefinierte Amazon-ECS-optimierte AMIs ... 253
Amazon EFS auf AL2023 ... 253
`amazon-efs-utils` ... 254
Ein Amazon-EFS-Dateisystems mounten ... 254
Amazon EMR in AL2023 ... 254
Auf AL2023 basierende Amazon EMR-Versionen ... 254
Auf AL2023 basiertes Amazon EMR auf EKS ... 255
AL2023 ein AWS Lambda ... 255
`provided.al2023`-Lambda-Laufzeit ... 255
Auf AL2023 basierende Laufzeiten ... 255
Tutorials ... 256
Installieren Sie LAMP auf AL2023 ... 256
Schritt 1: Vorbereiten des LAMP-Servers ... 257
Schritt 2: Testen Ihres Lamp-Servers ... 262
Schritt 3: Sichern des Datenbankservers ... 264
Schritt 4: (Optional) Installieren phpMyAdmin ... 265
Fehlerbehebung ... 268

Verwandte Themen 269
Konfigurieren Sie SSL/TLS auf AL2023 270
Voraussetzungen 271
Schritt 1: Aktivieren von TLS auf dem Server 272
Schritt 2: Abrufen eines CA-signierten Zertifikats 275
Schritt 3: Testen und Verstärken der Sicherheitskonfiguration 284
Fehlerbehebung 288
Hosten Sie einen WordPress Blog auf AL2023 289
Voraussetzungen 289
Installieren WordPress 290
Nächste Schritte 301
Hilfe! Mein öffentlicher DNS-Name hat sich geändert und jetzt funktioniert mein Blog nicht mehr. 302
AL2023 außerhalb von Amazon EC2 304
Laden Sie AL2023-VM-Images herunter 304
Unterstützte Konfigurationen 304
KVM-Anforderungen 305
VMwareVoraussetzungen 307
Hyper-V-Anforderungen 310
AL2023-VM-Konfiguration 312
NoCloud `seed.iso`-basierte Konfiguration 313
VMwareKonfiguration auf Guestinfo-Basis 316
AL2023-Paketlistenvergleich für das Standard-AMI- und KVM-Image 319
AL2023-Paketlistenvergleich für das Standard-AMI- und das VMware-OVA-Image 343
AL2023-Paketlistenvergleich für das Standard-AMI- und Hyper-V-Image 369
Aktualisierung von AL2023 395
Erhalten Sie Benachrichtigungen über neue Updates 395
Verwalten von Aktualisierungen 396
Prüfen auf verfügbare Paket-Updates 397
Anwenden von Sicherheits-Updates mithilfe von DNF- und Repository-Versionen 398
Automatischer Neustart des Dienstes nach (Sicherheits-) Updates 401
Starten einer Instance mit aktivierter neuester Repository-Version 402
Abrufen von Paketunterstützunginformationen 403
Prüfen auf neuere Repository-Versionen 404
Hinzufügen, aktivieren oder deaktivieren neuer Repositorys 407
Hinzufügen von Repositorys mit cloud-init 409

Verwendung deterministischer Upgrades über ein versioniertes Repository auf AL2023 410
Kontrolle über die Updates, die Sie aus Haupt- und Nebenversionen erhalten 411
Unterschiede zwischen Haupt- und Nebenversions-Upgrades 411
Kontrollieren Sie die Paket-Updates, die in den AL2023-Repositorien verfügbar sind 412
Deterministische Upgrades durch Nutzung versionierter Repositorys 412
Kernel-Live-Patching 418
Einschränkungen 419
Unterstützte Konfigurationen und Voraussetzungen 419
Arbeiten mit Kernel-Live-Patching 420
Programmiersprachen und Laufzeiten 426
C/C++ und Fortran 426
Go 427
AL2023 Lambda-Funktion: Go 428
Java 428
Perl 429
Perl-Module 429
PHP 429
Migration zu neuen PHP-Versionen 429
Migration aus PHP 7.x 430
PHP-Module 430
Python 430
Python-Module 431
Rust 431
AL2023 Lambda-Funktion: Rust 432
Sicherheit und Compliance 433
Sicherheitshinweise 434
ALAS Announcements 434
ALAS FAQs 435
Einstellung der SELinux-Modi für AL2023 435
Standard-SELinux-Status und -Modi für AL2023 435
Wechseln in den `enforcing`-Modus 436
Option zur Deaktivierung von SELinux 438
Aktivieren Sie den FIPS-Modus auf AL2023 439
Kernel-Hardening 441
Kernel-Hardening-Optionen (architekturunabhängig) 441
Spezifische Kernel-Hardening-Optionen für x86-64 454

aarch64-spezifische Kernel-Hardening-Optionen 457
UEFI Secure Boot auf AL2023 458
Aktivieren Sie UEFI Secure Boot auf AL2023 459
Registrierung einer vorhandenen Instance 459
Image aus einem Snapshot registrieren 460
Widerruf-Updates 461
Wie funktioniert UEFI Secure Boot auf AL2023 461
Eigene Schlüssel registrieren 462
..... cdlxiii

Was ist Amazon Linux 2023?

Amazon Linux 2023 (AL2023) ist die nächste Generation von Amazon Linux von Amazon Web Services (AWS). Mit AL2023 können Sie Cloud- und Unternehmensanwendungen in einer sicheren, stabilen und leistungsstarken Laufzeitumgebung entwickeln und ausführen. Außerdem erhalten Sie eine Anwendungsumgebung, die langfristigen Support mit Zugriff auf die neuesten Linux-Innovationen bietet. AL2023 wird Benutzern ohne Zusatzkosten angeboten.

AL2023 ist der Nachfolger von Amazon Linux 2 (AL2). Informationen zu den Unterschieden zwischen AL2023 und AL2 finden Sie unter Vergleich von AL2 und AL2023 und Paketänderungen in AL2023.

Themen

- Release-Taktfrequenz
- Benennung und Versionsverwaltung
- Leistungs- und Betriebsoptimierungen
- Beziehung zu Fedora
- Anpassen von cloud-init
- Sicherheits-Updates und Features
- Netzwerkdienst
- Core-Toolchain-Paketeglibc, gcc und binutils
- Paketmanagement-Tool
- Standard-SSH-Serverkonfiguration

Release-Taktfrequenz

Eine neue Hauptversion von Amazon Linux wird alle zwei Jahre veröffentlicht und beinhaltet fünf Jahre Support. Jede Version beinhaltet Support in zwei Phasen. Die Standard-Supportphase umfasst die ersten zwei Jahre. Als Nächstes folgt eine Wartungsphase, während der für weitere drei Jahre Support angeboten wird.

In der Standard-Supportphase erhält die Version vierteljährliche kleinere Versions-Updates. Während der Wartungsphase erhält eine Version nur Sicherheits-Updates und kritische Bugfixes, die veröffentlicht werden, sobald sie verfügbar sind.

Jahr	Amazon Linux 2023	Amazon Linux 2025	Amazon Linux 2027	Amazon Linux 2029
2023	 - Standard-Support			
2024	 - Standard-Support			
2025	Wartung	 - Standard-Support		
2026	Wartung	 - Standard-Support		
2027	Wartung	Wartung	 - Standard-Support	
2028	 - EOL	Wartung	 - Standard-Support	

Jahr	Amazon Linux 2023	Amazon Linux 2025	Amazon Linux 2027	Amazon Linux 2029
2029	 - EOL	Wartung	Wartung	 - Standard-Support
2030	 - EOL	 - EOL	Wartung	 - Standard-Support
2031	 - EOL	 - EOL	Wartung	Wartung

Haupt- und Nebenversionen

Mit jeder neuen Amazon-Linux-Version (Hauptversion, Nebenversion oder Sicherheits-Update) veröffentlichen wir ein neues Linux Amazon Machine Image (AMI).

- Hauptversion - Beinhaltet neue Features und Verbesserungen in Bezug auf Sicherheit und Leistung im gesamten Stack. Die Verbesserungen können größere Änderungen am Kernel, der Toolchain, Glib C, OpenSSL und allen anderen Systembibliotheken und Dienstprogrammen beinhalten. Hauptversionen von Amazon Linux basieren teilweise auf der aktuellen Version der Fedora-Linux-Upstream-Distribution. AWS kann ggf. bestimmte Pakete aus anderen Nicht-Fedora-Upstreams hinzufügen oder ersetzen.
- Nebenversion - Ein vierteljährliches Update mit Sicherheits-Updates, Bugfixes sowie neue Features und Paketen. Jede Nebenversion enthält kumulative Updates, die neben neuen Funktionen und Paketen auch Sicherheits-Updates und Bugfixes enthält. Diese Versionen können aktuelle Sprachlaufzeiten enthalten, z. B. PHP. Sie könnten auch andere beliebte Softwarepakete wie Ansible und Docker enthalten.

Aufnahme neuer Versionen

Updates werden über eine Kombination aus neuen Amazon-Machine-Image-(AMI)-Versionen und entsprechenden neuen Repositorys bereitgestellt. Standardmäßig werden ein neues AMI und das Repository, auf das es verweist, miteinander verknüpft. Sie können Ihre laufenden Amazon-EC2-Instances auch Schritt für Schritt auf neuere Repository-Versionen verweisen lassen, um die laufenden Instances zu aktualisieren. Sie können ein Update auch durch das Starten neuer Instances der neuesten AMIs durchführen.

Langfristige Support-Richtlinie

Amazon Linux stellt Updates für alle Ihre Pakete bereit und gewährleistet die Kompatibilität innerhalb einer Hauptversion für alle Ihre auf Amazon Linux aufbauenden Anwendungen. Kernpakete wie die glibc-Bibliothek, OpenSSL, OpenSSH und der DNF-Paketmanager erhalten Support für die gesamte Lebensdauer der AL2023-Hauptversion. Pakete, die nicht Teil der Kernpakete sind, werden auf der Basis ihrer jeweiligen Upstream-Quellen unterstützt. Führen Sie folgenden Befehl aus, um den Support-Status und -Zeitraum einzelner Pakete anzuzeigen.

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

Mit folgendem Befehl erhalten Sie Informationen über alle aktuell installierten Pakete.

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

Die vollständige Liste der Kernpakete wird in der Vorschau zusammengestellt. Lassen Sie es uns wissen, falls sie weitere Pakete als Kernpakete sehen möchten. Wir sammeln Feedback für weitergehende Analysen. Sie können Ihr Feedback zu AL2023 über Ihren zuständigen AWS-Mitarbeiter abgeben oder ein Problem im amazon-lnux-2023-repo auf GitHub einreichen.

Benennung und Versionsverwaltung

AL2023 bietet während der zweijährigen Standardunterstützung alle drei Monate eine Nebenversion. Jede Version wird durch eine Erhöhung von 0 bis N gekennzeichnet. 0 bezieht sich auf die ursprüngliche Hauptversion für diese Iteration. Alle Versionen werden Amazon Linux 2023 heißen. Wenn Amazon Linux 2025 veröffentlicht wird, wird AL2023 den erweiterten Support mit Aktualisierungen für Sicherheits-Updates und kritische Bugfixes erhalten.

AL2023-Nebenversionen haben beispilsweise das folgende Format:

- `2023.0.20230301`
- `2023.1.20230601`
- `2023.2.20230901`

Die ensprechenden AL2023-AMIs haben das folgende Format:

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

In einzelnen Nebenversionen werden reguläre AMI-Versionen mit einem Zeitstempel des Datums des AMI-Release veröffentlicht.

- `al2023-ami-2023.0.20230301.0-kernel-6.1-x86_64`
- `al2023-ami-2023.0.20230410.0-kernel-6.1-x86_64`
- `al2023-ami-2023.0.20230520.0-kernel-6.1-x86_64`

Die empfohlene Methode zur Identifizierung einer AL2- oder AL2023-Instanz beginnt mit dem Lesen der CPE-Zeichenfolge (Common Platform Enumeration) von. `/etc/system-release-cpe` Teilen Sie dann die Zeichenfolge in die einzelnen Felder auf. Lesen Sie abschließend die Plattform- und Versionswerte.

In AL2023 wurden außerdem auch neue Dateien zur Plattformidentifikation eingeführt:

- `/etc/amazon-linux-release`-Symlinks zu `/etc/system-release`
- `/etc/amazon-linux-release-cpe`-Symlinks zu `/etc/system-release-cpe`

Diese beiden Dateien weisen darauf hin, dass es sich um eine Amazon-Linux-Instance handelt. Sie brauchen kine Datei lesen oder die Zeichenfolge in Felder aufzuteilen, wenn Sie die spezifischen Plattform- und Versionswerte nicht benötigen.

Leistungs- und Betriebsoptimierungen

Amazon Linux 6.1-Kernel

- AL2023 verwendet die neuesten Treiber für Elastic Network Adapter (ENA) - und Elastic Fabric Adapter (EFA) -Geräte. AL2023 konzentriert sich auf Leistungs- und Funktions-Backports für Hardware in der Amazon EC2 EC2-Infrastruktur.
- Kernel-Live-Patching ist für die Instance-Typen `x86_64` und `aarch64` verfügbar. Dadurch werden weniger Neustarts erforderlich.
- Alle Kernel-Build- und Runtime-Konfigurationen beinhalten viele der gleichen Leistungs- und Betriebsoptimierungen wie AL2.

Auswahl der Basis-Toolchain und Standard-Build-Flags

- AL2023-Pakete werden mit standardmäßig aktivierten Compiler-Optimierungen () erstellt `-O2`
- AL2023-Pakete werden mit der Anforderung für `x86-64-v2` für `x86-64`-Systeme (`-march=x86-64-v2`), und Graviton 2 oder höher für `aarch64` (`-march=armv8.2-a+crypto -mtune=neoverse-n1`) erstellt.
- AL2023-Pakete werden mit aktivierter automatischer Vektorisierung (`-ftree-vectorize`) erstellt.
- AL2023-Pakete werden mit aktivierter Link Time Optimization (LTO) erstellt.
- AL2023 verwendet die aktualisierten Versionen von Rust, Clang/LLVM und Go.

Paketauswahl und Versionen

- Bestimmte Backports zu wichtigen Systemkomponenten beinhalten mehrere Leistungsverbesserungen für die Ausführung auf der Amazon-EC2-Infrastruktur, insbesondere Graviton-Instances.
- AL2023 ist mit mehreren Funktionen integriert. AWS-Services Dazu gehören der SSM-Agent AWS CLI, der Amazon Kinesis Kinesis-Agent und. CloudFormation
- AL2023 verwendet Amazon Corretto als Java Development Kits (JDK).
- AL2023 bietet Datenbank-Engines und Laufzeitupdates für Programmiersprachen in neueren Versionen, sobald diese von Upstream-Projekten veröffentlicht werden. Laufzeiten für Programmiersprachen in neuen Versionen werden hinzugefügt, wenn sie veröffentlicht werden.

Einsatz in einer Cloud-Umgebung

- Das AL2023-Basis-AMI und die Container-Images werden häufig aktualisiert, um den Austausch von Patch-Instances zu unterstützen.
- Kernel-Updates sind in AL2023-AMI-Updates enthalten. Das bedeutet, dass Sie keine Befehle wie `yum update` und `reboot` für Aktualisierungen Ihres Kernels verwenden müssen.
- Neben dem Standard-AL2023-AMI ist auch ein minimales AMI- und Container-Image verfügbar. Wählen Sie das minimale AMI, um eine Umgebung mit der minimalen Anzahl von Paketen auszuführen, die für den Betrieb Ihres Dienstes erforderlich ist.
- Standardmäßig sind AL2023-AMIs und Container an eine bestimmte Version der Paket-Repositorys gebunden. Es gibt kein automatisches Update, wenn sie gestartet werden. Das bedeutet, dass Sie immer die Kontrolle darüber haben, wann Sie ein Paket-Update aufnehmen. Sie können jederzeit in einer Beta-/Gamma-Umgebung testen, bevor Sie das Update in die Produktion aufnehmen. Wenn ein Problem auftritt, können Sie den vorvalidierten Rollback-Pfad verwenden.

Beziehung zu Fedora

AL2023 unterhält unabhängig von Fedora eigene Release- und Support-Lebenszyklen. AL2023 bietet aktualisierte Versionen von Open-Source-Software, eine größere Auswahl an Paketen sowie häufige Veröffentlichungen. Dadurch bleiben die vertrauten RPM-basierten Betriebssysteme erhalten.

Die allgemein verfügbare (GA-) Version von AL2023 ist nicht direkt mit einer bestimmten Fedora-Version vergleichbar. Die GA-Version von AL2023 enthält Komponenten von Fedora 34, 35 und 36. Einige der Komponenten sind dieselben wie die Komponenten in Fedora, andere sind modifiziert. Weitere Komponenten ähneln eher den Komponenten in CentOS 9 Streams oder wurden unabhängig entwickelt. Der Amazon-Linux-Kernel basiert auf den langfristigen Support-Optionen auf kernel.org, die unabhängig von Fedora ausgewählt wurden.

Anpassen von cloud-init

Das cloud-init-Paket ist eine Open-Source-Anwendung für das Bootstrapping von Linux-Images in einer Cloud-Computing-Umgebung. Weitere Informationen finden Sie in der Cloud-Init-Dokumentation.

AL2023 enthält eine angepasste Version von cloud-init. cloud-init erlaubt Ihnen festzulegen, was während des Bootvorgangs mit Ihrer Instance geschehen soll.

Wenn Sie eine Instance starten, können Sie die Benutzerdatenfelder verwenden, um Aktionen an sie zu übergeben. cloud-init Das bedeutet, dass Sie für viele Anwendungsfälle gängige Amazon Machine Images (AMI) verwenden, und diese beim Starten einer Instance dynamisch konfigurieren können. AL2023 verwendet außerdem cloud-init zum konfigurieren des `ec2-user`-Kontos.

AL2023 nutzt die cloud-init-Aktionen in `/etc/cloud/cloud.cfg.d` und `/etc/cloud/cloud.cfg`. Sie können Ihre eigenen cloud-init-Aktionsdateien im `/etc/cloud/cloud.cfg.d`-Verzeichnis erstellen. Cloud-init liest alle Dateien in diesem Verzeichnis in lexikografischer Reihenfolge. Spätere Dateien überschreiben Werte in früher gelesenen Dateien. Wenn cloud-init eine Instanz startet, führt das cloud-init-Paket die folgenden Konfigurationsaufgaben aus:

- Festlegung des Standard-Gebietsschemas
- Festlegung des Hostnamens.
- Parsen und verarbeiten der Benutzerdaten
- Generierung privater SSH-Schlüssel für den Host
- Hinzufügung öffentlicher SSH-Schlüssel eines Benutzers zu `.ssh/authorized_keys` für vereinfachte Anmeldung und Verwaltung
- Vorbereitung der Repositorys für die Paketverwaltung.
- Durchführung von in Benutzerdaten definierten Paketaktionen
- Ausführung von Benutzerskripts in Benutzerdaten
- Mounten von Instance-Speicher-Volumes (wo zutreffend)
  - Das `ephemeral0`-Instance-Speicher-Volume ist standardmäßig vorhanden und enthält ein gültiges Dateisystem. Das Instance-Speicher-Volume wird unter `/media/ephemeral0` gemountet. Sonst wird es nicht gemountet.
  - Standardmäßig werden alle Swap-Volumes für die Instance-Typen `m1.small` und `c1.medium` gemountet, die der Instance zugeordnet sind.
  - Sie können das Mounting für ein Standard-Instance-Speicher-Volume mithilfe der folgenden cloud-init-Anweisung überschreiben:

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

    Weitere Informationen zum Steuern von Mountvorgängen finden Sie unter Mounts in der cloud-init-Dokumentation.

- Wenn eine Instance gestartet wird, werden Instance-Speicher-Volumes, die TRIM unterstützen, nicht formatiert. Sie müssen Instance-Speicher-Volumes partitionieren und formatieren, bevor Sie diese mounten können.

  Weitere Informationen finden Sie unter TRIM-Unterstützung für Instance Store Volume im Amazon EC2 EC2-Benutzerhandbuch.

- Wenn Sie Ihre Instances starten, können Sie das `disk_setup`-Modul zur Partitionierung und Formatierung Ihrer Instance-Speicher-Volumes verwenden.

  Weitere Informationen finden Sie unter Festplatteneinrichtung in der cloud-init-Dokumentation.

Weitere Informationen zur Verwendung von cloud-init mit SELinux finden Sie unter Den cloud-init-Modus mit `enforcing` aktivieren.

Informationen zu cloud-init-Benutzerdatenformaten finden Sie unter Benutzerdatenformate in der cloud-init-Dokumentation.

Sicherheits-Updates und Features

AL2023 bietet viele Sicherheitsupdates und -lösungen.

Themen

- Verwaltete Updates
- Sicherheit in der Cloud
- SELinux-Modi
- Compliance-Programm
- SSH-Server-Standard
- Hauptfunktionen von OpenSSL 3

Verwaltete Updates

Wenden Sie Sicherheitsupdates mithilfe von DNF Repository-Versionen an. Weitere Informationen finden Sie unter Paket- und Betriebssystemupdates in AL2023 verwalten.

Sicherheit in der Cloud

Sicherheit ist eine gemeinsame Verantwortung zwischen Ihnen AWS und Ihnen. Das Modell der gemeinsamen Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud. Weitere Informationen finden Sie unter Sicherheit und Compliance in Amazon Linux 2.

SELinux-Modi

Standardmäßig ist SELinux aktiviert und in AL2023 auf den permissiven Modus gesetzt. Im permissiven Modus werden Zugriffsverweigerungen protokolliert, aber nicht durchgesetzt.

Die SELinux-Richtlinien definieren Berechtigungen für Benutzer, Prozesse, Programme, Dateien und Geräte. Mit SELinux können Sie eine von zwei Richtlinien auswählen. Bei den Richtlinien handelt es sich um gezielte oder mehrstufige Sicherheitsrichtlinien (MLS).

Weitere Informationen zu den SELinux-Modi und -Richtlinien finden Sie unter Einstellung der SELinux-Modi für AL2023 und im SELinux-Projekt-Wiki.

Compliance-Programm

Unabhängige Prüfer bewerten die Sicherheit und Konformität von AL2023 sowie vieler AWS Compliance-Programme.

SSH-Server-Standard

AL2023 beinhaltet OpenSSH 8.7. OpenSSH 8.7 deaktiviert standardmäßig den `ssh-rsa`-Schlüsselaustausch-Algorithmus. Weitere Informationen finden Sie unter Standard-SSH-Serverkonfiguration.

Hauptfunktionen von OpenSSL 3

- Das Certificate Management Protocol (CMP, RFC 4210) umfasst sowohl CRMF (RFC 4211) als auch HTTP-Übertragung (RFC 6712).
- Ein HTTP- oder HTTPS-Client in libcrypto unterstützt GET- und POST-Aktionen, Umleitungen, Klartext- und ASN.1-verschlüsselte Inhalte, Proxys und Timeouts.
- EVP_KDF arbeitet mit Schlüsselableitungsfunktionen.
- EVP_MAC API arbeitet mit MACs.
- Linux-Kernel-TLS-Unterstützung.

Weitere Informationen finden Sie im OpenSSL-Migrationshandbuch.

Netzwerkdienst

Das Open-Source-Projekt `systemd-networkd` ist in modernen Linux-Distributionen weit verbreitet. Das Projekt verwendet eine deklarative Konfigurationssprache, die dem Rest des `systemd`-Frameworks ähnelt. Die wichtigsten Konfigurationsdateitypen sind `.network`- und `.link`-Dateien.

Das `amazon-ec2-net-utils`-Paket generiert schnittstellenspezifische Konfigurationen im `/run/systemd/network`-Verzeichnis. Diese Konfigurationen ermöglichen sowohl IPv4- als auch IPv6-Netzwerke auf Schnittstellen, wenn diese mit einer Instance verbunden sind. Diese Konfigurationen installieren außerdem Richtlinien-Routing-Regeln, die sicherstellen, dass der lokale Datenverkehr über die Netzwerkschnittstelle der entsprechenden Instance an das Netzwerk weitergeleitet wird. Diese Regeln stellen sicher, dass der richtige Traffic von den zugehörigen Adressen oder Präfixen über das Elastic Network Interface (ENI) geleitet wird. Weitere Informationen zur Verwendung von ENI finden Sie unter Verwenden von ENI im Amazon EC2 EC2-Benutzerhandbuch.

Sie können dieses Netzwerkverhalten anpassen, indem Sie eine benutzerdefinierte Konfigurationsdatei im `/etc/systemd/network`-Verzeichnis hinterlegen, um die Standardkonfigurationseinstellungen in `/run/systemd/network` zu überschreiben.

In der systemd.network-Dokumentation wird beschrieben, wie der `systemd-networkd`-Dienst die Konfiguration für eine bestimmte Schnittstelle festlegt. Es generiert auch alternative Namen, sogenanntealtnames, für die ENI-gestützten Schnittstellen, um die Eigenschaften verschiedener Ressourcen widerzuspiegeln. AWS Diese ENI-gestützten Schnittstelleneigenschaften sind die Felder `ENI ID` und `DeviceIndex` im ENI-Anhang. Sie können bei Nutzung verschiedener Tools (z. B. dem `ip`-Befehl) mithilfe ihrer Eigenschaften auf diese Schnittstellen verweisen.

Die Namen der AL203-Instanzschnittstellen werden mithilfe des `systemd` Slot-Benennungsschemas generiert. Weitere Informationen finden Sie unter systemd.net-Namensschema.

Darüber hinaus verwendet AL2023 standardmäßig den `fq_codel`-Planungsalgorithmus für Netzwerkübertragungen für die aktive Warteschlangenverwaltung. Weitere Informationen finden Sie in der CoDelÜbersicht.

Core-Toolchain-Paketeglibc, gcc und binutils

Eine Teilmenge der Pakete in Amazon Linux wird als Core-Toolchain-Pakete bezeichnet. Als Hauptbestandteil von AL2023 erhalten Kernpakete fünf Jahre Support. Die Version eines Pakets

kann eventuell verändert werden, aber der langfristige Support gilt für das jeweilige in der Amazon-Linux-Version enthaltene Paket.

Diese drei Core-Pakete stellen eine System-Toolchain bereit, mit der die meiste Software in der Amazon Linux-Distribution erstellt wird.

Paket	Definition	Zweck
glibc 2.34	System-C-Bibliothek	Wird von den meisten Binärprogrammen verwendet , die Standardfunktionen bereitstellen sowie von der Schnittstelle zwischen Programmen und dem Kernel.
gcc 11.2	gcc-Compiler-Suite	Kompiliert C, C++ und Fortran.
binutils 2.35	Assembler und Linker sowie andere binäre Tools	Manipuliert oder untersucht Binärprogramme.

Wir empfehlen jeweils nach einer Aktualisierung der glibc-Bibliotheken einen Neustart durchzuführen. Bei Aktualisierungen der Pakete, die einen Service steuern, reicht möglicherweise ein Neustart des Service aus, um die Aktualisierungen zu aktivieren. Ein Systemneustart stellt jedoch sicher, dass alle vorherigen Paket- und Bibliotheks-Updates abgeschlossen werden.

Paketmanagement-Tool

Das Standardtool zur Verwaltung von Softwarepaketen in AL2023 ist. DNF DNFist der Nachfolger des Paketverwaltungstools in AL2. YUM

Die Nutzungsweisen von DNF und YUM sind ähnlich. Viele DNF Befehle und Befehlsoptionen sind mit YUM Befehlen identisch. In einem Befehlszeilenschnittstellen-(CLI)-Befehl ersetzt `dnf` `yum` in den meisten Fällen.

Zum Beispiel für die folgenden `yum` AL2-Befehle:

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

```
$ sudo yum remove packagename
```

In AL2023 werden sie zu den folgenden Befehlen:

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

Der `yum`-Befehl ist in AL2023 immer noch verfügbar, jedoch als Verweis auf den `dnf`-Befehl. Wenn der `yum`-Befehl also in der Shell oder in einem Skript verwendet wird, sind alle Befehle und Optionen dieselben wie bei DNF CLI. Weitere Informationen zu den Unterschieden zwischen YUM CLI und DNF CLI finden Sie unter Änderungen in DNF CLI im Vergleich zu YUM.

Eine vollständige Liste aller Befehle und Optionen für den `dnf`-Befehl finden Sie in der Startseite `man dnf`. Weitere Informationen finden Sie in der DNFBefehlsreferenz.

Standard-SSH-Serverkonfiguration

Wenn Sie SSH-Clients haben, die mehrere Jahre alt sind, wird möglicherweise ein Fehler angezeigt, wenn Sie eine Verbindung zu einer Instance herstellen. Wenn Ihnen der Fehler anzeigt, dass kein passender Host-Schlüsseltyp gefunden wurde, sollten Sie Ihren SSH-Hostschlüssel aktualisieren.

Standardmäßige Deaktivierung von **`ssh-rsa`**-Signaturen

AL2023 enthält eine Standardkonfiguration, die den alten `ssh-rsa` Hostschlüsselalgorithmus deaktiviert und einen reduzierten Satz von Hostschlüsseln generiert. Clients müssen den `ssh-ed25519`- oder `-ecdsa-sha2-nistp256`-Host-Schlüsselalgorithmus unterstützen.

Die Standardkonfiguration akzeptiert jeden der folgenden Schlüsselaustauschalgorithmen:

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

- diffie-hellman-group16-sha512
- diffie-hellman-group18-sha512

Standardmäßig generiert AL2023 ed25519- und ECDSA-Hostschlüssel. Clients unterstützen entweder den ssh-ed25519- oder -ecdsa-sha2-nistp256-Host-Schlüsselalgorithmus. Wenn Sie eine Verbindung zu einer Instance über SSH herstellen, müssen Sie einen Client verwenden, der einen kompatiblen Algorithmus unterstützt, z. B. ssh-ed25519 oder ecdsa-sha2-nistp256. Wenn Sie andere Schlüsseltypen verwenden müssen, überschreiben Sie die Liste der generierten Schlüssel mit einem cloud-config-Fragment in den Benutzerdaten.

Im folgenden Beispiel generiert cloud-config einen rsa Hostschlüssel mit den ed25519- und ecdsa-Schlüsseln.

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

Wenn Sie ein RSA-Schlüsselpaar zur Authentifizierung eines öffentlichen Schlüssels verwenden, muss Ihr SSH-Client eine rsa-sha2-256- oder rsa-sha2-512-Signatur unterstützen. Wenn Sie einen inkompatiblen Client verwenden und daher kein Upgrade durchführen können, aktivieren Sie den ssh-rsa-Support für Ihre Instance erneut. Um die ssh-rsa Unterstützung wieder zu aktivieren, aktivieren Sie die LEGACY System-Kryptorichtlinie mit den folgenden Befehlen.

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

Weitere Informationen zur Verwaltung von Hostschlüsseln finden Sie unter Amazon Linux-Hostschlüssel.

Veraltete Funktionalität in AL2023

Funktionen, die in AL2 veraltet sind und in AL2023 nicht vorhanden sind, sind hier dokumentiert. Dabei handelt es sich um Funktionen wie Funktionen und Pakete, die in AL2, aber nicht in AL2023 vorhanden sind und nicht zu AL2023 hinzugefügt werden. Weitere Informationen darüber, wie lange die Funktionalität in AL2 unterstützt wird, finden Sie unter Veraltete Funktionen in AL2.

Es gibt auch Funktionen in AL2023, die veraltet sind und in einer future Version entfernt werden. In diesem Kapitel wird beschrieben, was diese Funktionalität ist, wann sie nicht mehr unterstützt wird und wann sie aus Amazon Linux entfernt wird. Wenn Sie die veralteten Funktionen verstehen, können Sie AL2023 bereitstellen und sich auf die nächste Hauptversion von Amazon Linux vorbereiten.

Themen

- compat--Pakete
- Veraltete Funktionalität in AL1 eingestellt, in AL2 entfernt
- Funktionalität in AL2 veraltet und in AL2023 entfernt
- In AL2023 veraltet

`compat-`-Pakete

Alle Pakete in AL2 mit dem Präfix von `compat-` werden aus Gründen der Binärkompatibilität mit älteren Binärdateien bereitgestellt, die noch nicht für moderne Versionen des Pakets neu erstellt wurden. Jede neue Hauptversion von Amazon Linux wird keine `compat-` Pakete aus früheren Versionen übernehmen.

Alle `compat-` Pakete in einer Version von Amazon Linux (z. B. AL2) sind veraltet und in der nachfolgenden Version (z. B. AL2023) nicht vorhanden. Wir empfehlen dringend, die Software anhand der aktualisierten Versionen der Bibliotheken neu zu erstellen.

Veraltete Funktionalität in AL1 eingestellt, in AL2 entfernt

In diesem Abschnitt werden Funktionen beschrieben, die in AL1 verfügbar sind und in AL2 nicht mehr verfügbar sind.

> **Note**
>
> Im Rahmen der Wartungsunterstützungsphase von AL1 hatten einige Pakete ein end-of-life (EOL-) Datum, das vor dem EOL von AL1 lag. Weitere Informationen finden Sie in den Unterstützungserklärungen für das AL1-Paket.

> **Note**
>
> Einige AL1-Funktionen wurden in früheren Versionen eingestellt. Informationen finden Sie in den AL1-Versionshinweisen.

Themen

- 32-Bit-x86-AMIs (i686)
- aws-apitools-*ersetzt durch AWS CLI
- systemdersetzt in AL2 upstart

32-Bit-x86-AMIs (i686)

Im Rahmen der Version 2014.09 von AL1 kündigte Amazon Linux an, dass dies die letzte Version sein wird, die 32-Bit-AMIs produziert. Daher unterstützt Amazon Linux ab Version 2015.03 von AL1 nicht mehr die Ausführung des Systems im 32-Bit-Modus. AL2 bietet eingeschränkte Laufzeitunterstützung für 32-Bit-Binärdateien auf x86-64-Hosts und stellt keine Entwicklungspakete zur Verfügung, um die Erstellung neuer 32-Bit-Binärdateien zu ermöglichen. AL2023 enthält keine 32-Bit-User-Space-Pakete mehr. Wir empfehlen Benutzern, ihre Umstellung auf 64-Bit-Code abzuschließen, bevor sie zu AL2023 migrieren.

Wenn Sie 32-Bit-Binärdateien auf AL2023 ausführen müssen, ist es möglich, den 32-Bit-Userspace von AL2 in einem AL2-Container zu verwenden, der auf AL2023 läuft.

`aws-apitools-*`ersetzt durch AWS CLI

Vor der Veröffentlichung von AWS CLI im September 2013 wurde eine Reihe von Befehlszeilendienstprogrammen zur Verfügung AWS gestellt, die in implementiert wurden und es Benutzern ermöglichtenJava, Amazon EC2 EC2-API-Aufrufe zu tätigen. Diese Tools wurden 2015 eingestellt und wurden zur bevorzugten Methode, über die Befehlszeile mit Amazon EC2 EC2-APIs

zu interagieren. AWS CLI Der Satz an Befehlszeilen-Hilfsprogrammen umfasst die folgenden `aws-apitools-*` Pakete.

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

Der Upstream-Support für die `aws-apitools-*` Pakete endete im März 2017. Trotz des Mangels an Upstream-Unterstützung lieferte Amazon Linux weiterhin einige dieser Befehlszeilenprogramme aus, z. B. `aws-apitools-ec2` um Benutzern Abwärtskompatibilität zu bieten. Es AWS CLI ist ein robusteres und vollständigeres Tool als die `aws-apitools-*` Pakete, da es aktiv gewartet wird und die Möglichkeit bietet, alle AWS APIs zu nutzen.

Die `aws-apitools-*` Pakete wurden im März 2017 als veraltet eingestuft und werden keine weiteren Updates erhalten. Alle Benutzer eines dieser Pakete sollten AWS CLI so schnell wie möglich auf das migrieren. Diese Pakete sind in AL2023 nicht vorhanden.

AL1 stellte auch die `aws-apitools-rds` Pakete `aws-apitools-iam` und bereit, die in AL1 veraltet waren und ab AL2 nicht mehr in Amazon Linux vorhanden sind.

`systemd`ersetzt in AL2 **`upstart`**

AL2 war die erste Amazon Linux-Version, die das `systemd` Init-System verwendete und AL1 `upstart` ersetzte. Jede `upstart` spezifische Konfiguration muss im Rahmen der Migration von AL1 auf eine neuere Version von Amazon Linux geändert werden. Die Verwendung `systemd` auf AL1 ist nicht möglich, daher `systemd` kann der Wechsel von `upstart` zu nur im Rahmen der Umstellung auf eine neuere Hauptversion von Amazon Linux wie AL2 oder AL2023 erfolgen.

Funktionalität in AL2 veraltet und in AL2023 entfernt

In diesem Abschnitt werden Funktionen beschrieben, die in AL2 verfügbar und in AL2023 nicht mehr verfügbar sind.

Themen

- 32-Bit-x86-Pakete (i686)
- aws-apitools-*ersetzt durch AWS CLI
- bzrRevisionskontrollsystem
- cgroup v1
- log4jHotpatch () log4j-cve-2021-44228-hotpatch
- lsb_release und das system-lsb-core-Paket
- mcrypt
- OpenJDK (7) java-1.7.0-openjdk
- Python 2.7
- rsyslog-opensslersetzt rsyslog-gnutls
- Netzwerkinformationsdienst (NIS)/yp

32-Bit-x86-Pakete (i686)

Im Rahmen der Version 2014.09 von AL1 haben wir angekündigt, dass dies die letzte Version sein wird, die 32-Bit-AMIs produziert. Daher unterstützt Amazon Linux ab Version 2015.03 von AL1 nicht mehr die Ausführung des Systems im 32-Bit-Modus. AL2 bietet eingeschränkte Laufzeitunterstützung für 32-Bit-Binärdateien auf x86-64-Hosts und stellt keine Entwicklungspakete zur Verfügung, um die Erstellung neuer 32-Bit-Binärdateien zu ermöglichen. AL2023 enthält keine 32-Bit-Userspace-Pakete mehr. Wir empfehlen unseren Kunden, die Umstellung auf 64-Bit-Code abzuschließen.

Wenn Sie 32-Bit-Binärdateien auf AL2023 ausführen müssen, ist es möglich, den 32-Bit-Userspace von AL2 in einem AL2-Container zu verwenden, der auf AL2023 läuft.

aws-apitools-*ersetzt durch AWS CLI

Vor der Veröffentlichung von AWS CLI im September 2013 wurde eine Reihe von Befehlszeilendienstprogrammen zur Verfügung AWS gestellt, die in implementiert wurden und es Kunden ermöglichtenJava, Amazon EC2 EC2-API-Aufrufe zu tätigen. Diese Tools wurden 2015 als veraltet eingestuft und wurden zur bevorzugten Methode für AWS CLI die Interaktion mit Amazon EC2 EC2-APIs über die Befehlszeile. Dies beinhaltet die folgenden Pakete. `aws-apitools-*`

- `aws-apitools-as`
- `aws-apitools-cfn`
- `aws-apitools-common`

- `aws-apitools-ec2`
- `aws-apitools-elb`
- `aws-apitools-mon`

Der Upstream-Support für die `aws-apitools-*` Pakete endete im März 2017. Trotz des Mangels an Upstream-Unterstützung lieferte Amazon Linux weiterhin einige dieser Befehlszeilenprogramme (wie`aws-apitools-ec2`) aus, um Kunden Abwärtskompatibilität zu bieten. Es AWS CLI ist ein robusteres und vollständigeres Tool als die `aws-apitools-*` Pakete, da es aktiv gewartet wird und die Möglichkeit bietet, alle AWS APIs zu nutzen.

Die `aws-apitools-*` Pakete wurden im März 2017 als veraltet eingestuft und werden keine weiteren Updates erhalten. Alle Benutzer eines dieser Pakete sollten AWS CLI so schnell wie möglich auf das migrieren. Diese Pakete sind in AL2023 nicht vorhanden.

`bzr`Revisionskontrollsystem

Das Revisionskontrollsystem GNU Bazaar(`bzr`) wurde in AL2 eingestellt und ist in AL2023 nicht mehr vorhanden.

Benutzern von wird `bzr` empfohlen, ihre Repositorien zu migrieren. `git`

cgroup v1

AL2023 wechselt zur Unified Control Group-Hierarchie (cgroup v2), wohingegen AL2 cgroup v1 verwendet. Da AL2 cgroup v2 nicht unterstützt, muss diese Migration im Rahmen der Umstellung auf AL2023 abgeschlossen werden.

log4jHotpatch () **`log4j-cve-2021-44228-hotpatch`**

> **Note**
>
> Das `log4j-cve-2021-44228-hotpatch` Paket ist in AL2 veraltet und wurde in AL2023 entfernt.

Als Reaktion auf CVE-2021-44228 veröffentlichte Amazon Linux eine RPM-Paketversion des Hotpatches für Apache Log4j für AL1 und AL2. In der Ankündigung der Hinzufügung des Hotpatches

zu Amazon Linux haben wir festgestellt, dass „die Installation des Hotpatches kein Ersatz für die Aktualisierung auf eine log4j-Version ist, die CVE-2021-44228 oder CVE-2021-45046 abmildert“.

Der Hotpatch diente lediglich als Abhilfemaßnahme, um mehr Zeit für den `log4j`-Patch zu gewinnen. Die erste allgemein verfügbare Version von AL2023 wurde 15 Monate nach CVE-2021-44228 veröffentlicht, sodass AL2023 nicht mit dem Hotpatch ausgeliefert wird (aktiviert oder nicht).

Kunden, die ihre eigenen `log4j`-Versionen auf Amazon Linux ausführen, sollten sicherstellen, dass sie auf Versionen aktualisiert haben, die nicht von CVE-2021-44228 oder CVE-2021-45046 betroffen sind.

`lsb_release` und das `system-lsb-core`-Paket

In der Vergangenheit wurde der `lsb_release`-Befehl (in AL2 im `system-lsb-core`-Paket enthalten) von einigen Programmen aufgerufen, um Informationen über die Linux-Distribution zu erhalten, auf der sie ausgeführt wurden. Dieser Befehl wurde von Linux Standards Base (LSB) eingeführt und wurden von den Linux-Distributionen übernommen. Linux-Distributionen haben sich weiterentwickelt, sodass der einfachere Standard für die Speicherung dieser Informationen in `/etc/os-release` und anderen verwandten Dateien verwendet wird.

Der `os-release`-Standard stammt aus `systemd`. Weitere Informationen finden Sie in der systemd os-Versionsdokumentation.

AL2023 wird nicht mit dem `lsb_release`-Befehl ausgeliefert und beinhaltet auch nicht das `system-lsb-core`-Paket. Die Software sollte die Umstellung auf den `os-release`-Standard abschließen, um die Kompatibilität mit Amazon Linux und anderen wichtigen Linux-Distributionen aufrechtzuerhalten.

`mcrypt`

Die `mcrypt` Bibliothek und die zugehörige PHP Erweiterung waren in AL2 veraltet und sind in AL2023 nicht mehr vorhanden.

Upstream PHP hat die `mcrypt` Erweiterung in PHP 7.1, die erstmals im Dezember 2016 veröffentlicht wurde und im Oktober 2019 endgültig veröffentlicht wurde, als veraltet eingestuft.

Die `mcrypt` Upstream-Bibliothek wurde zuletzt 2007 veröffentlicht und hat nicht die Migration von der `cvs` Versionskontrolle vorgenommen, die 2017 für neue Commits SourceForge erforderlich war. Der letzte Commit (und nur für 3 Jahre davor) stammt aus dem Jahr 2011, wodurch die Erwähnung, dass das Projekt einen Betreuer hat, weggelassen wurde.

Allen verbleibenden Benutzern von wird `mcrypt` empfohlen, ihren Code auf AL2023 zu portierenOpenSSL, da dieser nicht zu AL2023 hinzugefügt `mcrypt` wird.

OpenJDK (7) `java-1.7.0-openjdk`

> **Note**
>
> AL2023 bietet mehrere Versionen von Amazon Corretto zur Unterstützung Java von basierten Workloads. Die OpenJDK 7-Pakete sind in AL2 veraltet und in AL2023 nicht mehr vorhanden. Das älteste in AL2023 verfügbare JDK wird von Corretto 8 bereitgestellt.

Weitere Informationen zu Java auf Amazon Linux finden Sie unterJava in AL2023.

Python 2.7

> **Note**
>
> In AL2023 wurde Python 2.7 komplett entfernt, sodass alle Python-abhängigen Betriebssystemkomponenten entsprechend umgeschrieben wurden, damit sie jetzt mit Python 3 arbeiten. Wenn Sie also weiterhin eine von Amazon Linux bereitgestellte und unterstützte Python-Version verwenden möchten, müssen Sie Ihren Python-2-Code in Python 3 konvertieren.

Weitere Informationen zu Python auf Amazon Linux finden Sie unterPython in AL2023.

`rsyslog-openssl`ersetzt `rsyslog-gnutls`

Das `rsyslog-gnutls` Paket ist in AL2 veraltet und in AL2023 nicht mehr vorhanden. Das `rsyslog-openssl` Paket sollte ein direkter Ersatz für jegliche Nutzung des Pakets sein. `rsyslog-gnutls`

Netzwerkinformationsdienst (NIS)/`yp`

Der Network Information Service (NIS), ursprünglich Yellow Pages genannt oder YP ist in AL2 veraltet und in AL2023 nicht mehr vorhanden. Dies beinhaltet die folgenden Pakete:`ypbind`, und. `ypserv` `yp-tools` Bei anderen Paketen, die sich in integrieren NIS lassen, wurde diese Funktionalität in AL2023 entfernt.

In AL2023 veraltet

In diesem Abschnitt werden Funktionen beschrieben, die in AL2023 vorhanden sind und wahrscheinlich in einer future Version von Amazon Linux entfernt werden. In jedem Abschnitt wird beschrieben, um welche Funktionen es sich handelt und wann sie voraussichtlich aus Amazon Linux entfernt werden.

> **Note**
>
> Dieser Abschnitt wird im Laufe der Zeit aktualisiert, da sich das Linux-Ökosystem weiterentwickelt und future Hauptversionen von Amazon Linux kurz vor der Veröffentlichung stehen.

Themen

- 32-Bit-x86-Laufzeitunterstützung (i686)
- Berkeley-Datenbank () libdb
- cron
- IMDSv1
- pcre Version 1
- System V init (sysvinit)

32-Bit-x86-Laufzeitunterstützung (i686)

AL2023 behält die Fähigkeit, 32-Bit-x86-Binärdateien (i686) auszuführen. Es ist wahrscheinlich, dass die nächste Hauptversion von Amazon Linux die Ausführung von 32-Bit-User-Space-Binärdateien nicht mehr unterstützt.

Berkeley-Datenbank () `libdb`

AL2023 wird mit Version 5.3.28 der Berkeley DB () -Bibliothek ausgeliefert. `libdb` Dies ist die letzte Version von Berkeley DB, bevor die Lizenz von der weniger restriktiven Sleepycat-Lizenz zur GNU Affero GPLv3 (AGPL) -Lizenz geändert wurde.

Es gibt nur wenige Pakete in AL2023, die weiterhin auf Berkeley DB (`libdb`) angewiesen sind, und die Bibliothek wird in der nächsten Hauptversion von Amazon Linux entfernt.

> **Note**
>
> Der `dnf` Paketmanager in AL2023 unterstützt weiterhin schreibgeschützte Datenbanken im Berkeley DB (BDB) -Format. `rpm` Diese Unterstützung wird in der nächsten Hauptversion von Amazon Linux entfernt.

`cron`

Das `cronie`-Paket wurde auf dem AL2-AMI standardmäßig installiert und bot Unterstützung für die herkömmliche `crontab`-Methode der Planung periodischer Aufgaben. In AL2023 `cronie` ist sie standardmäßig nicht enthalten. Daher `crontab` wird die Unterstützung für standardmäßig nicht mehr bereitgestellt.

In AL2023 können Sie das `cronie` Paket optional installieren, um klassische `cron` Jobs zu verwenden. Aufgrund der zusätzlichen Funktionalitäten in `systemd` empfehlen wir, auf `systemd`-Timer zu migrieren.

Es ist möglich, dass eine future Version von Amazon Linux, möglicherweise die nächste Hauptversion, keine Unterstützung für klassische `cron` Jobs mehr bietet und die Umstellung auf `systemd` Timer abschließt. Wir empfehlen Ihnen, von der Verwendung `cron` wegzugehen.

IMDSv1

Standardmäßig sind AL2023-AMIs so konfiguriert, dass sie im Modus „IMDSv2Nur“ gestartet werden, wodurch die Verwendung von deaktiviert wird. IMDSv1 Es besteht weiterhin die Möglichkeit, AL2023 mit aktiviertem IMDSv1 zu verwenden. Eine future Version von Amazon Linux wird wahrscheinlich IMDSv2 -only erzwingen.

Weitere Informationen zur IMDS-Konfiguration für AMIs finden Sie unter Configure the AMI im Amazon EC2 EC2-Benutzerhandbuch.

`pcre` Version 1

Das `pcre` Legacy-Paket ist veraltet und wird in der nächsten Hauptversion von Amazon Linux entfernt. Das `pcre2`-Paket ist der Nachfolger. Obwohl die ersten Versionen von AL2023 mit einer begrenzten Anzahl von Paketen ausgeliefert wurden, die darauf aufbauen`pcre`, werden diese Pakete innerhalb von AL2023 migriert. `pcre2` Die veraltete `pcre` Bibliothek wird in AL2023 weiterhin verfügbar sein.

> **Note**
>
> Die veraltete Version von `pcre` wird während der gesamten Lebensdauer von AL2023 keine Sicherheitsupdates erhalten. Weitere Informationen über den `pcre` Support-Lebenszyklus und den Zeitraum, in dem das Paket Sicherheitsupdates erhält, finden Sie in den Support-Anweisungen zum Paket. `pcre`

System V init (`sysvinit`)

Obwohl AL2023 die Abwärtskompatibilität mit System V service (init) -Skripten beibehält, kündigte das `systemd` Upstream-Projekt im Rahmen seiner Version v254 an, dass die Unterstützung für System V-Dienstskripte eingestellt wird, und gab an, dass die Unterstützung in einer future Version von entfernt wird. `systemd` Weitere Informationen finden Sie unter systemd.

AL2023 wird die Abwärtskompatibilität mit System V service (init) -Skripten beibehalten, aber Benutzern wird empfohlen, auf die Verwendung nativer `systemd` Unit-Dateien umzusteigen, um darauf vorbereitet zu sein, wenn die Unterstützung für System V service (init) -Skripte aus Amazon Linux entfernt wird, was wahrscheinlich in der nächsten Hauptversion der Fall sein wird.

Vergleich von AL2 und AL2023

In den folgenden Themen werden die wichtigsten Unterschiede zwischen AL2 und AL2023 beschrieben.

Themen

- Hinzugefügte, aktualisierte und entfernte Pakete
- Support für die einzelnen Versionen
- Änderungen bei der Benennung und Versionierung
- Optimierungen
- Python 2.7 wurden durch Python 3 ersetzt
- Sicherheits-Updates
- Deterministische Upgrades für Stabilität
- Aus mehreren Upstream-Quellen
- AMI-Root-Dateisystem und standardmäßiger Amazon-EBS-Volume-Typ
- Netzwerksystem-Dienst
- Vereinheitlichte Kontrollgruppenhierarchie (cgroup v2)
- Aufgabenplanung
- Pakete für glibc, gcc und binutils
- Paketmanager
- Protokollierungssystem
- Paketänderungen für curl und libcurl
- GNU Privacy Guard (GNUPG)
- Amazon Corretto als Standard-JVM
- AWS CLI v2
- UEFI Preferred
- Änderungen der Standardkonfiguration des SSH-Servers
- Extra Packages for Enterprise Linux (EPEL)
- Verwenden von cloud-init
- Grafische Desktop-Unterstützung

- Compiler-Triplet
- 32-Bit x86-(i686)-Pakete
- lsb_release und das system-lsb-core-Paket
- AL2023-Kerneländerungen gegenüber AL2
- Vergleich der auf Amazon Linux 2 und Amazon Linux 2023 AMIs installierten Pakete
- Vergleich der auf Amazon Linux 2 und Amazon Linux 2023 Minimal AMIs installierten Pakete
- Vergleich der auf Amazon Linux 2 und Amazon Linux 2023 Basis-Container-Images installierten Pakete

Hinzugefügte, aktualisierte und entfernte Pakete

AL2023 enthält tausende von Software-Paketen, die Sie nutzen können. Eine vollständige Liste aller Pakete, die in AL2023 im Vergleich zu früheren Amazon Linux-Versionen hinzugefügt, aktualisiert oder entfernt wurden, finden Sie unter Paketänderungen in AL2023.

Um zu beantragen, dass ein Paket in AL2023 hinzugefügt oder geändert wird, melden Sie ein Problem im Amazon-Linux-2023-Repo unter. GitHub

Support für die einzelnen Versionen

Für AL2023 bieten wir fünf Jahre Support.

Weitere Informationen finden Sie unter Release-Taktfrequenz.

Änderungen bei der Benennung und Versionierung

AL2023 unterstützt dieselben Mechanismen wie AL2 zur Plattformidentifikation. In AL2023 wurden außerdem auch neue Dateien zur Plattformidentifikation eingeführt.

Weitere Informationen finden Sie unter Benennung und Versionsverwaltung.

Optimierungen

In AL2023 wurde die Startzeit optimiert, um die Zeit vom Start der Instance bis zur Ausführung des Kunden-Workloads zu verkürzen. Diese Optimierungen umfassen die Amazon-EC2-

Kernelkonfiguration, `cloud-init`-Konfigurationen und Features, die in Pakete im Betriebssystem integriert wurden, z. B. `kmod` und `systemd`.

Weitere Informationen zu diesen Optimierungen finden Sie unter Leistungs- und Betriebsoptimierungen.

Python 2.7 wurden durch Python 3 ersetzt

AL2 bietet bis Juni 2025 Support und Sicherheits-Patches für Python 2.7 als Teil unseres langfristigen Support-Versprechens (LTS) für AL2-Core-Pakete. Diese Unterstützung geht über die Upstream-Python-Community-Erklärung von Python 2.7 end-of-life vom Januar 2020 hinaus.

AL2 verwendet den `yum` Paketmanager, der stark von Python 2.7 abhängig ist. In AL2023 wurde der `dnf`-Paketmanager auf Python 3 migriert und benötigt Python 2.7 nicht mehr. AL2023 wurde komplett auf Python 3 umgestellt.

> **Note**
>
> In AL2023 wurde Python 2.7 komplett entfernt, sodass alle Python-abhängigen Betriebssystemkomponenten entsprechend umgeschrieben wurden, damit sie jetzt mit Python 3 arbeiten. Wenn Sie also weiterhin eine von Amazon Linux bereitgestellte und unterstützte Python-Version verwenden möchten, müssen Sie Ihren Python-2-Code in Python 3 konvertieren.

Weitere Informationen zu Python auf Amazon Linux finden Sie unter Python in AL2023.

Sicherheits-Updates

SELinux

Standardmäßig ist Security Enhanced Linux (SELinux) für AL2023 `enabled` und befindet sich im `permissive`-Modus. Im `permissive`-Modus werden Zugriffsverweigerungen protokolliert, aber nicht durchgesetzt.

SELinux ist ein Sicherheits-Feature des Amazon-Linux-Kernel und war in AL2 `disabled`. SELinux ist eine Sammlung von Kernel-Features und Hilfsprogrammen, die eine Zugriffskontrollarchitektur (MAC) für wichtige Untersysteme des Kernels erzwingt.

Weitere Informationen finden Sie unter Einstellung der SELinux-Modi für AL2023.

Weitere Informationen zu SELinux-Repositorys, -Tools und -Richtlinien finden Sie unter SELinux Notebook, SELinux-Richtlinientypen und unter SELinux-Projekt.

OpenSSL 3

AL2023 enthält das Open Secure Sockets Layer version 3 (OpenSSL 3)-Crypto-Toolkit. AL2023 unterstützt TLS 1.3- und TLS 1.2-Netzwerkprotokolle.

AL2 wird standardmäßig mit OpenSSL 1.0.2 geliefert. Sie können Anwendungen für OpenSSL 1.1.1 erstellen.

Weitere Informationen zu OpenSSL finden Sie im OpenSSL-Migrationshandbuch.

Weitere Informationen zu Sicherheit finden Sie unter Sicherheits-Updates und Features.

IMDSv2

Standardmäßig benötigen alle mit dem AL2023 AMI gestarteten Instances IMDSv2 nur -only und Ihr Standard-Hop-Limit wird auf 2 gesetzt, um die Unterstützung von containerisierten Workloads zu ermöglichen. Hierfür setzen Sie den `imds-support`-Parameter auf `v2.0`. Weitere Informationen finden Sie unter Configure the AMI im Amazon EC2 EC2-Benutzerhandbuch.

> **Note**
>
> Die Gültigkeitsdauer des Sitzungstokens kann zwischen 1 Sekunde und 6 Stunden liegen. Die Adressen, an die die API-Anfragen für IMDSv2-Anforderungen weitergeleitet werden, lauten wie folgt:
>
> - IPv4:169.254.169.254
> - IPv6: fd00:ec2:254

Sie können diese Einstellungen manuell überschreiben und IMDSv1 mithilfe der Starteigenschaften der Instance-Metadaten aktivieren. Sie können auch IAM-Steuerelemente verwenden, um verschiedene IMDS Einstellungen durchzusetzen. Weitere Informationen zum Einrichten und Verwenden des Instance-Metadaten-Service finden Sie unter Verwenden IMDSv2, Konfigurieren von Instance-Metadatenoptionen für neue Instances und Ändern von Instance-Metadatenoptionen für bestehende Instances im Amazon EC2 EC2-Benutzerhandbuch.

Entfernen des log4j-Hotpatch (`log4j-cve-2021-44228-hotpatch`)

> **Note**
>
> AL2023 wird nicht mit dem `log4j-cve-2021-44228-hotpatch`-Paket geliefert.

Als Reaktion auf CVE-2021-44228 veröffentlichte Amazon Linux eine RPM-Paketversion des Hotpatches für Apache Log4j für AL1 und AL2. In der Ankündigung des zusätzlichen Hotpatch für Amazon erwähnten wir, dass „die Hotpatch-Installations kein Ersatz für die Aktualisierung auf eine log4j-Version darstellt, und keine Abhilfe für CVE-2021-44228 oder CVE-2021-45046 bietet“.

Der Hotpatch diente lediglich als Abhilfemaßnahme, um mehr Zeit für den `log4j`-Patch zu gewinnen. Die erste allgemein verfügbare (General Availability, GA) Version von AL2023 erschien 15 Monate nach CVE-2021-44228, weshalb AL2023 nicht mit dem Hotpatch ausgeliefert wird (aktiviert oder nicht).

Benutzer, die ihre eigenen `log4j` Versionen auf Amazon Linux ausführen, sollten sicherstellen, dass sie auf Versionen aktualisiert haben, die nicht von CVE-2021-44228 oder CVE-2021-45046 betroffen sind.

AL2023 bietet Anleitungen zu Aktualisierung von AL2023, damit Sie über Sicherheits-Patches auf dem Laufenden bleiben. Sicherheitsempfehlungen werden im Amazon Linux Security Center veröffentlicht.

Deterministische Upgrades für Stabilität

Mit der Funktion „Deterministische Upgrades durch versionierte Repositorys“ ist jedes AL203-AMI standardmäßig an eine bestimmte Repository-Version gebunden. Mit deterministische Upgrades schaffen Sie eine bessere Konsistenz zwischen Paketversionen und Updates. Jede Haupt- oder Nebenversion enthält eine bestimmte Repository-Version.

Neu in AL2023 ist, dass das deterministische Upgrade standardmäßig aktiviert ist. Dies ist eine Verbesserung gegenüber der manuellen, inkrementellen Sperrmethode, die in AL2 und anderen früheren Versionen verwendet wurde.

Weitere Informationen finden Sie unter Verwendung deterministischer Upgrades über ein versioniertes Repository auf AL2023.

Aus mehreren Upstream-Quellen

AL2023 ist RPM-basiert und enthält Komponenten, die aus mehreren Versionen von Fedora und anderen Distributionen wie CentOS 9 Stream stammen. Der Amazon Linux-Kernel stammt aus den Long-Term-Support-(LTS)-Versionen direkt von kernel.org, die unabhängig von anderen Distributionen ausgewählt wurden.

Weitere Informationen finden Sie unter Beziehung zu Fedora.

AMI-Root-Dateisystem und standardmäßiger Amazon-EBS-Volume-Typ

AL2023 AMI und AL2 verwenden beide das XFS-Dateisystem auf dem Root-Dateisystem. Für AL2023 wurden die `mkfs`-Optionen für Amazon EC2 für das Root-Geräte-Dateisystem weiter optimiert. AL2023 unterstützt außerdem auch eine Reihe anderer Dateisysteme, die Sie je nach Bedarf auf anderen Volumes verwenden können.

AL2023-AMIs verwenden standardmäßig Amazon-EBS-`gp3`-Volumes, wohingegen AL2-AMIs standardmäßig Amazon-EBS-`gp2`-Volumes verwenden. Sie können den Volume-Typ beim Start einer Instance wechseln.

Weitere Information zu Amazon-EBS-Volume-Typen finden Sie unter Amazon-EBS-Allzweck-Volumes.

Weitere Informationen zum Starten einer Amazon EC2 EC2-Instance finden Sie unter Launch an Instance im Amazon EC2 EC2-Benutzerhandbuch.

Netzwerksystem-Dienst

Der `systemd-networkd`-Systemdienst verwaltet die Netzwerkschnittstellen in AL2023. Dies ist eine Änderung gegenüber AL2, bei dem ISC dhclient oder `dhclient` verwendet wird.

Weitere Informationen finden Sie unter Netzwerkdienst.

Vereinheitlichte Kontrollgruppenhierarchie (cgroup v2)

Eine Kontrollgruppe (cgroup) ist eine Linux-Kernelfunktion zur hierarchischen Organisation von Prozessen und zur Verteilung von entsprechenden Systemressourcen. Kontrollgruppen werden häufig zur Implementierung einer Container-Laufzeit und von `systemd` verwendet.

AL2 unterstützt und cgroupv1 AL2023 unterstützt. cgroupv2 Dies ist gut zu wissen, wenn containerisierte Workloads ausgeführt werden, z. B. bei Verwendung von AL2023-basierten Amazon ECS-AMIs zum Hosten containerisierter Workloads.

AL2023 enthält zwar weiterhin Code, mit dem das System ausgeführt werden kanncgroupv1, dies ist jedoch keine empfohlene oder unterstützte Konfiguration und wird in einer future Hauptversion von Amazon Linux vollständig entfernt.

Es gibt umfangreiche Dokumentation zu Low-Level-Linux-Kernel-Schnittstellen sowie zur Delegierung von systemd cgroup.

Ein häufiger Anwendungsfall außerhalb von Containern ist die Erstellung von `systemd` Einheiten, bei denen die Systemressourcen, die sie verwenden können, begrenzt sind. Weitere Informationen finden Sie unter systemd.resource-control.

Aufgabenplanung

Das `cronie`-Paket wurde auf dem AL2-AMI standardmäßig installiert und bot Unterstützung für die herkömmliche `crontab`-Methode der Planung periodischer Aufgaben. In AL2023 `cronie` ist es standardmäßig nicht enthalten. Daher `crontab` wird die Unterstützung für standardmäßig nicht mehr bereitgestellt.

Sie können das `cronie`-Paket optional installieren, um klassische `cron`-Aufträge zu nutzen. Aufgrund der zusätzlichen Funktionalitäten in `systemd` empfehlen wir, auf `systemd`-Timer zu migrieren.

Pakete für glibc, gcc und binutils

AL2023 enthält viele der gleichen Kernpakete wie AL2.

Wir haben die folgenden drei Kern-Toolchain-Pakete für AL2023 aktualisiert.

Package name	AL2	AL2023
glibc	2,26	2,34
gcc	7.3	11,3
binutils	2,29	2,39

Weitere Informationen finden Sie unter Core-Toolchain-Paketeglibc, gcc und binutils.

Paketmanager

In Amazon Linux 2023 (AL2023) ist DNF das Standardverwaltungs-Tool für Softwarepakete. DNF ist der Nachfolger von YUM, dem Paketmanagement-Tool in AL2.

Weitere Informationen finden Sie unter Paketmanagement-Tool.

Protokollierungssystem

In AL2023 wurde das Protokollierungsystem-Paket gegenüber AL2 geändert. In AL2023 wird `rsyslog` nicht standardmäßig installiert, sodass die textbasierten Protokolldateien (z. B. `/var/log/messages`) nicht mehr wie in AL2 standardmäßig verfügbar sind. Die Standardkonfiguration für AL2023 ist `systemd-journal`, was mithilfe von `journalctl` geprüft werden kann. Obwohl es sich bei `rsyslog` um ein optionales Paket in AL2023 handelt, empfehlen wir die neue `systemd`-basierte `journalctl`-Schnittstelle und zugehörige Pakete. Weitere Informationen finden Sie auf der `journalctl` Seite im Handbuch.

Paketänderungen für **`curl`** und **`libcurl`**

AL2023 unterteilt die gemeinsamen Protokolle und Funktionalitäten der `curl`- und `libcurl`-Pakete in `curl-minimal` und `libcurl-minimal`. Dies reduziert den erforderlichen Platz auf der Festplatte, dem Arbeitsspeicher sowie Abhängigkeiten für die meisten Benutzer, und ist daher das Standardpaket für AL2023-AMIs und Container.

Wenn die volle `curl`-Funktionalität benötigt wird, z. B. für `gopher://`-Support, können Sie mithilfe der folgenden Befehle die `curl-full`- und `libcurl-full`-Pakete installieren.

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

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

GNU Privacy Guard (GNUPG)

AL2023 unterteilt die minimale und vollständige Funktionalität des `gnupg2`-Pakets in `gnupg2-minimal`- und `gnupg2-full`-Pakete. Nur das `gnupg2-minimal`-Paket ist standardmäßig

installiert. So wird die minimale Funktionalität zur Prüfung der digitalen Signaturen für `rpm`-Pakete bereitgestellt.

Vergewissern Sie sich, dass das `gnupg2-full`-Paket installiert ist, wenn Sie weitere `gnupg2`-Funktionalitäten benötigen, z. B. die Möglichkeit, Schlüssel von einem Schlüsselserver herunterzuladen. Mit folgendem Befehl tauschen Sie `gnupg2-minimal` gegen `gnupg2-full` ein.

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

Amazon Corretto als Standard-JVM

AL2023 wird mit Amazon Corretto als standardmäßigem (und einzigen) Java Development Kit (JDK) geliefert. Alle Java auf AL2023 basierenden Pakete sind alle mit gebaut. Amazon Corretto 17

Wenn Sie von AL2 migrieren, können Sie problemlos von der entsprechenden `OpenJDK` Version auf AL2 zu wechseln. Amazon Corretto

AWS CLI v2

AL2023 wird mit AWS CLI Version 2 geliefert, während AL2 mit Version 1 von geliefert wird. AWS CLI

UEFI Preferred

Standardmäßig werden alle Instances, die mit dem AL2023 AMI auf Instance-Typen gestartet werden und die UEFI-Firmware unterstützen, im UEFI-Modus gestartet. Hierzu setzen Sie den Boot-Modus-AMI-Parameter auf `uefi-preferred`. Weitere Informationen finden Sie unter Startmodi im Amazon EC2 EC2-Benutzerhandbuch.

Änderungen der Standardkonfiguration des SSH-Servers

Für das AL2023 AMI haben wir die `sshd`-Hostschlüssel-Typen geändert, die wir mit der Version generieren. Wir haben auch einige Legacy-Schlüsseltypen gestrichen, damit sie nicht versehentlich beim Starten generiert werden. Clients müssen die `rsa-sha2-512`- und `rsa-sha2-256`-Protokolle oder `ssh-ed25519` unter Verwendung eines `ed25519`-Schlüssels unterstützen. Standardmäßig werden `ssh-rsa`-Signaturen deaktiviert.

Außerdem enthalten die AL2023-Konfigurationseinstellungen `UseDNS=no` in der `sshd_config`-Standarddatei. Mit dieser neuen Einstellung ist es weniger wahrscheinlich, dass DNS-Beeinträchtigungen die Möglichkeit, `ssh`-Sitzungen mit Ihren Instances aufzubauen, blockieren könnten. Der Nachteil hierbei ist, dass die `from=`*`hostname.domain`*`,`*`hostname.domain`*-Zeileneinträge in Ihren `authorized_keys`-Dateien nicht aufgelöst werden. Da `sshd` nicht mehr versucht, die DNS-Namen aufzulösen, muss jeder durch Kommas getrennte *`hostname.domain`*-Wert in einen entsprechenden IP address-Wert übersetzt werden.

Weitere Informationen finden Sie unter Standard-SSH-Serverkonfiguration.

Extra Packages for Enterprise Linux (EPEL)

Extra Packages for Enterprise Linux (EPEL) ist ein Projekt der Fedora-Community mit dem Ziel, eine breite Auswahl an Paketen für Linux-Betriebssysteme auf Enterprise-Ebene zu erstellen. Das Projekt hat bisher hauptsächlich RHEL- und CentOS-Pakete produziert. AL2 zeichnet sich durch ein hohes Maß an Kompatibilität mit CentOS 7 aus. Daher funktionieren viele EPEL7-Pakete unter AL2. AL2023 unterstützt jedoch keine EPEL- oder EPEL-ähnliche Repositorys.

Verwenden von cloud-init

In AL2023 wird das Paket-Repository von cloud-init verwaltet. In früheren Versionen von Amazon Linux wurden standardmäßig Sicherheitsupdates über cloud-init installiert. Dies ist nicht der Standard für AL2023. Die neuen deterministischen Upgrade-Features für `releasever`-Updates beim Systemstart beschreiben die AL2023-Methode, Paketaktualisierungen beim Start zu aktivieren. Weitere Informationen finden Sie unter Paket- und Betriebssystemupdates in AL2023 verwalten und Deterministische Upgrades für Stabilität.

Bei AL2023 können Sie cloud-init mit SELinux verwenden. Weitere Informationen finden Sie unter Den cloud-init-Modus mit `enforcing` aktivieren.

Cloud-init lädt mithilfe von HTTP(S) Konfigurationsinhalte mit cloud-init von entfernten Standorten. In früheren Versionen gibt Amazon Linux keine Warnung aus, wenn Remote-Ressourcen nicht verfügbar sind. In AL2023 führen nicht verfügbare Remote-Ressourcen zu einem schwerwiegenden Fehler und die cloud-init-Ausführung schlägt fehl. Diese Verhaltensänderung gegenüber AL2 bietet ein sichereres „fail closed“-Standardverhalten.

Weitere Informationen finden Sie unter Anpassen von cloud-init und in der cloud-init-Dokumentation.

Grafische Desktop-Unterstützung

AL2023 ist Cloud-zentriert und für die Nutzung von Amazon EC2 optimiert. Derzeit wird keine Grafik- oder Desktop-Umgebung bereitgestellt. Wenn Sie Feedback dazu geben möchten GitHub, besuchen Sie https://github.com/.

Compiler-Triplet

AL2023 setzt das Compiler-Triplett für GCC und `LLVM` um anzuzeigen, dass es sich um den Anbieter `amazon` handelt.

Somit wird AL2 `aarch64-redhat-linux-gcc` zu `aarch64-amazon-linux-gcc` unter AL2023.

Dies sollte für die meisten Benutzer völlig transparent sein und betrifft möglicherweise nur diejenigen, die Compiler auf AL2023 erstellen.

32-Bit x86-(i686)-Pakete

Im Rahmen der Version 2014.09 von AL1 wurde angekündigt, dass dies die letzte Version sein wird, die 32-Bit-AMIs produziert. Ab der AL1-Version 2015.03 unterstützt Amazon Linux daher keine Systemausführung im 32-Bit-Modus mehr. AL2 bot lediglich eingeschränkten Laufzeitunterstützung für 32-Bit-Binärdateien auf x86-64-Hosts, und stellte keine Entwicklungspakete für neue 32-Bit-Binärdateien bereit. AL2023 enthält keine 32-Bit-Userspace-Pakete mehr. Wir empfehlen Ihnen, die Umstellung auf 64-Bit-Code abzuschließen.

Wenn Sie 32-Bit-Binärdateien auf AL2023 ausführen müssen, kann der 32-Bit-Userspace von AL2 in einem AL2-Container auf AL2023 ausgeführt werden.

`lsb_release` und das **`system-lsb-core`**-Paket

In der Vergangenheit wurde der `lsb_release`-Befehl (in AL2 im `system-lsb-core`-Paket enthalten) von einigen Programmen aufgerufen, um Informationen über die Linux-Distribution zu erhalten, auf der sie ausgeführt wurden. Dieser Befehl wurde von Linux Standards Base (LSB) eingeführt und wurden von den Linux-Distributionen übernommen. Linux-Distributionen haben sich weiterentwickelt, sodass der einfachere Standard für die Speicherung dieser Informationen in `/etc/os-release` und anderen verwandten Dateien verwendet wird.

Der `os-release`-Standard stammt aus `systemd`. Weitere Informationen finden Sie in der systemd os-Versionsdokumentation.

AL2023 wird nicht mit dem `lsb_release`-Befehl ausgeliefert und beinhaltet auch nicht das `system-lsb-core`-Paket. Die Software sollte die Umstellung auf den `os-release`-Standard abschließen, um die Kompatibilität mit Amazon Linux und anderen wichtigen Linux-Distributionen aufrechtzuerhalten.

AL2023-Kerneländerungen gegenüber AL2

AL2023 enthält den 6.1-Kernel sowie viele Konfigurationsänderungen, um Amazon Linux für die Cloud weiter zu optimieren. Für die meisten Benutzer sollten diese Änderungen völlig transparent sein.

Sicherheitsorientierte Änderungen an der Kernel-Konfiguration

CONFIG-Option	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_BUG_ON_DATA_CORRUPTION	n	y	n	y	y	y
CONFIG_DEFAULT_MMAP_MIN_ADDR	4096	4096	4096	4096	65536	65536
CONFIG_DEVMEM	n	y	n	y	n	n
CONFIG_DEVPORT	n	y	n	y	n	n
CONFIG_FORTIFY_SOURCE	n	y	n	y	y	y
CONFIG_HARDENED_US	N/A	–	y	y	–	–

CONFIG-Option	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
ERCOPY_FALLBACK						
CONFIG_INIT_ON_ALLOC_DEFAULT_ON	–	–	n	n	n	n
CONFIG_INIT_ON_FREE_DEFAULT_ON	–	–	n	n	n	n
CONFIG_IOMMU_DEFAULT_DMA_STRICT	–	–	–	–	n	n
CONFIG_LDISC_AUTOLOAD	y	y	y	y	n	n
CONFIG_SCHED_CORE	–	–	–	–	–	y
CONFIG_SCHED_STACK_END_CHECK	n	y	n	y	y	y
CONFIG_SECURITY_DMESG_RESTRICT	n	n	n	n	y	y

CONFIG-Option	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_SECURITY_SELINUX_DISABLE	y	y	y	y	n	n
CONFIG_SHUFFLE_PAGE_ALLOCATOR	–	N/A	y	y	y	y
CONFIG_SLAB_FREELIST_HARDENED	n	y	y	y	y	y
CONFIG_SLAB_FREELIST_RANDOM	n	n	y	y	y	y

Änderungen der Kernel-Konfigurationx für 86-64 Specific Security

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

aarch64 (ARM/Graviton) Kernel-Konfigurationsänderungen hinsichtlich Specific Security

CONFIG-Option	AL2/4.14/aarch64	AL2/5.10/aarch64	AL2023/6.1/aarch64
CONFIG_ARM64_PTR_AUTH	N/A	y	y
CONFIG_ARM64_PTR_AUTH_KERNEL	–	N/A	y
CONFIG_ARM64_SW_TTBR0_PAN	y	y	y

/dev/mem, **/dev/kmem** und **/dev/port**

Amazon Linux 2023 deaktiviert `/dev/mem` und baut `/dev/port` (`CONFIG_DEVMEM`und`CONFIG_DEVPORT`) vollständig auf den Einschränkungen auf, die bereits in AL2 gelten.

Der `/dev/kmem` Code wurde im 5.13-Kernel vollständig aus Linux entfernt, und obwohl er in AL2 deaktiviert war, gilt er jetzt nicht mehr für AL2023.

Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

FORTIFY_SOURCE

AL2023 ist auf allen unterstützten Architekturen `CONFIG_FORTIFY_SOURCE` aktiviert. Dieses Feature ist eine Funktion zur Erhöhung der Sicherheit. Wenn der Compiler die Puffergrößen ermitteln und validieren kann, erkennt dieses Feature Pufferüberläufe in gängigen String- und Speicherfunktionen.

Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Liniendisziplin automatisch laden () **CONFIG_LDISC_AUTOLOAD**

Der AL2023-Kernel lädt Zeilendisziplinen nicht automatisch, z. B. von Software, die die verwendet `TIOCSETDioctl`, es sei denn, die Anfrage stammt von einem Prozess mit den entsprechenden Berechtigungen. `CAP_SYS_MODULE`

Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

dmesgZugriff für unprivilegierte Benutzer () **CONFIG_SECURITY_DMESG_RESTRICT**

Standardmäßig erlaubt AL2023 unberechtigten Benutzern keinen Zugriff auf. `dmesg`

Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

SELinux deaktivieren **selinuxfs**

AL2023 deaktiviert die veraltete `CONFIG_SECURITY_SELINUX_DISABLE` Kerneloption, die eine Laufzeitmethode zur Deaktivierung von SELinux vor dem Laden der Richtlinie aktiviert hat.

Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Weitere Änderungen in der Kernelkonfiguration

CONFIG-Option	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_HZ	100	250	100	250	100	100
CONFIG_NR_CPUS	4096	8192	4096	8192	512	512
CONFIG_PANIC_ON_OOPS	y	n	y	n	y	y
CONFIG_PANIC_ON_OOPS_VALUE	1	0	1	0	1	1

CONFIG-Option	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_PPP	m	m	m	m	n	n
CONFIG_SLIP	m	m	m	m	n	n
CONFIG_XEN_PV	N/A	y	–	n	N/A	n

CONFIG_HZ

AL2023 wird auf beiden Plattformen auf 100 gesetzt. `CONFIG_HZ` `x86-64` `aarch64`

CONFIG_NR_CPUS

AL2023 setzt `CONFIG_NR_CPUS` auf eine Zahl, die näher an der maximalen Anzahl von CPU-Kernen in Amazon EC2 liegt.

-Panic bei OOPS

Der AL2023-Kernel gerät in Panik, wenn er ausfällt. Dieses Feature entspricht dem Booten mit `oops=panic` in der Kernel-Befehlszeile.

Bei einem Kernel-Oops hat der Kernel einen internen Fehler entdeckt, der die weitere Zuverlässigkeit des Systems beeinträchtigen kann.

-PPP- und SLIP-Unterstützung

AL2023 unterstützt die PPP- oder SLIP-Protokolle nicht.

Xen-PV-Gast-Support

AL2023 unterstützt nicht die Ausführung als Xen-PV-Gast.

Unterstützung für das Kernel-Dateisystem

Es wurden mehrere Änderungen an den Dateisystemen vorgenommen, die der Kernel in AL2 beim Einhängen unterstützen wird, ebenso wie Änderungen an den Partitionierungsschemata, die der Kernel analysiert.

CONFIG-Option	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_AFS_FS	n	m	n	m	n	n
CONFIG_AF_RXRPC	n	m	n	m	n	n
CONFIG_BSD_DISKLABEL	y	y	y	y	n	n
CONFIG_CRAMFS	m	m	m	m	n	n
CONFIG_CRAMFS_BLOCKDEV	N/A	–	y	n	–	–
CONFIG_DM_CLONE	–	–	n	n	n	n
CONFIG_DM_ERA	m	n	m	n	n	n
CONFIG_DM_INTEGRITY	n	m	n	m	m	m
CONFIG_DM_LOG_WRITES	n	n	m	m	m	m

CONFIG-Option	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_DM_SWITCH	m	n	m	n	n	n
CONFIG_DM_VERITY	m	n	m	n	n	n
CONFIG_ECRYPT_FS	n	m	n	m	n	n
CONFIG_EXFAT_FS	–	N/A	m	m	m	m
CONFIG_EXT2_FS	n	m	n	m	n	n
CONFIG_EXT3_FS	n	m	n	m	n	n
CONFIG_GFS2_FS	m	m	m	m	n	n
CONFIG_HFSPLUS_FS	n	m	n	m	n	n
CONFIG_HFS_FS	n	m	n	m	n	n
CONFIG_JFS_FS	n	m	n	m	n	n
CONFIG_LDM_PARTITION	n	y	n	y	n	n

CONFIG-Option	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_MAC_PARTITION	n	y	n	y	n	n
CONFIG_NFS_V2	n	m	n	m	n	n
CONFIG_NTFS_FS	n	m	n	n	n	n
CONFIG_ROMFS_FS	n	m	n	m	n	n
CONFIG_SOLARIS_X86_PARTITION	n	y	n	y	n	n
CONFIG_SQUASHFS_ZSTD	n	y	n	y	y	y
CONFIG_SUN_PARTITION	n	y	n	y	n	n

-Andrew-Dateisystem-Unterstützung (AFS)

Der Kernel wird nicht mehr mit Unterstützung für das `afs`-Dateisystem erstellt. AL2 wurde nicht mit User-Space-Unterstützung für ausgeliefert. `afs`

-cramfs-Unterstützung

Der Kernel wird nicht mehr mit Unterstützung für das `cramfs`-Dateisystem erstellt. Der Nachfolger von AL2023 ist das `squashfs` Dateisystem.

Unterstützung für BSD-disklabel

Der Kernel bietet keine Unterstützung für BSD-Disk-Labels mehr. Wenn Volumes mit BSD-Disk-Labels gelesen werden müssen, stehen hierfür verschiedene andere BSDs zur Verfügung.

-Device Mapper-Änderungen

Es wurden mehrere Änderungen an den im AL2023-Kernel konfigurierten Device Mapper-Zielen vorgenommen.

eCryptFs Unterstützung

Das `ecryptfs`-Dateisystem wird in Amazon Linux nicht mehr unterstützt. Die Userspace-Komponenten von `ecryptfs` waren in AL1 vorhanden, wurden in AL2 entfernt, und AL2023 baut den Kernel nicht mehr mit Unterstützung. `ecryptfs`

exFAT

Support für das `exFAT` Dateisystem wurde im 5.10-Kernel in AL2 hinzugefügt. Es war beim Start von AL2 mit einem 4.14-Kernel nicht vorhanden. AL2023 unterstützt weiterhin das Dateisystem. `exFAT`

Die Dateisysteme -ext2, -ext3 und -ext4

AL2023 wird mit der `CONFIG_EXT4_USE_FOR_EXT2` Option ausgeliefert, was bedeutet, dass der `ext4` Dateisystemcode zum Lesen älterer `ext2` Dateisysteme verwendet wird.

CONFIG_GFS2_FS

Der Kernel wird nicht mehr mit CONFIG_GFS2_FS erstellt.

Unterstützung für das Extended-HFS-Dateisystem von Apple (HFS+)

In AL2 wurden nur die `x86-64` Kernel mit `hfsplus` Dateisystemunterstützung gebaut. Der AL2 5.15-Kernel bietet keine `hfsplus` Unterstützung für jede Architektur. In AL2023 schließen wir die Einstellung der `hfsplus` Unterstützung in Amazon Linux ab.

Unterstützung für das HFS-Dateisystem

In AL2 wurden nur die `x86-64` Kernel mit Dateisystemunterstützung erstellt. `hfs` Der AL2 5.15-Kernel bietet keine `hfs` Unterstützung für jede Architektur. In AL2023 schließen wir die Einstellung der `hfs` Unterstützung in Amazon Linux ab.

Unterstützung für das JFS-Dateisystem

In AL2 wurden nur die `x86-64` Kernel mit Dateisystemunterstützung erstellt. `jfs` Der AL2 5.15-Kernel bietet keine `jfs` Unterstützung für jede Architektur. Weder AL1 noch AL2 wurden mit dem JFS-Userspace ausgeliefert. In AL2023 schließen wir die Einstellung der `jfs` Unterstützung in Amazon Linux ab.

Der Upstream-Linux-Kernel erwägt die Entfernung von. `JFS` Wenn Sie Daten in einem `JFS` Dateisystem haben, sollten Sie diese daher in ein anderes Dateisystem migrieren.

WindowsUnterstützung für Logical Disk Manager (dynamische Festplatte) (**`CONFIG_LDM_PARTITION`**)

AL2023 unterstützt Windows 2000 keine Windows Vista dynamischen Festplatten mit MS-DOS Stilpartitionen mehr. Windows XP Dieser Code unterstützte nie die neueren GPT-basierten dynamischen Festplatten, die mit eingeführt wurden. Windows Vista

Unterstützung für Macintosh-Partitionszuordnungen

AL2023 unterstützt die klassische Macintosh-Partitionsübersicht nicht mehr. Moderne macOS-Versionen erstellen standardmäßig moderne GPT-Partitionstabellen über diesem älteren Typ.

Unterstützung für NFSv2

AL2023 unterstützt NFSv2 nicht mehr, unterstützt aber weiterhin NFSv3, NFSv4, NFSv4.1 und NFSv4.2. Wir empfehlen Ihnen, auf NFSv3 oder neuer zu migrieren.

NTFS (**`CONFIG_NTFS_FS`**)

Der `ntfs3` Code wurde `ntfs` für den Zugriff auf NTFS-Dateisysteme unter Amazon Linux ab dem 5.10-Kernel in AL2 ersetzt. AL2023 enthält den `ntfs` Code nicht mehr und stützt sich ausschließlich auf den `ntfs3` Code für den Zugriff auf NTFS-Dateisysteme.

romfs-Dateisystem

Das `squashfs`-Dateisystem ist der Nachfolger des `romfs`-Dateisystems in Amazon Linux, und der AL2023-Kernel wird nicht mehr mit Unterstützung für `romfs` erstellt.

Solaris-x86-Festplattenpartitionsformat

AL2023 unterstützt das Solaris x86-Festplattenpartitionsformat nicht mehr.

`squashfs`-zstd-Komprimierung

AL2023 bietet Unterstützung für zstd komprimierte `squashfs` Dateisysteme auf allen unterstützten Architekturen.

Unterstützung für Sun-Partitionstabellen

AL2023 bietet keine Unterstützung mehr für das Sun-Partitionstabellenformat (). `CONFIG_SUN_PARTITION`

Vergleich der auf Amazon Linux 2 und Amazon Linux 2023 AMIs installierten Pakete

Ein Vergleich der RPMs, die auf den Standard-AMIs von Amazon Linux 2 und AL2023 vorhanden sind.

Paket	AL2 AMI	AL2023 AMI
`acl`	2.2.51	2.3.1
`acpid`	2.0.19	2.0.32
`alternatives`		1.15
`amazon-chrony-config`		4.3
`amazon-ec2-net-utils`		2.4.1
`amazon-linux-extras`	2.0.3	
`amazon-linux-extras-yum-plugin`	2.0.3	
`amazon-linux-repo-s3`		2023,4.20240513
`amazon-linux-sb-keys`		2023,1
`amazon-rpm-config`		228
`amazon-ssm-agent`	3.3.131,0	3.3.380,0

Paket	AL2 AMI	AL2023 AMI
at	3.1.13	3.1.23
attr	2,4,46	2.5.1
audit	2.8.1	3.0.6
audit-libs	2.8.1	3.0.6
authconfig	6.2.8	
aws-cfn-bootstrap	2,0	2.0
awscli	1.18,147	
awscli-2		2,15,30
basesystem	10.0	11
bash	4.2,46	5.2.15
bash-completion	2.1	2.11
bc	1,06,95	1,07,1
bind-export-libs	9,11,4	
bind-libs	9.11,4	9,16,48
bind-libs-lite	9,11,4	
bind-license	9.11,4	9,16,48
bind-utils	9,11,4	9,16,48
binutils	2,29,1	2,39
blktrace	1.0.5	
boost-date-time	1,53,0 (x86_64)	

Paket	AL2 AMI	AL2023 AMI
boost-filesystem		1,75,0
boost-system	1,53,0 (x86_64)	1,75,0
boost-thread	1,53,0 (x86_64)	1,75,0
bridge-utils	1.5	
bzip2	1.0.6	1.0.8
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023,2,64	2023,2,64
c-ares		1.19.0
checkpolicy		3.4
chkconfig	1,7.4	1.15
chrony	4.2	4.3
cloud-init	19,3	22.2.2
cloud-init-cfg-ec2		22.2.2
cloud-utils-growpart	0,31	0,31
coreutils	8,22	8,32
coreutils-common		8,32
cpio	2.12	2,13
cracklib	2.9.0	2.9.6
cracklib-dicts	2.9.0	2.9.6
cronie	1.4.11	

Paket	AL2 AMI	AL2023 AMI
cronie-anacron	1.4.11	
crontabs	1.11	1.11
crypto-policies		20220428
crypto-policies-scripts		20220428
cryptsetup	1,7.4	2.6.1
cryptsetup-libs	1.7.4	2.6.1
curl	8.3.0	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.26	2.1.27
cyrus-sasl-plain	2.1.26	2.1.27
dbus	1.10,24	1.12,28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.10.24	1.12,28
device-mapper	1,02,170	1,02,185
device-mapper-event	1,02,170	
device-mapper-event-libs	1,02,170	
device-mapper-libs	1,02,170	1,02,185

Paket	AL2 AMI	AL2023 AMI
device-mapper-persistent-data	0.7.3	
dhclient	4.2,5	
dhcp-common	4.2.5	
dhcp-libs	4.2.5	
diffutils	3.3	3.8
dmidecode	3.2	
dmraid	1.0.0.rc16	
dmraid-events	1.0.0.rc16	
dnf		4.14.0
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0
dnf-plugin-support-info		1.2
dnf-utils		4.3.0
dosfstools	3.0.20	4.2
dracut	033	055
dracut-config-ec2	2.0	3.0
dracut-config-generic	033	055

Paket	AL2 AMI	AL2023 AMI
dwz		0,14
dyninst	9.3.1 (x86_64)	10.2.1
e2fsprogs	1.42,9	1,46,5
e2fsprogs-libs	1,42,9	1,46,5
ec2-hibinit-agent	1.0.8	1.0.8
ec2-instance-connect	1.1	1.1
ec2-instance-connect-selinux	1.1	1.1
ec2-net-utils	1.7.3	
ec2-utils	1.2	2.2.0
ed	1.9	1.14.2
efibootmgr	15 (aarch64)	
efi-filesystem		5
efi-srpm-macros		5
efivar		38
efivar-libs	31 (aarch64)	38
elfutils-debuginfod-client		0.188
elfutils-default-yama-scope	0,176	0.188
elfutils-libelf	0,176	0.188

Paket	AL2 AMI	AL2023 AMI
elfutils-libs	0,176	0.188
ethtool	4,8	5,15
expat	2.1.0	2.5.0
file	5,11	5,39
file-libs	5,11	5,39
filesystem	3.2	3,14
findutils	4.5.11	4.8.0
fipscheck	1.4.1	
fipscheck-lib	1.4.1	
fonts-srpm-macros		2.0.5
freetype	2.8	
fstrm		0.6.1
fuse-libs	2.9.2	2.9.9
gawk	4.0.2	5.1.0
gdbm	1.13	
gdbm-libs		1,19
gdisk	0,8,10	1.0.8
generic-logos	18.0.0	
GeoIP	1.5.0	
gettext	0.19.8.1	0,21

Paket	AL2 AMI	AL2023 AMI
gettext-libs	0.19.8.1	0,21
ghc-srpm-macros		1.5.0
glib2	2,56,1	2,74,7
glibc	2,26	2,34
glibc-all-langpacks	2,26	2,34
glibc-common	2,26	2,34
glibc-gconv-extra		2,34
glibc-locale-source	2,26	2,34
glibc-minimal-langpack	2,26	
gmp	6.0.0	6.2.1
gnupg2	2.0.22	
gnupg2-minimal		2.3.7
gnutls		3.8.0
go-srpm-macros		3.2.0
gpgme	1.3.2	1.15.1
gpm-libs	1,20,7	1,20,7
grep	2,20	3.8
groff-base	1,22,2	1.22,4
grub2	2,06	
grub2-common	2,06	2,06

Paket	AL2 AMI	AL2023 AMI
grub2-efi-aa64	2,06 (aarch64)	
grub2-efi-aa64-ec2	2,06 (aarch64)	2,06 (aarch64)
grub2-efi-aa64-modules	2.06 (Noarch)	
grub2-efi-x64-ec2	2,06 (x86_64)	2,06 (x86_64)
grub2-pc	2,06 (x86_64)	
grub2-pc-modules	2.06 (Noarch)	2,06
grub2-tools	2,06	2,06
grub2-tools-minimal	2,06	2,06
grubby	8,28	8,40
gssproxy	0.7.0	0.8.4
gzip	1.5	1.12
hardlink	1.3	
hibagent	1.1.0	
hostname	3.13	3,23
hunspell	1.3.2	1.7.0
hunspell-en	0,20121024	0,20140811,1
hunspell-en-GB	0,20121024	0,20140811,1
hunspell-en-US	0,20121024	0,20140811,1
hunspell-filesystem		1.7.0
hwdata	0,252	0,353

Paket	AL2 AMI	AL2023 AMI
info	5.1	6.7
inih		49
initscripts	9,49,47	10,09
iproute	5.10.0	5.10.0
iptables	1.8.4	
iptables-libs	1.8.4	
iputils	20180629	20210202
irqbalance	1.7.0	1.9.0
jansson	(2.10)	2.14
jbigkit-libs	2.0	
jitterentropy		3.4.1
jq		1.7.1
json-c	0,11	0,14
kbd	1,1,5	2.4.0
kbd-legacy	1,1,5	
kbd-misc	1,1,5	2.4.0
kernel	5.10.215	6.1,90
kernel-livepatch-repo-s3		2023,4,20240513
kernel-srpm-macros		1,0
kernel-tools	5,10.215	6.1,90

Paket	AL2 AMI	AL2023 AMI
keyutils	1.5.8	1.6.3
keyutils-libs	1.5.8	1.6.3
kmod	25	29
kmod-libs	25	29
kpartx	0.4.9	
kpatch-runtime	0.9.4	0.9.7
krb5-libs	1.15.1	1,21
langtable	0.0.31	
langtable-data	0.0.31	
langtable-python	0.0.31	
less	458	608
libacl	2,2,51	2.3.1
libaio	0,3.109	0,3.111
libarchive		3,5.3
libargon2		20171227
libassuan	2.1.0	2,5.5
libattr	2,4,46	2.5.1
libbasicobjects	0.1.1	0.1.1
libblkid	2,30,2	2,37,4
libcap	2,54	2,48

Paket	AL2 AMI	AL2023 AMI
libcap-ng	0,7,5	0.8.2
libcbor		0.7.0
libcollection	0.7.0	0.7.0
libcom_err	1,42,9	1,46,5
libcomps		0,120
libconfig	1.4.9	1.7.2
libcroco	0.6.12	
libcrypt	2,26	
libcurl	8.3.0	
libcurl-minimal		8.5.0
libdaemon	0,14	
libdb	5.3.21	5.3.28
libdb-utils	5.3.21	
libdhash		0.5.0
libdnf		0,69,0
libdrm	2,4,97	
libdwarf	20130207 (x86_64)	
libeconf		0,4,0
libedit	3.0	3.1
libestr	0.1.9	

Paket	AL2 AMI	AL2023 AMI
libev		4,33
libevent	2.0.21	2.1.12
libfastjson	0,99,4	
libfdisk	2,30,2	2,37,4
libffi	3.0,13	3.4.4
libfido2		1.10.0
libgcc	7.3.1	11.4.1
libgcrypt	1.5.3	1.10.2
libgomp	7.3.1	11.4.1
libgpg-error	1.12	1,42
libibverbs		48,0
libicu	50,2	
libidn	1,28	
libidn2	2.3.0	2.3.2
libini_config	1.3.1	1.3.1
libjpeg-turbo	2,090	
libkcapi		1.4.0
libkcapi-hmaccalc		1.4.0
libldb		2.6.2
libmaxminddb		1.5.2

Paket	AL2 AMI	AL2023 AMI
libmetalink	0.1.3	0.1.3
libmnl	1.0.3	1.0.4
libmodulemd		2.13.0
libmount	2,30,2	2,37,4
libnetfilter_conntrack	1.0.6	
libnfnetlink	1.0.1	
libnfsidmap	0,25	2.5.4
libnghttp2	1.41,0	1,59,0
libnl3	3.2,28	3.5.0
libnl3-cli	3.2.28	
libpath_utils	0.2.1	0.2.1
libpcap	1.5.3	1.10.1
libpciaccess	0,14 (x86_64)	
libpipeline	1.2.3	1.5.3
libpkgconf		1.8.0
libpng	1,5,13	
libpsl	0,21,5	0,21,1
libpwquality	1.2.3	1.4.4
libref_array	0,15	0.1.5
librepo		1,14,5

Paket	AL2 AMI	AL2023 AMI
libreport-filesystem		2.15,2
libseccomp	2.5.2	2.5.3
libselinux	2.5	3.4
libselinux-utils	2.5	3.4
libsemanage	2.5	3.4
libsepol	2.5	3.4
libsigsegv		2,13
libsmartcols	2,30,2	2,37,4
libsolv		0,7.22
libss	1,42,9	1,46,5
libssh2	1.4.3	
libsss_certmap		2.9.4
libsss_idmap	1.16,5	2.9.4
libsss_nss_idmap	1.16,5	2.9.4
libsss_sudo		2.9.4
libstdc++	7.3.1	11.4.1
libstoragemgmt	1.6.1	1.9.4
libstoragemgmt-python	1.6.1	
libstoragemgmt-python-clibs	1.6.1	

Paket	AL2 AMI	AL2023 AMI
libsysfs	2.1.0	
libtalloc		2.3.4
libtasn1	4,10	4.19,0
libtdb		1.4.7
libteam	1,27	
libtevent		0.13.0
libtextstyle		0,21
libtiff	4,0,3	
libtirpc	0.2.4	1.3.3
libunistring	0.9.3	0.9.10
libuser	0,60	0,63
libutempter	1.1.6	1.2.1
libuuid	2,30,2	2,37,4
libuv		1.47.0
libverto	0,2,5	0.3.2
libverto-libev		0.3.2
libverto-libevent	0.2.5	
libwebp	0.3.0	
libxcrypt		4,4,33
libxml2	2.9.1	2.10.4

Paket	AL2 AMI	AL2023 AMI
libxml2-python	2.9.1	
libyaml	0.1.4	0.2.5
libzstd		1.5.5
lm_sensors-libs	3.4.0	3.6.0
lmdb-libs		0.9.29
logrotate	3.8.6	3,20,1
lsof	4,87	4,94,0
lua	5.1.4	
lua-libs		5.4.4
lua-srpm-macros		1
lvm2	2.02.187	
lvm2-libs	2,02.187	
lz4	1.7.5	
lz4-libs		1.9.4
make	3,82	
man-db	2.6.3	2.9.3
man-pages	3,53	5,10
man-pages-overrides	7.5.2	
mariadb-libs	5.5,68	
mdadm	4,0	

Paket	AL2 AMI	AL2023 AMI
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)
mlocate	0,26	
mpfr		4.1.0
mtr	0.92	
nano	2,9,8	5,8
ncurses	6.0	6.2
ncurses-base	6.0	6.2
ncurses-libs	6.0	6.2
nettle	2.7.1	3.8
net-tools	2,0	2.0
newt	0,52,15	0,52,21
newt-python	0,52,15	
nfs-utils	1.3.0	2,5.4
npth		1,6
nspr	4,35,0	4,35,0
nss	3,90,0	3,90,0
nss-pem	1.0.3	
nss-softokn	3,90,0	3,90,0
nss-softokn-freebl	3,90,0	3,90,0
nss-sysinit	3,90,0	3,90,0

Paket	AL2 AMI	AL2023 AMI
nss-tools	3,90,0	
nss-util	3,90,0	3,90,0
ntsysv	1.7.4	1.15
numactl-libs	2.0.9	2.0.14
ocaml-srpm-macros		6
oniguruma		6.9.7.1
openblas-srpm-macros		2
openldap	2.4.44	2,4,57
openssh	7,4p1	8,7 p1
openssh-clients	7,4p1	8,7 p1
openssh-server	7,4p1	8,7 p1
openssl	1,2k	3.0.8
openssl-libs	1,2k	3.0.8
openssl-pkcs11		0.4.12
os-prober	1.58	1,77
p11-kit	0,23,22	0,24,1
p11-kit-trust	0,23,22	0,24,1
package-notes-srpm-macros		0.4
pam	1.1.8	1.5.1
parted	3.1	3.4

Paket	AL2 AMI	AL2023 AMI
passwd	0,79	0,80
pciutils	3.5.1	3.7.0
pciutils-libs	3.5.1	3.7.0
pcre	8,32	
pcre2	10,23	10,40
pcre2-syntax		10,40
perl	5,16.3	
perl-Carp	1,26	1,50
perl-Class-Struct		0,66
perl-constant	1,27	1,33
perl-DynaLoader		1,47
perl-Encode	2,51	3,15
perl-Errno		1,30
perl-Exporter	5,68	5,74
perl-Fcntl		1.13
perl-File-Basename		2,85
perl-File-Path	2,09	2,18
perl-File-stat		1,09
perl-File-Temp	0,23,01	0,231,100
perl-Filter	1,49	

Paket	AL2 AMI	AL2023 AMI
perl-Getopt-Long	2,40	2,52
perl-Getopt-Std		1.12
perl-HTTP-Tiny	0,033	0,078
perl-if		0,60,800
perl-interpreter		5,32,1
perl-IO		1,43
perl-IPC-Open3		1,21
perl-libs	5.16,3	5.32,1
perl-macros	5.16,3	
perl-MIME-Base64		3,16
perl-mro		1,23
perl-overload		1,31
perl-overloading		0,02
perl-parent	0,225	0,238
perl-PathTools	3,40	3,78
perl-Pod-Escapes	1.04	1,07
perl-podlators	2.5.1	4,14
perl-Pod-Perldoc	3,20	3,28,01
perl-Pod-Simple	3,28	3,42
perl-Pod-Usage	1,63	2,01

Paket	AL2 AMI	AL2023 AMI
perl-POSIX		1,94
perl-Scalar-List-Utils	1,27	1,56
perl-SelectSaver		1.02
perl-Socket	2,010	2,032
perl-srpm-macros		1
perl-Storable	2,45	3,21
perl-subs		1,03
perl-Symbol		1,08
perl-Term-ANSIColor		5,01
perl-Term-Cap		1,17
perl-Text-ParseWords	3,29	3,30
perl-Text-Tabs+Wrap		2021,0726
perl-threads	1,87	
perl-threads-shared	1,43	
perl-Time-HiRes	1,9725	
perl-Time-Local	1,2300	1,300
perl-vars		1,05
pinentry	0.8.1	
pkgconf		1.8.0
pkgconfig	0,27,1	

Paket	AL2 AMI	AL2023 AMI
pkgconf-m4		1.8.0
pkgconf-pkg-config		1.8.0
plymouth	0,8,9	
plymouth-core-libs	0,8,9	
plymouth-scripts	0,8,9	
pm-utils	1.4.1	
policycoreutils	2.5	3.4
policycoreutils-python-utils		3.4
popt	1.13	1,18
postfix	2.10.1	
procps-ng	3.3.10	3.3.17
protobuf-c		1.4.1
psacct	6.6.1	6.6.4
psmisc	22,20	23,4
pth	2.0.7	
publicsuffix-list-dafsa	20240208	20240212
pygpgme	0.3	
pyliblzma	0,5,3	
pystache	0,5.3	

Paket	AL2 AMI	AL2023 AMI
python	2.7.18	
python2-botocore	1.18.6	
python2-colorama	0.3.9	
python2-cryptography	1.7.2	
python2-dateutil	2.6.1	
python2-futures	3.0.5	
python2-jmespath	0.9.3	
python2-jsonschema	2.5.1	
python2-oauthlib	2.0.1	
python2-pyasn1	0.1.9	
python2-rpm	4.11.3	
python2-rsa	3.4.1	
python2-s3transfer	0.3.3	
python2-setuptools	41,2,0	
python2-six	1.11.0	
python3	3.7.16	3.9,16
python3-attrs		20,3,0
python3-audit		3.0.6
python3-awscrt		0.19,19
python3-babel		2.9.1

Paket	AL2 AMI	AL2023 AMI
python3-cffi		1.14,5
python3-chardet		4.0.0
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36,1
python3-daemon	2.2.3	2.3.0
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils	0,14	0,16
python3-gpg		1.15.1
python3-hawkey		0,69,0
python3-idna		(2.10)
python3-jinja2		2.11,3
python3-jmespath		0.10.0
python3-jsonpatch		1,21
python3-jsonpointer		2.0
python3-jsonschema		3.2.0

Paket	AL2 AMI	AL2023 AMI
python3-libcomps		0,120
python3-libdnf		0,69,0
python3-libs	3.7,16	3.9,16
python3-libselinux		3.4
python3-libsemanage		3.4
python3-libstorage mgmt		1.9.4
python3-lockfile	0.11.0	0.12.2
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip	20.2.2	
python3-pip-wheel		21.3.1
python3-ply		3.11
python3-policycore utils		3.4
python3-prettytable		0.7.2
python3-prompt-too lkit		3.0,24
python3-pycparser		2,20
python3-pyrsistent		0,17,3

Paket	AL2 AMI	AL2023 AMI
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pystache	0.5.4	
python3-pytz		2022.7.1
python3-pyyaml		5.4.1
python3-requests		2,25,1
python3-rpm		4.16.1,3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools	49.1,3	59,6,0
python3-setuptools-wheel		59,6,0
python3-simplejson	3.2.0	
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1,25,10
python3-wcwidth		0,2,5
python-babel	0.9.6	
python-backports	1,0	

Paket	AL2 AMI	AL2023 AMI
python-backports-ssl_match_hostname	3.5.0.1	
python-cffi	1.6.0	
python-chardet	2.2.1	
python-chevron		0.13.1
python-configobj	4.7.2	
python-daemon	1,6	
python-devel	2.7.18	
python-docutils	0,12	
python-enum34	1.0.4	
python-idna	2.4	
python-iniparse	0.4	
python-ipaddress	1.0.16	
python-jinja2	2.7.2	
python-jsonpatch	1.2	
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	

Paket	AL2 AMI	AL2023 AMI
python-pillow	2.0.0	
python-ply	3.4	
python-pycparser	2.14	
python-pycurl	7,19,0	
python-repoze-lru	0.4	
python-requests	2.6.0	
python-simplejson	3.2.0	
python-srpm-macros		3.9
python-urlgrabber	3,10	
python-urllib3	1,25,9	
pyxattr	0.5.1	
PyYAML	3,10	
qrencode-libs	3.4.1	
quota	4,01	4,06
quota-nls	4,01	4,06
rdate	1.4	
readline	6.2	8,1
rng-tools	6.8	6,14
rootfiles	8.1	8.1
rpcbind	0.2.0	1.2.6

Paket	AL2 AMI	AL2023 AMI
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
rsync	3.1.2	3.2.6
rsyslog	8.24,0	
rust-srpm-macros		21
sbsigntools		0.9.4
scl-utils	20130529	
screen	4.1.0	4.8.0
sed	4.2.2	4.8
selinux-policy	3.13,1	37,22
selinux-policy-targeted	3,13,1	37,22
setserial	2,17	
setup	2,8,71	2.13,7
setuptool	1.19,11	
sgpio	1.2.0.10	

Paket	AL2 AMI	AL2023 AMI
shadow-utils	4.1.5.1	4,9 bis 4,9
shared-mime-info	1.8	
slang	2.2.4	2.3.2
sqlite	3.7.17	
sqlite-libs		3,40,0
sssd-client	1,16,5	2.9.4
sssd-common		2.9.4
sssd-kcm		2.9.4
sssd-nfs-idmap		2.9.4
strace	4,26	6.8
sudo	1,8,23	1.9,15
sysctl-defaults	1,0	1,0
sysstat	10.1.5	12.5.6
systemd	219	252,16
systemd-libs	219	252,16
systemd-networkd		252,16
systemd-pam		252,16
systemd-resolved		252,16
systemd-sysv	219	
systemd-udev		252,16

Paket	AL2 AMI	AL2023 AMI
system-release	2	2023,4,20240513
systemtap-runtime	4,5	4,8
sysvinit-tools	2,88	
tar	1,26	1,34
tbb		2020,3
tcp_wrappers	7.6	
tcp_wrappers-libs	7.6	
tcpdump	4.9.2	4,99,1
tcsh	6,18,01	6,24,07
teamd	1,27	
time	1,7	1.9
traceroute	2.0.22	2.1.3
tzdata	2024a	2024a
unzip	6.0	6.0
update-motd	1.1.2	2.2
usermode	1,111	
userspace-rcu		0.12.1
ustr	1.0.4	
util-linux	2,30,2	2,37,4
util-linux-core		2,37,4

Paket	AL2 AMI	AL2023 AMI
vim-common	9,0,2153	9,0,2153
vim-data	9,0,2153	9,0,2153
vim-enhanced	9,0,2153	9,0,2153
vim-filesystem	9,0,2153	9,0,2153
vim-minimal	9,0,2153	9,0,2153
virt-what	1,18	
wget	1.14	1,21,3
which	2,20	2,21
words	3.0	3.0
xfsdump	3.1.8	3.1.11
xfsprogs	5.0.0	5.18.0
xxd	9,0,2153	9,0,2153
xxhash-libs		0.8.0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yajl	2.0.4	
yum	3.4.3	4.14.0
yum-langpacks	0.4.2	
yum-metadata-parser	1.1.4	
yum-plugin-priorities	1.1.31	

Paket	AL2 AMI	AL2023 AMI
yum-utils	1.1.31	
zip	3.0	3.0
zlib	1.2.7	1.2.11
zram-generator		1.1.2
zram-generator-defaults		1.1.2
zstd		1.5.5

Vergleich der auf Amazon Linux 2 und Amazon Linux 2023 Minimal AMIs installierten Pakete

Ein Vergleich der RPMs, die auf den Minimal-AMIs von Amazon Linux 2 und AL2023 vorhanden sind.

Paket	AL2 Minimal	AL2023 Minimal
acl	2.2.51	
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.4.1
amazon-linux-extras	2.0.3	
amazon-linux-repo-s3		2023,4,20240513
amazon-linux-sb-keys		2023,1
audit	2.8.1	3.0.6
audit-libs	2.8.1	3.0.6

Paket	AL2 Minimal	AL2023 Minimal
authconfig	6.2.8	
awscli-2		2,15,30
basesystem	10.0	11
bash	4.2,46	5.2.15
bind-export-libs	9,11,4	
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023,2,64	2023,2,64
checkpolicy		3.4
chkconfig	1,7.4	
chrony	4.2	4.3
cloud-init	19,3	22.2.2
cloud-init-cfg-ec2		22.2.2
cloud-utils-growpart	0,31	0,31
coreutils	8,22	8,32
coreutils-common		8,32
cpio	2.12	2,13
cracklib	2.9.0	2.9.6
cracklib-dicts	2.9.0	2.9.6
cronie	1.4.11	
cronie-anacron	1.4.11	

Paket	AL2 Minimal	AL2023 Minimal
crontabs	1.11	
crypto-policies		20220428
cryptsetup-libs	1,7.4	2.6.1
curl	8.3.0	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.26	2.1.27
dbus	1.10,24	1.12,28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.10.24	1.12,28
device-mapper	1,02,170	1,02,185
device-mapper-libs	1,02,170	1,02,185
dhclient	4.2,5	
dhcp-common	4.2.5	
dhcp-libs	4.2.5	
diffutils	3.3	3.8
dnf		4.14.0
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0

Paket	AL2 Minimal	AL2023 Minimal
dnf-plugin-support-info		1.2
dracut	033	055
dracut-config-ec2	2.0	3.0
dracut-config-generic	033	055
e2fsprogs	1,42,9	1,46,5
e2fsprogs-libs	1,42,9	1,46,5
ec2-utils	1.2	2.2.0
efibootmgr	15 (aarch64)	
efi-filesystem		5
efivar		38
efivar-libs	31 (aarch64)	38
elfutils-default-yama-scope	0,176	0.188
elfutils-libelf	0,176	0.188
elfutils-libs	0,176	0.188
expat	2.1.0	2.5.0
file	5,11	5,39
file-libs	5,11	5,39
filesystem	3.2	3,14

Paket	AL2 Minimal	AL2023 Minimal
findutils	4.5.11	4.8.0
fipscheck	1.4.1	
fipscheck-lib	1.4.1	
freetype	2.8	
fuse-libs	2.9.2	2.9.9
gawk	4.0.2	5.1.0
gdbm	1.13	
gdbm-libs		1,19
gdisk	0,8,10	1.0.8
gettext	0.19.8.1	0,21
gettext-libs	0.19.8.1	0,21
glib2	2,56,1	2,74,7
glibc	2,26	2,34
glibc-all-langpacks	2,26	2,34
glibc-common	2,26	2,34
glibc-locale-source	2,26	2,34
glibc-minimal-lang pack	2,26	
gmp	6.0.0	6.2.1
gnupg2	2.0.22	
gnupg2-minimal		2.3.7

Paket	AL2 Minimal	AL2023 Minimal
gnutls		3.8.0
gpgme	1.3.2	1.15.1
grep	2,20	3.8
groff-base	1,22,2	1.22,4
grub2	2,06	
grub2-common	2,06	2,06
grub2-efi-aa64	2,06 (aarch64)	
grub2-efi-aa64-ec2	2,06 (aarch64)	2,06 (aarch64)
grub2-efi-aa64-modules	2.06 (Noarch)	
grub2-efi-x64-ec2	2,06 (x86_64)	2,06 (x86_64)
grub2-pc	2,06 (x86_64)	
grub2-pc-modules	2.06 (Noarch)	2,06
grub2-tools	2,06	2,06
grub2-tools-minimal	2,06	2,06
grubby	8,28	8,40
gzip	1.5	1.12
hardlink	1.3	
hostname	3.13	3,23
hwdata		0,353
info	5.1	

Paket	AL2 Minimal	AL2023 Minimal
inih		49
initscripts	9,49,47	10,09
iproute	5.10.0	5.10.0
iptables	1.8.4	
iptables-libs	1.8.4	
iputils	20180629	20210202
irqbalance	1.7.0	1.9.0
jansson		2.14
jitterentropy		3.4.1
jq		1.7.1
json-c		0,14
kbd		2.4.0
kbd-misc		2.4.0
kernel	4,14,343	6.1,90
kernel-livepatch-repo-s3		2023,4,20240513
keyutils-libs	1,5.8	1.6.3
kmod	25	29
kmod-libs	25	29
kpartx	0.4.9	
krb5-libs	1.15.1	1,21

Paket	AL2 Minimal	AL2023 Minimal
less	458	608
libacl	2,2,51	2.3.1
libarchive		3.5.3
libargon2		20171227
libassuan	2.1.0	2,5.5
libattr	2,4,46	2.5.1
libblkid	2,30,2	2,37,4
libcap	2,54	2,48
libcap-ng	0,7,5	0.8.2
libcbor		0.7.0
libcom_err	1,42,9	1,46,5
libcomps		0,120
libcroco	0.6.12	
libcrypt	2,26	
libcurl	8.3.0	
libcurl-minimal		8.5.0
libdb	5.3.21	5.3.28
libdb-utils	5.3.21	
libdnf		0,69,0
libeconf		0,4,0

Paket	AL2 Minimal	AL2023 Minimal
libedit	3.0	3.1
libestr	0.1.9	
libfastjson	0,99,4	
libfdisk	2,30,2	2,37,4
libffi	3.0,13	3.4.4
libfido2		1.10.0
libgcc	7.3.1	11.4.1
libgcrypt	1.5.3	1.10.2
libgomp	7.3.1	11.4.1
libgpg-error	1.12	1,42
libicu	50,2	
libidn	1,28	
libidn2	2.3.0	2.3.2
libkcapi		1.4.0
libkcapi-hmaccalc		1.4.0
libmetalink	0.1.3	
libmnl	1.0.3	1.0.4
libmodulemd		2.13.0
libmount	2,30,2	2,37,4
libnetfilter_conntrack	1.0.6	

Paket	AL2 Minimal	AL2023 Minimal
libnfnetlink	1.0.1	
libnghttp2	1,41,0	1,59,0
libpcap	1.5.3	
libpipeline	1.2.3	1.5.3
libpng	1.5,13	
libpsl	0,21,5	0,21,1
libpwquality	1.2.3	1.4.4
librepo		1,14,5
libreport-filesystem		2.15,2
libseccomp	2.5.2	2.5.3
libselinux	2.5	3.4
libselinux-utils	2.5	3.4
libsemanage	2.5	3.4
libsepol	2.5	3.4
libsigsegv		2,13
libsmartcols	2,30,2	2,37,4
libsolv		0,7.22
libss	1,42,9	1,46,5
libssh2	1.4.3	
libstdc++	7.3.1	11.4.1

Paket	AL2 Minimal	AL2023 Minimal
libsysfs	2.1.0	
libtasn1	4,10	4.19,0
libtextstyle		0,21
libunistring	0.9.3	0,9,10
libuser	0,60	0,63
libutempter	1.1.6	1.2.1
libuuid	2,30,2	2,37,4
libverto	0,2,5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libyaml	0.1.4	0.2.5
libzstd		1.5.5
logrotate	3.8.6	3,20,1
lua	5.1.4	
lua-libs		5.4.4
lz4	1.7.5	
lz4-libs		1.9.4
make	3,82	
man-db	2.6.3	2.9.3
mariadb-libs	5.5,68	

Paket	AL2 Minimal	AL2023 Minimal
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)
mpfr		4.1.0
ncurses	6.0	6.2
ncurses-base	6.0	6.2
ncurses-libs	6.0	6.2
nettle	2.7.1	3.8
net-tools	2,0	2.0
newt	0,52,15	
newt-python	0,52,15	
npth		1,6
nspr	4,35,0	
nss	3,90,0	
nss-pem	1.0.3	
nss-softokn	3,90,0	
nss-softokn-freebl	3,90,0	
nss-sysinit	3,90,0	
nss-tools	3,90,0	
nss-util	3,90,0	
numactl-libs	2.0.9	2.0.14
oniguruma		6.9.7.1

Paket	AL2 Minimal	AL2023 Minimal
openldap	2.4.44	2,4,57
openssh	7,4p1	8,7 p1
openssh-clients	7,4p1	8,7 p1
openssh-server	7,4p1	8,7 p1
openssl	1,2k	3.0.8
openssl-libs	1,2k	3.0.8
openssl-pkcs11		0.4.12
os-prober	1.58	1,77
p11-kit	0,23,22	0,24,1
p11-kit-trust	0,23,22	0,24,1
pam	1.1.8	1.5.1
passwd	0,79	0,80
pciutils		3.7.0
pciutils-libs		3.7.0
pcre	8,32	
pcre2	10,23	10,40
pcre2-syntax		10,40
pinentry	0.8.1	
pkgconfig	0,27,1	
policycoreutils	2.5	3.4

Paket	AL2 Minimal	AL2023 Minimal
popt	1.13	1,18
postfix	2.10.1	
procps-ng	3.3.10	3.3.17
psmisc	22,20	23,4
pth	2.0.7	
publicsuffix-list-dafsa	20240208	20240212
pygpgme	0.3	
pyliblzma	0,5,3	
python	2.7.18	
python2-cryptography	1.7.2	
python2-jsonschema	2.5.1	
python2-oauthlib	2.0.1	
python2-pyasn1	0.1.9	
python2-rpm	4.11.3	
python2-setuptools	41,2,0	
python2-six	1.11.0	
python3		3.9,16
python3-attrs		20,3,0
python3-audit		3.0.6
python3-awscrt		0.19,19

Paket	AL2 Minimal	AL2023 Minimal
python3-babel		2.9.1
python3-cffi		1.14,5
python3-chardet		4.0.0
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36,1
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils		0,16
python3-gpg		1.15.1
python3-hawkey		0,69,0
python3-idna		(2.10)
python3-jinja2		2.11,3
python3-jmespath		0.10.0
python3-jsonpatch		1,21
python3-jsonpointer		2.0
python3-jsonschema		3.2.0

Paket	AL2 Minimal	AL2023 Minimal
python3-libcomps		0,120
python3-libdnf		0,69,0
python3-libs		3,9,16
python3-libselinux		3.4
python3-libsemanage		3.4
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip-wheel		21.3.1
python3-ply		3.11
python3-policycoreutils		3.4
python3-prettytable		0.7.2
python3-prompt-toolkit		3.0,24
python3-pycparser		2,20
python3-pyrsistent		0,17,3
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pytz		2022.7.1
python3-pyyaml		5.4.1

Paket	AL2 Minimal	AL2023 Minimal
python3-requests		2,25,1
python3-rpm		4.16.1,3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools		59,6,0
python3-setuptools-wheel		59,6,0
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1,25,10
python3-wcwidth		0,2,5
python-babel	0.9.6	
python-backports	1,0	
python-backports-ssl_match_hostname	3.5.0.1	
python-cffi	1.6.0	
python-chardet	2.2.1	
python-configobj	4.7.2	
python-devel	2.7.18	

Paket	AL2 Minimal	AL2023 Minimal
python-enum34	1.0.4	
python-idna	2.4	
python-iniparse	0.4	
python-ipaddress	1.0.16	
python-jinja2	2.7.2	
python-jsonpatch	1.2	
python-jsonpointer	1.9	
python-jwcrypto	0.4.2	
python-libs	2.7.18	
python-markupsafe	0,11	
python-ply	3.4	
python-pycparser	2.14	
python-pycurl	7,19,0	
python-repoze-lru	0.4	
python-requests	2.6.0	
python-urlgrabber	3,10	
python-urllib3	1,25,9	
pyxattr	0.5.1	
PyYAML	3,10	
qrencode-libs	3.4.1	

Paket	AL2 Minimal	AL2023 Minimal
readline	6.2	8.1
rng-tools	6.8	6,14
rootfiles	8.1	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
rsyslog	8.24,0	
sbsigntools		0.9.4
sed	4.2.2	4.8
selinux-policy	3.13,1	37,22
selinux-policy-targeted	3,13,1	37,22
setup	2,8,71	2.13,7
shadow-utils	4.1.5.1	4,9 bis 4,9
shared-mime-info	1.8	
slang	2.2.4	
sqlite	3.7.17	

Paket	AL2 Minimal	AL2023 Minimal
sqlite-libs		3,40,0
sudo	1,8,23	1.9,15
sysctl-defaults	1,0	1,0
systemd	219	252,16
systemd-libs	219	252,16
systemd-networkd		252,16
systemd-pam		252,16
systemd-resolved		252,16
systemd-sysv	219	
systemd-udev		252,16
system-release	2	2023,4,20240513
sysvinit-tools	2,88	
tar	1,26	1,34
tcp_wrappers-libs	7.6	
tzdata	2024a	2024a
update-motd	1.1.2	2.2
userspace-rcu		0.12.1
ustr	1.0.4	
util-linux	2,30,2	2,37,4
util-linux-core		2,37,4

Paket	AL2 Minimal	AL2023 Minimal
vim-data	9,0,2153	9,0,2153
vim-minimal	9,0,2153	9,0,2153
which	2,20	2,21
xfsprogs	5.0.0	5,18,0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	1.1.4	
yum-plugin-priorities	1.1.31	
zlib	1.2.7	1.2.11
zram-generator		1.1.2
zram-generator-defaults		1.1.2
zstd		1.5.5

Vergleich der auf Amazon Linux 2 und Amazon Linux 2023 Basis-Container-Images installierten Pakete

Ein Vergleich der RPMs, die auf den Amazon Linux 2- und AL2023-Basiscontainer-Images vorhanden sind.

Paket	AL2-Behälter	AL2023 Behälter
alternatives		1.15

Paket	AL2-Behälter	AL2023 Behälter
amazon-linux-extras	2.0.3	
amazon-linux-repo-cdn		2023.4.20240513
audit-libs		3.0.6
basesystem	10.0	11
bash	4.2,46	5.2.15
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023,2,64	2023,2,64
chkconfig	1,7.4	
coreutils	8,22	
coreutils-single		8,32
cpio	2.12	
crypto-policies		20220428
curl	8,3,0	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.26	
diffutils	3.3	
dnf		4.14.0
dnf-data		4.14.0
elfutils-default-yama-scope		0.188

Paket	AL2-Behälter	AL2023 Behälter
elfutils-libelf	0,176	0.188
elfutils-libs		0.188
expat	2.1.0	2.5.0
file-libs	5,11	5,39
filesystem	3.2	3,14
findutils	4.5.11	
gawk	4.0.2	5.1.0
gdbm	1.13	
gdbm-libs		1,19
glib2	2,56,1	2,74,7
glibc	2,26	2,34
glibc-common	2,26	2,34
glibc-langpack-en	2,26	
glibc-minimal-langpack	2,26	2,34
gmp	6.0.0	6.2.1
gnupg2	2.0.22	
gnupg2-minimal		2.3.7
gpgme	1.3.2	1.15.1
grep	2,20	3.8
info	5.1	

Paket	AL2-Behälter	AL2023 Behälter
json-c		0,14
keyutils-libs	1,5.8	1.6.3
krb5-libs	1.15.1	1,21
libacl	2,2,51	2.3.1
libarchive		3.5.3
libassuan	2.1.0	2.5.5
libattr	2,4,46	2.5.1
libblkid	2,30,2	2,37,4
libcap	2,54	2,48
libcap-ng		0.8.2
libcom_err	1,42,9	1,46,5
libcomps		0,120
libcrypt	2,26	
libcurl	8.3.0	
libcurl-minimal		8.5.0
libdb	5.3.21	
libdb-utils	5.3.21	
libdnf		0,69,0
libffi	3.0,13	3.4.4
libgcc	7.3.1	11.4.1

Paket	AL2-Behälter	AL2023 Behälter
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.12	1,42
libidn2	2.3.0	2.3.2
libmetalink	0.1.3	
libmodulemd		2.13.0
libmount	2,30,2	2,37,4
libnghttp2	1,41,0	1,59,0
libpsl	0,21,5	0,21,1
librepo		1,14,5
libreport-filesystem		2.15,2
libselinux	2.5	3.4
libsepol	2.5	3.4
libsigsegv		2,13
libsmartcols		2,37,4
libsolv		0,7.22
libssh2	1.4.3	
libstdc++	7.3.1	11,4,1
libtasn1	4,10	4.19,0
libunistring	0.9.3	0.9.10

Paket	AL2-Behälter	AL2023 Behälter
libuuid	2,30,2	2,37,4
libverto	0,2,5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libyaml		0.2.5
libzstd		1.5.5
lua	5.1.4	
lua-libs		5.4.4
lz4-libs		1.9.4
mpfr		4.1.0
ncurses	6.0	
ncurses-base	6.0	6.2
ncurses-libs	6.0	6.2
npth		1,6
nspr	4,35,0	
nss	3,90,0	
nss-pem	1.0.3	
nss-softokn	3,90,0	
nss-softokn-freebl	3,90,0	
nss-sysinit	3,90,0	

Paket	AL2-Behälter	AL2023 Behälter
nss-tools	3,90,0	
nss-util	3,90,0	
openldap	2,4,44	
openssl-libs	1,2k	3.0.8
p11-kit	0,23,22	0,24,1
p11-kit-trust	0,23,22	0,24,1
pcre	8,32	
pcre2		10,40
pcre2-syntax		10,40
pinentry	0.8.1	
popt	1.13	1,18
pth	2.0.7	
publicsuffix-list-dafsa	20240208	20240212
pygpgme	0.3	
pyliblzma	0,5,3	
python	2.7.18	
python2-rpm	4.11.3	
python3		3.9.16
python3-dnf		4.14.0
python3-gpg		1.15.1

Paket	AL2-Behälter	AL2023 Behälter
python3-hawkey		0,69,0
python3-libcomps		0,120
python3-libdnf		0,69,0
python3-libs		3,9,16
python3-pip-wheel		21.3.1
python3-rpm		4.16.1.3
python3-setuptools-wheel		59,6,0
python-iniparse	0.4	
python-libs	2.7,18	
python-pycurl	7.19.0	
python-urlgrabber	3,10	
pyxattr	0.5.1	
readline	6.2	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
sed	4.2.2	4.8
setup	2,8,71	2.13,7
shared-mime-info	1.8	

Paket	AL2-Behälter	AL2023 Behälter
sqlite	3.7.17	
sqlite-libs		3,40,0
system-release	2	2023,4,20240513
tzdata	2024a	2024a
vim-data	9.0.2153	
vim-minimal	9,0,2153	
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	1.1.4	
yum-plugin-ovl	1.1.31	
yum-plugin-priorities	1.1.31	
zlib	1.2.7	1.2.11

Vergleich von AL1 und AL2023

In den folgenden Themen werden die wichtigsten Unterschiede zwischen AL1 und AL2023 beschrieben, die im Vergleich mit AL2 noch nicht behandelt wurden.

> **Note**
>
> AL1 hat seinen end-of-life (EOL) am 31. Dezember 2023 erreicht und wird ab dem 1. Januar 2024 keine Sicherheitsupdates oder Bugfixes mehr erhalten. Weitere Informationen zu AL1 EOL und Wartungsunterstützung finden Sie im Blogbeitrag Update on Amazon Linux AMI. end-of-life Wir empfehlen Ihnen, Ihre Anwendungen auf AL2023 zu aktualisieren, was langfristigen Support bis 2028 beinhaltet.

Themen

- Support für die einzelnen Versionen
- systemd ersetzt upstart als init-System
- Python 2.6 und 2.7 wurden durch Python 3 ersetzt
- OpenJDK 8 als ältestes JDK
- AL2023-Kernel-Änderungen gegenüber Amazon Linux 1 (AL1)
- Vergleich der auf Amazon Linux 1 (AL1) und Amazon Linux 2023 AMIs installierten Pakete
- Vergleich der auf Amazon Linux 1 (AL1) und Amazon Linux 2023 Minimal AMIs installierten Pakete
- Vergleich der auf Amazon Linux 1 (AL1) und Amazon Linux 2023Basis-Container-Images installierten Pakete

Support für die einzelnen Versionen

Für AL2023 bieten wir fünf Jahre Support ab dem Veröffentlichungsdatum an. AL1 hat den Standardsupport zum 31. Dezember 2020 und den Wartungssupport zum 31. Dezember 2023 eingestellt.

Weitere Informationen finden Sie unter Release-Taktfrequenz.

systemd ersetzt **upstart** als **init**-System

In AL2 `upstart` wurde das System durch `systemd` AS ersetzt. `init` AL2023 verwendet auch `systemd` als `init` System, wodurch weitere neue Merkmale und Funktionen von übernommen werden. `systemd`

Python 2.6 und 2.7 wurden durch Python 3 ersetzt

Obwohl AL1 Python 2.6 mit der Version 2018.03 als EOL markierte, waren die Pakete weiterhin in den Repositorys zur Installation verfügbar. AL2 wurde mit Python 2.7 als frühester unterstützter Python-Version ausgeliefert, und AL2023 schließt den Übergang zu Python 3 ab. In den AL2023-Repositorien sind keine Python 2.x-Versionen enthalten.

Weitere Informationen zu Python auf Amazon Linux finden Sie unter Python in AL2023.

OpenJDK 8 als ältestes JDK

AL2023 wird mit Amazon Corretto als standardmäßigem (und einzigen) Java Development Kit (JDK) geliefert. Alle Java auf AL2023 basierenden Pakete werden mit erstellt. Amazon Corretto 17

In AL1 wurde OpenJDK 1.6.0 (`java-1.6.0-openjdk`) mit der ersten Version 2018.03 als EOL eingestuft, und OpenJDK 1.7.0 (`java-1.7.0-openjdk`) wurde Mitte 2020 als EOL eingestuft, obwohl beide Versionen in den AL1-Repositorys verfügbar waren. Die früheste in AL2023 verfügbare OpenJDK-Version ist OpenJDK 8, bereitgestellt von. Amazon Corretto 8

AL2023-Kernel-Änderungen gegenüber Amazon Linux 1 (AL1)

Kernel-Live-Patching

Sowohl AL2023 als auch AL2 bieten Unterstützung für Kernel-Live-Patching-Funktionen. Auf diese Weise können Sie kritische und wichtige Sicherheitslücken im Linux-Kernel ohne Neustart oder Ausfallzeiten patchen. Weitere Informationen finden Sie unter Kernel-Live-Patching auf AL2023.

Unterstützung für das Kernel-Dateisystem

Es wurden mehrere Änderungen an den Dateisystemen vorgenommen, die der Kernel in AL1 beim Einhängen unterstützen wird, sowie Änderungen an den Partitionierungsschemata, die der Kernel analysieren wird.

CONFIG-Option	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_AFS_FS	m	n	n
CONFIG_AF_RXRPC	m	n	n
CONFIG_BSD_DISKLABEL	y	n	n
CONFIG_CRAMFS	m	n	n
CONFIG_CRAMFS_BLOCKDEV	N/A	–	–
CONFIG_DM_CLONE	–	n	n
CONFIG_DM_ERA	n	n	n
CONFIG_DM_INTEGRITY	m	m	m
CONFIG_DM_LOG_WRITES	n	m	m
CONFIG_DM_SWITCH	n	n	n
CONFIG_DM_VERITY	n	n	n
CONFIG_ECRYPT_FS	m	n	n
CONFIG_EXFAT_FS	N/A	m	m
CONFIG_EXT2_FS	m	n	n
CONFIG_EXT3_FS	m	n	n
CONFIG_GFS2_FS	n	n	n

CONFIG-Option	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_HFSPLUS_FS	m	n	n
CONFIG_HFS_FS	m	n	n
CONFIG_JFS_FS	m	n	n
CONFIG_LDM_PARTITION	y	n	n
CONFIG_MAC_PARTITION	y	n	n
CONFIG_NFS_V2	m	n	n
CONFIG_NTFS_FS	m	n	n
CONFIG_ROMFS_FS	m	n	n
CONFIG_SOLARIS_X86_PARTITION	y	n	n
CONFIG_SQUASHFS_ZSTD	y	y	y
CONFIG_SUN_PARTITION	y	n	n

Sicherheitsorientierte Änderungen an der Kernel-Konfiguration

CONFIG-Option	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_BUG_ON_DATA_CORRUPTION	y	y	y

CONFIG-Option	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_DEFAULT_MMAP_MIN_ADDR	4096	65536	65536
CONFIG_DEVMEM	y	n	n
CONFIG_DEVPORT	y	n	n
CONFIG_FORTIFY_SOURCE	y	y	y
CONFIG_HARDENED_USERCOPY_FALLBACK	N/A	–	–
CONFIG_INIT_ON_ALLOC_DEFAULT_ON	–	n	n
CONFIG_INIT_ON_FREE_DEFAULT_ON	–	n	n
CONFIG_IOMMU_DEFAULT_DMA_STRICT	–	n	n
CONFIG_LDISC_AUTOLOAD	y	n	n
CONFIG_SCHED_CORE	–	–	y
CONFIG_SCHED_STACK_END_CHECK	y	y	y

CONFIG-Option	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_SECURITY_DMESG_RESTRICT	n	y	y
CONFIG_SECURITY_SELINUX_DISABLE	y	n	n
CONFIG_SHUFFLE_PAGE_ALLOCATOR	N/A	y	y
CONFIG_SLAB_FREELIST_HARDENED	y	y	y
CONFIG_SLAB_FREELIST_RANDOM	n	y	y

Weitere Änderungen in der Kernelkonfiguration

CONFIG-Option	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_HZ	250	100	100
CONFIG_NR_CPUS	8192	512	512
CONFIG_PANIC_ON_OOPS	n	y	y
CONFIG_PANIC_ON_OOPS_VALUE	0	1	1

CONFIG-Option	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_PPP	m	n	n
CONFIG_SLIP	m	n	n
CONFIG_XEN_PV	y	N/A	n

Vergleich der auf Amazon Linux 1 (AL1) und Amazon Linux 2023 AMIs installierten Pakete

Ein Vergleich der auf den Standard-AMIs AL1 und AL2023 vorhandenen RPMs.

Paket	AL1 AMI	AL2023 AMI
acl	2.2,49	2.3.1
acpid	2.0.19	2.0.32
alsa-lib	1.0.22	
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.4.1
amazon-linux-repo-s3		2023,4.20240513
amazon-linux-sb-keys		2023,1
amazon-rpm-config		228
amazon-ssm-agent	3,2,2222,0	3.3.380,0
at	3.1.10	3.1.23
attr	2,4,46	2.5.1

Paket	AL1 AMI	AL2023 AMI
audit	2.6.5	3.0.6
audit-libs	2.6.5	3.0.6
authconfig	6.2.8	
aws-amitools-ec2	1.5.13	
aws-cfn-bootstrap	1.4	2.0
aws-cli	1.18,107	
awscli-2		2,15,30
basesystem	10.0	11
bash	4.2,46	5.2.15
bash-completion		2.11
bc	1,06,95	1,07,1
bind-libs	9.8.2	9,16,48
bind-license		9,16,48
bind-utils	9.8.2	9,16,48
binutils	2,27	2,39
boost-filesystem		1,75,0
boost-system		1,75,0
boost-thread		1,75,0
bzip2	1.0.6	1.0.8
bzip2-libs	1.0.6	1.0.8

Paket	AL1 AMI	AL2023 AMI
ca-certificates	2023,2,62	2023,2,64
c-ares		1.19.0
checkpolicy	2.1.10	3.4
chkconfig	1.3.49,3	1.15
chrony		4.3
cloud-disk-utils	0,27	
cloud-init	0,7.6	22.2.2
cloud-init-cfg-ec2		22.2.2
cloud-utils-growpart		0,31
copy-jdk-configs	3,3	
coreutils	8,22	8,32
coreutils-common		8,32
cpio	(2.10)	2,13
cracklib	2.8,16	2.9.6
cracklib-dicts	2.8,16	2.9.6
cronie	1.4.4	
cronie-anacron	1.4.4	
crontabs	1.10	1.11
crypto-policies		20220428
crypto-policies-scripts		20220428

Paket	AL1 AMI	AL2023 AMI
cryptsetup	1.6.7	2.6.1
cryptsetup-libs	1.6.7	2.6.1
curl	7,61,1	
curl-minimal		8,5,0
cyrus-sasl	2.1.23	
cyrus-sasl-lib	2.1.23	2.1.27
cyrus-sasl-plain	2.1.23	2.1.27
dash	0.5.5.1	
db4	4.7.25	
db4-utils	4,7,25	
dbus	1.6.12	1.12.28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.6.12	1.12.28
dejavu-fonts-common	2,33	
dejavu-sans-fonts	2,33	
dejavu-serif-fonts	2,33	
device-mapper	1,02,135	1,02,185
device-mapper-event	1,02,135	
device-mapper-event-libs	1,02,135	

Paket	AL1 AMI	AL2023 AMI
device-mapper-libs	1,02,135	1,02,185
device-mapper-persistent-data	0,6.3	
dhclient	4.1.1	
dhcp-common	4.1.1	
diffutils	3.3	3.8
dmraid	1.0.0.rc16	
dmraid-events	1.0.0.rc16	
dnf		4.14.0
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0
dnf-plugin-support-info		1.2
dnf-utils		4.3.0
dosfstools		4.2
dracut	004	055
dracut-config-ec2		3.0
dracut-config-generic		055

Paket	AL1 AMI	AL2023 AMI
dracut-modules-growroot	0.20	
dump	0.4	
dwz		0,14
dyninst		10.2.1
e2fsprogs	1.43,5	1,46,5
e2fsprogs-libs	1,43,5	1,46,5
ec2-hibinit-agent	1.0.0	1.0.8
ec2-instance-connect		1.1
ec2-instance-connect-selinux		1.1
ec2-net-utils	0.7	
ec2-utils	0.7	2.2.0
ed	1.1	1.14.2
efi-filesystem		5
efi-srpm-macros		5
efivar		38
efivar-libs		38
elfutils-debuginfod-client		0.188
elfutils-default-yama-scope		0.188

Paket	AL1 AMI	AL2023 AMI
elfutils-libelf	0,168	0.188
elfutils-libs		0.188
epel-release	6	
ethtool	3,15	5,15
expat	2.1.0	2.5.0
file	5,37	5,39
file-libs	5,37	5,39
filesystem	2,4,30	3,14
findutils	4.4.2	4.8.0
fipscheck	1.3.1	
fipscheck-lib	1.3.1	
fontconfig	2.8.0	
fontpackages-filesystem	1,41	
fonts-srpm-macros		2.0.5
freetype	2.3.11	
fstrm		0.6.1
fuse-libs	2.9.4	2.9.9
gawk	3.1.7	5.1.0
gdbm	1.8.0	
gdbm-libs		1,19

Paket	AL1 AMI	AL2023 AMI
gdisk	0,8,10	1.0.8
generic-logos	17.0.0	
get_reference_source	1.2	
gettext		0,21
gettext-libs		0,21
ghc-srpm-macros		1.5.0
giflib	4.1,6	
glib2	2,36,3	2,74,7
glibc	2,17	2,34
glibc-all-langpacks		2,34
glibc-common	2,17	2,34
glibc-gconv-extra		2,34
glibc-locale-source		2,34
gmp	6.0.0	6.2.1
gnupg2	2.0.28	
gnupg2-minimal		2.3.7
gnutls		3.8.0
go-srpm-macros		3.2.0
gpgme	1.4.3	1.15.1
gpm-libs	1,20,6	1,20,7

Paket	AL1 AMI	AL2023 AMI
grep	2,20	3.8
groff	1,22,2	
groff-base	1.22.2	1.22,4
grub	0,97	
grub2-common		2,06
grub2-efi-x64-ec2		2,06
grub2-pc-modules		2,06
grub2-tools		2,06
grub2-tools-minimal		2,06
grubby	7,0,15	8,40
gssproxy		0.8.4
gzip	1.5	1.12
hesiod	3.1.0	
hibagent	1.0.0	
hmaccalc	0,9,12	
hostname		3,23
hunspell		1.7.0
hunspell-en		0,20140811,1
hunspell-en-GB		0,20140811,1
hunspell-en-US		0,20140811,1

Paket	AL1 AMI	AL2023 AMI
hunspell-filesystem		1.7.0
hwdata	0,233	0,353
info	5.1	6.7
inih		49
initscripts	9,03.58	10,09
iproute	4.4.0	5.10.0
iptables	1,4,21	
iputils	20121221	20210202
irqbalance	1.5.0	1.9.0
jansson		2.14
java-1.7.0-openjdk	1.7.0.321	
javapackages-tools	0.9.1	
jitterentropy		3.4.1
jpackage-utils	1.7.5	
jq		1.7.1
json-c		0,14
kbd	1.15	2.4.0
kbd-misc	1.15	2.4.0
kernel	4,14,336	6.1,90
kernel-livepatch-repo-s3		2023,4,20240513

Paket	AL1 AMI	AL2023 AMI
kernel-srpm-macros		1,0
kernel-tools	4,14,336	6.1,90
keyutils	1.5.8	1.6.3
keyutils-libs	1.5.8	1.6.3
kmod	14	29
kmod-libs	14	29
kpartx	0.4.9	
kpatch-runtime		0.9.7
krb5-libs	1.15.1	1,21
lcms2	2.6	
less	436	608
libacl	2,2,49	2.3.1
libaio	0,3.109	0,3.111
libarchive		3,5.3
libargon2		20171227
libassuan	2.0.3	2,5.5
libattr	2,4,46	2.5.1
libbasicobjects		0.1.1
libblkid	2,23,2	2,37,4
libcap	2,16	2,48

Paket	AL1 AMI	AL2023 AMI
libcap54	2,54	
libcap-ng	0,7,5	0.8.2
libcbor		0.7.0
libcgroup	0,40.rc1	
libcollection		0.7.0
libcom_err	1,43,5	1,46,5
libcomps		0,120
libconfig		1.7.2
libcurl	7,61,1	
libcurl-minimal		8,5,0
libdb		5.3.28
libdhash		0.5.0
libdnf		0,69,0
libeconf		0,4,0
libedit	2.11	3.1
libev		4,33
libevent	2.0.21	2.1.12
libfdisk		2,37,4
libffi	3.0,13	3.4.4
libfido2		1.10.0

Paket	AL1 AMI	AL2023 AMI
libfontenc	1.0.5	
libgcc		11.4.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.11	1,42
libgssglue	0.1	
libibverbs		48,0
libICE	1.0.6	
libicu	50,2	
libidn	1,18	
libidn2	2.3.0	2.3.2
libini_config		1.3.1
libjpeg-turbo	1,2,90	
libkcapi		1.4.0
libkcapi-hmaccalc		1.4.0
libldb		2.6.2
libmaxminddb		1.5.2
libmetalink		0.1.3
libmnl	1.0.3	1.0.4

Paket	AL1 AMI	AL2023 AMI
libmodulemd		2.13.0
libmount	2,23,2	2,37,4
libnetfilter_connt rack	1.0.4	
libnfnetlink	1.0.1	
libnfsidmap	0,25	2.5.4
libnghttp2	1.33,0	1,59,0
libnih	1.0.1	
libnl	1.1.4	
libnl3		3.5.0
libpath_utils		0,2.1
libpcap		1.10.1
libpipeline	1.2.3	1.5.3
libpkgconf		1.8.0
libpng	1.2,49	
libpsl	0.6.2	0,21,1
libpwquality	1.2.3	1.4.4
libref_array		0,15
librepo		1,14,5
libreport-filesystem		2.15,2
libseccomp		2.5.3

Paket	AL1 AMI	AL2023 AMI
libselinux	2.1.10	3.4
libselinux-utils	2.1.10	3.4
libsemanage	2.1.6	3.4
libsepol	2.1.7	3.4
libsigsegv		2,13
libSM	1.2.1	
libsmartcols	2,23,2	2,37,4
libsolv		0,7.22
libss	1,43,5	1,46,5
libssh2	1.4.2	
libsss_certmap		2.9.4
libsss_idmap		2.9.4
libsss_nss_idmap		2.9.4
libsss_sudo		2.9.4
libstdc++		11.4.1
libstdc++72	7.2.1	
libstoragemgmt		1.9.4
libsysfs	2.1.0	
libtalloc		2.3.4
libtasn1	2.3	4.19.0

Paket	AL1 AMI	AL2023 AMI
libtdb		1.4.7
libtevent		0.13.0
libtextstyle		0,21
libtirpc	0,2,4	1.3.3
libudev	173	
libunistring	0.9.3	0.9.10
libuser	0,60	0,63
libutempter	1.1.5	1.2.1
libuuid	2,23,2	2,37,4
libuv		1.47.0
libverto	0,2,5	0.3.2
libverto-libev		0.3.2
libX11	1.6.0	
libX11-common	1.6.0	
libXau	1.0.6	
libxcb	1.11	
libXcomposite	0.4.3	
libxcrypt		4.4.33
libXext	1.3.2	
libXfont	1.4.5	

Paket	AL1 AMI	AL2023 AMI
libXi	1.7.2	
libxml2	2.9.1	2.10.4
libxml2-python27	2.9.1	
libXrender	0.9.8	
libxslt	1.1.28	
libXtst	1.2.2	
libyaml	0.1.6	0.2.5
libzstd		1.5.5
lm_sensors-libs		3.6.0
lmdb-libs		0.9.29
log4j-cve-2021-44228-hotpatch	1.3	
logrotate	3.7.8	3.20.1
lsof	4,82	4,94,0
lua	5.1.4	
lua-libs		5.4.4
lua-srpm-macros		1
lvm2	2.02.166	
lvm2-libs	2,02.166	
lz4-libs		1.9.4
mailcap	2.1.31	

Paket	AL1 AMI	AL2023 AMI
make	3,82	
man-db	2.6.3	2.9.3
man-pages	4,10	5,10
mdadm	3.2.6	
microcode_ctl	2.1	2.1
mingetty	1,08	
mpfr		4.1.0
nano	2.5.3	5,8
nc	1,84	
ncurses	5,7	6.2
ncurses-base	5,7	6.2
ncurses-libs	5,7	6.2
nettle		3.8
net-tools	1,60	2.0
newt	0,52,11	0,52,21
newt-python27	0,52,11	
nfs-utils	1.3.0	2,5.4
npth		1,6
nspr	4.25,0	4,35,0
nss	3,53,1	3,90,0

Paket	AL1 AMI	AL2023 AMI
nss-pem	1.0.3	
nss-softokn	3,53,1	3,90,0
nss-softokn-freebl	3,53,1	3,90,0
nss-sysinit	3,53,1	3,90,0
nss-tools	3,53,1	
nss-util	3,53,1	3,90,0
ntp	4.2.8 p 15	
ntpdate	4.2.8p15	
ntsysv	1.3.49,3	1.15
numactl	2.0.7	
numactl-libs		2.0.14
ocaml-srpm-macros		6
oniguruma		6.9.7.1
openblas-srpm-macros		2
openldap	2.4.40	2,4,57
openssh	7,4p1	8,7 p1
openssh-clients	7,4p1	8,7 p1
openssh-server	7,4p1	8,7 p1
openssl	1,2k	3.0.8
openssl-libs		3.0.8

Paket	AL1 AMI	AL2023 AMI
openssl-pkcs11		0.4.12
os-prober		1,77
p11-kit	0,18,5	0,24,1
p11-kit-trust	0,18,5	0,24,1
package-notes-srpm-macros		0.4
pam	1.1.8	1.5.1
pam_ccreds	10	
pam_krb5	2.3.11	
pam_passwdqc	1.0.5	
parted	2.1	3.4
passwd	0,79	0,80
pciutils	3.1.10	3.7.0
pciutils-libs	3.1.10	3.7.0
pcre	8,21	
pcre2		10,40
pcre2-syntax		10,40
perl	5,16.3	
perl-Carp	1,26	1,50
perl-Class-Struct		0,66
perl-constant	1,27	1,33

Paket	AL1 AMI	AL2023 AMI
perl-Digest	1,17	
perl-Digest-HMAC	1,03	
perl-Digest-MD5	2,52	
perl-Digest-SHA	5,85	
perl-DynaLoader		1,47
perl-Encode	2,51	3,15
perl-Errno		1,30
perl-Exporter	5,68	5,74
perl-Fcntl		1.13
perl-File-Basename		2,85
perl-File-Path	2,09	2,18
perl-File-stat		1,09
perl-File-Temp	0,23,01	0,231,100
perl-Filter	1,49	
perl-Getopt-Long	2,40	2,52
perl-Getopt-Std		1.12
perl-HTTP-Tiny	0,033	0,078
perl-if		0,60,800
perl-interpreter		5,32,1
perl-IO		1,43

Paket	AL1 AMI	AL2023 AMI
perl-IPC-Open3		1,21
perl-libs	5.16,3	5.32,1
perl-macros	5.16,3	
perl-MIME-Base64		3,16
perl-mro		1,23
perl-overload		1,31
perl-overloading		0,02
perl-parent	0,225	0,238
perl-PathTools	3,40	3,78
perl-Pod-Escapes	1.04	1,07
perl-podlators	2.5.1	4,14
perl-Pod-Perldoc	3,20	3,28,01
perl-Pod-Simple	3,28	3,42
perl-Pod-Usage	1,63	2,01
perl-POSIX		1,94
perl-Scalar-List-Utils	1,27	1,56
perl-SelectSaver		1.02
perl-Socket	2,010	2,032
perl-srpm-macros		1
perl-Storable	2,45	3,21

Paket	AL1 AMI	AL2023 AMI
perl-subs		1,03
perl-Symbol		1,08
perl-Term-ANSIColor		5,01
perl-Term-Cap		1,17
perl-Text-ParseWords	3,29	3,30
perl-Text-Tabs+Wrap		2021,0726
perl-threads	1,87	
perl-threads-shared	1,43	
perl-Time-HiRes	1,9725	
perl-Time-Local	1,2300	1,300
perl-vars		1,05
pinentry	0,7.6	
pkgconf		1.8.0
pkgconfig	0,27,1	
pkgconf-m4		1.8.0
pkgconf-pkg-config		1.8.0
pm-utils	1.4.1	
policycoreutils	2.1.12	3.4
policycoreutils-python-utils		3.4
popt	1.13	1,18

Paket	AL1 AMI	AL2023 AMI
procmail	3,22	
procps	3.2.8	
procps-ng		3.3.17
protobuf-c		1.4.1
psacct	6.3.2	6.6.4
psmisc	22,20	23,4
pth	2.0.7	
publicsuffix-list-dafsa		20240212
python27	2.7,18	
python27-babel	0.9.4	
python27-backports	1,0	
python27-backports-ssl_match_hostname	3.4.0.2	
python27-boto	2,48,0	
python27-botocore	1,17,31	
python27-chardet	2.0.1	
python27-colorama	0.4.1	
python27-configobj	4.7.2	
python27-crypto	2.6.1	
python27-daemon	1.5.2	

Paket	AL1 AMI	AL2023 AMI
python27-dateutil	2.1	
python27-devel	2.7.18	
python27-docutils	0,11	
python27-ecdsa	0,11	
python27-futures	3.0.3	
python27-imaging	1.1.6	
python27-iniparse	0.3.1	
python27-jinja2	2.7.2	
python27-jmespath	0.9.2	
python27-jsonpatch	1.2	
python27-jsonpointer	1,0	
python27-kitchen	1.1.1	
python27-libs	2.7.18	
python27-lockfile	0.8	
python27-markupsafe	0,11	
python27-paramiko	1.15.1	
python27-pip	9.0.3	
python27-ply	3.4	
python27-pyasn1	0.1.7	
python27-pycurl	7.19.0	

Paket	AL1 AMI	AL2023 AMI
python27-pygpgme	0.3	
python27-pyliblzma	0.5.3	
python27-pystache	0,5.3	
python27-pyxattr	0.5.0	
python27-PyYAML	3,10	
python27-requests	1.2.3	
python27-rsa	3.4.1	
python27-setuptools	36,2,7	
python27-simplejson	3.6.5	
python27-six	1.8.0	
python27-urlgrabber	3,10	
python27-urllib3	1,24,3	
python27-virtualenv	15.1,0	
python3		3.9,16
python3-attrs		20,3,0
python3-audit		3.0.6
python3-awscrt		0.19,19
python3-babel		2.9.1
python3-cffi		1.14,5
python3-chardet		4.0.0

Paket	AL1 AMI	AL2023 AMI
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36,1
python3-daemon		2.3.0
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils		0,16
python3-gpg		1.15.1
python3-hawkey		0,69,0
python3-idna		(2.10)
python3-jinja2		2.11,3
python3-jmespath		0.10.0
python3-jsonpatch		1,21
python3-jsonpointer		2.0
python3-jsonschema		3.2.0
python3-libcomps		0,120
python3-libdnf		0,69,0

Paket	AL1 AMI	AL2023 AMI
python3-libs		3,9,16
python3-libselinux		3.4
python3-libsemanage		3.4
python3-libstorage mgmt		1.9.4
python3-lockfile		0.12.2
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip-wheel		21.3.1
python3-ply		3.11
python3-policycore utils		3.4
python3-prettytable		0.7.2
python3-prompt-too lkit		3.0,24
python3-pycparser		2,20
python3-pyrsistent		0,17,3
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pytz		2022.7.1

Paket	AL1 AMI	AL2023 AMI
python3-pyyaml		5.4.1
python3-requests		2,25,1
python3-rpm		4.16.1,3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools		59,6,0
python3-setuptools-wheel		59,6,0
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1,25,10
python3-wcwidth		0,2,5
python-chevron		0.13.1
python-srpm-macros		3.9
quota	4,00	4,06
quota-nls	4,00	4,06
readline	6.2	8,1
rmt	0.4	
rng-tools	5	6,14

Paket	AL1 AMI	AL2023 AMI
rootfiles	8.1	8.1
rpcbind	0.2.0	1.2.6
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit		4.16.1.3
rpm-python27	4.11.3	
rpm-sign-libs		4.16.1.3
rsync	3.0.6	3.2.6
rsyslog	5.8.10	
ruby	2.0	
ruby20	2.0.0.648	
ruby20-irb	2.0.0.648	
ruby20-libs	2.0.0.648	
rubygem20-bigdecimal	1.2.0	
rubygem20-json	1.8.3	
rubygem20-psych	2.0.0	
rubygem20-rdoc	4.2.2	
rubygems20	2.0.14.1	

Paket	AL1 AMI	AL2023 AMI
rust-srpm-macros		21
sbsigntools		0.9.4
screen	4.0.3	4.8.0
sed	4.2.1	4.8
selinux-policy		37,22
selinux-policy-targeted		37,22
sendmail	8,1,4	
setserial	2,17	
setup	2.8,14	2.13,7
sgpio	1.2.0.10	
shadow-utils	4.1.4.2	4,9 bis 4,9
shared-mime-info	1.1	
slang	2.2.1	2.3.2
sqlite	3.7.17	
sqlite-libs		3,40,0
sssd-client		2.9.4
sssd-common		2.9.4
sssd-kcm		2.9.4
sssd-nfs-idmap		2.9.4
strace		6.8

Paket	AL1 AMI	AL2023 AMI
sudo	1.8.23	1.9,15
sysctl-defaults	1,0	1,0
sysfsutils	2.1.0	
sysstat		12,5.6
systemd		252,16
systemd-libs		252,16
systemd-networkd		252,16
systemd-pam		252,16
systemd-resolved		252,16
systemd-udev		252,16
system-release	2018,03	2023,4.20240513
systemtap-runtime		4.8
sysvinit	2,87	
tar	1,26	1,34
tbb		2020,3
tcp_wrappers	7.6	
tcp_wrappers-libs	7.6	
tcpdump		4,99,1
tcsh		6,24,07
time	1,7	1.9

Paket	AL1 AMI	AL2023 AMI
tmpwatch	2.9,16	
traceroute	2.0.14	2.1.3
ttmkfdir	3.0.9	
tzdata	2023c	2024a
tzdata-java	2023c	
udev	173	
unzip	6.0	6.0
update-motd	1.0.1	2.2
upstart	0,6,5	
userspace-rcu		0.12.1
ustr	1.0.4	
util-linux	2,23,2	2,37,4
util-linux-core		2,37,4
vim-common	9,0,2120	9,0,2153
vim-data	9,0,2120	9,0,2153
vim-enhanced	9,0,2120	9,0,2153
vim-filesystem	9,0,2120	9,0,2153
vim-minimal	9,0,2120	9,0,2153
wget	1,18	1,21,3
which	2,19	2,21

Paket	AL1 AMI	AL2023 AMI
words	3.0	3.0
xfsdump		3.1.11
xfsprogs		5.18.0
xorg-x11-fonts-Type1	7.2	
xorg-x11-font-utils	7.2	
xxd	9,0,2120	9,0,2153
xxhash-libs		0.8.0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	1.1.4	
yum-plugin-priorities	1.1.31	
yum-plugin-upgrade-helper	1.1.31	
yum-utils	1.1.31	
zip	3.0	3.0
zlib	1.2.8	1.2.11
zram-generator		1.1.2
zram-generator-defaults		1.1.2

Paket	AL1 AMI	AL2023 AMI
zstd		1.5.5

Vergleich der auf Amazon Linux 1 (AL1) und Amazon Linux 2023 Minimal AMIs installierten Pakete

Ein Vergleich der auf den Minimal-AMIs AL1 und AL2023 vorhandenen RPMs.

Paket	AL1 Minimal	AL2023 Minimal
acpid	2.0.19	
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.4.1
amazon-linux-repo-s3		2023,4.20240513
amazon-linux-sb-keys		2023,1
audit	2.6.5	3.0.6
audit-libs	2.6.5	3.0.6
authconfig	6.2.8	
awscli-2		2,15,30
basesystem	10.0	11
bash	4.2,46	5.2.15
binutils	2,27	
bzip2	1.0.6	

Paket	AL1 Minimal	AL2023 Minimal
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023,2,62	2023,2,64
checkpolicy	2.1.10	3.4
chkconfig	1.3.49,3	
chrony		4.3
cloud-disk-utils	0,27	
cloud-init	0,7.6	22.2.2
cloud-init-cfg-ec2		22.2.2
cloud-utils-growpart		0,31
coreutils	8,22	8,32
coreutils-common		8,32
cpio	(2.10)	2,13
cracklib	2.8,16	2.9.6
cracklib-dicts	2.8,16	2.9.6
cronie	1.4.4	
cronie-anacron	1.4.4	
crontabs	1.10	
crypto-policies		20220428
cryptsetup-libs		2.6.1
curl	7,61,1	

Paket	AL1 Minimal	AL2023 Minimal
curl-minimal		8,5,0
cyrus-sasl	2.1.23	
cyrus-sasl-lib	2.1.23	2.1.27
dash	0.5.5.1	
db4	4.7.25	
db4-utils	4,7,25	
dbus		1.12,28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.6.12	1.12.28
device-mapper		1,02,185
device-mapper-libs		1,02,185
dhclient	4.1.1	
dhcp-common	4.1.1	
diffutils	3.3	3.8
dnf		4.14.0
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0

Paket	AL1 Minimal	AL2023 Minimal
dnf-plugin-support-info		1.2
dracut	004	055
dracut-config-ec2		3.0
dracut-config-generic		055
dracut-modules-growroot	0.20	
e2fsprogs	1,43,5	1,46,5
e2fsprogs-libs	1,43,5	1,46,5
ec2-utils	0.7	2.2.0
ed	1.1	
efi-filesystem		5
efivar		38
efivar-libs		38
elfutils-default-yama-scope		0.188
elfutils-libelf	0,168	0.188
elfutils-libs		0.188
ethtool	3,15	
expat	2.1.0	2.5.0
file	5,37	5,39

Paket	AL1 Minimal	AL2023 Minimal
file-libs	5,37	5,39
filesystem	2,4,30	3,14
findutils	4.4.2	4.8.0
fipscheck	1.3.1	
fipscheck-lib	1.3.1	
fuse-libs	2.9.4	2.9.9
gawk	3.1.7	5.1.0
gdbm	1.8.0	
gdbm-libs		1,19
gdisk	0,8,10	1.0.8
generic-logos	17.0.0	
get_reference_source	1.2	
gettext		0,21
gettext-libs		0,21
glib2	2,36,3	2,74,7
glibc	2,17	2,34
glibc-all-langpacks		2,34
glibc-common	2,17	2,34
glibc-locale-source		2,34
gmp	6.0.0	6.2.1

Paket	AL1 Minimal	AL2023 Minimal
gnupg2	2.0.28	
gnupg2-minimal		2.3.7
gnutls		3.8.0
gpgme	1.4.3	1.15.1
grep	2,20	3.8
groff	1,22,2	
groff-base	1.22.2	1.22,4
grub	0,97	
grub2-common		2,06
grub2-efi-x64-ec2		2,06
grub2-pc-modules		2,06
grub2-tools		2,06
grub2-tools-minimal		2,06
grubby	7,0,15	8,40
gzip	1.5	1.12
hesiod	3.1.0	
hmaccalc	0,9,12	
hostname		3,23
hwdata	0,233	0,353
info	5.1	

Paket	AL1 Minimal	AL2023 Minimal
inih		49
initscripts	9,03.58	10,09
iproute	4.4.0	5.10.0
iptables	1,4,21	
iputils	20121221	20210202
irqbalance		1.9.0
jansson		2.14
jitterentropy		3.4.1
jq		1.7.1
json-c		0,14
kbd	1.15	2.4.0
kbd-misc	1.15	2.4.0
kernel	4,14,336	6.1,90
kernel-livepatch-repo-s3		2023,4,20240513
keyutils-libs	1,5.8	1.6.3
kmod	14	29
kmod-libs	14	29
krb5-libs	1.15.1	1,21
less	436	608
libacl	2,2,49	2.3.1

Paket	AL1 Minimal	AL2023 Minimal
libarchive		3.5.3
libargon2		20171227
libassuan	2.0.3	2,5.5
libattr	2,4,46	2.5.1
libblkid	2,23,2	2,37,4
libcap	2,16	2,48
libcap54	2,54	
libcap-ng	0,7,5	0.8.2
libcbor		0.7.0
libcgroup	0,40.rc1	
libcom_err	1,43,5	1,46,5
libcomps		0,120
libcurl	7,61,1	
libcurl-minimal		8,5,0
libdb		5.3.28
libdnf		0,69,0
libeconf		0,4,0
libedit	2.11	3.1
libfdisk		2,37,4
libffi	3.0,13	3.4.4

Paket	AL1 Minimal	AL2023 Minimal
libfido2		1.10.0
libgcc		11.4.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.11	1,42
libicu	50,2	
libidn	1,18	
libidn2	2.3.0	2.3.2
libkcapi		1.4.0
libkcapi-hmaccalc		1.4.0
libmnl	1.0.3	1.0.4
libmodulemd		2.13.0
libmount	2,23,2	2,37,4
libnetfilter_conntrack	1.0.4	
libnfnetlink	1.0.1	
libnghttp2	1,33,0	1,59,0
libnih	1.0.1	
libpipeline		1.5.3
libpsl	0.6.2	0,21,1

Paket	AL1 Minimal	AL2023 Minimal
libpwquality	1.2.3	1.4.4
librepo		1,14,5
libreport-filesystem		2.15,2
libseccomp		2.5.3
libselinux	2.1.10	3.4
libselinux-utils	2.1.10	3.4
libsemanage	2.1.6	3.4
libsepol	2.1.7	3.4
libsigsegv		2,13
libsmartcols	2,23,2	2,37,4
libsolv		0,7.22
libss	1,43,5	1,46,5
libssh2	1.4.2	
libstdc++		11.4.1
libstdc++72	7.2.1	
libsysfs	2.1.0	
libtasn1	2.3	4.19.0
libtextstyle		0,21
libudev	173	
libunistring	0.9.3	0,9,10

Paket	AL1 Minimal	AL2023 Minimal
libuser	0,60	0,63
libutempter	1.1.5	1.2.1
libuuid	2,23,2	2,37,4
libverto	0,2,5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libyaml	0.1.6	0.2.5
libzstd		1.5.5
logrotate	3.7.8	3.20.1
lua	5.1.4	
lua-libs		5.4.4
lz4-libs		1.9.4
make	3,82	
man-db		2.9.3
microcode_ctl	2.1	2.1
mingetty	1,08	
mpfr		4.1.0
ncurses	5,7	6.2
ncurses-base	5,7	6.2
ncurses-libs	5,7	6.2

Paket	AL1 Minimal	AL2023 Minimal
nettle		3.8
net-tools	1,60	2.0
newt	0,52,11	
newt-python27	0,52,11	
npth		1,6
nspr	4,25,0	
nss	3,53,1	
nss-pem	1.0.3	
nss-softokn	3,53,1	
nss-softokn-freebl	3,53,1	
nss-sysinit	3,53,1	
nss-tools	3,53,1	
nss-util	3,53,1	
ntp	4.2.8 p 15	
ntpdate	4.2.8p15	
numactl-libs		2.0.14
oniguruma		6.9.7.1
openldap	2.4.40	2,4,57
openssh	7,4p1	8,7 p1
openssh-clients		8,7 p1

Paket	AL1 Minimal	AL2023 Minimal
openssh-server	7,4p1	8,7 p1
openssl	1,2k	3.0.8
openssl-libs		3.0.8
openssl-pkcs11		0.4.12
os-prober		1,77
p11-kit	0,18,5	0,24,1
p11-kit-trust	0,18,5	0,24,1
pam	1.1.8	1.5.1
passwd	0,79	0,80
pciutils	3.1.10	3.7.0
pciutils-libs	3.1.10	3.7.0
pcre	8,21	
pcre2		10,40
pcre2-syntax		10,40
pinentry	0,7.6	
pkgconfig	0,27,1	
policycoreutils	2.1.12	3.4
popt	1.13	1,18
procmail	3,22	
procps	3.2.8	

Paket	AL1 Minimal	AL2023 Minimal
procps-ng		3.3.17
psmisc	22,20	23,4
pth	2.0.7	
publicsuffix-list-dafsa		20240212
python27	2.7,18	
python27-babel	0.9.4	
python27-backports	1,0	
python27-backports-ssl_match_hostname	3.4.0.2	
python27-chardet	2.0.1	
python27-configobj	4.7.2	
python27-iniparse	0.3.1	
python27-jinja2	2.7.2	
python27-jsonpatch	1.2	
python27-jsonpointer	1,0	
python27-libs	2.7.18	
python27-markupsafe	0,11	
python27-pycurl	7,19,0	
python27-pygpgme	0.3	
python27-pyliblzma	0.5.3	

Paket	AL1 Minimal	AL2023 Minimal
python27-pyxattr	0.5.0	
python27-PyYAML	3,10	
python27-requests	1.2.3	
python27-setuptools	36,2,7	
python27-six	1.8.0	
python27-urlgrabber	3,10	
python27-urllib3	1,24,3	
python3		3.9,16
python3-attrs		20,3,0
python3-audit		3.0.6
python3-awscrt		0.19,19
python3-babel		2.9.1
python3-cffi		1.14,5
python3-chardet		4.0.0
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36,1
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0

Paket	AL1 Minimal	AL2023 Minimal
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils		0,16
python3-gpg		1.15.1
python3-hawkey		0,69,0
python3-idna		(2.10)
python3-jinja2		2.11,3
python3-jmespath		0.10.0
python3-jsonpatch		1,21
python3-jsonpointer		2.0
python3-jsonschema		3.2.0
python3-libcomps		0,120
python3-libdnf		0,69,0
python3-libs		3,9,16
python3-libselinux		3.4
python3-libsemanage		3.4
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip-wheel		21.3.1

Paket	AL1 Minimal	AL2023 Minimal
python3-ply		3.11
python3-policycoreutils		3.4
python3-prettytable		0.7.2
python3-prompt-toolkit		3.0,24
python3-pycparser		2,20
python3-pyrsistent		0,17,3
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pytz		2022.7.1
python3-pyyaml		5.4.1
python3-requests		2,25,1
python3-rpm		4.16.1,3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools		59,6,0
python3-setuptools-wheel		59,6,0
python3-six		1.15.0

Paket	AL1 Minimal	AL2023 Minimal
python3-systemd		235
python3-urllib3		1,25,10
python3-wcwidth		0,2,5
readline	6.2	8.1
rng-tools		6,14
rootfiles	8.1	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit		4.16.1.3
rpm-python27	4.11.3	
rpm-sign-libs		4.16.1.3
rsyslog	5.8.10	
sbsigntools		0.9.4
sed	4.2.1	4.8
selinux-policy		37,22
selinux-policy-targeted		37,22
sendmail	8,1,4	

Paket	AL1 Minimal	AL2023 Minimal
setserial	2,17	
setup	2.8,14	2.13,7
shadow-utils	4.1.4.2	4,9 bis 4,9
shared-mime-info	1.1	
slang	2.2.1	
sqlite	3.7.17	
sqlite-libs		3,40,0
sudo	1,8,23	1.9,15
sysctl-defaults	1,0	1,0
sysfsutils	2.1.0	
systemd		252,16
systemd-libs		252,16
systemd-networkd		252,16
systemd-pam		252,16
systemd-resolved		252,16
systemd-udev		252,16
system-release	2018,03	2023,4.20240513
sysvinit	2,87	
tar	1,26	1,34
tcp_wrappers-libs	7.6	

Paket	AL1 Minimal	AL2023 Minimal
tzdata	2023c	2024a
udev	173	
update-motd	1.0.1	2.2
upstart	0,6,5	
userspace-rcu		0.12.1
ustr	1.0.4	
util-linux	2,23,2	2,37,4
util-linux-core		2,37,4
vim-data	9,0,2120	9,0,2153
vim-minimal	9,0,2120	9,0,2153
which	2,19	2,21
xfsprogs		5,18,0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	1.1.4	
yum-plugin-priorities	1.1.31	
yum-plugin-upgrade-helper	1.1.31	
zlib	1.2.8	1.2.11

Paket	AL1 Minimal	AL2023 Minimal
zram-generator		1.1.2
zram-generator-defaults		1.1.2
zstd		1.5.5

Vergleich der auf Amazon Linux 1 (AL1) und Amazon Linux 2023Basis-Container-Images installierten Pakete

Ein Vergleich der RPMs, die auf den Basiscontainer-Images AL1 und AL2023 vorhanden sind.

Paket	AL1-Behälter	AL2023 Behälter
alternatives		1.15
amazon-linux-repo-cdn		2023.4.20240513
audit-libs		3.0.6
basesystem	10.0	11
bash	4.2,46	5.2.15
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023,2,62	2023,2,64
chkconfig	1.3.49,3	
coreutils	8,22	
coreutils-single		8,32
crypto-policies		20220428

Paket	AL1-Behälter	AL2023 Behälter
curl	7,61,1	
curl-minimal		8,5,0
cyrus-sasl-lib	2.1.23	
db4	4.7,25	
db4-utils	4,7,25	
dnf		4.14.0
dnf-data		4.14.0
elfutils-default-yama-scope		0.188
elfutils-libelf	0,168	0.188
elfutils-libs		0.188
expat	2.1.0	2.5.0
file-libs	5,37	5,39
filesystem	2,4,30	3,14
gawk	3.1.7	5.1.0
gdbm	1.8.0	
gdbm-libs		1,19
glib2	2,36,3	2,74,7
glibc	2,17	2,34
glibc-common	2,17	2,34

Paket	AL1-Behälter	AL2023 Behälter
glibc-minimal-langpack		2,34
gmp	6.0.0	6.2.1
gnupg2	2.0.28	
gnupg2-minimal		2.3.7
gpgme	1.4.3	1.15.1
grep	2,20	3.8
gzip	1.5	
info	5.1	
json-c		0,14
keyutils-libs	1,5.8	1.6.3
krb5-libs	1.15.1	1,21
libacl	2,2,49	2.3.1
libarchive		3.5.3
libassuan	2.0.3	2.5.5
libattr	2,4,46	2.5.1
libblkid		2,37,4
libcap	2,16	2,48
libcap-ng		0.8.2
libcom_err	1,43,5	1,46,5
libcomps		0,120

Paket	AL1-Behälter	AL2023 Behälter
libcurl	7,61,1	
libcurl-minimal		8,5,0
libdnf		0,69,0
libffi	3.0,13	3.4.4
libgcc		11.4.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.11	1,42
libicu	50,2	
libidn2	2.3.0	2.3.2
libmodulemd		2.13.0
libmount		2,37,4
libnghttp2	1,33,0	1,59,0
libpsl	0.6.2	0,21,1
librepo		1,14,5
libreport-filesystem		2.15,2
libselinux	2.1.10	3.4
libsepol	2.1.7	3.4
libsigsegv		2,13

Paket	AL1-Behälter	AL2023 Behälter
libsmartcols		2,37,4
libsolv		0,7.22
libssh2	1.4.2	
libstdc++		11,4,1
libstdc++72	7.2.1	
libtasn1	2.3	4.19.0
libunistring	0.9.3	0.9.10
libuuid		2,37,4
libverto	0,2,5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libxml2-python27	2.9.1	
libyaml		0.2.5
libzstd		1.5.5
lua	5.1.4	
lua-libs		5.4.4
lz4-libs		1.9.4
make	3,82	
mpfr		4.1.0
ncurses	5,7	

Paket	AL1-Behälter	AL2023 Behälter
ncurses-base	5,7	6.2
ncurses-libs	5,7	6.2
npth		1,6
nspr	4,25,0	
nss	3,53,1	
nss-pem	1.0.3	
nss-softokn	3,53,1	
nss-softokn-freebl	3,53,1	
nss-sysinit	3,53,1	
nss-tools	3,53,1	
nss-util	3,53,1	
openldap	2,4,40	
openssl	1,2k	
openssl-libs		3.0.8
p11-kit	0,18,5	0,24,1
p11-kit-trust	0,18,5	0,24,1
pcre	8,21	
pcre2		10,40
pcre2-syntax		10,40
pinentry	0,7.6	

Paket	AL1-Behälter	AL2023 Behälter
pkgconfig	0,27,1	
popt	1.13	1,18
pth	2.0.7	
publicsuffix-list-dafsa		20240212
python27	2.7,18	
python27-chardet	2.0.1	
python27-iniparse	0.3.1	
python27-kitchen	1.1.1	
python27-libs	2.7.18	
python27-pycurl	7.19.0	
python27-pygpgme	0.3	
python27-pyliblzma	0.5.3	
python27-pyxattr	0.5.0	
python27-urlgrabber	3,10	
python3		3.9,16
python3-dnf		4.14.0
python3-gpg		1.15.1
python3-hawkey		0,69,0
python3-libcomps		0,120
python3-libdnf		0,69,0

Paket	AL1-Behälter	AL2023 Behälter
python3-libs		3,9,16
python3-pip-wheel		21.3.1
python3-rpm		4.16.1.3
python3-setuptools-wheel		59,6,0
readline	6.2	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-python27	4.11.3	
rpm-sign-libs		4.16.1.3
sed	4.2.1	4.8
setup	2.8,14	2.13,7
shared-mime-info	1.1	
sqlite	3.7.17	
sqlite-libs		3,40,0
sysctl-defaults	1,0	
system-release	2018,03	2023,4.20240513
tar	1,26	
tzdata	2023c	2024a
xz-libs	5.2.2	5.2.5

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

AL2023 Systemanforderungen

In diesem Abschnitt werden die Systemanforderungen für die Verwendung von AL2023 beschrieben.

Themen

- CPU-Anforderungen für die Ausführung von AL2023
- Speicheranforderungen (RAM) für die Ausführung von AL2023

CPU-Anforderungen für die Ausführung von AL2023

Um beliebigen AL203-Code ausführen zu können, muss der verwendete Prozessor bestimmte Mindestanforderungen erfüllen. Versuche, AL2023 auf CPUs auszuführen, die diese Anforderungen nicht erfüllen, können sehr früh in der Codeausführung zu illegalen Befehlsfehlern führen.

Die Mindestanforderungen gelten für AL2023 auf Amazon EC2AL2023 in Containern, und. AL2023 außerhalb von Amazon EC2

ARM-CPU-Anforderungen für AL2023

Alle AL2023 `aarch64` (ARM) -Binärdateien sind für 64-Bit konzipiert. Es sind keine ARM 32-Bit-Binärdateien verfügbar, daher ist eine ARM 64-Bit-CPU erforderlich.

> **Note**
>
> Für ARM-basierte Instances unterstützt AL2023 nur Instance-Typen, die Graviton2- oder neuere Prozessoren verwenden. AL2023 unterstützt keine A1-Instances.

AL2023 benötigt einen ARMv8.2-kompatiblen Prozessor mit Cryptography Extension (`ARMv8.2+crypto`). Alle AL2023-Pakete für `aarch64` werden mit dem `-march=armv8.2-a+crypto` Compiler-Flag erstellt. Wir versuchen zwar, ansprechende Fehlermeldungen zu drucken, wenn versucht wird, AL203-Code auf älteren ARM Prozessoren auszuführen, aber es ist möglich, dass es sich bei der ersten Fehlermeldung um einen unzulässigen Befehlsfehler handelt.

> **Note**
>
> Aufgrund der `aarch64` CPU-Basisanforderungen von AL2023 erfüllen alle Raspberry Pi Systeme vor dem Raspberry Pi 5 nicht die Mindestanforderungen an die CPU.

x86-64-CPU-Anforderungen für AL2023

Alle `x86-64` AL2023-Binärdateien wurden für die `x86-64v2` Überarbeitung der `x86-64` Architektur erstellt, indem sie an den Compiler `-march=x86-64-v2` übergeben werden.

Die `x86-64v2` Revision der Architektur fügt zusätzlich zur Basisarchitektur die folgenden CPU-Funktionen hinzu: `x86-64`

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

Dies entspricht in etwa `x86-64` Prozessoren, die 2009 oder später veröffentlicht wurden. Beispiele hierfür sind die Mikroarchitekturen Intel Nehalem AMD JaguarAtom Silvermont,,, VIA Nano sowie die Eden C Mikroarchitekturen.

In Amazon EC2 unterstützen alle `x86-64`-Instance-Typen `x86-64v2`, einschließlich der `M1`-, `C1`- und `M2`-Instance-Familien.

Es wurden keine 32-Bit-x86 (i686) -AL2023-Binärdateien erstellt. AL2023 unterstützt zwar weiterhin die Ausführung von 32-Bit-Userspace-Binärdateien, diese Funktionalität ist jedoch veraltet und könnte in einer future Hauptversion von Amazon Linux entfernt werden. Weitere Informationen finden Sie unter 32-Bit x86-(i686)-Pakete.

Speicheranforderungen (RAM) für die Ausführung von AL2023

Die Amazon EC2 `.nano` EC2-Familie von Instance-Typen (`t2.nano`, `t3.nanot3a.nano`, und`t4g.nano`) verfügt über 512 MB RAM, was die Mindestanforderung für AL2023 ist.

> **Note**
>
> Obwohl 512 MB die Mindestanforderung sind, sind diese Instance-Typen speicherbeschränkt und Funktionalität und Leistung können eingeschränkt sein.

AL203-Images wurden nicht auf Systemen mit weniger als 512 MB RAM getestet. Die Ausführung von AL2023-basierten Container-Images in weniger als 512 MB RAM hängt von der containerisierten Arbeitslast ab.

Für einige Workloads, z. B. `dnf update` zwischen einigen AL2023-Versionen, können mehr als 512 MB RAM erforderlich sein. Aus diesem Grund wurde mit der Version AL2023.3 die standardmäßige Aktivierung `zram` für Instances mit weniger als 800 MB RAM eingeführt. Für containerisierte Workloads bedeutet dies, dass einige Workloads auf AL2023-Instances mit dieser Speichermenge möglicherweise problemlos ausgeführt werden, aber fehlschlagen, wenn sie in einem Container ausgeführt werden, der auf diese Speicherbelegung beschränkt ist.

Für Instance-Typen mit weniger als 800 MB RAM aktiviert AL2023 (ab AL2023.3 oder höher) ab sofort standardmäßig einen `zram`-basierten Swap. Beispiele für Amazon EC2 EC2-Instance-Typen mit weniger als 800 MB Arbeitsspeicher sind`t4g.nano`,`t3a.nano`, `t3.nanot2.nano`, und`t1.micro`. Für diese Instance-Typen bedeutet das weniger Szenarien mit unzureichendem Arbeitsspeicher, da AL2023 Speicherseiten bei Bedarf komprimiert und dekomprimiert. Dies ermöglicht Workloads, für die andernfalls ein Instance-Typ mit mehr Arbeitsspeicher erforderlich wäre, und zwar auf Kosten der für die Komprimierung erforderlichen CPU-Auslastung.

Verwenden von AL2023 auf AWS

Sie können AL2023 für die Verwendung mit anderen einrichten. AWS-Services Sie können beispielsweise ein AL2023 AMI wählen, wenn Sie eine Amazon Elastic Compute Cloud (Amazon EC2) -Instance starten.

Für diese Einrichtungsverfahren verwenden Sie den AWS Identity and Access Management (IAM) -Service. Umfassende Informationen zu IAM finden Sie in den folgenden Referenzmaterialien:

- AWS Identity and Access Management (IAM)
- IAM Benutzerhandbuch

Themen

- Erste Schritte mit AWS
- AL2023 auf Amazon EC2
- Verwendung von AL2023 in Containern
- AL2023 ein AWS Elastic Beanstalk
- Verwendung von AL2023 in AWS CloudShell
- Verwendung von AL2023-basierten Amazon ECS-AMIs zum Hosten containerisierter Workloads
- Verwenden von Amazon Elastic File System auf AL2023
- Verwenden von Amazon EMR, das auf AL2023 basiert
- Verwendung von AL2023 in AWS Lambda

Erste Schritte mit AWS

Melde dich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Online-Anweisungen.

Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich ist.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu https://aws.amazon.com/ auf und klicken Sie auf Mein Konto.

Erstellen Sie einen Benutzer mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS Management Consoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen Sie einen Benutzer mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.
2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis im AWS IAM Identity Center Benutzerhandbuch.

Melden Sie sich als Benutzer mit Administratorzugriff an

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Zugriffsportal.

Weisen Sie weiteren Benutzern Zugriff zu

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

   Anweisungen finden Sie im Benutzerhandbuch unter Einen Berechtigungssatz erstellen.AWS IAM Identity Center
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Anweisungen finden Sie im AWS IAM Identity Center Benutzerhandbuch unter Gruppen hinzufügen.

Erteilen programmgesteuerten Zugriffs

Benutzer benötigen programmatischen Zugriff, wenn sie mit AWS außerhalb des interagieren möchten. AWS Management Console Die Art und Weise, wie programmatischer Zugriff gewährt wird, hängt vom Benutzertyp ab, der zugreift. AWS

Um Benutzern programmgesteuerten Zugriff zu gewähren, wählen Sie eine der folgenden Optionen.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
Mitarbeiteridentität (Benutzer, die in IAM Identity Center verwaltet werden)	Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an die AWS CLI, AWS SDKs oder APIs zu signieren. AWS	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen zu den AWS CLI finden Sie unter Konfiguration der AWS CLI zu AWS IAM Identity Center verwendenden im AWS Command Line Interface Benutzerhandbuch. • Informationen zu AWS SDKs, Tools und AWS APIs finden Sie unter IAM Identity Center-Authentifizierung im Referenzhandbuch für AWS SDKs und Tools.
IAM	Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an die AWS CLI, AWS SDKs oder APIs zu signieren. AWS	Folgen Sie den Anweisungen unter Verwenden temporäre r Anmeldeinformationen mit AWS Ressourcen im IAM-Benutzerhandbuch.
IAM	(Nicht empfohlen) Verwenden Sie langfristige Anmeldeinformationen, um programmatische Anfragen an die AWS CLI, AWS SDKs oder APIs zu signieren. AWS	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen dazu finden Sie unter Authentifizierung mithilfe von IAM-Benut zeranmeldedaten im Benutzerhandbuch. AWS

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
		CLIAWS Command Line Interface • Informationen zu AWS SDKs und Tools finden Sie unter Authentifizieren mit langfristigen Anmeldeinformationen im Referenzhandbuch für AWS SDKs und Tools. • Informationen zu AWS APIs finden Sie unter Verwaltung von Zugriffsschlüsseln für IAM-Benutzer im IAM-Benutzerhandbuch.

AL2023 auf Amazon EC2

Verwenden Sie eines der folgenden Verfahren, um eine Amazon EC2 EC2-Instance mit einem AL2023 AMI zu starten. Wählen Sie entweder das Standard-AMI oder das Minimal-AMI aus. Weitere Informationen zu den Unterschieden zwischen dem Standard-AMI und dem Minimal-AMI finden Sie unter Vergleich von AL2023 Standard (Standardversion) und Minimal-AMIs.

Themen

- AL2023 mit der Amazon EC2 EC2-Konsole starten
- Starten von AL2023 mit dem SSM-Parameter und AWS CLI
- Starten des neuesten AL2023 AMI mit AWS CloudFormation
- AL2023 mit einer bestimmten AMI-ID starten
- AL2023 AMI: Veraltete Version und Lebenszyklus
- Verbindung zu AL203-Instances herstellen
- Vergleich von AL203-Standard- und Minimal-AMIs

AL2023 mit der Amazon EC2 EC2-Konsole starten

Starten einer AL2023-Instance in der Amazon-EC2-Konsole.

> **Note**
>
> Für ARM-basierte Instances unterstützt AL2023 nur Instance-Typen, die Graviton2- oder neuere Prozessoren verwenden. AL2023 unterstützt keine A1-Instances.

Führen Sie die folgenden Schritte aus, um eine Amazon-EC2-Instance mit AL2023 AMI in der Amazon-EC2-Konsole zu starten.

So starten Sie eine EC2-Instance mit einem AL2023 AMI

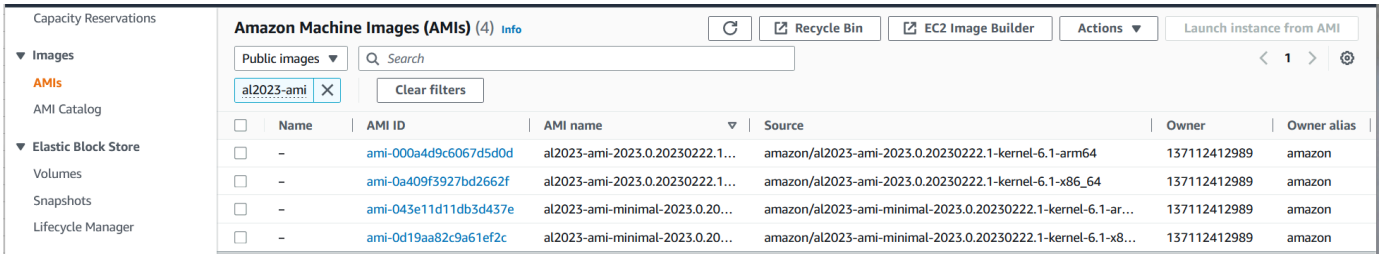
1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich die Option AMIs.
3. Wählen Sie im Dropdown-Menü Öffentliche Abbilder aus.
4. Geben Sie im Suchfeld **`al2023-ami`** ein.

   > **Note**
   >
   > Vergewissern Sie sich, dass in der Spalte Eigentümer-Alias Amazon angezeigt wird.

5. Wählen Sie eine Abbildung aus der Liste aus. Unter Quelle können Sie festlegen, ob es sich bei dem AMI um ein Standard- oder ein Minimal-AMI handeln soll. Ein AL2023-AMI-Name kann mit diesem Format interpretiert werden:

   ```
   'al2023-[ami || ami-minimal]-2023.0.[release build date].[build
   number]-kernel-[version number]-[arm64 || x86_64]'
   ```

6. In der folgenden Abbildung sehen Sie eine Teilliste der AL2023-AMIs.



Weitere Informationen zum Starten von Amazon EC2 EC2-Instances finden Sie unter Erste Schritte mit Amazon EC2 EC2-Linux-Instances im Amazon EC2 EC2-Benutzerhandbuch.

Starten von AL2023 mit dem SSM-Parameter und AWS CLI

In der AWS CLI können Sie den SSM-Parameterwert eines AMI verwenden, um eine neue Instanz von AL2023 zu starten. Genauer gesagt, sollten Sie einen der dynamischen SSM-Parameterwerte aus der folgenden Liste verwenden und `/aws/service/ami-amazon-linux-latest/` vor dem SSM-Parameterwert hinzufügen. Hiermit starten Sie die Container-Instance in AWS CLI.

- `al2023-ami-kernel-default-arm64` für arm64-Architektur
- `al2023-ami-minimal-kernel-default-arm64` für arm64-Architektur (Minimal AMI)
- `al2023-ami-kernel-default-x86_64` für x86_64-Architektur
- `al2023-ami-minimal-kernel-default-x86_64` für x86_64-Architektur (Minimal AMI)

> **Note**
>
> Jede *kursive* Position ist ein Beispielparameter. Ersetzen Sie diese mit Ihren eigenen Daten.

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

Die `--image-id`-Markierung gibt den SSM-Parameterwert an.

Die `--instance-type`-Markierung gibt den Typ und die Größe der Instance an. Diese Markierung muss mit dem ausgewählten AMI-Typ kompatibel sein.

Das `--region` Flag gibt an AWS-Region , wo Sie Ihre Instance erstellen.

Das `--key-name` Flag gibt den AWS-Region Schlüssel an, der für die Verbindung mit der Instance verwendet wird. Wenn Sie keinen Schlüssel angeben, der in der gewünschten Instance-Region bereits existiert, können Sie sich nicht über SSH mit der Instance verbinden.

Die `--security-group-ids`-Markierung gibt die Sicherheitsgruppe an, die die Zugriffsberechtigungen für ein- und ausgehenden Netzwerkverkehr festlegt.

> **Important**
>
> Das AWS CLI erfordert, dass Sie eine bestehende Sicherheitsgruppe angeben, die den Zugriff auf die Instanz von Ihrem Remote-Computer aus über den Port ermöglichtTCP:22. Geben Sie keine Sicherheitsgruppe an, so wird Ihre neue Instanz in eine Standardsicherheitsgruppe aufgenommen. In einer Standardsicherheitsgruppe kann Ihre Instance nur Verbindungen zu den anderen Instances innerhalb Ihrer VPC herstellen.

Weitere Informationen finden Sie unter Starten, Auflisten und Beenden von Amazon-EC2-Instances im AWS Command Line Interface -Benutzerhandbuch.

Starten des neuesten AL2023 AMI mit AWS CloudFormation

Verwenden Sie eine der folgenden Vorlagen AWS CloudFormation, um ein AL2023 AMI mit zu starten.

> **Note**
>
> Die `x86_64`- und `Arm64`-AMIs setzen unterschiedliche Instance-Typen voraus. Weitere Informationen finden Sie unter Amazon-EC2-Instance-Typen.

`JSON`-Vorlage:

```
{
  "Parameters": {
    "LatestAmiId": {
      "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
      "Default": "/aws/service/ami-amazon-linux-latest/al2023-ami-minimal-kernel-
default-x86_64"
    }
  },
  "Resources": {
    "MyEC2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
```

```
        "InstanceType": "t2.large",
        "ImageId": {
          "Ref": "LatestAmiId"
        }
      }
    }
  }
}
```

YAML-Vorlage:

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

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

Stellen Sie sicher, dass Sie den AMI-Parameter am Ende des Abschnitts „Standard“ falls erforderlich ersetzen. Die folgenden Parameterwerte können verwendet werden:

- `al2023-ami-kernel-6.1-arm64` für arm64-Architektur
- `al2023-ami-minimal-kernel-6.1-arm64` für arm64-Architektur (Minimal AMI)
- `al2023-ami-kernel-6.1-x86_64` für x86_64-Architektur
- `al2023-ami-minimal-kernel-6.1-x86_64` für x86_64-Architektur (Minimal AMI)

Im Folgenden sind dynamische Kernelspezifikationen aufgeführt. Die Standard-Kernel-Version ändert sich automatisch mit jedem größeren Kernel-Versionsupdate.

- `al2023-ami-kernel-default-arm64` für arm64-Architektur
- `al2023-ami-minimal-kernel-default-arm64` für arm64-Architektur (Minimal AMI)
- `al2023-ami-kernel-default-x86_64` für x86_64-Architektur
- `al2023-ami-minimal-kernel-default-x86_64` für x86_64-Architektur (Minimal AMI)

AL2023 mit einer bestimmten AMI-ID starten

Mithilfe der AMI-ID können Sie eine bestimmte AL2023-AMI starten. Ermitteln Sie die korrekte AL2023-AMI-ID anhand der AMI-Liste, die Sie in der Amazon-EC2-Konsole finden. Oder Sie können verwenden. AWS Systems Manager Wenn Sie Systems Manager verwenden, müssen Sie den AMI-Alias aus der Liste im vorigen Abschnitt auswählen. Weitere Informationen finden Sie unter Abfragen der neuesten Amazon Linux-AMI-IDs mithilfe des AWS Systems Manager Parameterspeichers.

AL2023 AMI: Veraltete Version und Lebenszyklus

Jede neue AL2023-Version enthält ein neues AMI. Wenn das AMI registriert ist, ist es mit einem Beendigungsdatum gekennzeichnet. Das Beendigungsdatum für jedes AL2023-AMI liegt 90 Tage ab dem Zeitpunkt der Veröffentlichung, also entsprechend dem Zeitraum, der für jede Kernel-Live-Patching auf AL2023-Kernel-Version angeboten wird.

> **Note**
>
> Das 90-Tage-Beendigungsdatum bezieht sich auf ein einzelnes AMI und nicht auf den AL2023-Release-Taktfrequenz oder Produktsupport-Zeitraum.

Weitere Informationen zu veralteten AMIs finden Sie unter Deprecate an AMI im Amazon EC2 EC2-Benutzerhandbuch.

Eine regelmäßige Verwendung aktualisierter AMI zum Starten einer Instance stellt sicher, dass die Instance mit den neuesten Sicherheitsupdates (einschließlich eines aktualisierten Kernels) gestartet wird. Wenn Sie eine frühere Version eines AMI starten und Updates anwenden, wird die Instance zeitweise nicht die neuesten Sicherheitsupdates verfügbar haben. Wir empfehlen die Verwendung von SSM-Parametern, um sicherzustellen, dass Sie das neueste AMI verwenden.

Weitere Informationen zur Nutzung von SSM-Parametern zum Starten einer Instance finden Sie unter:

- Starten von AL2023 mit dem SSM-Parameter und AWS CLI
- Starten des neuesten AL2023 AMI mit AWS CloudFormation

Verbindung zu AL203-Instances herstellen

Verwenden Sie SSH oder, um eine Verbindung AWS Systems Manager zu Ihrer AL2023-Instance herzustellen.

Herstellung einer Verbindung zu Ihrer Instance mit SSH

Anweisungen zur Verwendung von SSH zum Herstellen einer Connect zu einer Instance finden Sie unter Verbindung zu Ihrer Linux-Instance mithilfe von SSH herstellen im Amazon EC2 EC2-Benutzerhandbuch.

Connect zu Ihrer Instance her mit AWS Systems Manager

Anweisungen zum Herstellen einer Connect einer AL203-Instance finden Sie unter Herstellen einer Verbindung zu Ihrer Linux-Instance mithilfe von Session Manager im Amazon EC2 EC2-Benutzerhandbuch. AWS Systems Manager

Verwenden von Amazon EC2 Instance Connect

Beim AL2023 AMI, mit Ausnahme des Minimal-AMI, ist der EC2 Instance Connect-Agent standardmäßig installiert. Um EC2 Instance Connect mit einer AL2023-Instance zu verwenden, die über das Minimal-AMI gestartet wurde, müssen Sie das `ec2-instance-connect` Paket installieren. Anweisungen zur Verwendung von EC2 Instance Connect finden Sie unter Mit EC2 Instance Connect eine Verbindung zu Ihrer Linux-Instance Connect im Amazon EC2 EC2-Benutzerhandbuch.

Vergleich von AL203-Standard- und Minimal-AMIs

Sie können eine Amazon EC2 EC2-Instance entweder mit einem Standard- (Standard) oder einem minimalen AL2023-AMI starten. Anweisungen zum Starten einer Amazon EC2 EC2-Instance mit dem Standard- oder Minimal-AMI-Typ finden Sie unterAL2023 auf Amazon EC2.

Im Standard AL2023 AMI sind alle am häufigsten verwendeten Anwendungen und Tools installiert. Wir empfehlen das Standard-AMI zu verwenden, wenn Sie schnell starten möchten und nicht an einer Anpassung des AMI interessiert sind.

Das minimale AL2023 AMI ist die vereinfachte Basisversion, die nur die grundlegendsten Tools und Dienstprogramme enthält, die für den Betrieb des Betriebssystems (OS) erforderlich sind. Wir empfehlen das Minimal-AMI zu verwenden, wenn Sie das Betriebssystem so wenig wie möglich auslasten möchten. Das Minimal-AMI bietet eine leicht verringerte Festplattenspeicherauslastung und eine bessere langfristige Kosteneffizienz. Das Minimal-AMI eignet sich gut, wenn Sie ein kleineres

Betriebssystem wünschen und es Ihnen nichts ausmacht, Tools und Anwendungen manuell zu installieren.

Das Container-Image ist dem Minimal-AMI-AL2023-Paketsatz am ähnlichsten.

Vergleichen von Paketen, die auf Amazon Linux 2023 Images installiert sind

Ein Vergleich der RPMs, die auf den AL2023 AMI-, Minimal AMI- und Container-Images vorhanden sind.

Paket	AMI	Minimal-AMI	Container
`acl`	2.3.1		
`acpid`	2.0.32		
`alternatives`	1.15	1.15	1.15
`amazon-chrony-config`	4.3	4.3	
`amazon-ec2-net-utils`	2.4.1	2.4.1	
`amazon-linux-repo-cdn`			2023,4.20240513
`amazon-linux-repo-s3`	2023,4.20240513	2023,4.20240513	
`amazon-linux-sb-keys`	2023,1	2023,1	
`amazon-rpm-config`	228		
`amazon-ssm-agent`	3.3.380,0		
`at`	3.1.23		

Paket	AMI	Minimal-AMI	Container
attr	2.5.1		
audit	3.0.6	3.0.6	
audit-libs	3.0.6	3.0.6	3.0.6
aws-cfn-bootstrap	2.0		
awscli-2	2,15,30	2,15,30	
basesystem	11	11	11
bash	5.2,15	5.2.15	5.2.15
bash-completion	2.11		
bc	1,07,1		
bind-libs	9,16,48		
bind-license	9,16,48		
bind-utils	9,16,48		
binutils	2,39		
boost-filesystem	1,75,0		
boost-system	1,75,0		
boost-thread	1,75,0		
bzip2	1.0.8		
bzip2-libs	1.0.8	1.0.8	1.0.8
ca-certificates	2023,2,64	2023,2,64	2023,2,64

Paket	AMI	Minimal-AMI	Container
c-ares	1.19.0		
checkpolicy	3.4	3.4	
chkconfig	1.15		
chrony	4.3	4.3	
cloud-init	22,2,2	22.2.2	
cloud-init-cfg-ec2	22.2.2	22.2.2	
cloud-utils-growpart	0,31	0,31	
coreutils	8,32	8,32	
coreutils-common	8,32	8,32	
coreutils-single			8,32
cpio	2,13	2,13	
cracklib	2.9.6	2.9.6	
cracklib-dicts	2.9.6	2.9.6	
crontabs	1.11		
crypto-policies	20220428	20220428	20220428
crypto-policies-scripts	20220428		
cryptsetup	2.6.1		

Paket	AMI	Minimal-AMI	Container
cryptsetup-libs	2.6.1	2.6.1	
curl-minimal	8,5,0	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27	
cyrus-sasl-plain	2.1.27		
dbus	1.12,28	1.12.28	
dbus-broker	32	32	
dbus-common	1.12.28	1.12.28	
dbus-libs	1.12.28	1.12.28	
device-mapper	1,02,185	1,02,185	
device-mapper-libs	1,02,185	1,02,185	
diffutils	3.8	3.8	
dnf	4.14.0	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2	
dnf-plugins-core	4.3.0	4.3.0	
dnf-plugin-support-info	1.2	1.2	
dnf-utils	4.3.0		

Paket	AMI	Minimal-AMI	Container
dosfstools	4,2		
dracut	055	055	
dracut-config-ec2	3.0	3.0	
dracut-config-generic	055	055	
dwz	0,14		
dyninst	10.2.1		
e2fsprogs	1,46,5	1,46,5	
e2fsprogs-libs	1,46,5	1,46,5	
ec2-hibinit-agent	1.0.8		
ec2-instance-connect	1.1		
ec2-instance-connect-selinux	1.1		
ec2-utils	2.2.0	2.2.0	
ed	1.14.2		
efi-filesystem	5	5	
efi-srpm-macros	5		
efivar	38	38	
efivar-libs	38	38	

Paket	AMI	Minimal-AMI	Container
elfutils-debuginfod-client	0.188		
elfutils-default-yama-scope	0.188	0.188	0.188
elfutils-libelf	0.188	0.188	0.188
elfutils-libs	0.188	0.188	0.188
ethtool	5,15		
expat	2.5.0	2.5.0	2.5.0
file	5,39	5,39	
file-libs	5,39	5,39	5,39
filesystem	3,14	3,14	3,14
findutils	4.8.0	4.8.0	
fonts-srpm-macros	2.0.5		
fstrm	0.6.1		
fuse-libs	2.9.9	2.9.9	
gawk	5.1.0	5.1.0	5.1.0
gdbm-libs	1,19	1,19	1,19
gdisk	1.0.8	1.0.8	
gettext	0,21	0,21	

Paket	AMI	Minimal-AMI	Container
gettext-libs	0,21	0,21	
ghc-srpm-macros	1.5.0		
glib2	2,74,7	2,74,7	2,74,7
glibc	2,34	2,34	2,34
glibc-all-langpacks	2,34	2,34	
glibc-common	2,34	2,34	2,34
glibc-gconv-extra	2,34		
glibc-locale-source	2,34	2,34	
glibc-minimal-langpack			2,34
gmp	6.2.1	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7	2.3.7
gnutls	3.8.0	3.8.0	
go-srpm-macros	3.2.0		
gpgme	1.15.1	1.15.1	1.15.1
gpm-libs	1.20.7		
grep	3.8	3.8	3.8
groff-base	1,22,4	1.22,4	
grub2-common	2,06	2,06	

Paket	AMI	Minimal-AMI	Container
grub2-efi-aa64-ec2	2,06 (aarch64)	2,06 (aarch64)	
grub2-efi-x64-ec2	2,06 (x86_64)	2,06 (x86_64)	
grub2-pc-modules	2,06	2,06	
grub2-tools	2,06	2,06	
grub2-tools-minimal	2,06	2,06	
grubby	8,40	8,40	
gssproxy	0.8.4		
gzip	1.12	1.12	
hostname	3,23	3,23	
hunspell	1.7.0		
hunspell-en	0,20140811,1		
hunspell-en-GB	0,20140811,1		
hunspell-en-US	0,20140811,1		
hunspell-filesystem	1.7.0		
hwdata	0,353	0,353	
info	6.7		
inih	49	49	

Paket	AMI	Minimal-AMI	Container
initscripts	10,09	10,09	
iproute	5.10.0	5.10.0	
iputils	20210202	20210202	
irqbalance	1.9.0	1.9.0	
jansson	2.14	2.14	
jitterentropy	3.4.1	3.4.1	
jq	1.7.1	1.7.1	
json-c	0,14	0,14	0,14
kbd	2.4.0	2.4.0	
kbd-misc	2.4.0	2.4.0	
kernel	6.1,90	6.1,90	
kernel-livepatch-repo-s3	2023,4,20240513	2023,4.20240513	
kernel-srpm-macros	1,0		
kernel-tools	6.1,90		
keyutils	1.6.3		
keyutils-libs	1.6.3	1.6.3	1.6.3
kmod	29	29	
kmod-libs	29	29	

Paket	AMI	Minimal-AMI	Container
kpatch-runtime	0.9.7		
krb5-libs	1,21	1,21	1,21
less	608	608	
libacl	2.3.1	2.3.1	2.3.1
libaio	0,3,111		
libarchive	3,5.3	3.5.3	3.5.3
libargon2	20171227	20171227	
libassuan	2,5.5	2.5.5	2.5.5
libattr	2.5.1	2.5.1	2.5.1
libbasicobjects	0.1.1		
libblkid	2,37,4	2,37,4	2,37,4
libcap	2,48	2,48	2,48
libcap-ng	0.8.2	0.8.2	0.8.2
libcbor	0.7.0	0.7.0	
libcollection	0.7.0		
libcom_err	1,46,5	1,46,5	1,46,5
libcomps	0,120	0,120	0,120
libconfig	1.7.2		
libcurl-minimal	8.5.0	8.5.0	8.5.0
libdb	5.3.28	5.3.28	

Paket	AMI	Minimal-AMI	Container
libdhash	0.5.0		
libdnf	0,69,0	0,69,0	0,69,0
libeconf	0,4,0	0,4,0	
libedit	3.1	3.1	
libev	4,33		
libevent	2.1.12		
libfdisk	2,37,4	2,37,4	
libffi	3.4.4	3.4.4	3.4.4
libfido2	1.10.0	1.10.0	
libgcc	11.4.1	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2	1.10.2
libgomp	11.4.1	11.4.1	11.4.1
libgpg-error	1,42	1,42	1,42
libibverbs	48,0		
libidn2	2.3.2	2.3.2	2.3.2
libini_config	1.3.1		
libkcapi	1.4.0	1.4.0	
libkcapi-hmaccalc	1.4.0	1.4.0	
libldb	2.6.2		
libmaxminddb	1.5.2		

Paket	AMI	Minimal-AMI	Container
libmetalink	0.1.3		
libmnl	1.0.4	1.0.4	
libmodulemd	2.13.0	2.13.0	2.13.0
libmount	2,37,4	2,37,4	2,37,4
libnfsidmap	2.5.4		
libnghttp2	1,59,0	1,59,0	1,59,0
libnl3	3.5.0		
libpath_utils	0,2.1		
libpcap	1.10.1		
libpipeline	1.5.3	1.5.3	
libpkgconf	1.8.0		
libpsl	0,21,1	0,21,1	0,21,1
libpwquality	1.4.4	1.4.4	
libref_array	0,15		
librepo	1,14,5	1.14,5	1.14,5
libreport-filesystem	2.15,2	2.15,2	2.15,2
libseccomp	2.5.3	2.5.3	
libselinux	3.4	3.4	3.4
libselinux-utils	3.4	3.4	

Paket	AMI	Minimal-AMI	Container
libsemanage	3.4	3.4	
libsepol	3.4	3.4	3.4
libsigsegv	2,13	2,13	2,13
libsmartcols	2,37,4	2,37,4	2,37,4
libsolv	0,7.22	0.7.22	0.7.22
libss	1,46,5	1,46,5	
libsss_certmap	2.9.4		
libsss_idmap	2.9.4		
libsss_nss_idmap	2.9.4		
libsss_sudo	2.9.4		
libstdc++	11.4.1	11.4.1	11.4.1
libstoragemgmt	1.9.4		
libtalloc	2.3.4		
libtasn1	4.19.0	4.19,0	4.19,0
libtdb	1.4.7		
libtevent	0.13.0		
libtextstyle	0,21	0,21	
libtirpc	1.3.3		
libunistring	0,9,10	0.9,10	0.9,10
libuser	0,63	0,63	

Paket	AMI	Minimal-AMI	Container
libutempter	1.2.1	1.2.1	
libuuid	2,37,4	2,37,4	2,37,4
libuv	1.47.0		
libverto	0.3.2	0.3.2	0.3.2
libverto-libev	0.3.2		
libxcrypt	4.4.33	4.4.33	4.4.33
libxml2	2.10,4	2.10.4	2.10.4
libyaml	0.2.5	0,2,5	0,2,5
libzstd	1.5.5	1.5.5	1.5.5
lm_sensors-libs	3.6.0		
lmdb-libs	0.9.29		
logrotate	3,20,1	3,20,1	
lsof	4,94,0		
lua-libs	5.4.4	5.4.4	5.4.4
lua-srpm-macros	1		
lz4-libs	1.9.4	1.9.4	1.9.4
man-db	2.9.3	2.9.3	
man-pages	5,10		
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)	
mpfr	4.1.0	4.1.0	4.1.0

Paket	AMI	Minimal-AMI	Container
nano	5,8		
ncurses	6.2	6.2	
ncurses-base	6.2	6.2	6.2
ncurses-libs	6.2	6.2	6.2
nettle	3.8	3.8	
net-tools	2,0	2.0	
newt	0,52,21		
nfs-utils	2,5.4		
npth	1,6	1,6	1,6
nspr	4,35,0		
nss	3,90,0		
nss-softokn	3,90,0		
nss-softokn-freebl	3,90,0		
nss-sysinit	3,90,0		
nss-util	3,90,0		
ntsysv	1.15		
numactl-libs	2.0.14	2.0.14	
ocaml-srpm-macros	6		
oniguruma	6.9.7.1	6.9.7.1	

Paket	AMI	Minimal-AMI	Container
openblas-srpm-macros	2		
openldap	2.4.57	2,4,57	
openssh	8,7p1	8,7 p1	
openssh-clients	8,7 p1	8,7 p1	
openssh-server	8,7 p1	8,7 p1	
openssl	3.0.8	3.0.8	
openssl-libs	3.0.8	3.0.8	3.0.8
openssl-pkcs11	0.4.12	0.4.12	
os-prober	1,77	1,77	
p11-kit	0,24,1	0,24,1	0,24,1
p11-kit-trust	0,24,1	0,24,1	0,24,1
package-notes-srpm-macros	0.4		
pam	1.5.1	1.5.1	
parted	3.4		
passwd	0,80	0,80	
pciutils	3.7.0	3.7.0	
pciutils-libs	3.7.0	3.7.0	
pcre2	10,40	10,40	10,40
pcre2-syntax	10,40	10,40	10,40

Paket	AMI	Minimal-AMI	Container
perl-Carp	1,50		
perl-Class-Struct	0,66		
perl-constant	1,33		
perl-DynaLoader	1,47		
perl-Encode	3,15		
perl-Errno	1,30		
perl-Exporter	5,74		
perl-Fcntl	1.13		
perl-File-Basename	2,85		
perl-File-Path	2,18		
perl-File-stat	1,09		
perl-File-Temp	0,231,100		
perl-Getopt-Long	2,52		
perl-Getopt-Std	1.12		
perl-HTTP-Tiny	0,078		
perl-if	0,60,800		
perl-interpreter	5,32,1		
perl-IO	1,43		

Paket	AMI	Minimal-AMI	Container
perl-IPC-Open3	1,21		
perl-libs	5,32,1		
perl-MIME-Base64	3,16		
perl-mro	1,23		
perl-overload	1,31		
perl-overloading	0,02		
perl-parent	0,238		
perl-PathTools	3,78		
perl-Pod-Escapes	1,07		
perl-podlators	4,14		
perl-Pod-Perldoc	3,28,01		
perl-Pod-Simple	3,42		
perl-Pod-Usage	2,01		
perl-POSIX	1,94		
perl-Scalar-List-Utils	1,56		
perl-SelectSaver	1.02		
perl-Socket	2,032		

Paket	AMI	Minimal-AMI	Container
perl-srpm-macros	1		
perl-Storable	3,21		
perl-subs	1,03		
perl-Symbol	1,08		
perl-Term-ANSIColor	5,01		
perl-Term-Cap	1,17		
perl-Text-ParseWords	3,30		
perl-Text-Tabs+Wrap	2021,0726		
perl-Time-Local	1,300		
perl-vars	1,05		
pkgconf	1.8.0		
pkgconf-m4	1.8.0		
pkgconf-pkg-config	1.8.0		
policycoreutils	3.4	3.4	
policycoreutils-python-utils	3.4		
popt	1,18	1,18	1,18

Paket	AMI	Minimal-AMI	Container
procps-ng	3.3,17	3.3.17	
protobuf-c	1.4.1		
psacct	6.6.4		
psmisc	23,4	23,4	
publicsuffix-list-dafsa	20240212	20240212	20240212
python3	3,9,16	3.9,16	3.9,16
python3-attrs	20,3,0	20.3,0	
python3-audit	3.0.6	3.0.6	
python3-awscrt	0.19,19	0,19,19	
python3-babel	2.9.1	2.9.1	
python3-cffi	1.14,5	1.14,5	
python3-chardet	4.0.0	4.0.0	
python3-colorama	0.4.4	0.4.4	
python3-configobj	5.0.6	5.0.6	
python3-cryptography	36,1	36,1	
python3-daemon	2.3.0		
python3-dateutil	2.8.1	2.8.1	

Paket	AMI	Minimal-AMI	Container
python3-dbus	1.2.18	1.2.18	
python3-distro	1.5.0	1.5.0	
python3-dnf	4.14.0	4.14.0	4.14.0
python3-dnf-plugins-core	4.3.0	4.3.0	
python3-docutils	0,16	0,16	
python3-gpg	1.15.1	1.15.1	1.15.1
python3-hawkey	0,69,0	0,69,0	0,69,0
python3-idna	(2.10)	(2.10)	
python3-jinja2	2.11,3	2.11.3	
python3-jmespath	0.10.0	0.10.0	
python3-jsonpatch	1,21	1,21	
python3-jsonpointer	2,0	2.0	
python3-jsonschema	3.2.0	3.2.0	
python3-libcomps	0,120	0,120	0,120
python3-libdnf	0,69,0	0,69,0	0,69,0
python3-libs	3,9,16	3.9,16	3.9,16

Paket	AMI	Minimal-AMI	Container
python3-libselinux	3.4	3.4	
python3-libsemanage	3.4	3.4	
python3-libstoragemgmt	1.9.4		
python3-lockfile	0.12.2		
python3-markupsafe	1.1.1	1.1.1	
python3-netifaces	0.10.6	0.10.6	
python3-oauthlib	3.0.2	3.0.2	
python3-pip-wheel	21.3.1	21.3.1	21.3.1
python3-ply	3.11	3,11	
python3-policycoreutils	3.4	3.4	
python3-prettytable	0.7.2	0.7.2	
python3-prompt-toolkit	3.0,24	3,0,24	
python3-pycparser	2,20	2,20	

Paket	AMI	Minimal-AMI	Container
python3-pyrsistent	0,17,3	0,17,3	
python3-pyserial	3.4	3.4	
python3-pysocks	1.7.1	1.7.1	
python3-pytz	2022.7.1	2022,7.1	
python3-pyyaml	5.4.1	5.4.1	
python3-requests	2,25,1	2,25,1	
python3-rpm	4.16.1,3	4.16.1.3	4.16.1.3
python3-ruamel-yaml	0.16.6	0,16,6	
python3-ruamel-yaml-clib	0.1.2	0.1.2	
python3-setools	4.4.1	4.4.1	
python3-setuptools	59,6,0	59,6,0	
python3-setuptools-wheel	59,6,0	59,6,0	59,6,0
python3-six	1.15.0	1.15.0	
python3-systemd	235	235	
python3-urllib3	1,25,10	1,25,10	

Paket	AMI	Minimal-AMI	Container
python3-wcwidth	0,2,5	0,2,5	
python-chevron	0.13.1		
python-srpm-macros	3.9		
quota	4,06		
quota-nls	4,06		
readline	8,1	8.1	8.1
rng-tools	6,14	6,14	
rootfiles	8.1	8.1	
rpcbind	1.2.6		
rpm	4.16.1.3	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3	4.16.1.3
rpm-libs	4.16.1.3	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3	4.16.1.3	
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3	
rpm-sign-libs	4.16.1.3	4.16.1.3	4.16.1.3
rsync	3.2.6		
rust-srpm-macros	21		
sbsigntools	0.9.4	0.9.4	

Paket	AMI	Minimal-AMI	Container
screen	4.8.0		
sed	4.8	4,8	4,8
selinux-policy	37,22	37,22	
selinux-policy-targeted	37,22	37,22	
setup	2,13,7	2.13,7	2.13,7
shadow-utils	4,9 bis 4,9	4,9 bis 4,9	
slang	2.3.2		
sqlite-libs	3,40,0	3,40,0	3,40,0
sssd-client	2.9.4		
sssd-common	2.9.4		
sssd-kcm	2.9.4		
sssd-nfs-idmap	2.9.4		
strace	6.8		
sudo	1.9,15	1.9,15	
sysctl-defaults	1,0	1,0	
sysstat	12,5.6		
systemd	252,16	252,16	
systemd-libs	252,16	252,16	
systemd-networkd	252,16	252,16	

Paket	AMI	Minimal-AMI	Container
systemd-pam	252,16	252,16	
systemd-resolved	252,16	252,16	
systemd-udev	252,16	252,16	
system-release	2023,4,20240513	2023,4.20240513	2023,4.20240513
systemtap-runtime	4.8		
tar	1,34	1,34	
tbb	2020,3		
tcpdump	4,99,1		
tcsh	6,24,07		
time	1.9		
traceroute	2.1.3		
tzdata	2024a	2024a	2024a
unzip	6.0		
update-motd	2.2	2.2	
userspace-rcu	0.12.1	0.12.1	
util-linux	2,37,4	2,37,4	
util-linux-core	2,37,4	2,37,4	
vim-common	9,0,2153		
vim-data	9,0,2153	9,0,2153	

Paket	AMI	Minimal-AMI	Container
vim-enhanced	9,0,2153		
vim-filesystem	9,0,2153		
vim-minimal	9,0,2153	9,0,2153	
wget	1,21,3		
which	2,21	2,21	
words	3.0		
xfsdump	3.1.11		
xfsprogs	5.18.0	5.18,0	
xxd	9,0,2153		
xxhash-libs	0.8.0		
xz	5.2,5	5.2.5	
xz-libs	5.2.5	5.2.5	5.2.5
yum	4.14.0	4.14.0	4.14.0
zip	3.0		
zlib	1.2.11	1.2.11	1.2.11
zram-generator	1.1.2	1.1.2	
zram-generator-defaults	1.1.2	1.1.2	
zstd	1.5.5	1.5.5	

Verwendung von AL2023 in Containern

> **Note**
>
> Weitere Informationen zur Verwendung von AL2023 zum Hosten von containerisierten Workloads auf Amazon ECS finden Sie unter. AL2023 für Amazon ECS-Container-Hosts

Je nach Anwendungsfall gibt es mehrere Möglichkeiten, AL2023 in Containern zu verwenden. Das AL2023-Basiscontainer-Image ist einem Amazon Linux 2-Container-Image und dem minimalen AMI AL2023 am ähnlichsten.

Für fortgeschrittene Benutzer bieten wir ein minimales Container-Image an, das in der Version AL2023.2 eingeführt wurde, zusammen mit einer Dokumentation, in der beschrieben wird, wie Bare-Bone-Container erstellt werden.

AL2023 kann auch zum Hosten von containerisierten Workloads von AL2023-basierten Container-Images oder von Containern, die auf anderen Linux-Versionen basieren verwendet werden. Hierzu können Sie AL2023 für Amazon ECS-Container-Hosts verwenden oder die enthaltenen Container-Runtime-Pakete direkt verwenden. Die `docker`-, `containerd`- und `nerdctl`-Pakete sind für Installationen auf AL2023 verfügbar.

Themen

- Verwenden des AL2023-Basis-Container-Images
- AL2023 Minimales Container-Image
- Erstellung einfacher AL203-Container-Images
- Vergleich der auf Amazon-Linux-2023-Images installierten Pakete
- Vergleich der auf Amazon-Linux-2023–Minimal-AMI und Container-Images installierten Pakete

Verwenden des AL2023-Basis-Container-Images

Das AL203-Container-Image besteht aus denselben Softwarekomponenten, die im AL2023-AMI enthalten sind. Es kann in jeder Umgebung als Basis-Image für Docker-Workloads verwendet werden. Wenn Sie das Amazon-Linux-AMI für Anwendungen in Amazon Elastic Compute Cloud (Amazon EC2) verwenden, können Sie Ihre Anwendungen mit dem Amazon-Linux-Container-Image containerisieren.

Verwenden Sie das Amazon Linux-Container-Image in Ihrer lokalen Entwicklungsumgebung und übertragen Sie dann Ihre AWS Anwendung auf Amazon Elastic Container Service (Amazon ECS). Weitere Informationen finden Sie unter Verwenden von Amazon-ECR-Images mit Amazon ECS im Amazon-Elastic-Container-Registry-Benutzerhandbuch.

Das Amazon-Linux-Container-Image ist unter Amazon ECR Public verfügbar. Sie können Ihr Feedback zu AL2023 über Ihren zuständigen AWS Vertreter abgeben oder indem Sie ein Problem im Amazon-Linux-2023-Repo unter einreichen. GitHub

So rufen Sie das Amazon-Linux-Container-Image von Amazon ECR Public ab

1. Authentifizieren Sie Ihren Docker-Client beim Amazon-Linux-Public-Registry. Authentifizierungs-Token sind 12 Stunden lang gültig. Weitere Informationen finden Sie unter Registry-Authentifizierung im Benutzerhandbuch zum Amazon Elastic Container Registry.

   > **Note**
   >
   > Der get-login-password Befehl wird mit der neuesten Version von Version 2 unterstützt. AWS CLI Weitere Informationen finden Sie unter Installieren der AWS Command Line Interface im AWS Command Line Interface -Benutzerhandbuch.

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

   Die Ausgabe sieht wie folgt aus.

   ```
   Login succeeded
   ```

2. Rufen Sie das Amazon-Linux-Container-Image mit dem docker pull-Befehl ab. Um das Amazon Linux-Container-Image in der Amazon ECR Public Gallery anzuzeigen, siehe Amazon ECR Public Gallery – amazonlinux.

   > **Note**
   >
   > Wenn Sie das AL2023 Docker-Container-Image abrufen, können Sie die Tags in einem der folgenden Formate verwenden:

- Für die neueste Version des AL2023-Container-Images verwenden Sie das `:2023`-Tag.
- Wenn Sie eine bestimmte Version von AL2023 möchten, können Sie das folgende Format verwenden:
  - `:2023.[0-7 release quarter].[release date].[build number]`

In den folgenden Beispielen wird das `:2023`-Tag verwendet, um das neueste verfügbare AL2023-Container-Image abzurufen.

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

3. (Optional) Führen Sie den Container lokal aus.

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

So rufen Sie das AL2023 -Container-Image aus Docker-Hub ab

1. Rufen Sie das AL2023-Container-Image mit dem docker pull-Befehl ab.

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

2. (Optional) Führen Sie den Container lokal aus.

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

> **Note**
>
> Mit dem AL2023-Container-Image kann nur der `dnf`-Paketmanager zur Installation von Softwarepaketen verwendet werden. Das bedeutet, dass es keinen `amazon-linux-extras`- oder gleichwertigen Befehl gibt, der für zusätzliche Software verwendet werden kann.

AL2023 Minimales Container-Image

> **Note**
>
> Die standardmäßigen AL203-Container-Images sind für die meisten Anwendungsfälle geeignet, und die Anpassung an das minimale Container-Image ist wahrscheinlich aufwändiger als die Anpassung an das AL2023-Basiscontainer-Image.

Das in AL2023.2 eingeführte AL2023-Minimal-Container-Image unterscheidet sich vom Basis-Container-Image dadurch, dass es nur die Mindestpakete enthält, die für die Installation anderer Pakete erforderlich sind. Das Minimal-Container-Image ist so konzipiert, dass es sich um eine minimale Anzahl von Paketen handelt, nicht um eine praktische Zusammenstellung von Paketen.

Das AL2023-Minimal-Container-Image basiert auf Softwarekomponenten, die bereits in AL2023 vorhanden sind. Der wesentliche Unterschied beim Minimal-Container-Image besteht darin, den `dnf` Paketmanager `microdnf` zur Verfügung zu stellen, und nicht das Image mit vollem Python Funktionsumfang`dnf`. Dadurch kann das minimale Container-Image kleiner sein, mit dem Nachteil, dass nicht der gesamte Funktionsumfang des `dnf` Paketmanagers zur Verfügung steht, der in den AL2023-AMIs und dem Basis-Container-Image enthalten ist.

Das minimale Container-Image AL2023 bildet die Basis der `provided.al2023` AWS Lambda-Laufzeitumgebung.

Eine ausführliche Liste der Pakete, die im Minimal-Container-Image enthalten sind, finden Sie unter. Vergleich der auf Amazon-Linux-2023-Images installierten Pakete

Größe des Minimal-Container-Image

Da das minimale Container-Image AL2023 weniger Pakete enthält als das AL2023-Basiscontainer-Image, ist es auch deutlich kleiner. In der folgenden Tabelle werden die Container-Image-Optionen aktueller und früherer Versionen von Amazon Linux verglichen.

> **Note**
>
> Die Größe des Image wird unter Amazon Linux in der Amazon ECR Public Gallery angezeigt.

Image	Version	Größe des Image	Hinweis
Amazon Linux 1 (AL1)	2018.03.0.20230918.0	62,3 MB	Nur `x86-64`
Amazon Linux 2	2.0.20230926.0	64,2 MB	`aarch64` ist 1,6 MB größer als `x86-64`
Base-Container-Image für Amazon Linux 2023	2023,2.20231002,0	52,4 MB	
Minimal-Container-Image für Amazon Linux 2023	2023.2.20231002.0-minimal	35,2 MB	

Verwendung des AL2023-Minimal-Container-Image

Das AL2023-Minimal-Container-Image ist verfügbar auf ECR und das `2023-minimal` Tag verweist immer auf das neueste AL2023-basierte Minimal-Container-Image, obwohl das `minimal` Tag auf eine neuere Version von Amazon Linux als AL2023 aktualisiert werden kann.

Sie können diese Tags anhand des folgenden `docker` Beispiels abrufen:

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

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

Das folgende Beispiel zeigt a`Dockerfile`, das das minimale Container-Image verwendet und GCC darüber installiert:

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

Erstellung einfacher AL203-Container-Images

Das AL203-Container-Image besteht aus denselben Softwarekomponenten, die im AL2023-AMI enthalten sind. Es enthält eine Software, die es der Basis-Container-Ebene ermöglicht, sich ähnlich zu verhalten wie die Ausführung auf einer Amazon EC2 EC2-Instance, z. B. dem Paketmanager`dnf`. In diesem Abschnitt wird erklärt, wie Sie einen Container von Grund auf neu erstellen können, der nur die für eine Anwendung erforderlichen Mindestabhängigkeiten enthält.

> **Note**
>
> Die Standard-AL203-Container-Images sind für die meisten Anwendungsfälle geeignet. Mit den Standard-Container-Images ist es einfach, auf Ihrem eigenen Image aufzubauen. Ein bloßes Container-Image macht es schwieriger, auf Ihrem Image aufzubauen.

So erstellen Sie einen Container mit minimalen Abhängigkeiten für eine Anwendung

1. Stellen Sie Ihre Laufzeitabhängigkeiten fest. Diese sind von Ihrer Anwendung abhängig.
2. Konstruieren Sie ein `Dockerfile`/`Containerfile`, das `FROM scratch` aufbaut. Das folgende `Dockerfile`-Beispiel kann verwendet werden, um einen Container zu erstellen, der nur die `bash`-Shell und deren Abhängigkeiten enthält.

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

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

   - Dieses `Dockerfile` funktioniert wie folgt:

1. Starten Sie einen AL2023-Container mit dem Namen `build`. Dieser Container wird für das Bootstrapping des minimalen Containers verwendet. Dieser Container wird selbst nicht bereitgestellt, sondern generiert den Container, der bereitgestellt werden soll.
2. Erstellen des `/sysroot`-Verzeichnisses. In diesem Verzeichnis installiert der `build`-Container die für den minimalen Container benötigten Abhängigkeiten. Im nächsten Schritt wird der `/sysroot`-Pfad so gepackt, dass er zum Stammverzeichnis unseres minimalen Images wird.

   Wir erstellen dann die anderen AL2023-Images, indem wir die `--installroot`-Option für `dnf` verwenden. Dies ist ein Feature von `dnf`, mit dem Installationsprogramme und Tools zur Image-Erstellung arbeiten können.
3. Aufruf von `dnf` um Pakete in `/sysroot` zu installieren.

   Der `rpm -q system-release --qf '%{VERSION}'`-Befehl fragt (`-q`) das `system-release`-Paket ab, wobei das Abfrageformat (`--qf`) festgelegt wird, mit dem die Version des abgefragten Pakets ausgedruckt wird (die `%{VERSION}`-Variable ist die `rpm`-Variable für die `RPM`-Version).

   Durch eine Festlegung des `--releasever`-Arguments von `dnf` auf die `system-release`-Version im `build`-Container, kann dieses `Dockerfile` verwendet werden, um den minimalen Container jedes Mal neu zu erstellen, wenn ein aktualisiertes Container-Basis-Image von Amazon Linux veröffentlicht wird.

   Es ist möglich, das `--releasever` auf eine beliebige Amazon Linux 2023-Version einzustellen, z. B. 2023.4.20240513. Dies würde bedeuten, dass der `build` Container als neueste AL2023-Version ausgeführt wird, der Barebones-Container jedoch unabhängig von der aktuellen AL2023-Version ab 2023.4.20240513 erstellt wird.

   Die `--setopt=install_weak_deps=False`-Konfigurationsoption informiert `dnf`, dass nur erforderliche Abhängigkeiten installiert werden sollen und nicht solche, die empfohlen oder vorgeschlagen werden.
4. Kopieren des installierten Systems in das Stammverzeichnis eines leeren (`FROM scratch`-) Containers.
5. `ENTRYPOINT` auf den gewünschten Binärwert setzen, hier `/bin/bash`.

3. Erstellen Sie ein leeres Verzeichnis und fügen Sie den Inhalt des Beispiels in Schritt 2 zu einer Datei mit dem Namen `Dockerfile` hinzu.

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

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

4. Erstellen Sie den Container, indem Sie den folgenden Befehl ausführen.

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

5. Führen Sie den Container mit dem folgenden Befehl aus, um zu sehen, wie minimal ein Nur-bash-Container ist.

```
$ docker run -it --rm al2023-barebones-bash-example
bash-5.2# rpm
bash: rpm: command not found
bash-5.2# du -sh /usr/
bash: du: command not found
bash-5.2# ls
bash: ls: command not found
bash-5.2# echo /bin/*
/bin/alias /bin/bash /bin/bashbug /bin/bashbug-64 /bin/bg /bin/catchsegv /bin/cd /
bin/command /bin/fc /bin/fg /bin/gencat /bin/getconf /bin/getent /bin/getopts /
bin/hash /bin/iconv /bin/jobs /bin/ld.so /bin/ldd /bin/locale /bin/localedef /
bin/pldd /bin/read /bin/sh /bin/sotruss /bin/sprof /bin/type /bin/tzselect /bin/
ulimit /bin/umask /bin/unalias /bin/wait /bin/zdump
```

Ein praktischeres Beispiel sehen Sie im folgenden Verfahren, in dem ein Container für eine C-Anwendung erstellt wird, die `Hello World!` anzeigt.

1. Erstellen Sie ein leeres Verzeichnis und fügen Sie den C-Quellcode und `Dockerfile` hinzu.

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

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

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

2. Erstellen Sie den Container mithilfe des folgenden Befehls.

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

3. Führen Sie den Container mithilfe des folgenden Befehls aus.

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

```
Hello World!
```

Vergleich der auf Amazon-Linux-2023-Images installierten Pakete

Ein Vergleich der RPMs, die auf dem AL2023-Basiscontainer-Image vorhanden sind, mit den RPMs, die auf dem AL2023-Minimal-Container-Image vorhanden sind.

Paket	Container	Minimaler Container
alternatives	1.15	1.15
amazon-linux-repo-cdn	2023.4.20240513	2023,4.20240513
audit-libs	3.0.6	3.0.6
basesystem	11	11
bash	5.2.15	5.2.15
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023,2,64	2023,2,64
coreutils-single	8,32	8,32
crypto-policies	20220428	20220428
curl-minimal	8,5,0	8.5.0
dnf	4.14.0	
dnf-data	4.14.0	4.14.0
elfutils-default-yama-scope	0.188	
elfutils-libelf	0.188	

Paket	Container	Minimaler Container
elfutils-libs	0.188	
expat	2.5.0	
file-libs	5,39	5,39
filesystem	3,14	3,14
gawk	5.1.0	5.1.0
gdbm-libs	1,19	
glib2	2,74,7	2,74,7
glibc	2,34	2,34
glibc-common	2,34	2,34
glibc-minimal-langpack	2,34	2,34
gmp	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7
gobject-introspection		1,73,0
gpgme	1.15.1	1.15.1
grep	3.8	3.8
json-c	0,14	0,14
keyutils-libs	1.6.3	1.6.3
krb5-libs	1,21	1,21
libacl	2.3.1	2.3.1

Paket	Container	Minimaler Container
libarchive	3.5.3	3.5.3
libassuan	2.5.5	2.5.5
libattr	2.5.1	2.5.1
libblkid	2,37,4	2,37,4
libcap	2,48	2,48
libcap-ng	0.8.2	0.8.2
libcom_err	1,46,5	1,46,5
libcomps	0,120	
libcurl-minimal	8.5.0	8.5.0
libdnf	0,69,0	0,69,0
libffi	3.4.4	3.4.4
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	
libgpg-error	1,42	1,42
libidn2	2.3.2	2.3.2
libmodulemd	2.13.0	2.13.0
libmount	2,37,4	2,37,4
libnghttp2	1,59,0	1,59,0
libpeas		1.32.0

Paket	Container	Minimaler Container
libpsl	0,21,1	0,21,1
librepo	1,14,5	1.14,5
libreport-filesystem	2.15,2	2.15,2
libselinux	3.4	3.4
libsepol	3.4	3.4
libsigsegv	2,13	2,13
libsmartcols	2,37,4	2,37,4
libsolv	0,7.22	0.7.22
libstdc++	11,4,1	11.4.1
libtasn1	4.19.0	4.19,0
libunistring	0.9.10	0.9,10
libuuid	2,37,4	2,37,4
libverto	0.3.2	0.3.2
libxcrypt	4.4.33	
libxml2	2.10,4	2.10.4
libyaml	0.2.5	0,2,5
libzstd	1.5.5	1.5.5
lua-libs	5.4.4	5.4.4
lz4-libs	1.9.4	1.9.4
microdnf		3.8.1

Paket	Container	Minimaler Container
microdnf-dnf		3.8.1
mpfr	4.1.0	4.1.0
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
npth	1,6	1,6
openssl-libs	3.0.8	3.0.8
p11-kit	0,24,1	0,24,1
p11-kit-trust	0,24,1	0,24,1
pcre2	10,40	10,40
pcre2-syntax	10,40	10,40
popt	1,18	1,18
publicsuffix-list-dafsa	20240212	20240212
python3	3,9,16	
python3-dnf	4.14.0	
python3-gpg	1.15.1	
python3-hawkey	0,69,0	
python3-libcomps	0,120	
python3-libdnf	0,69,0	
python3-libs	3,9,16	
python3-pip-wheel	21.3.1	

Paket	Container	Minimaler Container
python3-rpm	4.16.1.3	
python3-setuptools-wheel	59,6,0	
readline	8.1	8.1
rpm	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	
rpm-libs	4.16.1.3	4.16.1.3
rpm-sign-libs	4.16.1.3	
sed	4.8	4,8
setup	2,13,7	2.13,7
sqlite-libs	3,40,0	3,40,0
system-release	2023,4,20240513	2023,4.20240513
tzdata	2024a	
xz-libs	5.2.5	5.2.5
yum	4.14.0	
zlib	1.2.11	1.2.11

Vergleich der auf Amazon-Linux-2023–Minimal-AMI und Container-Images installierten Pakete

Ein Vergleich der auf dem AL2023 Minimial AMI vorhandenen RPMs mit den RPMs auf der AL2023-Basis und den Minimal-Container-Images.

Paket	Minimal-AMI	Container	Minimaler Container
alternatives	1.15	1.15	1.15
amazon-chrony-config	4.3		
amazon-ec2-net-utils	2.4.1		
amazon-linux-repo-cdn		2023.4.20240513	2023,4.20240513
amazon-linux-repo-s3	2023,4.20240513		
amazon-linux-sb-keys	2023,1		
audit	3.0.6		
audit-libs	3.0.6	3.0.6	3.0.6
awscli-2	2,15,30		
basesystem	11	11	11
bash	5.2,15	5.2.15	5.2.15
bzip2-libs	1.0.8	1.0.8	1.0.8
ca-certificates	2023,2,64	2023,2,64	2023,2,64
checkpolicy	3.4		
chrony	4.3		
cloud-init	22,2,2		
cloud-init-cfg-ec2	22.2.2		

Paket	Minimal-AMI	Container	Minimaler Container
cloud-utils-growpart	0,31		
coreutils	8,32		
coreutils-common	8,32		
coreutils-single		8,32	8,32
cpio	2,13		
cracklib	2.9.6		
cracklib-dicts	2.9.6		
crypto-policies	20220428	20220428	20220428
cryptsetup-libs	2.6.1		
curl-minimal	8,5,0	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27		
dbus	1.12,28		
dbus-broker	32		
dbus-common	1.12.28		
dbus-libs	1.12.28		
device-mapper	1,02,185		
device-mapper-libs	1,02,185		
diffutils	3.8		

Paket	Minimal-AMI	Container	Minimaler Container
dnf	4.14.0	4.14.0	
dnf-data	4.14.0	4.14.0	4.14.0
dnf-plugin-release-notification	1.2		
dnf-plugins-core	4.3.0		
dnf-plugin-support-info	1.2		
dracut	055		
dracut-config-ec2	3.0		
dracut-config-generic	055		
e2fsprogs	1,46,5		
e2fsprogs-libs	1,46,5		
ec2-utils	2.2.0		
efi-filesystem	5		
efivar	38		
efivar-libs	38		
elfutils-default-yama-scope	0.188	0.188	

Paket	Minimal-AMI	Container	Minimaler Container
elfutils-libelf	0.188	0.188	
elfutils-libs	0.188	0.188	
expat	2.5.0	2.5.0	
file	5,39		
file-libs	5,39	5,39	5,39
filesystem	3,14	3,14	3,14
findutils	4.8.0		
fuse-libs	2.9.9		
gawk	5.1.0	5.1.0	5.1.0
gdbm-libs	1,19	1,19	
gdisk	1.0.8		
gettext	0,21		
gettext-libs	0,21		
glib2	2,74,7	2,74,7	2,74,7
glibc	2,34	2,34	2,34
glibc-all-langpacks	2,34		
glibc-common	2,34	2,34	2,34
glibc-locale-source	2,34		

Paket	Minimal-AMI	Container	Minimaler Container
glibc-minimal-langpack		2,34	2,34
gmp	6.2.1	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7	2.3.7
gnutls	3.8.0		
gobject-introspection			1,73,0
gpgme	1.15.1	1.15.1	1.15.1
grep	3.8	3.8	3.8
groff-base	1,22,4		
grub2-common	2,06		
grub2-efi-aa64-ec2	2,06 (aarch64)		
grub2-efi-x64-ec2	2,06 (x86_64)		
grub2-pc-modules	2,06		
grub2-tools	2,06		
grub2-tools-minimal	2,06		
grubby	8,40		
gzip	1.12		
hostname	3,23		

Paket	Minimal-AMI	Container	Minimaler Container
hwdata	0,353		
inih	49		
initscripts	10,09		
iproute	5.10.0		
iputils	20210202		
irqbalance	1.9.0		
jansson	2.14		
jitterentropy	3.4.1		
jq	1.7.1		
json-c	0,14	0,14	0,14
kbd	2.4.0		
kbd-misc	2.4.0		
kernel	6.1,90		
kernel-livepatch-repo-s3	2023,4,20240513		
keyutils-libs	1.6.3	1.6.3	1.6.3
kmod	29		
kmod-libs	29		
krb5-libs	1,21	1,21	1,21
less	608		

Paket	Minimal-AMI	Container	Minimaler Container
libacl	2.3.1	2.3.1	2.3.1
libarchive	3,5.3	3.5.3	3.5.3
libargon2	20171227		
libassuan	2,5.5	2.5.5	2.5.5
libattr	2.5.1	2.5.1	2.5.1
libblkid	2,37,4	2,37,4	2,37,4
libcap	2,48	2,48	2,48
libcap-ng	0.8.2	0.8.2	0.8.2
libcbor	0.7.0		
libcom_err	1,46,5	1,46,5	1,46,5
libcomps	0,120	0,120	
libcurl-minimal	8.5.0	8.5.0	8.5.0
libdb	5.3.28		
libdnf	0,69,0	0,69,0	0,69,0
libeconf	0,4,0		
libedit	3.1		
libfdisk	2,37,4		
libffi	3.4.4	3.4.4	3.4.4
libfido2	1.10.0		
libgcc	11.4.1	11.4.1	11.4.1

Paket	Minimal-AMI	Container	Minimaler Container
libgcrypt	1.10.2	1.10.2	1.10.2
libgomp	11.4.1	11.4.1	
libgpg-error	1,42	1,42	1,42
libidn2	2.3.2	2.3.2	2.3.2
libkcapi	1.4.0		
libkcapi-hmaccalc	1.4.0		
libmnl	1.0.4		
libmodulemd	2.13.0	2.13.0	2.13.0
libmount	2,37,4	2,37,4	2,37,4
libnghttp2	1,59,0	1,59,0	1,59,0
libpeas			1.32.0
libpipeline	1.5.3		
libpsl	0,21,1	0,21,1	0,21,1
libpwquality	1.4.4		
librepo	1,14,5	1.14,5	1.14,5
libreport-filesystem	2.15,2	2.15,2	2.15,2
libseccomp	2.5.3		
libselinux	3.4	3.4	3.4

Paket	Minimal-AMI	Container	Minimaler Container
libselinux-utils	3.4		
libsemanage	3.4		
libsepol	3.4	3.4	3.4
libsigsegv	2,13	2,13	2,13
libsmartcols	2,37,4	2,37,4	2,37,4
libsolv	0,7.22	0.7.22	0.7.22
libss	1,46,5		
libstdc++	11.4.1	11.4.1	11.4.1
libtasn1	4.19.0	4.19,0	4.19,0
libtextstyle	0,21		
libunistring	0,9,10	0.9,10	0.9,10
libuser	0,63		
libutempter	1.2.1		
libuuid	2,37,4	2,37,4	2,37,4
libverto	0.3.2	0.3.2	0.3.2
libxcrypt	4.4.33	4.4.33	
libxml2	2.10,4	2.10.4	2.10.4
libyaml	0.2.5	0,2,5	0,2,5
libzstd	1.5.5	1.5.5	1.5.5
logrotate	3.20.1		

Paket	Minimal-AMI	Container	Minimaler Container
lua-libs	5.4.4	5.4.4	5.4.4
lz4-libs	1.9.4	1.9.4	1.9.4
man-db	2.9.3		
microcode_ctl	2.1 (x86_64)		
microdnf			3.8.1
microdnf-dnf			3.8.1
mpfr	4.1.0	4.1.0	4.1.0
ncurses	6.2		
ncurses-base	6.2	6.2	6.2
ncurses-libs	6.2	6.2	6.2
nettle	3.8		
net-tools	2.0		
npth	1,6	1,6	1,6
numactl-libs	2.0.14		
oniguruma	6.9.7.1		
openldap	2.4.57		
openssh	8,7p1		
openssh-clients	8,7 p1		
openssh-server	8,7 p1		
openssl	3.0.8		

Paket	Minimal-AMI	Container	Minimaler Container
openssl-libs	3.0.8	3.0.8	3.0.8
openssl-pkcs11	0.4.12		
os-prober	1,77		
p11-kit	0,24,1	0,24,1	0,24,1
p11-kit-trust	0,24,1	0,24,1	0,24,1
pam	1.5.1		
passwd	0,80		
pciutils	3.7.0		
pciutils-libs	3.7.0		
pcre2	10,40	10,40	10,40
pcre2-syntax	10,40	10,40	10,40
policycoreutils	3.4		
popt	1,18	1,18	1,18
procps-ng	3.3,17		
psmisc	23,4		
publicsuffix-list-dafsa	20240212	20240212	20240212
python3	3,9,16	3.9,16	
python3-attrs	20,3,0		
python3-audit	3.0.6		
python3-awscrt	0.19,19		

Paket	Minimal-AMI	Container	Minimaler Container
python3-babel	2.9.1		
python3-cffi	1.14,5		
python3-chardet	4.0.0		
python3-colorama	0.4.4		
python3-configobj	5.0.6		
python3-cryptography	36,1		
python3-dateutil	2.8.1		
python3-dbus	1.2.18		
python3-distro	1.5.0		
python3-dnf	4.14.0	4.14.0	
python3-dnf-plugins-core	4.3.0		
python3-docutils	0,16		
python3-gpg	1.15.1	1.15.1	
python3-hawkey	0,69,0	0,69,0	
python3-idna	(2.10)		
python3-jinja2	2.11,3		

Paket	Minimal-AMI	Container	Minimaler Container
python3-jmespath	0.10.0		
python3-jsonpatch	1,21		
python3-jsonpointer	2.0		
python3-jsonschema	3.2.0		
python3-libcomps	0,120	0,120	
python3-libdnf	0,69,0	0,69,0	
python3-libs	3,9,16	3.9,16	
python3-libselinux	3.4		
python3-libsemanage	3.4		
python3-markupsafe	1.1.1		
python3-netifaces	0.10.6		
python3-oauthlib	3.0.2		
python3-pip-wheel	21.3.1	21.3.1	
python3-ply	3.11		

Paket	Minimal-AMI	Container	Minimaler Container
python3-policycoreutils	3.4		
python3-prettytable	0.7.2		
python3-prompt-toolkit	3.0,24		
python3-pycparser	2,20		
python3-pyrsistent	0,17,3		
python3-pyserial	3.4		
python3-pysocks	1.7.1		
python3-pytz	2022.7.1		
python3-pyyaml	5.4.1		
python3-requests	2,25,1		
python3-rpm	4.16.1,3	4.16.1.3	
python3-ruamel-yaml	0.16.6		
python3-ruamel-yaml-clib	0.1.2		
python3-setools	4.4.1		

Paket	Minimal-AMI	Container	Minimaler Container
python3-setuptools	59,6,0		
python3-setuptools-wheel	59,6,0	59,6,0	
python3-six	1.15.0		
python3-systemd	235		
python3-urllib3	1,25,10		
python3-wcwidth	0,2,5		
readline	8.1	8.1	8.1
rng-tools	6,14		
rootfiles	8.1		
rpm	4.16.1.3	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3	
rpm-libs	4.16.1.3	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3		
rpm-plugin-systemd-inhibit	4.16.1.3		
rpm-sign-libs	4.16.1.3	4.16.1.3	
sbsigntools	0.9.4		
sed	4.8	4,8	4,8

Paket	Minimal-AMI	Container	Minimaler Container
selinux-policy	37,22		
selinux-policy-targeted	37,22		
setup	2,13,7	2.13,7	2.13,7
shadow-utils	4,9 bis 4,9		
sqlite-libs	3,40,0	3,40,0	3,40,0
sudo	1.9,15		
sysctl-defaults	1,0		
systemd	252,16		
systemd-libs	252,16		
systemd-networkd	252,16		
systemd-pam	252,16		
systemd-resolved	252,16		
systemd-udev	252,16		
system-release	2023,4,20240513	2023,4.20240513	2023,4.20240513
tar	1,34		
tzdata	2024a	2024a	
update-motd	2.2		
userspace-rcu	0.12.1		

Paket	Minimal-AMI	Container	Minimaler Container
util-linux	2,37,4		
util-linux-core	2,37,4		
vim-data	9,0,2153		
vim-minimal	9,0,2153		
which	2,21		
xfsprogs	5,18,0		
xz	5.2.5		
xz-libs	5.2.5	5.2.5	5.2.5
yum	4.14.0	4.14.0	
zlib	1.2.11	1.2.11	1.2.11
zram-generator	1.1.2		
zram-generator-defaults	1.1.2		
zstd	1.5.5		

AL2023 ein AWS Elastic Beanstalk

AWS Elastic Beanstalk ist ein Dienst für die Bereitstellung und Skalierung von Webanwendungen und -diensten. Sie laden Ihren Code hoch und Elastic Beanstalk übernimmt automatisch die Bereitstellung - von der Kapazitätsbereitstellung über Load-Balancing und Auto Scaling bis zur Statusüberwachung der Anwendung. Weitere Informationen finden Sie unter AWS Elastic Beanstalk.

Zum Verwenden von Elastic Beanstalk erstellen Sie eine Anwendung, laden eine Anwendungsversion in Form eines Anwendungs-Quell-Bundles (z. B. eine Java-WAR-Datei) in Elastic Beanstalk hoch und stellen einige Informationen zur Anwendung bereit. Elastic Beanstalk startet automatisch eine Umgebung und erstellt und konfiguriert die AWS Ressourcen, die für

die Ausführung Ihres Codes benötigt werden. Weitere Informationen finden Sie im AWS Elastic Beanstalk -Entwicklerhandbuch.

Elastic-Beanstalk-Linux-Plattformen verwenden Amazon-EC2-Instances und diese Instances führen Amazon Linux aus. Seit dem 4. August 2023 bietet Elastic Beanstalk die folgenden Plattformzweige, die auf Amazon Linux 2023 basieren: Docker, Tomcat, Java SE, Node.js, PHP und Python. Elastic Beanstalk hat derzeit Support für AL2023 für weitere Elastic-Beanstalk-Plattformen in Arbeit.

Eine ausführliche Liste aller unterstützten Elastic-Beanstalk-Plattformen sowie der Plattformen, die auf AL2023 aufbauen, finden Sie im Abschnitt Elastic-Beanstalk-Linux-Plattformen des Entwicklerhandbuchs für Elastic Beanstalk.

Die Release Notes für neue Elastic-Beanstalk-Plattformen und Versionen vorhandener Plattformen finden Sie in den Versionshinweisen für Elastic Beanstalk.

Verwendung von AL2023 in AWS CloudShell

AWS CloudShell ist eine browserbasierte, vorauthentifizierte Shell, die Sie direkt von der aus starten können. AWS Management Console Sie können auf verschiedene Arten CloudShell zu ihr navigieren. AWS Management Console Weitere Informationen finden Sie unter Wie fange ich an mit AWS CloudShell?

AWS CloudShell, das derzeit auf Amazon Linux 2 basiert, wird auf AL2023 migriert. Die Migration auf AL2023 wird AWS-Regionen ab dem 4. Dezember 2023 insgesamt eingeführt. Weitere Informationen zur CloudShell Migration zu AL2023 finden Sie unter AWS CloudShell Migration von Amazon Linux 2 zu Amazon Linux 2023.

Verwendung von AL2023-basierten Amazon ECS-AMIs zum Hosten containerisierter Workloads

> **Note**
>
> Weitere Informationen zur Verwendung von AL2023 in einem Container finden Sie unter. AL2023 in Containern

Amazon Elastic Container Service (Amazon ECS) ist ein vollständig verwalteter Container-Orchestrierungsservice, mit dem Sie containerisierte Anwendungen einfach bereitstellen, verwalten

und skalieren können. Als vollständig verwalteter Service bietet Amazon ECS integrierte Best Practices für AWS Konfiguration und Betrieb. Es ist AWS sowohl in Tools von Drittanbietern als auch in Tools von Drittanbietern wie Amazon Elastic Container Registry (Amazon ECR) und Docker integriert. Diese Integration erleichtert es Teams, sich auf die Erstellung der Anwendungen zu konzentrieren, nicht auf die Umgebung. Sie können Ihre Container-Workloads ohne die Komplexität der Verwaltung einer Steuerebene über AWS -Regionen hinweg in der Cloud und On-Premises ausführen und skalieren.

Sie können containerisierte Workloads auf AL2023 mithilfe des auf Amazon ECS optimierten AMI auf AL2023 hosten. Weitere Informationen finden Sie unter Amazon ECS-optimiertes AMI

Änderungen in AL2023 für Amazon ECS im Vergleich zu AL2

Wie AL2 stellt AL2023 die Basispakete bereit, die für die Ausführung als Amazon ECS-Linux-Instance erforderlich sind. In AL2 waren die `ecs-init` Pakete `containerd`und `docker`, und verfügbar über`amazon-linux-extras`, wohingegen AL2023 diese Pakete in den Core-Repositorys beinhaltet.

Mit der Funktion „Deterministische Upgrades durch versionierte Repositorys“ ist jedes AL203-AMI standardmäßig an eine bestimmte Repository-Version gebunden. Dies gilt auch für das für Amazon ECS optimierte AMI AL2023. Alle Updates für Ihre Umgebung können vor der Bereitstellung sorgfältig verwaltet und getestet werden. Außerdem bieten sie eine einfache Möglichkeit, im Falle eines Problems zum Inhalt eines früheren AMI zurückzukehren. Weitere Informationen über dieses AL2023-Feature finden Sie unter Verwendung deterministischer Upgrades über ein versioniertes Repository auf AL2023.

AL2023 wechselt über die in AL2 unterstützte cgroup v1-Schnittstelle zu cgroup v2. Weitere Informationen finden Sie unter Vereinheitlichte Kontrollgruppenhierarchie (cgroup v2).

> **Note**
>
> AL2023-Versionen vor 2023.2.20230920 (die erste Version von AL2023.2) enthielten einen Fehler bei der Handhabung von Out-of-Memory (OOM) innerhalb einer Cgroup. `systemd` Alle Prozesse in der Cgroup wurden immer beendet, anstatt dass der OOM-Killer einen Prozess nach dem anderen auswählte, was das beabsichtigte Verhalten ist.
> Dies war im Vergleich zum AL2-Verhalten eine Regression und wurde in der Version 2023.2.20230920 von AL2023 behoben.

Der Code zum Erstellen des Amazon ECS-optimierten AMI ist im amazon-ecs-ami GitHub Projekt verfügbar. In den Versionshinweisen wird beschrieben, welche AL2023-Version welcher Amazon ECS AMI-Version zugeordnet ist.

Anpassen des AL2023-basierten, Amazon-ECS-optimierten AMI

> **Important**
>
> Wir empfehlen Ihnen, das für Amazon ECS optimierte AL2023 AMI zu verwenden. Weitere Informationen finden Sie unter Amazon ECS-Optimized AMI im Amazon Elastic Container Service Developer Guide.

Sie können dieselben Build-Skripts verwenden, die Amazon ECS zur Erstellung benutzerdefinierter AMIs verwendet. Weitere Informationen finden Sie unter Amazon ECS-optimiertes Linux-AMI-Build-Skript.

Verwenden von Amazon Elastic File System auf AL2023

Amazon Elastic File System (Amazon EFS) bietet vollständig elastischen Serverless-Dateispeicher, sodass Sie Dateidaten gemeinsam nutzen können, ohne Speicherkapazität und Leistung bereitstellen oder verwalten zu müssen. Amazon EFS ist so konzipiert, dass es bei Bedarf auf Petabytes skaliert werden kann, ohne Anwendungen zu unterbrechen. Es wächst und schrumpft automatisch, wenn Sie Dateien hinzufügen oder entfernen. Da Amazon EFS über eine einfache Webservice-Schnittstelle verfügt, können Sie Dateisysteme schnell und einfach erstellen und konfigurieren. Der Service übernimmt die Verwaltung der gesamten Dateispeicherinfrastruktur für Sie. Auf diese Weise kann der Aufwand der Bereitstellung, des Patchings und der Wartung komplexer Dateisystemkonfigurationen vermieden werden.

Amazon EFS unterstützt das Protokoll Network File System Version 4 (NFSv4.1 und NFSv4.0), so dass Ihre Anwendungen und Tools nahtlos mit Amazon EFS zusammenarbeiten. Mehrere Recheninstanzen, darunter Amazon EC2, Amazon ECS und AWS Lambda, können gleichzeitig auf ein Amazon EFS-Dateisystem zugreifen. Daher kann ein EFS-Dateisystem eine gemeinsame Datenquelle für Workloads und Anwendungen bereitstellen, die auf mehr als einer Recheninstanz oder mehreren Servern ausgeführt werden.

amazon-efs-utils auf AL2023 installieren

Das `amazon-efs-utils` Paket ist in den AL2023-Repositorys verfügbar und kann für den Zugriff auf Amazon EFS-Dateisysteme installiert und verwendet werden.

Installieren Sie das **amazon-efs-utils**-Paket auf AL2023

- Installieren Sie es `amazon-efs-utils` mit dem folgenden Befehl.

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

Ein Amazon-EFS-Dateisystems auf AL2023 mounten

Nach `amazon-efs-utils` der Installation können Sie ein Amazon EFS-Dateisystem auf Ihrer AL2023-Instance mounten.

Ein Amazon-EFS-Dateisystems auf AL2023 mounten

- Verwenden Sie den folgenden Befehl, um mithilfe der Dateisystem-ID zu mounten.

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

Sie können das Dateisystem auch so mounten, dass Daten während der Übertragung mit TLS verschlüsselt werden, oder indem Sie den DNS-Namen oder die Mount-Ziel-IP anstelle der Dateisystem-ID verwenden. Weitere Informationen finden Sie unter Mounting auf Amazon-Linux-Instances mithilfe der EFS-Mountinghilfe.

Verwenden von Amazon EMR, das auf AL2023 basiert

Amazon EMR ist Webservice, der die effiziente Verarbeitung riesiger Datenmengen erleichtert. Dabei kommen Apache Hadoop und von AWS angebotene Services zum Einsatz.

Auf AL2023 basierende Amazon EMR-Versionen

Amazon EMR Version 7.0.0 war die erste Version, die auf AL2023 basiert. Mit dieser Version ist AL2023 das Basisbetriebssystem für Amazon EMR und bietet Amazon EMR alle Vorteile von AL2023. Weitere Informationen finden Sie in den Versionshinweisen zu Amazon EMR 7.0.0.

Auf AL2023 basiertes Amazon EMR auf EKS

Amazon EMR auf EKS 6.13 war die erste Version, in der AL2023 als Option vorgestellt wurde. Mit dieser Version können Sie Spark mit AL2023 als Betriebssystem zusammen mit Java 17 Runtime starten. Weitere Informationen finden Sie in den Versionshinweisen zu Amazon EMR on EKS 6.13 und allen Versionshinweisen zu Amazon EMR on EKS.

Verwendung von AL2023 in AWS Lambda

Mit können Sie Code ausführen AWS Lambda, ohne Server bereitstellen oder verwalten zu müssen. Sie zahlen nur für die genutzte Rechenzeit. Wenn Ihr Code nicht ausgeführt wird, wird auch nichts berechnet. Sie können Code für praktisch jeden Anwendungstyp oder Backend-Service ohne jeden Verwaltungsaufwand ausführen. Sie laden einfach Ihren Code hoch und Lambda kümmert sich darum, dass Ihr Code mit hoher Verfügbarkeit ausgeführt und skaliert wird.

Von AL2023 **`provided.al2023`** verwaltetes Runtime- und Container-Image

Die `provided.al2023` Basislaufzeit basiert auf dem minimalen Container-Image AL2023 und bietet eine auf AL2023 basierende, von Lambda verwaltete Laufzeit und ein Container-Basis-Image. Da die `provided.al2023` Runtime auf dem minimalen Container-Image AL2023 basiert, ist sie mit weniger als 40 MB wesentlich kleiner als die `provided.al2` Runtime mit etwa 109 MB.

Weitere Informationen finden Sie unter Lambda-Laufzeiten und Arbeiten mit Lambda-Container-Images.

AL203-basierte Lambda-Laufzeiten

Zukünftige Versionen verwalteter Sprachlaufzeiten, wie Node.js 20, Python 3.12, Java 21 und .NET 8, basieren auf AL2023 und werden als Basis-Image verwendet, `provided.al2023` wie in der Ankündigung von AL2023-basierten Laufzeiten beschrieben.

AL203-basierte Lambda-Funktionen

- AL2023 Lambda-Funktionen, geschrieben in Go
- AL2023 Lambda-Funktionen, geschrieben in Rust

Weitere Informationen finden Sie unter Lambda-Laufzeiten im AWS Lambda Developer Guide.

Tutorials

Die folgenden Tutorials zeigen Ihnen, wie Sie allgemeine Aufgaben mit Amazon EC2 EC2-Instances ausführen, auf denen Amazon Linux 2023 (AL2023) ausgeführt wird. Video-Tutorials finden Sie unter AWS Lehrvideos und Übungen.

Anweisungen zu AL2 finden Sie unter Tutorials für Amazon EC2 EC2-Instances, auf denen Linux ausgeführt wird, im Amazon EC2 EC2-Benutzerhandbuch.

Tutorials

- Tutorial: Installieren Sie einen LAMP-Server auf AL2023
- Tutorial: SSL/TLS auf AL2023 konfigurieren
- Tutorial: Einen WordPress Blog auf AL2023 hosten

Tutorial: Installieren Sie einen LAMP-Server auf AL2023

Die folgenden Verfahren helfen Ihnen bei der Installation eines Apache-Webservers mit Unterstützung für PHP und MariaDB (ein von der Community entwickelter Fork von MySQL) auf Ihrer AL203-Instance (manchmal auch LAMP-Webserver oder LAMP-Stack genannt). Sie können diesen Server dazu verwenden, eine statische Website zu hosten oder eine dynamische PHP-Anwendung bereitzustellen, die Informationen aus einer Datenbank liest und in diese schreibt.

> **Important**
>
> Diese Verfahren sind für die Verwendung mit AL2023 vorgesehen. Dieses Tutorial funktioniert nicht, wenn Sie versuchen, einen LAMP-Webserver auf einer anderen Verteilung, wie z. B. Ubuntu oder Red Hat Enterprise Linux, einzurichten. Informationen zu Ubuntu finden Sie in der folgenden Dokumentation der Ubuntu-Community: ApacheMySQLPHP. Andere Verteilungen finden Sie in der jeweiligen Dokumentation.

Aufgaben

- Schritt 1: Vorbereiten des LAMP-Servers
- Schritt 2: Testen Ihres Lamp-Servers
- Schritt 3: Sichern des Datenbankservers

- Schritt 4: (Optional) Installieren phpMyAdmin
- Fehlerbehebung
- Verwandte Themen

Schritt 1: Vorbereiten des LAMP-Servers

Voraussetzungen

- In diesem Tutorial wird davon ausgegangen, dass Sie mit AL2023 bereits eine neue Instanz mit einem öffentlichen DNS-Namen gestartet haben, der über das Internet erreichbar ist. Weitere Informationen finden Sie unter AL2023 auf Amazon EC2. Außerdem müssen Sie Ihre Sicherheitsgruppe so konfiguriert haben, dass Verbindungen über SSH (Port 22), HTTP (Port 80) und HTTPS (Port 443) erlaubt sind. Weitere Informationen zu diesen Voraussetzungen finden Sie unter Autorisieren des eingehenden Datenverkehrs für Ihre Linux-Instances im Amazon EC2 EC2-Benutzerhandbuch.
- Mit dem folgenden Verfahren wird die neueste PHP-Version installiert, die auf AL2023 verfügbar ist, derzeit 8.1. Falls Sie andere PHP-Anwendungen als die in diesem Tutorial beschriebenen verwenden möchten, prüfen Sie ihre Kompatibilität mit PHP 8.1.

Vorbereiten des LAMP-Servers

1. Verbinden Sie sich mit der Instance. Weitere Informationen finden Sie unter Verbindung zu AL203-Instances herstellen.
2. Um sicherzustellen, dass alle Ihre Softwarepakete aktuell sind, führen Sie ein schnelles Softwareupdate auf Ihrer Instance aus. Dieser Vorgang kann einige Minuten in Anspruch nehmen, ist aber wichtig, um sicherzustellen, dass Sie über die neuesten Sicherheitsupdates und Fehlerbehebungen verfügen.

   Mit der Option `-y` werden die Updates installiert, ohne um Bestätigung zu bitten. Wenn Sie die Aktualisierungen vor der Installation überprüfen möchten, können Sie diese Option auslassen.

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

3. Installieren Sie die neuesten Versionen des Apache-Webservers und der PHP-Pakete für AL2023.

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

4. Installieren Sie die MariaDB-Softwarepakete. Verwenden Sie den Befehl dnf install, um mehrere Softwarepakete und alle damit verbundenen Abhängigkeiten gleichzeitig zu installieren.

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

   Sie können die aktuellen Versionen dieser Pakete mit dem folgenden Befehl anzeigen:

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

   Beispiel:

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

5. Starten Sie den Apache-Webserver.

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

6. Konfigurieren Sie den Apache-Webserver mit dem Befehl systemctl so, dass er bei jedem Systemstart startet.

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

   Mit folgendem Befehl können Sie prüfen, ob der Befehl httpd ausgeführt wird:

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

7. Fügen Sie eine Sicherheitsregel hinzu, um eingehende HTTP-Verbindungen (Port 80) auf Ihre Instance zuzulassen, wenn Sie dies nicht bereits getan haben. Standardmäßig wurde für Ihre Instance während des Starts eine Sicherheitsgruppe launch-wizard-***N*** erstellt. Wenn Sie keine zusätzlichen Sicherheitsgruppenregeln hinzugefügt haben, enthält diese Gruppe nur eine einzige Regel, um SSH-Verbindungen zuzulassen.

   a. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.

   b. Wählen Sie im linken Navigator die Option Instances und wählen Sie Ihre Instance aus.

   c. Zeigen Sie auf der Registerkarte Sicherheit die Regeln für eingehenden Datenverkehr an. Sie sollten die folgende Regel sehen:

      ```
      Port range   Protocol     Source
      22           tcp          0.0.0.0/0
      ```

      > **Warning**
      >
      > Wenn Sie `0.0.0.0/0` verwenden, können alle IPv4-Adressen über SSH auf Ihre Instance zugreifen. Dies ist zwar für kurze Zeit in einer Testumgebung akzeptabel, aber für Produktionsumgebungen sehr unsicher. Für die Produktion wird nur eine bestimmte IP-Adresse bzw. ein bestimmter Adressbereich für den Zugriff auf Ihre Instance autorisiert.

   d. Wenn es keine eingehende Regel gibt, die HTTP-Verbindungen (Port 80) zulässt, müssen Sie jetzt die Regel hinzufügen. Wählen Sie den Link für die Sicherheitsgruppe aus. Fügen Sie mithilfe der Verfahren unter Autorisieren von eingehendem Datenverkehr für Ihre Linux-

Instances eine neue Sicherheitsregel für eingehenden Datenverkehr mit den folgenden Werten hinzu:

- Typ: HTTP
- Protocol (Protokoll): TCP
- Portbereich: 80
- Quelle: Benutzerdefiniert

8. Testen Sie Ihren Webserver. Geben Sie in einen Web-Browser die öffentliche DNS-Adresse (oder die öffentliche IP-Adresse) Ihrer Instance ein. Wenn in `/var/www/html` keine Inhalte vorhanden sind, sollten Sie die Testseite von Apache aufrufen, auf der die Meldung „Es funktioniert! “ angezeigt wird.

   Sie können den öffentlichen DNS für Ihre Instance über die Amazon-EC2-Konsole abrufen (prüfen Sie die Spalte Public IPv4 DNS; wenn diese Spalte ausgeblendet ist, wählen Sie Preferences (Präferenzen) (das zahnradförmige Symbol) und schalten Sie Public IPv4 DNS ein).

   Stellen Sie sicher, dass die Sicherheitsgruppe für die Instance eine Regel enthält, die HTTP-Datenverkehr auf Port 80 zulässt. Weitere Informationen finden Sie unter Regeln zur Sicherheitsgruppe hinzufügen.

   > **Important**
   >
   > Wenn Sie nicht Amazon Linux verwenden, müssen Sie möglicherweise auch die Firewall Ihrer Instance so konfigurieren, dass diese Verbindungen zugelassen werden. Weitere Informationen zum Konfigurieren der Firewall finden Sie in der Dokumentation für Ihre spezifische Verteilung.

Der Apache-Befehl httpd gilt für Dateien, die in einem Verzeichnis gespeichert sind, das als Apache-Dokumenten-Stammverzeichnis bezeichnet wird. Das Amazon Linux-Apache-Dokumenten-Stammverzeichnis ist `/var/www/html`, das standardmäßig Eigentum des Stammverzeichnisses ist.

Damit das `ec2-user`-Konto Dateien in diesem Verzeichnis bearbeiten kann, müssen Sie die Eigentümerschaft und die Berechtigungen des Verzeichnisses ändern. Es gibt viele Möglichkeiten, um diese Aufgabe zu erfüllen. In diesem Tutorial fügen Sie `ec2-user` zu der `apache`-Gruppe hinzu, um der `apache`-Gruppe das Eigentum an dem `/var/www`-Verzeichnis zu geben und ihr Schreibrechte zuzuweisen.

So richten Sie Dateiberechtigungen ein

1. Fügen Sie Ihren Benutzer (in diesem Fall `ec2-user`) zu der `apache`-Gruppe hinzu.

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

2. Melden Sie sich ab und anschließend wieder an, um die neue Gruppe auszuwählen, und verifizieren Sie dann Ihre Mitgliedschaft.

   a. Melden Sie sich ab (Sie können den Befehl exit verwenden oder das Terminal-Fenster schließen):

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

   b. Ihre Mitgliedschaft in der `apache`-Gruppe zu verifizieren, stellen Sie erneut die Verbindung zu Ihrer Instance her und führen Sie anschließend den folgenden Befehl aus:

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

3. Übertragen Sie die Eigentümerschaft der Datei `/var/www` und ihrer Inhalte auf die `apache`-Gruppe.

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

4. Um Schreibberechtigungen für die Gruppe hinzuzufügen und die Gruppen-ID für zukünftige Unterverzeichnisse einzurichten, ändern Sie die Verzeichnisberechtigungen von `/var/www` und deren Unterverzeichnisse.

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

5. Um Schreibberechtigungen für die Gruppe hinzuzufügen, ändern Sie die Dateiberechtigungen von `/var/www` und deren Unterverzeichnisse rekursiv.

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

Jetzt kann `ec2-user` (und jedes zukünftige Mitglied der `apache`-Gruppe) im Dokumenten-Stammverzeichnis von Apache Dateien hinzufügen, löschen und bearbeiten. Auf diese Weise können Sie Inhalte hinzufügen, beispielsweise eine statische Website oder eine PHP-Anwendung.

So sichern Sie Ihren Webserver (optional)

Ein Webserver, auf dem HTTP ausgeführt wird, bietet keine Transportsicherheit für die gesendeten oder empfangenen Daten. Wenn Sie über einen Webbrowser Verbindung zu einem HTTP-Server aufnehmen, sind die URLs, die Sie besuchen, die Inhalte von Webseiten, die Sie empfangen, und die Inhalte (einschließlich Passwörtern) aller HTML-Formulare, die Sie übermitteln, überall auf dem Netzwerk-Pfad für Lauscher zugänglich. Die beste Methode, Ihren Webserver abzusichern, besteht darin, Unterstützung für HTTPS (HTTP Secure) zu installieren, wodurch Ihre Daten mit der SSL/TLS-Verschlüsselung geschützt werden.

Informationen zur Aktivierung von HTTPS auf Ihrem Server finden Sie unter Tutorial: SSL/TLS auf AL2023 konfigurieren.

Schritt 2: Testen Ihres Lamp-Servers

Wenn Ihr Server installiert ist und läuft und Ihre Dateiberechtigungen korrekt eingestellt sind, müsste für Ihr `ec2-user`-Konto die Erstellung einer PHP-Datei im Verzeichnis `/var/www/html` möglich sein, auf die über das Internet zugegriffen werden kann.

So testen Sie Ihren LAMP-Server

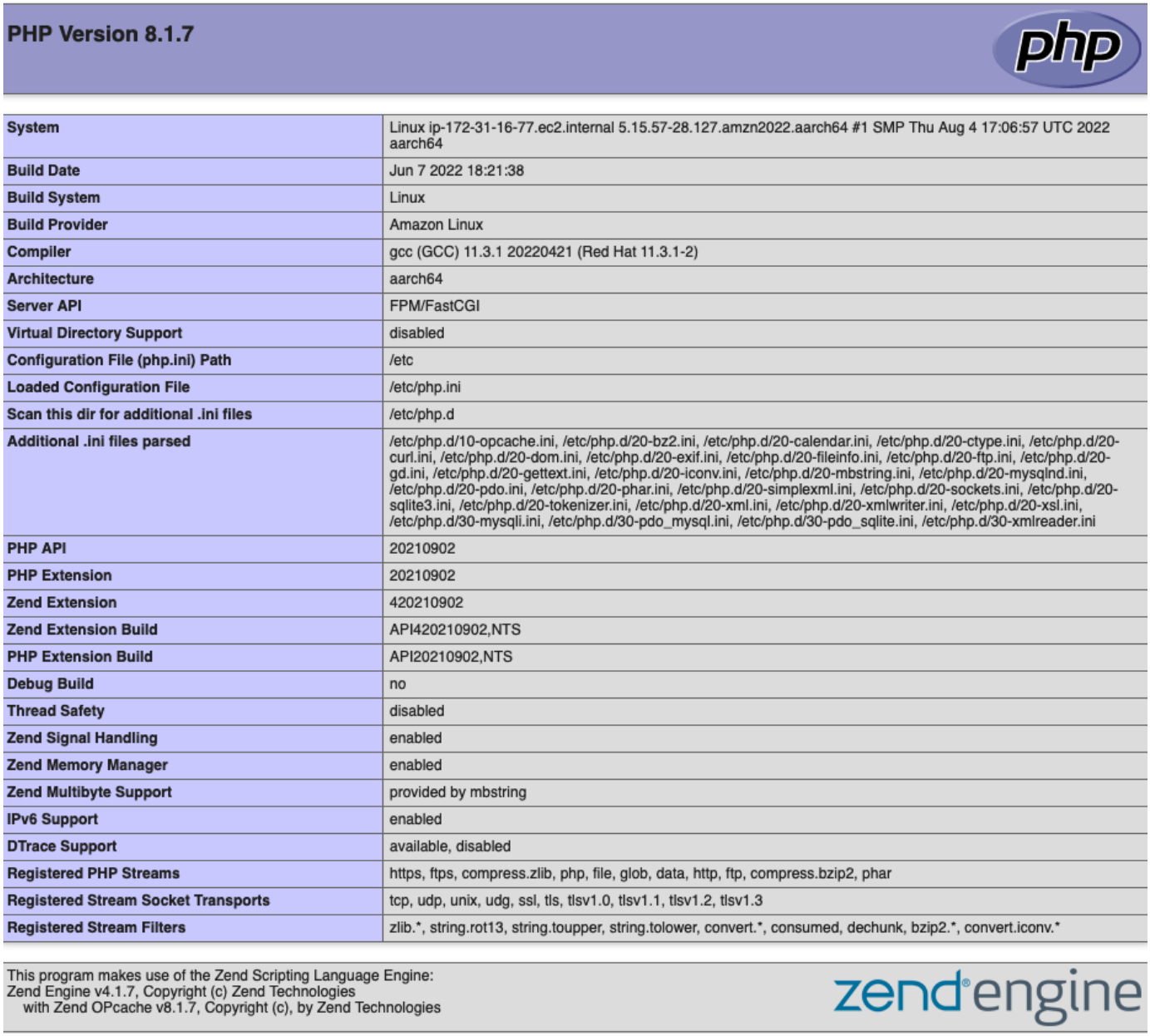
1. Erstellen Sie eine PHP-Datei im Dokumenten-Stammverzeichnis von Apache.

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

   Wenn beim Ausführen dieses Befehls der Fehler „Permission denied“ angezeigt wird, melden Sie sich ab und anschließend wieder an, damit die richtigen Gruppenberechtigungen übernommen werden, die Sie in konfiguriert habe So richten Sie Dateiberechtigungen ein.
2. Geben Sie in einem Webbrowser die URL der Datei ein, die Sie gerade erstellt haben. Diese URL ist die öffentliche DNS-Adresse Ihrer Instance, gefolgt von einem Schrägstrich und dem Dateinamen. Beispiel:

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

   Die PHP-Informationsseite wird angezeigt:



PHP Version 8.1.7

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

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

Wenn diese Seite nicht angezeigt wird, überprüfen Sie, ob die Datei `/var/www/html/phpinfo.php` im vorherigen Schritt ordnungsgemäß angelegt wurde. Mit dem folgenden Befehl können Sie auch überprüfen, ob alle erforderlichen Pakete installiert wurden.

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

Wenn eines der erforderlichen Pakete in Ihrem Ergebnis nicht aufgelistet ist, installieren Sie es mit dem Befehl sudo yum install ***package***.

3. Löschen Sie die Datei `phpinfo.php`. Obwohl sie nützliche Informationen enthalten könnte, sollte sie aus Sicherheitsgründen nicht über das Internet übertragen werden.

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

Sie sollten nun über einen voll funktionsfähigen LAMP-Webserver verfügen. Wenn Sie zum Dokumenten-Stammverzeichnis von Apache unter `/var/www/html` Inhalte hinzufügen, können Sie diese unter der öffentlichen DNS-Adresse für Ihre Instance anzeigen.

Schritt 3: Sichern des Datenbankservers

Die Standardinstallation des MariaDB-Servers verfügt über mehrere Funktionen, die hervorragend zum Testen und für die Entwicklung geeignet sind, aber bei Produktionsservern sollten Sie deaktiviert oder entfernt werden. Mit dem Befehl mysql_secure_installation rufen Sie eine Anleitung dazu auf, wie Sie ein Stammpasswort einrichten und die unsicheren Funktionen aus Ihrer Installation entfernen. Auch wenn Sie nicht vorhaben, den MariaDB-Server zu verwenden, empfehlen wir Ihnen die Durchführung dieses Verfahrens.

Sichern des MariaDB-Servers

1. Starten Sie den MariaDB-Server.

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

2. Führen Sie mysql_secure_installation.

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

   a. Geben Sie das Passwort für das Stammkonto ein, wenn Sie dazu aufgefordert werden.

      i. Geben Sie das aktuelle Stammpasswort ein. Standardmäßig ist für das Stammkonto kein Passwort eingerichtet. Drücken Sie die Eingabetaste.

      ii. Drücken Sie **Y**, um ein Passwort einzurichten, und geben Sie ein sicheres Passwort zweimal ein. Weitere Informationen zum Erstellen eines sicheren Passworts finden Sie unter https://identitysafe.norton.com/password-generator/. Bewahren Sie dieses Passwort an einem sicheren Ort auf.

Die Einrichtung eines Stammpassworts für MariaDB ist nur die grundlegendste Maßnahme, um Ihre Datenbank abzusichern. Wenn Sie eine datenbankgestützte Anwendung aufbauen oder installieren, legen Sie für diese Anwendung normalerweise einen Datenbank-Servicebenutzer an und nutzen das Stammkonto ausschließlich zur Datenbankverwaltung.

b. Geben Sie **Y** ein, um die anonymen Benutzerkonten zu entfernen.

c. Geben Sie **Y** ein, um die Root-Anmeldung per Remote-Zugriff zu deaktivieren.

d. Geben Sie **Y** ein, um die Testdatenbank zu entfernen.

e. Geben Sie **Y** ein, um die Tabellen mit den Berechtigungen neu zu laden. Speichern Sie anschließend Ihre Änderungen.

3. (Optional) Wenn Sie nicht vorhaben, den MariaDB-Server weiter zu verwenden, stoppen Sie ihn. Sie können ihn erneut starten, wenn Sie ihn wieder brauchen.

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

4. (Optional) Wenn Sie wollen, dass der MariaDB-Server bei jedem Systemstart gestartet wird, geben Sie den folgenden Befehl ein.

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

Schritt 4: (Optional) Installieren phpMyAdmin

phpMyAdminist ein webbasiertes Datenbankverwaltungstool, mit dem Sie die MySQL-Datenbanken auf Ihrer EC2-Instance anzeigen und bearbeiten können. Führen Sie die unten genannten Schritte durch, um `phpMyAdmin` auf Ihrer Amazon Linux-Instance zu installieren und zu konfigurieren.

> **Important**
>
> Es ist nicht empfehlenswert, `phpMyAdmin` zum Zugriff auf einen LAMP-Server zu verwenden, falls Sie nicht SSL/TLS in Apache aktiviert haben; andernfalls werden Ihr Datenbankadministrator-Passwort und andere Daten ungesichert im Internet übertragen. Sicherheitsempfehlungen der Entwickler finden Sie unter Sichern Ihrer phpMyAdmin Installation. Allgemeine Informationen zur Sicherung eines Webservers auf einer EC2-Instance finden Sie unter Tutorial: SSL/TLS auf AL2023 konfigurieren.

Um zu installieren phpMyAdmin

1. Installieren Sie die erforderlichen Abhängigkeiten.

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

2. Starten Sie Apache erneut.

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

3. Starten Sie `php-fpm` neu.

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

4. Navigieren Sie zum Stammverzeichnis von Apache unter `/var/www/html`.

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

5. Wählen Sie unter https://www.phpmyadmin.net/downloads ein Quellpaket für die neueste phpMyAdmin Version aus. Um die Datei direkt in Ihre Instance herunterzuladen, kopieren Sie den Link in einen wget-Befehl wie im folgenden Beispiel:

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

6. Erstellen Sie mit dem folgenden Befehl einen `phpMyAdmin`-Ordner und extrahieren Sie das Paket in diesen.

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

7. Löschen Sie den *phpMyAdminTarball latest-all-languages -.tar.gz.*

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

8. (Optional) Wenn der MySQL-Server nicht ausgeführt wird, starten Sie ihn jetzt.

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

9. Geben Sie in einem Webbrowser die URL Ihrer Installation ein. phpMyAdmin Diese URL ist die öffentliche DNS-Adresse (oder die öffentliche IP-Adresse) Ihrer Instance gefolgt von einem Schrägstrich und dem Namen wie im folgenden Beispiel: Beispielsweise:

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

   Sie sollten die phpMyAdmin Anmeldeseite sehen:

10. Melden Sie sich mit dem `root` Benutzernamen und dem MySQL-Root-Passwort, das Sie zuvor erstellt haben, bei Ihrer phpMyAdmin Installation an.

    Ihre Installation muss vor der Inbetriebnahme noch konfiguriert werden. Wir schlagen vor, dass Sie zunächst die Konfigurationsdatei wie folgt manuell erstellen:

a. Um mit einer minimalen Konfigurationsdatei zu beginnen, erstellen Sie mit Ihrem bevorzugten Texteditor eine neue Datei und kopieren Sie dann den Inhalt von `config.sample.inc.php` hinein.

b. Speichern Sie die Datei `config.inc.php` in dem phpMyAdmin Verzeichnis, das enthält`index.php`.

c. Weitere Einstellungen finden Sie in den Anweisungen nach der Dateierstellung im Abschnitt Verwenden des Setup-Skripts der phpMyAdmin Installationsanweisungen.

Informationen zur Verwendung phpMyAdmin finden Sie im phpMyAdmin Benutzerhandbuch.

Fehlerbehebung

In diesem Abschnitt finden Sie Vorschläge zur Lösung von Problemen, die beim Einrichten eines neuen LAMP-Servers auftreten können.

Ich kann zu meinem Server keine Verbindung über einen Webbrowser herstellen

Führen Sie die folgende Prüfungen durch, um zu sehen, ob Ihr Apache-Webserver ausgeführt wird und auf ihn zugegriffen werden kann.

- Wird der Webserver ausgeführt?

  Mit folgendem Befehl können Sie prüfen, ob der Befehl httpd ausgeführt wird:

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

  Wenn der httpd-Prozess nicht ausgeführt wird, wiederholen Sie die unter Vorbereiten des LAMP-Servers beschriebenen Schritte.

- Ist die Firewall richtig konfiguriert?

  Stellen Sie sicher, dass die Sicherheitsgruppe für die Instance eine Regel enthält, die HTTP-Datenverkehr auf Port 80 zulässt. Weitere Informationen finden Sie unter Regeln zur Sicherheitsgruppe hinzufügen.

Ich kann über HTTPS keine Verbindung zu meinem Server herstellen

Führen Sie die folgende Prüfungen durch, um zu sehen, ob Ihr Apache-Webserver konfiguriert ist, HTTPS zu unterstützen.

- Ist der Webserver richtig konfiguriert?

  Nach der Installation von Apache ist der Server für HTTP-Verkehr konfiguriert. Um HTTPS zu unterstützen, aktivieren Sie TLS auf dem Server und installieren Sie ein SSL-Zertifikat. Weitere Informationen finden Sie unter Tutorial: SSL/TLS auf AL2023 konfigurieren.

- Ist die Firewall richtig konfiguriert?

  Stellen Sie sicher, dass die Sicherheitsgruppe für die Instance eine Regel enthält, die HTTPS-Datenverkehr auf Port 443 zulässt. Weitere Informationen finden Sie unter Autorisieren von eingehendem Datenverkehr für Ihre Linux-Instances.

Verwandte Themen

Weitere Informationen zum Übertragen von Dateien auf Ihre Instance oder zum Installieren eines WordPress Blogs auf Ihrem Webserver finden Sie in der folgenden Dokumentation:

- Übertragen Sie Dateien mithilfe von WinSCP auf Ihre Linux-Instance im Amazon EC2 EC2-Benutzerhandbuch.
- Übertragen Sie Dateien mithilfe eines SCP-Clients auf Linux-Instances im Amazon EC2 EC2-Benutzerhandbuch.
- Tutorial: Einen WordPress Blog auf AL2023 hosten

Weitere Informationen über die in diesem Tutorial verwendete(n) Befehle und Software finden Sie auf den folgenden Webseiten:

- Apache-Webserver: http://httpd.apache.org/
- MariaDB-Datenbankserver: https://mariadb.org/
- PHP-Programmiersprache: http://php.net/

Weitere Informationen zum Registrieren eines Domänennamens für Ihren Webserver oder zum Übertragen eines bestehenden Domänennamens auf diesen Host finden Sie unter Erstellen und

Migrieren von Domänen und Subdomänen zu Amazon Route 53 im Entwicklerhandbuch für Amazon Route 53.

Tutorial: SSL/TLS auf AL2023 konfigurieren

Secure Sockets Layer/Transport Layer Security (SSL/TLS) erstellt einen verschlüsselten Kanal zwischen einem Webserver und einem Webclient, der Daten in der Übertragung davor schützt, abgefangen zu werden. In diesem Tutorial wird erklärt, wie Sie manuell Unterstützung für SSL/TLS auf einer EC2-Instance mit AL2023 und Apache-Webserver hinzufügen. In diesem Tutorial wird davon ausgegangen, dass Sie keinen Load Balancer verwenden. Wenn Sie Elastic Load Balancing verwenden, können Sie im Load Balancer SSL-Offload konfigurieren und stattdessen ein Zertifikat aus AWS Certificate Manager verwenden.

Aus historischen Gründen wird die Webverschlüsselung häufig einfach als SSL bezeichnet. Auch wenn Webbrowser SSL weiterhin unterstützen, ist das Nachfolgeprotokoll TLS weniger anfällig für Angriffe. AL2023 deaktiviert standardmäßig die serverseitige Unterstützung für alle Versionen von SSL. Gremien für Sicherheitsstandards erachten TLS 1.0 als unsicher. TLS 1.0 und TLS 1.1 wurden im März 2021 formell veraltet. Dieses Tutorial enthält Empfehlungen, die ausschließlich auf der Aktivierung von TLS 1.2 basieren. TLS 1.3 wurde 2018 fertiggestellt und ist in AL2 verfügbar, sofern die zugrunde liegende TLS-Bibliothek (OpenSSL in diesem Tutorial) unterstützt und aktiviert ist. Kunden müssen spätestens zum 28. Juni 2023 TLS 1.2 oder höher unterstützen. Weitere Informationen zum aktualisierten Verschlüsselungsstandard finden Sie unter RFC 7568 und RFC 8446.

Dieses Tutorial bezieht sich auf TLS als moderne Web-Verschlüsselung.

> **Important**
>
> Diese Verfahren sind für die Verwendung mit AL2023 vorgesehen. Wenn Sie versuchen, eine EC2-Instance mit einer anderen Verteilung einzurichten oder eine Instance mit einer alten Version von Amazon Linux einzurichten, funktionieren möglicherweise einige Verfahren in diesem Tutorial nicht. Für Ubuntu lesen Sie bitte die folgende Community-Dokumentation: Open SSL auf Ubuntu. Informationen zu Red Hat Enterprise Linux finden Sie im Thema Apache-HTTP-Webserver einrichten. Andere Verteilungen finden Sie in der jeweiligen Dokumentation.

> **Note**
>
> Alternativ können Sie AWS Certificate Manager (ACM) für AWS Nitro-Enklaven verwenden. Dabei handelt es sich um eine Enklave-Anwendung, mit der Sie öffentliche und private SSL/TLS-Zertifikate für Ihre Webanwendungen und Server verwenden können, die auf Amazon EC2 EC2-Instances mit Nitro Enclaves ausgeführt werden. AWS Nitro Enclaves ist eine Amazon-EC2-Funktion, die die Erstellung isolierter Rechenumgebungen ermöglicht, um hochsensible Daten wie SSL-/TLS-Zertifikate und private Schlüssel zu schützen und sicher zu verarbeiten.
> ACM for Nitro Enclaves arbeitet mit nginx zusammen, das auf Ihrer Amazon-EC2-Linux-Instance ausgeführt wird, um private Schlüssel zu erstellen, Zertifikate und private Schlüssel zu verteilen und Zertifikatverlängerungen zu verwalten.
> Um ACM for Nitro Enclaves verwenden zu können, müssen Sie eine Enclave-fähige Linux-Instance nutzen.
> Weitere Informationen AWS finden Sie unter Was ist Nitro Enclaves? und AWS Certificate Manager für Nitro Enclaves im Nitro Enclaves-Benutzerhandbuch.AWS

Inhalt

- Voraussetzungen
- Schritt 1: Aktivieren von TLS auf dem Server
- Schritt 2: Abrufen eines CA-signierten Zertifikats
- Schritt 3: Testen und Verstärken der Sicherheitskonfiguration
- Fehlerbehebung

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, führen Sie die folgenden Schritte aus:

- Starten Sie eine EBS-gestützte AL203-Instance. Weitere Informationen finden Sie unter AL2023 auf Amazon EC2.
- Konfigurieren Sie Ihre Sicherheitsgruppen so, dass Ihre Instance Verbindungen auf den folgenden TCP-Ports akzeptieren kann:
  - SSH (Port 22)
  - HTTP (Port 80)

- HTTPS (Port 443)

  Weitere Informationen finden Sie unter Autorisieren von eingehendem Datenverkehr für Ihre Linux-Instances im Amazon EC2 EC2-Benutzerhandbuch.

- Installieren Sie den Apache-Webserver. step-by-step Anweisungen finden Sie unter. Tutorial: Installieren Sie einen LAMP-Server auf AL2023 Es werden nur das httpd-Paket und die zugehörigen Abhängigkeiten benötigt, sodass die Anleitungen mit PHP und MariaDB ignoriert werden können.
- Zum Identifizieren und Authentifizieren von Websites verwendet die Public Key-Infrastruktur (PKI) TLS das Domain Name System (DNS). Wenn Sie Ihre EC2-Instance zum Hosten einer öffentlichen Website verwenden möchten, müssen Sie einen Domain-Namen für Ihren Webserver registrieren oder einen vorhandenen Domain-Namen an Ihren Amazon-EC2-Host übertragen. Dafür sind zahlreiche Drittanbieterservices für die Domain-Registrierung und das DNS-Hosting verfügbar. Oder Sie verwenden Amazon Route 53.

Schritt 1: Aktivieren von TLS auf dem Server

Dieses Verfahren führt Sie durch den Prozess der Einrichtung von TLS auf AL2023 mit einem selbstsignierten digitalen Zertifikat.

> **Note**
>
> Ein selbstsigniertes Zertifikat kann zu Testzwecken, jedoch nicht für die Produktion verwendet werden. Wenn Sie Ihr selbstsigniertes Zertifikat im Internet bereitstellen, werden den Besuchern Ihrer Website Sicherheitswarnungen angezeigt.

So aktivieren Sie TLS auf einem Server

1. Verbinden Sie sich mit der Instance und stellen Sie sicher, dass Apache ausgeführt wird. Weitere Informationen finden Sie unter Verbindung zu AL203-Instances herstellen.

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

   Wenn der zurückgegebene Wert nicht „enabled“ (aktiviert) ist, starten Sie Apache und richten es so ein, dass es bei jedem Neustart des Systems gestartet wird.

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

2. Um sicherzustellen, dass alle Ihre Softwarepakete aktuell sind, führen Sie ein schnelles Softwareupdate auf Ihrer Instance aus. Dieser Vorgang kann einige Minuten dauern. Es ist jedoch wichtig, sicherzustellen, dass Sie über die aktuellen Sicherheitsaktualisierungen und Fehlerbehebungen verfügen.

   > **Note**
   >
   > Mit der Option `-y` werden die Updates installiert, ohne um Bestätigung zu bitten. Wenn Sie die Aktualisierungen vor der Installation überprüfen möchten, können Sie diese Option auslassen.

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

3. Nachdem Sie den folgenden Befehl eingegeben haben, werden Sie zu einer Aufforderung weitergeleitet, in der Sie Informationen zu Ihrer Website eingeben können.

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

   Dadurch wird eine neue Datei `apache-selfsigned.crt` im Verzeichnis `/etc/pki/tls/certs/` erstellt. Der angegebene Dateiname entspricht dem Standard, der in der SSLCertificateFile-Direktive in `/etc/httpd/conf.d/ssl.conf` zugewiesen ist.

   Ihre Instance verfügt nun über die folgenden Dateien, mit denen Sie Ihren sicheren Server konfigurieren und ein Zertifikat zum Testen erstellen:

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

     Die Konfigurationsdatei für mod_ssl. Diese enthält Richtlinien, die Apache mitteilen, wo Verschlüsselungsschlüssel und Zertifikate, die zu genehmigenden TLS-Protokollversionen und die zu akzeptierenden Verschlüsselungschiffren gefunden werden können. Dies wird Ihre lokale Zertifikatdatei sein:

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

Die Datei enthält sowohl ein selbstsigniertes Zertifikat als auch den privaten Schlüssel des Zertifikats. Für Apache müssen das Zertifikat und der Schlüssel im PEM-Format sein. Diese bestehen aus Base64-kodierten ASCII-Zeichen, die durch „BEGIN” und „END”-Zeilen eingerahmt werden, wie im folgendem, verkürzten Beispiel dargestellt.

```
-----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-----
```

Die Dateinamen und Erweiterungen dienen der Einfachheit und haben keinerlei Auswirkungen auf die Funktion. Sie können beispielsweise ein Zertifikat mit `cert.crt`, `cert.pem` oder einem beliebigen anderen Dateinamen benennen, solange die zugehörige Richtlinie in der Datei `ssl.conf` denselben Namen verwendet.

> **Note**
>
> Wenn Sie die TLS-Standarddateien mit Ihren eigenen benutzerdefinierten Dateien ersetzen, müssen diese das PEM-Format aufweisen.

4. Starten Sie Apache erneut.

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

> **Note**
>
> Stellen Sie wie zuvor beschrieben sicher, dass auf den TCP-Port 443 über Ihre EC2-Instance zugegriffen werden kann.

5. Ihr Apache-Webserver sollte jetzt HTTPS (sicheres HTTP) über Port 443 unterstützen. Dies können Sie testen, indem Sie die IP-Adresse oder den vollständig qualifizierten Domain-Namen Ihrer EC2-Instance mit dem Präfix **`https://`** in einer Browser-URL-Leiste eingeben.

   Da Sie eine Verbindung mit einer Website mit einem selbstsignierten, nicht vertrauenswürdigen Host-Zertifikat herstellen, zeigt Ihr Browser möglicherweise eine Reihe von Sicherheitswarnungen an. Setzen Sie die Warnmeldungen außer Kraft und fahren Sie mit der Website fort.

   Wenn die Apache-Standardtestseite geöffnet wird, bedeutet dies, dass Sie TLS erfolgreich auf Ihrem Server konfiguriert haben. Alle Daten, die zwischen dem Browser und dem Server übertragen werden, sind nun verschlüsselt.

   > **Note**
   >
   > Damit den Besuchern keine Warnbildschirme angezeigt werden, müssen Sie ein vertrauenswürdiges, CA-signiertes Zertifikat abrufen, das nicht nur verschlüsselt, sondern Sie auch öffentlich als den Besitzer der Website authentifiziert.

Schritt 2: Abrufen eines CA-signierten Zertifikats

Sie können das folgende Verfahren verwenden, um ein CA-signiertes Zertifikat zu erhalten:

- Erzeugen Sie aus dem privaten Schlüssel eine Zertifikatssignierungsanforderung (Certificate Signing Request, CSR)
- Senden Sie die CSR an eine Zertifizierungsstelle (CA)
- Sie erhalten ein signiertes Host-Zertifikat

- Konfigurieren Sie Apache, um das Zertifikat zu verwenden

Ein selbstsigniertes TLS-X.509-Host-Zertifikat ist kryptologisch mit einem CA-signierten Zertifikat identisch. Der Unterschied liegt im sozialen, nicht im mathematischen Bereich. Eine CA validiert zumindest den Besitzer einer Domain, bevor ein Zertifikat für einen Antragsteller ausgegeben wird. Jeder Webbrowser enthält eine Liste von CAs, die der Browseranbieter dafür als vertrauenswürdig erachtet. Ein X.509-Zertifikat besteht hauptsächlich aus einem öffentlichen Schlüssel, der Ihrem privaten Serverschlüssel entspricht, sowie einer Signatur durch die CA, die kryptografisch an den öffentlichen Schlüssel gebunden ist. Wenn ein Browser eine Verbindung mit einem Webserver über HTTPS herstellt, stellt der Server ein Zertifikat für den Browser bereit, das anhand der Liste der vertrauenswürdigen CAs überprüft wird. Wenn sich der Aussteller auf der Liste befindet oder über eine Vertrauenskette aus anderen vertrauenswürdigen Ausstellern zugänglich ist, handelt der Browser einen schnellen verschlüsselten Datenkanal mit dem Server aus und lädt die Seite.

Im Allgemeinen sind Zertifikate aufgrund der Arbeit im Zusammenhang mit der Validierung der Anforderungen kostenpflichtig, deshalb lohnt es sich, die Angebote zu vergleichen. Einige CAs bieten grundlegende Zertifikate kostenlos an. Die namhafteste dieser CAs ist das Let's Encrypt-Projekt, das auch die Automatisierung des Prozesses zur Erstellung und Verlängerung von Zertifikaten unterstützt. Weitere Informationen zur Verwendung eines Let's Encrypt-Zertifikats finden Sie unter Get Certbot.

Wenn Sie beabsichtigen, kommerzielle Dienstleistungen anzubieten, ist AWS Certificate Manager eine gute Option.

Dem Host-Zertifikat liegt der Schlüssel zugrunde. Seit 2019 empfehlen Regierungs- und Branchengruppen eine Schlüssel(-Modul)-Mindestgröße von 2048 Bits für RSA-Schlüssel, die Dokumente bis 2030 schützen sollen. Die von OpenSSL in AL2023 generierte Standardmodulgröße beträgt 2048 Bit, was für die Verwendung in einem CA-signierten Zertifikat geeignet ist. Im folgenden Verfahren ist ein optionaler Schritt für diejenigen vorgesehen, die einen benutzerdefinierten Schlüssel verwenden möchten, z.B. einen mit einem größeren Modul oder mit einem anderen Verschlüsselungsalgorithmus.

> **Important**
>
> Diese Anweisungen zum Erwerb eines CA-signierten Host-Zertifikats funktioniert nur, wenn Sie eine registrierte und gehostete DNS-Domain besitzen.

So rufen Sie ein CA-signierten Zertifikat ab

1. Verbinden Sie sich mit der Instance und navigieren Sie zu /etc/pki/tls/private/. Dies ist das Verzeichnis, in dem Sie den privaten Schlüssel des Servers für TLS speichern. Wenn Sie lieber Ihren vorhandenen Host-Schlüssel zum Generieren der CSR verwenden möchten, fahren Sie mit Schritt 3 fort. Weitere Informationen zum Herstellen einer Verbindung mit Ihrer Instance finden Sie unter Verbindung zu AL203-Instances herstellen
2. (Optional) Generieren Sie einen neuen privaten Schlüssel. Hier sind einige Beispiele für Schlüsselkonfigurationen. Jeder der resultierenden Schlüssel funktioniert mit Ihrem Webserver, aber sie unterscheiden sich durch den Grad und die Art der Sicherheit, die sie implementieren.

   - Beispiel 1: Erstellen Sie einen Standard-RSA-Hostschlüssel. Bei der erstellten Datei, **`custom.key`**, handelt es sich um einen privaten 2048-Bit-RSA-Schlüssel.

     ```
     [ec2-user ~]$ sudo openssl genrsa -out custom.key
     ```

   - Beispiel 2: Erstellen Sie einen stärkeren RSA-Schlüssel mit einem größeren Modul. Bei der erstellten Datei, **`custom.key`**, handelt es sich um einen privaten 4096-Bit-RSA-Schlüssel.

     ```
     [ec2-user ~]$ sudo openssl genrsa -out custom.key 4096
     ```

   - Beispiel 3: Erstellen Sie einen 4096-Bit-verschlüsselten RSA-Schlüssel mit Passwortschutz. Die resultierende Datei, **`custom.key`**, ist ein privater 4096-Bit-RSA-Schlüssel, der mit der AES-128-Verschlüsselung verschlüsselt ist.

     > **Important**
     >
     > Die Verschlüsselung des Schlüssels bietet höhere Sicherheit. Da für einen verschlüsselten Schlüssel ein Passwort erforderlich ist, können von diesem abhängige Services jedoch nicht automatisch gestartet werden. Jedes Mal, wenn Sie diesen Schlüssel verwenden, müssen Sie das Passwort (im vorhergehenden Beispiel „abcde12345“) über eine SSH-Verbindung bereitstellen.

     ```
     [ec2-user ~]$ sudo openssl genrsa -aes128 -passout pass:abcde12345 -out
      custom.key 4096
     ```

- Beispiel 4: Erstellen Sie einen Schlüssel mit einer Nicht-RSA-Verschlüsselung. Die RSA-Kryptografie kann aufgrund der Größe ihrer öffentlichen Schlüssel, die auf dem Produkt aus zwei großen Primzahlen basieren, relativ langsam sein. Es ist jedoch möglich, Schlüssel für TLS zu erstellen, die andere Verschlüsselungschiffren als RSA verwenden. Schlüssel, die auf der Mathematik von Ellipsenkurven basieren, sind kleiner und bieten eine schnellere Rechenleistung bei der Bereitstellung einer gleichwertigen Sicherheitsebene.

  ```
  [ec2-user ~]$ sudo openssl ecparam -name prime256v1 -out custom.key -genkey
  ```

  Das Ergebnis ist ein privater Ellipsenkurvenschlüssel mit 256-Bit, der prime256v1 verwendet, einer „benannten Kurve“, die OpenSSL unterstützt. Die kryptografische Stärke ist hierbei laut NIST etwas höher als bei einem 2048-Bit-RSA-Schlüssel.

  > **Note**
  >
  > Nicht alle Zertifizierungsstellen bieten dieselbe Unterstützung für elliptic-curve-based Schlüssel wie für RSA-Schlüssel.

Stellen Sie sicher, dass der neue private Schlüssel stark einschränkende Eigentümerschaft und Berechtigungen aufweist (Eigentümer=root, Gruppe=root, Lesen/Schreiben nur für Eigentümer). Führen Sie die Befehle wie im folgenden Beispiel veranschaulicht aus.

```
[ec2-user ~]$ sudo chown root:root custom.key
[ec2-user ~]$ sudo chmod 600 custom.key
[ec2-user ~]$ ls -al custom.key
```

Die vorhergehenden Befehle erzeugen das folgende Ergebnis.

```
-rw------- root root custom.key
```

Wenn Sie einen zufriedenstellenden Schlüssel erstellt und konfiguriert haben, können Sie eine CSR erstellen.

3. Erstellen Sie eine CSR mit Ihrem bevorzugten Schlüssel. Im folgenden Beispiel wird verwende **`custom.key`**.

   ```
   [ec2-user ~]$ sudo openssl req -new -key custom.key -out csr.pem
   ```

OpenSSL öffnet einen Dialog und fordert Sie auf, die in der folgenden Tabelle aufgeführten Informationen einzugeben. Alle Felder außer Common Name (Allgemeiner Name) sind bei einem grundlegenden, Domain-validierten Host-Zertifikat optional.

Name	Beschreibung	Beispiel
Ländername	Die zweistellige ISO-Abkürzung für Ihr Land	US (=United States, Vereinigte Staaten)
State or Province Name	Der Name des Bundesstaats oder der Provinz, in dem bzw. der sich Ihre Organisation befindet. Dieser Name darf nicht abgekürzt werden.	Washington
Locality Name	Der Standort Ihrer Organisation, wie beispielsweise eine Stadt.	Seattle
Name der Organisation	Der vollständige, offizielle Name Ihrer Organisation. Kürzen Sie den Namen Ihrer Organisation nicht ab.	Beispielunternehmen
Organizational Unit Name	Zusätzliche Informationen zu Ihrer Organisation, sofern vorhanden.	Beispielabteilung
Common Name	Dieser Wert muss genau der Webadresse entsprechen, die Ihre Benutzer in einen Browser eingeben sollen. Dies ist in der Regel ein Domain-Name mit einem vorangestellten Hostnamen oder Alias in der Form **`www.example.com`** . Bei Tests mit einem selbstsignierten Zertifikat und ohne DNS-Auflösung kann dieser allgemeine Name nur aus dem Hostnamen bestehen. CAs bieten darüber hinaus teurere Zertifikate an, die Platzhalternamen wie beispielsweise akzeptiere **`*.example.com`** .	www.example.com

Name	Beschreibung	Beispiel
Email Address	Die E-Mail-Adresse des Serveradministrators.	someone@example.com

Zuletzt fordert OpenSSL Sie zur Eingabe eines optionalen Challenge-Passworts auf. Dieses Passwort gilt nur für die CSR und für Transaktionen zwischen Ihnen und Ihrer CA. Befolgen Sie daher die Empfehlungen Ihrer CA diesbezüglich und in Bezug auf das andere optionale Feld, den optionalen Unternehmensnamen. Das CSR-Challenge-Passwort wirkt sich nicht auf den Serverbetrieb aus.

Die erstellte Datei **`csr.pem`** enthält Ihren öffentlichen Schlüssel, die digitale Signatur Ihres öffentlichen Schlüssels und die von Ihnen eingegebenen Metadaten.

4. Übermitteln Sie die CSR an eine CA. Dies besteht in der Regel daraus, Ihre CSR-Datei in einem Texteditor zu öffnen und den Inhalt in ein Webformular zu kopieren. Zu diesem Zeitpunkt werden Sie möglicherweise aufgefordert, einen oder mehrere alternative „Subject”-Namen (SANs) bereitzustellen, die auf dem Zertifikat angegeben werden sollen. Wenn **`www.example.com`** der allgemeine Name ist, wäre **`example.com`** ein guter SAN und umgekehrt. Ein Besucher Ihrer Website, der einen dieser Namen eingibt, wird eine fehlerfreie Verbindung sehen. Wenn dies auf Ihrem CA-Webformular möglich ist, geben Sie den allgemeinen Namen in der Liste der SANs an. Einige CAs fügen diesen automatisch ein.

   Nachdem Ihre Anfrage genehmigt wurde, erhalten Sie ein neues, von der CA unterzeichnetes Host-Zertifikat. Möglicherweise werden Sie auch dazu aufgefordert, eine Zwischenzertifikatsdatei herunterzuladen, die zusätzliche Zertifikate enthält, welche zum Fertigstellen der Vertrauenskette der CA benötigt werden.

   > **Note**
   >
   > Ihre CA kann Ihnen Dateien in verschiedenen Formaten für verschiedene Zwecke zusenden. Für dieses Tutorial sollten Sie nur eine Zertifikatsdatei im PEM-Format verwenden, die in der Regel (aber nicht immer) mit der Dateierweiterung `.pem` oder `.crt` gekennzeichnet ist. Wenn Sie sich nicht sicher sind, welche Datei Sie verwenden sollen, öffnen Sie die Dateien mit einem Texteditor und suchen Sie die Datei, die einen oder mehrere Blöcke enthält, die mit der folgenden Zeile beginnen.

```
- - - - -BEGIN CERTIFICATE - - - - -
```

Die Datei sollte darüber hinaus mit der folgenden Zeile enden.

```
- - - -END CERTIFICATE - - - - -
```

Sie können die Datei auch in der Befehlszeile testen, wie im Folgenden gezeigt.

```
[ec2-user certs]$ openssl x509 -in certificate.crt -text
```

Vergewissern Sie sich, dass diese Zeilen in der Datei erscheinen. Verwenden Sie keine Dateien, die mit `.p7b`, `.p7c` oder ähnlichen Dateiendungen enden.

5. Platzieren Sie ein neues CA-signiertes Zertifikat und alle Zwischenzertifikate im `/etc/pki/tls/certs`-Verzeichnis.

> **Note**
>
> Es gibt mehrere Möglichkeiten, für den Upload Ihres neuen Zertifikat in Ihre EC2-Instance. Der einfachste und informativste Weg ist jedoch, einen Texteditor (vi, nano oder notepad usw.) sowohl auf Ihrem lokalen Computer als auch auf Ihrer Instance zu öffnen, und dann den Dateiinhalt zwischen ihnen zu kopieren und einzufügen. Sie benötigen Root [sudo]-Berechtigungen, wenn Sie diese Operationen auf der EC2-Instance ausführen. Auf diese Weise können Sie sofort erkennen, ob es Probleme mit Berechtigungen oder mit dem Pfad gibt. Achten Sie jedoch darauf, beim Kopieren der Inhalte keine zusätzlichen Zeilen einzufügen und die Inhalte nicht zu ändern.

Überprüfen Sie innerhalb des `/etc/pki/tls/certs` Verzeichnisses, ob die Einstellungen für Dateibesitz, Gruppen und Berechtigungen den äußerst restriktiven AL203-Standardeinstellungen entsprechen (owner=root, group=root, read/write only for owner). Das folgende Beispiel zeigt die zu verwendenden Befehle.

```
[ec2-user certs]$ sudo chown root:root custom.crt
[ec2-user certs]$ sudo chmod 600 custom.crt
[ec2-user certs]$ ls -al custom.crt
```

Diese Befehle sollten das folgende Ergebnis hervorrufen.

```
-rw------- root root custom.crt
```

Die Berechtigungen für die Zwischenzertifikatsdatei sind weniger strikt (Eigentümer=root, Gruppe=root, Eigentümer kann schreiben, Gruppe kann lesen, die restliche Welt kann lesen). Das folgende Beispiel zeigt die zu verwendenden Befehle.

```
[ec2-user certs]$ sudo chown root:root intermediate.crt
[ec2-user certs]$ sudo chmod 644 intermediate.crt
[ec2-user certs]$ ls -al intermediate.crt
```

Diese Befehle sollten das folgende Ergebnis hervorrufen.

```
-rw-r--r-- root root intermediate.crt
```

6. Legen Sie den privaten Schlüssel, den Sie zum Erstellen der CSR verwendet haben, in das Verzeichnis `/etc/pki/tls/private/`.

> **Note**
>
> Es gibt mehrere Möglichkeiten für den Upload Ihrer benutzerdefinierten Schlüssel in Ihre EC2-Instance. Der einfachste und informativste Weg ist jedoch, einen Texteditor (vi, nano oder notepad usw.) sowohl auf Ihrem lokalen Computer als auch auf Ihrer Instance zu öffnen, und dann den Dateiinhalt zwischen ihnen zu kopieren und einzufügen. Sie benötigen Root [sudo]-Berechtigungen, wenn Sie diese Operationen auf der EC2-Instance ausführen. Auf diese Weise können Sie sofort erkennen, ob es Probleme mit Berechtigungen oder mit dem Pfad gibt. Achten Sie jedoch darauf, beim Kopieren der Inhalte keine zusätzlichen Zeilen einzufügen und die Inhalte nicht zu ändern.

Verwenden Sie innerhalb des `/etc/pki/tls/private` Verzeichnisses die folgenden Befehle, um zu überprüfen, ob die Einstellungen für Dateibesitz, Gruppen und Berechtigungen den äußerst restriktiven AL203-Standardeinstellungen entsprechen (owner=root, group=root, read/write only for owner).

```
[ec2-user private]$ sudo chown root:root custom.key
```

```
[ec2-user private]$ sudo chmod 600 custom.key
[ec2-user private]$ ls -al custom.key
```

Diese Befehle sollten das folgende Ergebnis hervorrufen.

```
-rw------- root root custom.key
```

7. Bearbeiten Sie die Datei `/etc/httpd/conf.d/ssl.conf` so, dass sie Ihr neues Zertifikat und Ihre Schlüsseldateien widerspiegelt.

   a. Geben Sie den Pfad und Dateinamen des CA-signierten Host-Zertifikats im `SSLCertificateFile`-Verzeichnis von Apache an.

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

   b. Wenn Sie eine Zwischenzertifikatsdatei erhalten haben (`intermediate.crt` in diesem Beispiel), stellen Sie den entsprechenden Pfad und Dateinamen über das `SSLCACertificateFile`-Verzeichnis in Apache bereit:

   ```
   SSLCACertificateFile /etc/pki/tls/certs/intermediate.crt
   ```

   > **Note**
   >
   > Einige CAs kombinieren das Host-Zertifikat und die Zwischenzertifikate in einer einzelnen Datei, wodurch das `SSLCACertificateFile`-Verzeichnis überflüssig wird. Informieren Sie sich in den von Ihrer CA bereitgestellten Anweisungen.

   c. Geben Sie den Pfad und Dateinamen des privaten Schlüssels (in diesem Beispiel `custom.key`) in der `SSLCertificateKeyFile`-Direktive von Apache an:

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

8. Speichern Sie `/etc/httpd/conf.d/ssl.conf` und starten Sie Apache erneut.

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

9. Testen Sie Ihren Server, indem Sie Ihren Domain-Namen in eine Browser-URL-Leiste mit dem Präfix `https://` eingeben. Ihr Browser sollte die Testseite über HTTPS laden, ohne Fehler zu erzeugen.

Schritt 3: Testen und Verstärken der Sicherheitskonfiguration

Wenn Ihre TLS betriebsbereit und öffentlich zugänglich ist, sollten Sie testen, wie sicher sie wirklich ist. Dies ist ganz einfach möglich mithilfe von Online-Services wie beispielsweise Qualys SSL Labs, der eine kostenlose und gründliche Analyse Ihrer Sicherheitseinrichtung durchführt. Basierend auf den Ergebnissen entscheiden Sie sich möglicherweise dafür, die Standard-Sicherheitskonfiguration zu verstärken, indem Sie kontrollieren, welche Protokolle akzeptiert werden sollen, welche Chiffren Sie bevorzugen und welche ausgeschlossen werden soll. Um weitere Informationen zu erhalten, sehen Sie sich an, wie Qualys seine Skalen gestaltet.

> **Important**
>
> Reale Tests sind außerordentlich wichtig für die Sicherheit Ihres Servers. Kleine Konfigurationsfehler führen möglicherweise zu ernsten Sicherheitsverstößen und Datenverlusten. Da sich die empfohlenen Sicherheitsmaßnahmen aufgrund von Forschungen und neuartigen Bedrohungen ständig ändern, sind regelmäßige Sicherheitsprüfungen wichtig für eine gute Serveradministration.

Geben Sie auf der Website von Qualys SSL Labs den vollständigen Domain-Namen Ihres Servers ein, in der Form **`www.example.com`**. Nach ungefähr zwei Minuten erhalten Sie eine Note (von A bis F) für Ihre Website sowie eine detaillierte Auflistung der Ergebnisse. In der folgenden Tabelle wird der Bericht für eine Domain mit Einstellungen zusammengefasst, die mit der Apache-Standardkonfiguration auf AL2023 identisch sind, und mit einem Certbot-Standardzertifikat.

Gesamtbewertung	B
Zertifikat	100 %
Protokollunterstützung	95 %
Schlüsselaustausch	70 %
Chiffrestärke	90 %

Obwohl die Übersicht zeigt, dass die Konfiguration größtenteils intakt ist, zeigt der detaillierte Bericht einige potenzielle Probleme, die hier nach Schweregrad geordnet aufgelistet werden:

✗ Die RC4-Chiffre kann von bestimmten älteren Browsern verwendet werden. Eine Chiffre ist der mathematische Kern eines Verschlüsselungsalgorithmus. RC4, eine schnelle Chiffre zur Verschlüsselung von TLS-Daten-Streams, ist für gravierende Schwachstellen bekannt. Wenn Sie nicht sehr gute Gründe haben, veraltete Browser zu unterstützen, sollten Sie dies deaktivieren.

✗ Alte TLS-Versionen werden unterstützt. Die Konfiguration unterstützt TLS 1.0 (bereits veraltet) und TLS 1.1 (demnächst veraltet). Seit 2018 wurde nur TLS 1.2 empfohlen.

✗ Forward Secrecy wird nicht vollständig unterstützt. Forward Secrecy ist ein Feature von Algorithmen zur Verschlüsselung mit temporären (flüchtigen) Sitzungsschlüsseln, die von dem privaten Schlüssel abgeleitet werden. In der Praxis bedeutet dies, dass Angreifen HTTPS-Daten nicht entschlüsseln können, selbst wenn sie den langfristigen privaten Schlüssel eines Webservers besitzen.

So korrigieren Sie die TLS-Konfiguration und machen Sie zukunftssicher

1. Öffnen Sie die Konfigurationsdatei `/etc/httpd/conf.d/ssl.conf` in einem Texteditor und kommentieren Sie die folgende Zeile aus, indem Sie „#“ am Anfang der Zeile eingeben.

   ```
   #SSLProtocol all -SSLv3
   ```

2. Fügen Sie die folgende Richtlinie hinzu:

   ```
   #SSLProtocol all -SSLv3
   SSLProtocol -SSLv2 -SSLv3 -TLSv1 -TLSv1.1 +TLSv1.2
   ```

   Diese Richtlinie deaktiviert die SSL-Versionen 2 und 3 explizit sowie auch die TLS-Versionen 1.0 und 1.1. Der Server akzeptiert jetzt keine verschlüsselten Verbindungen mit Clients, die eine andere Version als TLS 1.2 verwenden. Der Verbose-Wortlaut in der Richtlinie teilt einem menschlichen Leser genauer mit, wofür der Server konfiguriert ist.

   > **Note**
   >
   > Durch eine solche Deaktivierung der TLS-Versionen 1.0 und 1.1 wird ein kleiner Prozentsatz von veralteten Webbrowsern daran gehindert, auf Ihre Website zuzugreifen.

So ändern Sie die Liste der zulässigen Chiffren

1. Suchen Sie in der Konfigurationsdatei `/etc/httpd/conf.d/ssl.conf` den Abschnitt mit der **`SSLCipherSuite`**-Richtlinie und kommentieren Sie die bestehende Zeile aus, indem Sie „#“ am Anfang der Zeile eingeben.

   ```
   #SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5
   ```

2. Geben Sie explizite Verschlüsselungssammlungen und eine Verschlüsselungsreihenfolge an, die Forward Secrecy unterstützt und unsichere Verschlüsselungen vermeidet. Die hier verwendete Richtlinie `SSLCipherSuite` basiert auf der Ausgabe aus dem Mozilla SSL-Konfigurationsgenerator, der eine TLS-Konfiguration an die spezifische Software, die auf Ihrem Server ausgeführt wird, angepasst wird. (Weitere Informationen finden Sie in der nützlichen Ressource Security/Server Side TLS von Mozilla.) Bestimmen Sie zunächst Ihre Apache- und OpenSSL-Versionen, indem Sie die Ausgabe der folgenden Befehle verwenden.

   ```
   [ec2-user ~]$ yum list installed | grep httpd

   [ec2-user ~]$ yum list installed | grep openssl
   ```

   Wenn die zurückgegebenen Informationen beispielsweise Apache 2.4.34 und OpenSSL 1.0.2 sind, geben wir diese in den Generator ein. Wenn Sie das „moderne“ Kompatibilitätsmodell auswählen, wird dadurch eine `SSLCipherSuite`-Richtlinie erstellt, die die Sicherheit aggressiv durchsetzt, aber dennoch für die meisten Browser funktioniert. Wenn die Modemkonfiguration von der Software nicht unterstützt wird, können Sie Ihre Software aktualisieren oder stattdessen die „fortgeschrittene“ Konfiguration wählen.

   ```
   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
   ```

   Die ausgewählten Chiffren weisen ECDHE im Namen auf, eine Abkürzung für Elliptic Curve Diffie-Hellman Ephemeral. Der Begriff Ephemeralität (Flüchtigkeit) gibt die "Forward Secrecy (Folgenlosigkeit)" an. Als Nebenprodukt unterstützten diese Chiffren RC4 nicht.

Wir empfehlen, eine explizite Liste von Chiffren zu verwenden, anstatt sich auf Standardeinstellungen oder knappe Richtlinien zu verlassen, deren Inhalt nicht sichtbar ist.

Kopieren Sie die erzeugte Richtlinie in `/etc/httpd/conf.d/ssl.conf`.

> **Note**
>
> Obwohl dies hier zur besseren Lesbarkeit auf mehrere Zeilen verteilt ist, muss die Richtlinie in einer einzelnen Zeile mit nur einem Doppelpunkt (ohne Leerstellen) aufgeführt werden, wenn sie nach `/etc/httpd/conf.d/ssl.conf` kopiert wird.

3. Entfernen Sie schließlich die Kommentarzeichen in der folgende Zeile, indem Sie das „#“ am Anfang der Zeile löschen.

```
#SSLHonorCipherOrder on
```

Diese Richtlinie zwingt den Server, hochrangige Chiffren zu bevorzugen, einschließlich derjenigen (in diesem Fall), die Forward Secrecy unterstützen. Wenn diese Richtlinie aktiviert ist, versucht der Server, eine hochgradig sichere Verbindung herzustellen, bevor er auf Chiffren mit geringerer Sicherheit zurückgreift.

Nach Abschluss dieser beiden Verfahren speichern Sie die Änderungen in `/etc/httpd/conf.d/ssl.conf` und starten Sie Apache neu.

Wenn Sie die Domain auf Qualys SSL Labs erneut testen, sollten Sie sehen, dass die Schwachstelle in Bezug auf RC4 und andere Warnungen nicht mehr vorhanden sind und dass die Übersicht in etwa wie folgt aussieht.

Gesamtbewertung	A
Zertifikat	100 %
Protokollunterstützung	100 %
Schlüsselaustausch	90 %
Chiffrestärke	90 %

Mit jeder Aktualisierung von OpenSSL werden neue Chiffren eingeführt und die Unterstützung für ältere entfernt. Behalten Sie Ihre EC2 AL203-Instance up-to-date, achten Sie auf Sicherheitsankündigungen von OpenSSL und achten Sie auf Berichte über neue Sicherheitslücken in der Fachpresse.

Fehlerbehebung

- Mein Apache-Webserver startet erst, wenn ich ein Passwort eingebe

  Dieses Verhalten wird erwartet, wenn Sie einen verschlüsselten, passwortgeschützten privaten Serverschlüssel installiert haben.

  Sie können die Verschlüsselungs- und Passwortanforderung vom Schlüssel entfernen. Angenommen, Sie haben einen privaten verschlüsselten RSA-Schlüssel namens `custom.key` im Standardverzeichnis und das Passwort darauf ist **`abcde12345`**, dann führen Sie die folgenden Befehle auf Ihrer EC2-Instance aus, um eine unverschlüsselte Version des Schlüssels zu erzeugen.

  ```
  [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
  ```

  Apache sollte jetzt starten, ohne Sie zur Eingabe eines Passworts aufzufordern.

- Ich erhalte Fehlermeldungen, wenn ich „sudo dnf install -y mod_ssl“ ausführe.

  Wenn Sie die für SSL erforderlichen Pakete installieren, treten möglicherweise Fehler wie die folgenden auf.

  ```
  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
  ```

  Dies bedeutet in der Regel, dass auf Ihrer EC2-Instance AL2023 nicht ausgeführt wird. Dieses Tutorial unterstützt nur Instances, die frisch von einem offiziellen AL2023 AMI erstellt wurden.

Tutorial: Einen WordPress Blog auf AL2023 hosten

Die folgenden Verfahren helfen Ihnen bei der Installation, Konfiguration und Sicherung eines WordPress Blogs auf Ihrer AL203-Instance. Dieses Tutorial ist eine gute Einführung in die Verwendung von Amazon EC2, da Sie die volle Kontrolle über einen Webserver haben, auf dem Ihr WordPress Blog gehostet wird, was bei einem herkömmlichen Hosting-Service nicht typisch ist.

Sie sind für das Aktualisieren der Softwarepakete und das Warten der Sicherheitspatches für Ihren Server verantwortlich. Für eine stärker automatisierte WordPress Installation, die keine direkte Interaktion mit der Webserver-Konfiguration erfordert, bietet der AWS CloudFormation Service eine WordPress Vorlage, mit der Sie auch schnell loslegen können. Weitere Informationen finden Sie unter Erste Schritte im AWS CloudFormation -Benutzerhandbuch. Wenn Sie Ihren WordPress Blog lieber auf einer Windows-Instance hosten möchten, finden Sie weitere Informationen unter Bereitstellen eines WordPress Blogs auf Ihrer Amazon EC2 EC2-Windows-Instance im Amazon EC2 EC2-Benutzerhandbuch. Wenn Sie eine Hochverfügbarkeitslösung mit einer entkoppelten Datenbank benötigen, finden Sie weitere Informationen unter Bereitstellen einer WordPress Hochverfügbarkeitswebsite im Entwicklerhandbuch.AWS Elastic Beanstalk

> **Important**
>
> Diese Verfahren sind für die Verwendung mit AL2023 vorgesehen. Weitere Informationen zu anderen Verteilungen finden Sie in der jeweiligen Dokumentation. Zahlreiche Schritte in diesem Tutorial funktionieren auf Ubuntu-Instances nicht. Hilfe zur Installation WordPress auf einer Ubuntu-Instanz finden Sie WordPressin der Ubuntu-Dokumentation. Sie können diese Aufgabe auch auf Amazon Linux-, macOS- oder Unix-Systemen ausführen. CodeDeploy

Themen

- Voraussetzungen
- Installieren WordPress
- Nächste Schritte
- Hilfe! Mein öffentlicher DNS-Name hat sich geändert und jetzt funktioniert mein Blog nicht mehr.

Voraussetzungen

Wir empfehlen dringend, dass Sie der Instance, die Sie zum Hosten eines WordPress Blogs verwenden, eine Elastic IP-Adresse (EIP) zuordnen. Dies verhindert, dass die öffentliche DNS-

Adresse für Ihre Instance geändert und Ihre Installation beschädigt wird. Wenn Sie einen Domain-Namen besitzen und für Ihren Blog verwenden möchten, können Sie den DNS-Eintrag für den Domain-Namen so aktualisieren, dass er auf Ihre EIP-Adresse verweist (wenden Sie sich an Ihre Domain-Namen-Registrierungsstelle, wenn Sie dabei Hilfe benötigen). Sie können eine EIP-Adresse kostenlos mit einer aktiven Instance verknüpfen. Weitere Informationen finden Sie unter Elastische IP-Adressen im Amazon-EC2-Benutzerhandbuch. Das Tutorial: Installieren Sie einen LAMP-Server auf AL2023-Tutorial enthält auch Schritte zum Konfigurieren einer Sicherheitsgruppe, um HTTP- und HTTPS-Datenverkehr zuzulassen, sowie mehrere Schritte zum Sicherstellen, dass die Dateiberechtigungen für Ihren Webserver richtig festgelegt sind. Informationen zum Hinzufügen von Regeln zu Ihrer Sicherheitsgruppe finden Sie unter Regeln zu einer Sicherheitsgruppe hinzufügen.

Wenn Sie noch keinen Domain-Namen für Ihren Blog haben, können Sie einen Domain-Namen bei Route 53 registrieren und die EIP-Adresse Ihrer Instance mit Ihrem Domain-Namen verknüpfen. Weitere Informationen finden Sie unter Registrieren von Domain-Namen mithilfe von Amazon Route 53 im Entwicklerhandbuch für Amazon Route 53.

Installieren WordPress

Connect zu Ihrer Instance her und laden Sie das WordPress Installationspaket herunter. Weitere Informationen zum Herstellen einer Verbindung mit Ihrer Instance finden Sie unter Verbindung zu AL203-Instances herstellen.

1. Laden Sie diese Pakete mit dem folgenden Befehl herunter und installieren Sie sie.

```
dnf install wget php-mysqlnd httpd php-fpm php-mysqli mariadb105-server php-json
 php php-devel -y
```

2. Möglicherweise wird eine Warnung mit ähnlichem Wortlaut in der Ausgabe angezeigt (die Versionen können im Laufe der Zeit variieren):

```
WARNING:
  A newer release of "Amazon Linux" is available.

  Available Versions:

dnf update --releasever=2023.0.20230202

    Release notes:
     https://aws.amazon.com
```

```
  Version 2023.0.20230204:
    Run the following command to update to 2023.0.20230204:

      dnf update --releasever=2023.0.20230204 ... etc
```

Als bewährte Methode empfehlen wir, das Betriebssystem so weit up-to-date wie möglich beizubehalten. Möglicherweise möchten Sie jedoch jede Version erneut durchlaufen, um sicherzustellen, dass es in Ihrer Umgebung keine Konflikte gibt. Wenn die Installation der in Schritt 1 genannten vorherigen Pakete fehlschlägt, müssen Sie möglicherweise auf eine der neueren aufgeführten Versionen aktualisieren und es erneut versuchen.

3. Laden Sie das neueste WordPress Installationspaket mit dem wget Befehl herunter. Mit dem folgenden Befehl sollte immer die aktuelle Version heruntergeladen werden.

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

4. Extrahieren Sie das Installationspaket. Der Installationsordner wird in einem Ordner namens extrahier `wordpress`.

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

Um einen Datenbankbenutzer und eine Datenbank für Ihre WordPress Installation zu erstellen

Ihre WordPress Installation muss Informationen wie Blogbeiträge und Benutzerkommentare in einer Datenbank speichern. Mit diesem Verfahren können Sie eine Datenbank für Ihren Blog und einen Benutzer mit der Berechtigung zum Lesen und Speichern von Informationen in dieser Datenbank erstellen.

1. Starten Sie den Datenbank- und Webserver.

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

2. Melden Sie sich auf dem Datenbankserver als `root`-Benutzer an. Geben Sie Ihr Datenbank-`root`-Passwort ein, wenn Sie dazu aufgefordert werden; dieses kann sich von Ihrem `root`-Systempasswort unterscheiden oder sogar leer bleiben, wenn Sie Ihren Datenbankserver nicht gesichert haben.

Wenn Sie Ihren Datenbankserver noch nicht gesichert haben, ist es wichtig, dass Sie diesen Schritt durchführen. Weitere Informationen finden Sie unter Schritt 3: Sichern des Datenbankservers (AL2023).

```
[ec2-user ~]$ mysql -u root -p
```

3. Erstellen Sie einen Benutzer und ein Passwort für Ihre MySQL-Datenbank. Ihre WordPress Installation verwendet diese Werte, um mit Ihrer MySQL-Datenbank zu kommunizieren. Geben Sie den folgenden Befehl ein, wobei Sie einen eindeutigen Benutzernamen und ein eindeutiges Passwort einsetzen.

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

Achten Sie darauf, ein sicheres Passwort für Ihren Benutzer zu erstellen. Verwenden Sie keine einfachen Anführungszeichen (') in Ihrem Passwort, da diese den vorhergehenden Befehl beschädigen. Verwenden Sie kein bereits vorhandenes Passwort und speichern Sie das Passwort an einem sicheren Ort.

4. Erstellen Sie Ihre Datenbank. Geben Sie Ihrer Datenbank einen aussagekräftigen Namen wie `wordpress-db`.

> **Note**
>
> Die Satzzeichen um den Datenbanknamen im folgenden Befehl heißen „einfache umgekehrte Anführungszeichen“. Die Taste für das einfache umgekehrte Anführungszeichen (`) befindet sich auf einer Standardtastatur in der Regel oberhalb der Tab-Taste. Einfache umgekehrte Anführungszeichen sind nicht immer erforderlich, sie ermöglichen Ihnen jedoch die Verwendung von Zeichen in Datenbanknamen, die andernfalls nicht zulässig wären, z. B. Bindestriche.

```
CREATE DATABASE `wordpress-db`;
```

5. Gewähren Sie dem WordPress Benutzer, den Sie zuvor erstellt haben, die vollen Rechte für Ihre Datenbank.

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

6. Löschen Sie die Datenbankrechte, damit alle Ihre Änderungen übernommen werden.

```
FLUSH PRIVILEGES;
```

7. Beenden Sie den `mysql`-Client.

```
exit
```

So erstellen und bearbeiten Sie die Datei „wp-config.php“

Der WordPress Installationsordner enthält eine Beispielkonfigurationsdatei namens`wp-config-sample.php`. In diesem Verfahren kopieren und bearbeiten Sie diese Datei, um sie an Ihre individuelle Konfiguration anzupassen.

1. Kopieren Sie die Datei `wp-config-sample.php` in eine Datei namens `wp-config.php`. Dadurch wird eine neue Konfigurationsdatei erstellt und die Originalversion der Beispieldatei als Sicherung aufbewahrt.

```
[ec2-user ~]$ cp wordpress/wp-config-sample.php wordpress/wp-config.php
```

2. Bearbeiten Sie die Datei `wp-config.php` mit Ihrem bevorzugten Texteditor (z. B. `nano`oder vim) und geben Sie Werte für Ihre Installation ein. Falls Sie keinen bevorzugten Texteditor haben, ist `nano` für den Einstieg geeignet.

```
[ec2-user ~]$ nano wordpress/wp-config.php
```

   a. Suchen Sie die Zeile, die `DB_NAME` definiert und ändern Sie `database_name_here` in den Namen der Datenbank, die Sie in Step 4 von Um einen Datenbankbenutzer und eine Datenbank für Ihre WordPress Installation zu erstellen erstellt haben.

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

   b. Suchen Sie die Zeile, die `DB_USER` definiert und ändern Sie `username_here` in den Namen des Datenbankbenutzers, den Sie in Step 3 von Um einen Datenbankbenutzer und eine Datenbank für Ihre WordPress Installation zu erstellen erstellt haben.

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

c. Suchen Sie die Zeile, die DB_PASSWORD definiert und ändern Sie password_here in das sichere Passwort, das Sie in Step 3 von Um einen Datenbankbenutzer und eine Datenbank für Ihre WordPress Installation zu erstellen erstellt haben.

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

d. Suchen Sie den Abschnitt Authentication Unique Keys and Salts. Diese KEY und SALT Werte bieten eine Verschlüsselungsebene für die Browser-Cookies, die WordPress Benutzer auf ihren lokalen Computern speichern. Grundsätzlich wird Ihre Website durch das Hinzufügen langer, zufälliger Werte sicherer. Unter https://api.wordpress.org/secret-key/1.1/salt/ lässt sich ein Satz von zufälligen Schlüsselwerten generieren, die Sie kopieren und in Ihre Datei vom Typ wp-config.php einfügen können. Zum Einfügen von Text in ein PuTTY-Terminal platzieren Sie den Mauszeiger dort, wo der Text eingefügt werden soll, und klicken mit der rechten Maustaste innerhalb des PuTTY-Terminals.

Weitere Informationen zu Sicherheitsschlüsseln finden Sie unter https://wordpress.org/support/article/editing-wp-config-php/#security -keys.

> **Note**
>
> Die folgenden Werte dienen nur als Beispiel; verwenden Sie diese Werte nicht für Ihre Installation.

```
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. Speichern Sie die Datei und beenden Sie den Texteditor.

Um Ihre WordPress Dateien im Apache Document Root zu installieren

- Nachdem Sie den Installationsordner entpackt, eine MySQL-Datenbank und einen MySQL-Benutzer erstellt und die WordPress Konfigurationsdatei angepasst haben, können Sie Ihre Installationsdateien in den Dokumentenstamm Ihres Webservers kopieren, damit Sie das Installationsskript ausführen können, das Ihre Installation abschließt. Der Speicherort dieser Dateien hängt davon ab, ob Ihr WordPress Blog im eigentlichen Stammverzeichnis Ihres Webservers (z. B.*my.public.dns.amazonaws.com*) oder in einem Unterverzeichnis oder Ordner unter dem Stammverzeichnis (z. B.) verfügbar sein soll. *my.public.dns.amazonaws.com/blog*
  - Wenn Sie es im Stammverzeichnis Ihres Dokuments ausführen WordPress möchten, kopieren Sie den Inhalt des WordPress-Installationsverzeichnisses (aber nicht das Verzeichnis selbst) wie folgt:

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

  - Wenn Sie in einem alternativen Verzeichnis unter dem Dokumentenstamm ausführen möchten WordPress , erstellen Sie zuerst dieses Verzeichnis und kopieren Sie dann die Dateien in dieses Verzeichnis. In diesem Beispiel WordPress wird aus dem Verzeichnis ausgeführt`blog`:

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

> **Important**
>
> Wenn Sie nicht umgehend mit dem nächsten Verfahren fortfahren, beenden Sie aus Sicherheitsgründen den Apache-Webserver (`httpd`) jetzt. Nachdem Sie Ihre Installation in das Apache Document Root verschoben haben, ist das WordPress Installationsskript ungeschützt und ein Angreifer könnte sich Zugriff auf Ihr Blog verschaffen, wenn der Apache-Webserver läuft. Zum Beenden des Apache-Webservers geben Sie den Befehl sudo service

httpd stop. Wenn Sie mit dem nächsten Verfahren fortfahren, müssen Sie den Apache-Webserver nicht beenden.

Um die Verwendung von WordPress Permalinks zu ermöglichen

WordPress Permalinks müssen `.htaccess` Apache-Dateien verwenden, um ordnungsgemäß zu funktionieren. Dies ist jedoch unter Amazon Linux standardmäßig nicht aktiviert. Verwenden Sie dieses Verfahren, um alle Überschreibungen im Dokumenten-Stammverzeichnis von Apache zuzulassen.

1. Öffnen Sie die Datei `httpd.conf` mit einem Texteditor Ihrer Wahl (z. B. nano oder vim). Falls Sie keinen bevorzugten Texteditor haben, ist `nano` für den Einstieg geeignet.

   ```
   [ec2-user ~]$ sudo vim /etc/httpd/conf/httpd.conf
   ```

2. Suchen Sie den Abschnitt, der mit beginn `<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. Ändern Sie die Zeile `AllowOverride None` im Abschnitt oben in `AllowOverride All`.

> **Note**
>
> Diese Datei enthält mehrere `AllowOverride`-Zeilen; achten Sie unbedingt darauf, die Zeile im Abschnitt `<Directory "/var/www/html">` zu ändern.

```
AllowOverride All
```

4. Speichern Sie die Datei und beenden Sie den Text-Editor.

Um die PHP-Grafikbibliothek auf AL2023 zu installieren

Mit der GD-Bibliothek für PHP können Sie Bilder bearbeiten. Installieren Sie diese Bibliothek wie folgt, wenn Sie das Header-Image für Ihren Blog zuschneiden müssen. Für die Version phpMyAdmin , die Sie installieren, ist möglicherweise eine bestimmte Mindestversion dieser Bibliothek erforderlich (z. B. Version 8.1).

Verwenden Sie den folgenden Befehl, um die PHP-Grafikbibliothek auf AL2023 zu installieren. Wenn Sie beispielsweise php8.1 von der Quelle als Teil der Installation des LAMP-Stacks installiert haben, installiert dieser Befehl Version 8.1 der PHP-Grafikzeichnungsbibliothek.

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

Verwenden Sie den folgenden Befehl, um die installierte Version zu überprüfen:

```
[ec2-user ~]$ sudo dnf list installed | grep php-gd
```

Ausgabebeispiel:

```
php-gd.x86_64                  8.1.30-1.amzn2           @amazonlinux
```

So installieren Sie die PHP-Grafikzeichenbibliothek auf dem Amazon Linux AMI:

Mit der GD-Bibliothek für PHP können Sie Bilder bearbeiten. Installieren Sie diese Bibliothek wie folgt, wenn Sie das Header-Image für Ihren Blog zuschneiden müssen. Für die Version phpMyAdmin , die Sie installieren, ist möglicherweise eine bestimmte Mindestversion dieser Bibliothek erforderlich (z. B. Version 8.1).

Um zu überprüfen, welche Versionen verfügbar sind, verwenden Sie den folgenden Befehl:

```
[ec2-user ~]$ dnf list | grep php
```

Im Folgenden finden Sie eine Beispielzeile aus der Ausgabe für die PHP-Grafikzeichnungsbibliothek (Version 8.1):

```
php8.1.aarch64                                                 8.1.7-1.amzn2023.0.1
                      @amazonlinux
php8.1-cli.aarch64                                             8.1.7-1.amzn2023.0.1
                      @amazonlinux
php8.1-common.aarch64                                          8.1.7-1.amzn2023.0.1
                      @amazonlinux
php8.1-devel.aarch64                                           8.1.7-1.amzn2023.0.1
                      @amazonlinux
php8.1-fpm.aarch64                                             8.1.7-1.amzn2023.0.1
                      @amazonlinux
php8.1-gd.aarch64                                              8.1.7-1.amzn2023.0.1
                      @amazonlinux
```

Verwenden Sie den folgenden Befehl, um eine bestimmte Version der PHP-Grafikzeichnungsbibliothek (z. B. Version php8.1) auf dem Amazon Linux AMI zu installieren:

```
[ec2-user ~]$ sudo dnf install -y php8.1-gd
```

So beheben Sie Probleme mit den Dateizugriffsberechtigungen für den Apache-Webserver

Für einige der verfügbaren Funktionen ist Schreibzugriff auf das Apache Document Root WordPress erforderlich (z. B. das Hochladen von Medien über die Administrationsbildschirme). Wenn Sie dies nicht bereits getan haben, wenden Sie die folgenden Gruppenmitgliedschaften und Berechtigungen an (diese werden im Tutorial zum LAMP-Webserver ausführlicher beschrieben).

1. Machen Sie den `/var/www`-Benutzer zum Eigentümer der Datei `apache` und ihrer Inhalte.

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

2. Machen Sie die `/var/www`-Gruppe zum Eigentümer der Datei `apache` und ihrer Inhalte.

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

3. Ändern Sie die Verzeichnisberechtigungen von `/var/www` und deren Unterverzeichnissen, indem Sie Schreibberechtigungen für die Gruppe hinzufügen und die Gruppen-ID für zukünftige Unterverzeichnisse einrichten.

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

4. Ändern Sie die Dateiberechtigungen von `/var/www` und deren Unterverzeichnissen rekursiv.

```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod 0644 {} \;
```

> **Note**
>
> Wenn Sie den Server auch WordPress als FTP-Server verwenden möchten, benötigen Sie hier großzügigere Gruppeneinstellungen. Bitte lesen Sie die empfohlenen Schritte und Sicherheitseinstellungen unter, WordPress um dies zu erreichen.

5. Starten Sie den Apache-Webserver neu, damit die neue Gruppe und die neuen Berechtigungen übernommen werden.

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

Um das WordPress Installationsskript mit AL2023 auszuführen

Sie sind bereit zur Installation WordPress. Welche Befehle zu verwenden sind, ist vom Betriebssystem abhängig. Die Befehle in diesem Verfahren sind für die Verwendung mit AL2023 vorgesehen. Wenden Sie das folgende Verfahren mit AL2023 AMI an.

1. Stellen Sie mit dem Befehl systemctl sicher, dass die `httpd`- und Datenbankdienste bei jedem Systemstart gestartet werden.

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

2. Überprüfen Sie, ob der Datenbankserver ausgeführt wird.

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

Wenn der Datenbankdienst nicht ausgeführt wird, starten Sie ihn.

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

3. Überprüfen Sie, ob Ihr Apache-Webserver (`httpd`) ausgeführt wird.

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

Wenn der `httpd`-Dienst nicht ausgeführt wird, starten Sie ihn.

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

4. Geben Sie in einem Webbrowser die URL Ihres WordPress Blogs ein (entweder die öffentliche DNS-Adresse für Ihre Instance oder die Adresse, gefolgt vom `blog` Ordner). Sie sollten das WordPress Installationsskript sehen. Geben Sie die für die WordPress Installation erforderlichen Informationen ein. Wählen Sie Installieren WordPress, um die Installation abzuschließen. Weitere Informationen finden Sie unter Schritt 5: Ausführen des Installationsskripts auf der WordPress Website.

Um das WordPress Installationsskript mit AL2023 AMI auszuführen

1. Stellen Sie mit dem Befehl chkconfig sicher, dass die `httpd`- und Datenbankdienste bei jedem Systemstart gestartet werden.

```
[ec2-user ~]$ sudo chkconfig httpd on && sudo chkconfig mariadb on
```

2. Überprüfen Sie, ob der Datenbankserver ausgeführt wird.

```
[ec2-user ~]$ sudo service mariadb status
```

Wenn der Datenbankdienst nicht ausgeführt wird, starten Sie ihn.

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

3. Überprüfen Sie, ob Ihr Apache-Webserver (`httpd`) ausgeführt wird.

```
[ec2-user ~]$ sudo service httpd status
```

Wenn der `httpd`-Dienst nicht ausgeführt wird, starten Sie ihn.

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

4. Geben Sie in einem Webbrowser die URL Ihres WordPress Blogs ein (entweder die öffentliche DNS-Adresse für Ihre Instance oder die Adresse, gefolgt vom `blog` Ordner). Sie sollten das WordPress Installationsskript sehen. Geben Sie die für die WordPress Installation erforderlichen Informationen ein. Wählen Sie Installieren WordPress, um die Installation abzuschließen. Weitere Informationen finden Sie unter Schritt 5: Ausführen des Installationsskripts auf der WordPress Website.

Nächste Schritte

Nachdem Sie Ihren WordPress Blog getestet haben, sollten Sie erwägen, seine Konfiguration zu aktualisieren.

Verwenden eines benutzerdefinierten Domain-Namens

Wenn ein Domain-Name mit der EIP-Adresse Ihrer EC2 Instance verknüpft ist, können Sie Ihren Blog für die Verwendung dieses Namens anstatt der öffentlichen DNS-Adresse von EC2 konfigurieren. Weitere Informationen finden Sie unter Ändern der Site-URL auf der WordPress Website.

Konfigurieren Ihres Blogs

Sie können Ihren Blog für die Verwendung verschiedener Designs und Plugins konfigurieren, um Ihren Lesern eine persönlich angepasste Umgebung zu bieten. Bisweilen kann der Installationsprozess jedoch fehlschlagen und zum Verlust des gesamten Blogs führen. Wir empfehlen dringend, eine Amazon Machine Image (AMI)-Sicherung Ihrer Instance zu erstellen, bevor Sie versuchen, Designs oder Plug-Ins zu installieren, damit Sie Ihren Blog wiederherstellen können, falls bei der Installation ein Fehler auftritt. Weitere Informationen finden Sie unter Create Your own AMI im Amazon EC2 EC2-Benutzerhandbuch.

Erhöhen der Kapazität

Wenn Ihr WordPress Blog populär wird und Sie mehr Rechenleistung oder Speicherplatz benötigen, sollten Sie die folgenden Schritte in Betracht ziehen:

- Erweitern Sie den Speicherplatz auf Ihrer Instance. Weitere Informationen finden Sie unter Amazon EBS Elastic Volumes.
- Verschieben Sie Ihre MySQL-Datenbank zu Amazon RDS, um die Möglichkeit zur einfachen Skalierung dieses Services zu nutzen.

Verbesserung der Netzwerkleistung Ihres Internetverkehrs

Wenn Sie erwarten, dass Ihr Blog den Traffic von Nutzern auf der ganzen Welt steigern wird, sollten Sie AWS Global Accelerator in Betracht ziehen. Global Accelerator hilft Ihnen dabei, die Latenz zu senken, indem es die Leistung des Internetverkehrs zwischen den Client-Geräten Ihrer Benutzer und Ihrer WordPress Anwendung, auf AWS der ausgeführt wird, verbessert. Global Accelerator nutzt das AWS globale Netzwerk, um den Datenverkehr an einen funktionierenden Anwendungsendpunkt in der AWS Region weiterzuleiten, die dem Client am nächsten ist.

Erfahren Sie mehr über WordPress

Die folgenden Links enthalten weitere Informationen zu WordPress.

- Informationen zu WordPress finden Sie in der WordPress Codex-Hilfedokumentation unter Codex.
- Weitere Informationen zur Fehlerbehebung bei Ihrer Installation finden Sie unter Häufige Installationsprobleme.
- Informationen dazu, wie Sie Ihr WordPress Blog sicherer machen können, finden Sie unter Hardening WordPress.
- Informationen zur Aufbewahrung Ihres WordPress Blogs up-to-date finden Sie unter Aktualisieren WordPress.

Hilfe! Mein öffentlicher DNS-Name hat sich geändert und jetzt funktioniert mein Blog nicht mehr.

Ihre WordPress Installation wird automatisch mit der öffentlichen DNS-Adresse für Ihre EC2-Instance konfiguriert. Wenn Sie die Instance anhalten und neu starten ändert sich die öffentliche DNS-Adresse (es sei denn, sie ist mit einer Elastic IP-Adresse) und Ihr Blog funktioniert nicht mehr, da er auf Ressourcen an einer Adresse verweist, die nicht mehr vorhanden (oder einer anderen EC2 Instance zugewiesen) ist. Eine detailliertere Beschreibung des Problems und mehrere mögliche Lösungen finden Sie unter https://wordpress.org/support/article/changing-the-site-url/.

Wenn dies bei Ihrer WordPress Installation passiert ist, können Sie Ihr Blog möglicherweise mit dem folgenden Verfahren wiederherstellen, bei dem die wp-cli Befehlszeilenschnittstelle für verwendet wird WordPress.

Um die URL Ihrer WordPress Website mit dem zu ändern wp-cli

1. Stellen Sie eine Verbindung mit Ihrer EC2 Instance über SSH her.
2. Notieren Sie die alte und die neue Website-URL für Ihre Instance. Die alte Site-URL ist wahrscheinlich der öffentliche DNS-Name für Ihre EC2-Instance bei der Installation WordPress. Die neue Website-URL ist der aktuelle öffentliche DNS-Name für Ihre EC2 Instance. Wenn Sie nicht sicher sind, was Ihre alte Website-URL ist, können Sie sie mit dem folgenden Befehl mithilfe von curl ermitteln.

   ```
   [ec2-user ~]$ curl localhost | grep wp-content
   ```

   In der Ausgabe, die folgendermaßen aussieht (alte Website-URL in rot) sollten Referenzen auf Ihren alten öffentlichen DNS-Namen enthalten sein:

   ```
   <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. Laden Sie das wp-cli mit dem folgenden Befehl herunter.

   ```
   [ec2-user ~]$ curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/
   phar/wp-cli.phar
   ```
4. Suchen und ersetzen Sie die alte Site-URL in Ihrer WordPress Installation durch den folgenden Befehl. Ersetzen Sie die alten und neuen Site-URLs für Ihre EC2-Instance und den Pfad zu Ihrer WordPress Installation (normalerweise `/var/www/html` oder`/var/www/html/blog`).

   ```
   [ec2-user ~]$ php wp-cli.phar search-replace 'old_site_url' 'new_site_url' --path=/
   path/to/wordpress/installation --skip-columns=guid
   ```
5. Geben Sie in einem Webbrowser die neue Site-URL Ihres WordPress Blogs ein, um zu überprüfen, ob die Website wieder ordnungsgemäß funktioniert. Ist dies nicht der Fall, finden Sie weitere Informationen unter Ändern der Site-URL und Häufige Installationsprobleme.

Verwenden von Amazon Linux 2023 außerhalb von Amazon EC2

Die Amazon Linux 2023-Container-Images können in kompatiblen Container-Laufzeitumgebungen ausgeführt werden. Weitere Informationen zur Verwendung von Amazon Linux 2023 in einem Container finden Sie unter AL2023 in Containern.

Amazon Linux 2023 (AL2023) kann zusätzlich zu einer Auführung direkt auf Amazon EC3 auch als virtualisierter Gast ausgeführt werden. Derzeit sind KVM (`qcow2`), VMware (`OVA`) und Hyper-V (`vhdx`) Images verfügbar.

> **Note**
>
> Die Konfiguration der Amazon-Linux-2023-Images ist anders als bei Amazon Linux 2. Wenn Sie an Amazon Linux 2 als virtuelle Maschine vor Ort ausführen gewöhnt sind, werden Sie Ihre Konfiguration anpassen müssen, damit sie mit AL2023 kompatibel ist.

Laden Sie Amazon Linux 2023-Images zur Verwendung mit KVM, VMware und Hyper-V herunter

Amazon Linux 2023-Festplatten-Images zur Verwendung mit KVM, VMware und Hyper-V können von cdn.amazonlinux.com heruntergeladen werden.

Unterstützte Amazon-Linux-2023-Konfigurationen zur Verwendung in virtualisierten Umgebungen außerhalb von Amazon EC2

In diesem Abschnitt werden die Anforderungen für die Ausführung von Amazon Linux 2023 in virtualisierten Umgebungen außerhalb von Amazon EC2 behandelt, z. B. auf KVM, VMware oder Hyper-V.

Die Basis-AL2023 Systemanforderungen gilt für alle virtualisierten Umgebungen außerhalb von Amazon EC2. Eine Liste der unterstützten Gerätemodelle für jede Hypervisor-Umgebung finden Sie in den folgenden Themen.

KVM, VMware und Hyper-V bieten viele Konfigurationsoptionen, und es ist Vorsicht geboten, sie für Ihre Sicherheits-, Leistungs- und Zuverlässigkeitsanforderungen zu konfigurieren. Weitere Informationen finden Sie in der Dokumentation Ihres Hypervisors.

Themen

- Voraussetzungen für die Ausführung von AL2023 auf KVM
- Voraussetzungen für den Betrieb von AL2023 auf VMware
- Anforderungen für die Ausführung von Amazon Linux 2023 auf Hyper-V

Voraussetzungen für die Ausführung von AL2023 auf KVM

In diesem Abschnitt werden die Anforderungen für die Ausführung von AL2023 auf KVM beschrieben. Die KVM-Images von AL2023 sind für `aarch64`- und `x86-64`-Architekturen verfügbar. Diese Anforderungen gelten zusätzlich zu den Basisanforderungen AL2023 Systemanforderungen für die KVM-Images.

Themen

- KVM-Hostanforderungen für die Ausführung von AL2023 auf KVM
- Geräteunterstützung für AL2023 auf KVM
- Unterstützung für den Startmodus (UEFIundBIOS) für AL2023 auf KVM
- Einschränkungen bei der Ausführung von AL2023 auf KVM

KVM-Hostanforderungen für die Ausführung von AL2023 auf KVM

Die KVM-Images sind derzeit auf einem Host qualifiziert, auf dem Ubuntu 22.04.3 LTS mit der von dieser Ubuntu-Version bereitgestellten `qemu` Version ausgeführt wird6.2+dfsg-2ubuntu6.15, wobei ein Maschinentyp für und ein `q35` Maschinentyp für verwendet werden. `x86-64` `virt` `aarch64`

Geräteunterstützung für AL2023 auf KVM

Folgende **`qemu`**-Gerätemodelle wurden für die Verwendung mit AL2023-KVM-Images (**`aarch64`** und **`x86-64`**) getestet:

- `virtio-blk` (`virtio`-Blockgerät)
- `virtio-scsi` (`virtio`-SCSI-Controller mit Datenträger)
- `virtio-net` (`virtio`-Netzwerkgerät)

- `ahci` (zur Verwendung mit dem virtuellen CD-ROM-Laufwerk)
- `usb-storage` (über `xhci`)

Folgende weitere **`qemu`** Gerätemodelle wurden in der AL2023 KVM-Image-Qualifizierung aktiviert, aber nicht intensiv beansprucht:

- VGA (`qemu` VGA) ausschließlich auf `x86-64`
- `virtio-rng` (virtueller Zufallszahlengenerator)
- Legacy-AT-Tastatur und PS/2-Mausgeräte
- Serielles Legacy-Gerät

Unterstützung für den Startmodus (UEFIundBIOS) für AL2023 auf KVM

Das `x86-64`-Image wurde sowohl im Legacy-BIOS sowie im UEFI-Boot-Modus getestet. Die `aarch64`-Images wurden im UEFI-Boot-Modus getestet.

> **Note**
>
> Wenn der UEFI Startmodus verwendet wird, stellen einige Manager virtueller Maschinen der VM standardmäßig Microsoft Secure Boot-Schlüssel zur Verfügung, wodurch Secure Boot aktiviert wird. AL2023 kann mit dieser Konfiguration nicht gestartet werden.
> Da der AL2023-Bootloader nicht von Microsoft signiert ist, muss die VM entweder ohne UEFI-Schlüssel oder mit den AL2023-Schlüsseln für Secure Boot bereitgestellt werden.

> **Important**
>
> Die Secure Boot-Unterstützung für KVM Images wurde noch nicht validiert.

Einschränkungen bei der Ausführung von AL2023 auf KVM

Es gibt einige bekannte Einschränkungen bei der Ausführung von AL2023 auf KVM.

> **Note**
>
> Code, der einige der aufgelisteten, nicht unterstützten Funktionen implementiert, ist möglicherweise in AL2023 vorhanden und funktioniert ordnungsgemäß. Die Liste der nicht unterstützten Funktionen ist vorhanden, sodass Sie fundierte Entscheidungen darüber treffen können, auf welche Funktionen Sie sich verlassen können und welche Funktionen das Amazon Linux-Team im Rahmen future Updates als funktionstüchtig einstufen wird.

Bekannte Einschränkungen bei der Ausführung von AL2023 auf KVM

- Der KVM-Gast-Agent ist derzeit nicht paketiert und wird auch nicht unterstützt.
- Das Hot-Pluggen und Trennen von CPUs, Speichern oder anderen Gerätetypen wird nicht unterstützt.
- Der Ruhezustand von virtuellen Computern wird nicht unterstützt.
- VM-Migration wird nicht unterstützt.
- Geräte-Passhthrough jeglicher Art (z. B. via PCI-Passthrough oder USB-Passthrough) wird nicht unterstützt.

Voraussetzungen für den Betrieb von AL2023 auf VMware

In diesem Abschnitt werden die Anforderungen für die Ausführung von AL2023 auf beschrieben. VMware Die VMware Images von AL2023 sind nur für die `x86-64` Architektur verfügbar. VMwareBilder für `aarch64` sind nicht verfügbar oder werden nicht unterstützt. Diese Anforderungen gelten zusätzlich zur Grundlage AL2023 Systemanforderungen für die VMware Bilder.

Themen

- VMwareHost-Anforderungen für die Ausführung von AL2023 auf VMware
- Geräteunterstützung für AL2023 auf VMware
- Unterstützung für den Startmodus (UEFIundBIOS) für AL2023 aktiviert VMware
- Einschränkungen bei der Ausführung von AL2023 auf VMware

VMwareHost-Anforderungen für die Ausführung von AL2023 auf VMware

Die AL2023 VMware OVA-Images sind derzeit für Folgendes qualifiziert:

- VMwareWorkstation 17.5.0 läuft auf Hosts mit einem Intel (R) Xeon (R) Platinum 8124M Prozessor
- VMwarevSphere 8.0 mit einem Intel (R) Xeon (R) Platinum 8275CL Prozessor

Die AL2023 VMware OVA-Images spezifizieren eine Maschinenhardwareversion von 13.

VMwareDie Maschinenhardwareversion 13 wird unterstützt von:

- ESXi 6.5 oder höher
- VMwareWorkstation 14 oder höher

Geräteunterstützung für AL2023 auf VMware

Die folgenden VMware Gerätemodelle wurden für die Verwendung mit AL2023 VMware OVA-Bildern getestet (**`x86-64`**nur):

- `vmw_pvscsi`(VMwareparavirtualisierter ControllerSCSI)
- `vmxnet3`(paravirtualisiertes Netzwerkgerät) VMware
- `ata_piix` (Legacy-IDE nur zur Verwendung mit dem virtuellen CD-ROM-Laufwerk)

Zusätzliche VMware Gerätemodelle, die für die VMware AL203-Bildqualifizierung aktiviert wurden, aber nicht stark beansprucht wurden:

- `vmw_vmci`und die zugehörige `vsock` Schnittstelle (virtueller Socket-Transport für den VMware Gastagenten)
- `vmw_balloon`-Memory-Ballooning-Gerät
- VMware`SVGA`Controller
- Legacy-AT-Tastatur und PS/2-Mausgeräte

Das VMware Gast-Agent-Paket (`open-vm-tools`) ist standardmäßig in den VMware AL2023-OVA-Images verfügbar und installiert.

Unterstützung für den Startmodus (UEFIundBIOS) für AL2023 aktiviert VMware

Ab der Version 2023.3.20231211 wurde das AL2023 VMware OVA-Image sowohl im Legacy- als auch im Startmodus validiert. BIOS UEFI Die OVA-Standardkonfiguration ist immer noch veraltet, kann BIOS aber vom Benutzer geändert werden.

> **Important**
>
> Secure Boot-Unterstützung ist erforderlichUEFI. Diese wurde nicht für die Ausführung von AL2023 validiert. VMware

Einschränkungen bei der Ausführung von AL2023 auf VMware

Es gibt einige bekannte Einschränkungen bei der Ausführung von AL2023 auf. VMware

> **Note**
>
> Es kann funktionierender Code in AL2023 existieren, der manche der aufgelisteten, nicht unterstützten Funktionen implementiert. Die Liste der nicht unterstützten Funktionen wird lediglich bereitgestellt, damit Kunden fundierte Entscheidungen darüber treffen können, was nachweislich bereits funktioniert, und welche Funktionen das Amazon Linux-Team im Rahmen zukünftiger Updates als funktionstüchtig einstufen wird.

Bekannte Einschränkungen bei der Ausführung von AL2023 auf VMware

- UEFISecure Boot ist derzeit nicht validiert, wenn AL2023 aktiviert ist. VMware
- Das Hot-Pluggen und Trennen von CPUs, Speichern oder anderen Gerätetypen wird nicht unterstützt.
- Der Ruhezustand von virtuellen Computern wird nicht unterstützt.
- VM-Migration wird nicht unterstützt.
- Geräte-Passhthrough jeglicher Art (z. B. via PCI-Passthrough oder USB-Passthrough) wird nicht unterstützt.

Anforderungen für die Ausführung von Amazon Linux 2023 auf Hyper-V

In diesem Abschnitt werden die Anforderungen für die Ausführung von Amazon Linux 2023 auf Hyper-V behandelt. Die Hyper-V-Images von AL2023 sind nur für die Architektur verfügbar. `x86-64` Hyper-V-Images für `aarch64` sind derzeit nicht verfügbar oder werden nicht unterstützt.

In diesem Abschnitt werden zusätzliche Anforderungen behandelt, die zusätzlich zu den Basisanforderungen AL2023 Systemanforderungen für Hyper-V-Images gelten.

Themen

- Hyper-V-Hostanforderungen für die Ausführung von Amazon Linux 2023 auf Hyper-V
- Geräteunterstützung für Amazon Linux 2023 auf Hyper-V
- Einschränkungen bei der Ausführung von Amazon Linux 2023 auf Hyper-V

Hyper-V-Hostanforderungen für die Ausführung von Amazon Linux 2023 auf Hyper-V

Die Hauptqualifikation von Amazon Linux 2023 auf Hyper-V erfolgt auf Windows Server 2022, das auf einer `c5.metal` EC2-Instance ausgeführt wird.

Geräteunterstützung für Amazon Linux 2023 auf Hyper-V

Amazon Linux 2023 wurde sowohl auf virtuellen Hyper-V-Maschinen der Generation 1 als auch der Generation 2 mit der folgenden virtualisierten Hardware getestet:

- VM der ersten Generation (Legacy-BIOS-Start)
- VM der Generation 2 (UEFI-Start — kein sicherer Start)
- Die folgenden Gerätemodelle wurden für die Verwendung mit AL2023 Hyper-V-Images getestet:
  - Virtueller Hyper-V-Speicher *`hv_storvsc`* für die Stammfestplatte und das emulierte CD-ROM-Laufwerk auf virtuellen Maschinen der zweiten Generation
  - Emulierte PIIX-IDE *`ata_piix`* für das virtuelle CD-ROM-Laufwerk auf virtuellen Maschinen der Generation 1
  - Virtuelles Hyper-V-Ethernet `hv_netvsc`
- Die folgenden Gerätemodelle sind aktiviert, wurden jedoch nur geringfügig getestet:
  - Legacy-VGA-Textmodus auf virtuellen Maschinen der 1. Generation
  - Framebuffer auf UEFI-Firmware-Basis auf VMs *`simpledrmfb`* der Generation 2
  - Hyper-V-Ballon `hv_balloon`

  - Hyper-V-Ballon `hv_balloon`
  - Hyper-V HID/Maus `hid_hyperv`
- Die folgenden Gerätemodi sind derzeit in AL2023 nicht aktiviert:
  - Hyper-V-PCI-Passthrough
  - Hyper-V-DRM-Grafik

> **Important**
>
> Für virtuelle Maschinen der Generation 2 wird Secure Boot nicht unterstützt und muss vor dem Start der virtuellen Maschine deaktiviert werden, damit Amazon Linux 2023 erfolgreich gestartet werden kann. Hyper-V unterstützt derzeit nur Secure Boot mit Softwarekomponenten, die mit eigenen Schlüsseln von Microsoft signiert sind, während der Amazon Linux-Bootloader mit einem privaten Amazon-Schlüssel signiert ist. Hyper-V unterstützt derzeit nicht den Import von Schlüsseln von Drittanbietern.

Einschränkungen bei der Ausführung von Amazon Linux 2023 auf Hyper-V

Im Folgenden sind einige bekannte Einschränkungen bei der Ausführung von Amazon Linux 2023 auf Hyper-V aufgeführt:

> **Note**
>
> Es kann funktionierender Code in AL2023 existieren, der manche der aufgelisteten, nicht unterstützten Funktionen implementiert. Die Liste der nicht unterstützten Funktionen wird lediglich bereitgestellt, damit Kunden fundierte Entscheidungen darüber treffen können, was nachweislich bereits funktioniert, und welche Funktionen das Amazon Linux-Team im Rahmen zukünftiger Updates als funktionstüchtig einstufen wird.

Bekannte Einschränkungen bei der Ausführung von AL2023 auf Hyper-V

- Der UEFI Secure Boot-Modus wird derzeit nicht unterstützt und funktioniert auch nicht mit AL2023 auf Hyper-V
- Das Hot-Pluggen und Trennen von CPUs, Speichern oder anderen Gerätetypen wird nicht unterstützt.

- VM-Hibernation wird nicht unterstützt.
- Die Migration virtueller Maschinen (VM) wird nicht unterstützt.
- Geräte-Passhthrough jeglicher Art (z. B. via PCI-Passthrough oder USB-Passthrough) wird nicht unterstützt.

Einrichtung und `cloud-init`-Konfiguration von Amazon Linux 2023 bei Verwendung außerhalb von Amazon EC2

In diesem Abschnitt wird beschrieben, wie Sie eine virtuelle Amazon Linux 2023-Maschine einrichten und konfigurieren, wenn sie nicht direkt auf Amazon EC2 ausgeführt wird, z. B. wenn sie auf KVM, VMware oder Hyper-V läuft.

Standardmäßig werden Images einer Amazon-Linux-2023-VM nicht mit einem Benutzerkennwort oder SSH-Schlüssel bereitgestellt. Ihre Netzwerkkonfiguration wird über DHCP an der ersten erkannten Netzwerkschnittstelle abgerufen. Das bedeutet, dass es ohne zusätzliche Konfiguration standardmäßig keine Möglichkeit gibt, eine Verbindung zur resultierenden virtuellen Maschine herzustellen.

Aus diesem Grund muss der virtuellen Maschine irgendeine Form von Konfiguration bereitgestellt werden. Der Standardmechanismus dafür für Amazon Linux erfolgt über `cloud-init`-Datenquellen.

Amazon Linux 2023 wurde mit den folgenden Datenquellen qualifiziert:

NoCloud

Dies ist die traditionelle Methode zur Konfiguration von lokalen Images mithilfe einer virtuellen CD-ROM, die ein Seed-ISO9660-Image mit `cloud-init`-Konfigurationsdateien enthält.

VMware

Amazon Linux 2023 unterstützt außerdem die Konfiguration von VMware-Images, die auf vSphere über die VMware-spezifische Datenquelle über VMware-`guestinfo.userdata` und -`guestinfo.metadata` ausgeführt wird.

> **Note**
>
> Die Konfiguration der Datenquellen kann sich von Amazon Linux 2 unterscheiden. Genauer gesagt verwendet Amazon Linux 2023 `systemd-networkd` für seine Konfiguration und

erfordert die Verwendung von `cloud-init`-„Networking Config Version 2", wie in der Dokumentation zur `cloud-init`-Netzwerkkonfiguration dokumentiert.

Die vollständige Dokumentation der `cloud-init`-Konfigurationsmechanismen für die in Amazon Linux 2023 enthaltene `cloud-init`-Version finden Sie in der Upstream-`cloud-init`-Dokumentation.

NoCloud (`seed.iso`) `cloud-init` Konfiguration für Amazon Linux 2023 auf KVM und VMWare

In diesem Abschnitt wird beschrieben, wie Sie ein `seed.iso` Image erstellen und verwenden, um Amazon Linux 2023 zu konfigurieren, das auf KVM oder ausgeführt wirdVMware. Da KVM VMware Umgebungen nicht über Amazon EC2 Instance Meta Data Service (IMDS) verfügen, ist eine alternative Methode zur Konfiguration von Amazon Linux 2023 erforderlich, und die Bereitstellung eines `seed.iso` Images ist eine dieser Methoden.

Das `seed.iso`-Start-Image enthält die Erstkonfigurationsinformationen, die zum Starten und Konfigurieren Ihrer neuen VM benötigt werden, einschließlich Netzwerkkonfiguration, Hostname und Benutzerdaten.

> **Note**
>
> Das `seed.iso`-Start-Image enthält nur die Konfigurationsinformationen, die zum Starten der VM benötigt werden. Amazon Linux 2-Betriebssystemdateien sind nicht enthalten.

Sie benötigen mindestens zwei Konfigurationsdateien zum Erstellen des `seed.iso`-Start-Images, manchmal sogar drei:

`meta-data`

Diese Datei enthält in der Regel den VM-Hostnamen.

`user-data`

Diese Datei konfiguriert in der Regel Benutzerkonten, deren Passwörter, ssh-Schlüsselpaare, und/oder Zugriffsmechanismen. Die Amazon Linux 2023 KVM- und Vmware-Images erstellen standardmäßig ein `ec2-user`-Benutzerkonto. Mithilfe der `user-data`-Konfigurationsdatei legen Se das Passwort für das Standard-Benutzerkonto fest.

`network-config` (optional)

Diese Datei stellt normalerweise eine Netzwerkkonfiguration für die virtuelle Maschine bereit, die die Standardkonfiguration überschreibt. Die Standardkonfiguration verwendet DHCP auf der ersten verfügbaren Netzwerkschnittstelle.

Erstellen Sie das **`seed.iso`**-Festplatten-Image

1. Auf einem Linux- oder macOS-Computer legen Sie einen neuen Ordner mit dem Namen `seedconfig` an und öffnen ihn.

   > **Note**
   >
   > Diese Schritte können auch unter Windows oder einem anderen Betriebssystem ausgeführt werden, jedoch benötigen Sie ein Tool, das `mkisofs` entspricht, um die Erstellung des `seed.iso`-Images abzuschließen.

2. Erstellen Sie die `meta-data`-Konfigurationsdatei.

   a. Erstellen Sie eine neue Datei mit dem Namen `meta-data`.

   b. Öffnen Sie die `meta-data`-Datei mit Ihrem bevorzugten Texteditor und fügen Sie Folgendes hinzu, indem Sie *`vm-hostname`* durch den Hostnamen der VM ersetzen:

   ```
   local-hostname: vm-hostname
   ```

   c. Speichern und schließen Sie die `meta-data`-Konfigurationsdatei.

3. Erstellen Sie die `user-data`-Konfigurationsdatei.

   a. Erstellen Sie eine neue Datei mit dem Namen `user-data`.

   b. Öffnen Sie die `user-data`-Datei mit Ihrem bevorzugten Texteditor und fügen Sie Folgendes hinzu, wobei Sie Inhalte nach Bedarf ersetzen:

   ```
   #cloud-config
   #vim:syntax=yaml
   users:
   # A user by the name 'ec2-user' is created in the image by default.
   - default
   - name: ec2-user
   ```

```
  ssh_authorized_keys:
   - ssh-rsa ssh-key
  # In the above line, replace ssh key with the content of your ssh public key.
```

c. Sie können der `user-data` Konfigurationsdatei optional weitere Benutzerkonten hinzufügen.

   Hierzu geben Sie weitere Benutzerkonten an, deren Zugriffsmechanismen, Passwörter und Schlüsselpaare. Weitere Informationen zu den unterstützten Richtlinien finden Sie in der Upstream-`cloud-init`Dokumentation.

d. Speichern und schließen Sie die `user-data`-Konfigurationsdatei.

4. (Optional) Erstellen der `network-config`-Konfigurationsdatei.

   a. Erstellen Sie eine neue Datei mit dem Namen `network-config`.

   b. Öffnen Sie die `network-config`-Datei mit Ihrem bevorzugten Texteditor und fügen Sie Folgendes hinzu, indem Sie die verschiedenen IP-Adressen durch die für Ihr System zutreffenden Adressen ersetzen.

```
version: 2
ethernets:
  enp1s0:
    addresses:
      - 192.168.122.161/24
    gateway4: 192.168.122.1
    nameservers:
      addresses: 192.168.122.1
```

> **Note**
>
> Die `cloud-init`-Netzwerkkonfiguration bietet Mechanismen für den Abgleich mit der MAC-Adresse der Schnittstelle. So wird kein Schnittstellenname angegeben, der sich je nach VM-Konfiguration ändern kann. Diese (und weitere) `cloud-init`-Funktionen für die Netzwerkkonfiguration werden in der Upstream-Dokumentation zu `cloud-init`-Network Config Version 2 ausführlicher beschrieben.

   c. Speichern und schließen Sie die `network-config`-Konfigurationsdatei.

5. Erstellen Sie das `seed.iso`-Festplatten-Image mithilfe von `meta-data` und `user-data`, sowie den optionalen `network-config`-Konfigurationsdateien, die Sie in den vorausgegangenen Schritten erstellt haben.

   Gehen Sie je nach dem Betriebssystem, unter dem Sie das `seed.iso`-Festplatten-Image erstellen, wie folgt vor.

   - Bei Linux-Systemen verwenden Sie ein Tool wie **`mkisofs`** oder **`genisoimage`**, um die fertige `seed.iso`-Datei zu generieren. Öffnen Sie den `seedconfig`-Ordner und führen Sie den folgenden Befehl aus:

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

   - Wenn Sie ein `network-config` verwenden, muss es in den Aufruf von **`mkisofs`** mit inbegriffen werden:

     ```
     $ mkisofs -output seed.iso -volid cidata -joliet -rock user-data meta-data
      network-config
     ```

   - Bei macOS-Systemen können Sie ein Tool wie **`hdiutil`** verwenden, um die fertige `seed.iso` Datei zu generieren. Da **`hdiutil`** mit einem Pfadnamen anstelle einer Dateiliste arbeitet, kann derselbe Aufruf unabhängig davon verwendet werden, ob eine `network-config`-Konfigurationsdatei erstellt wurde oder nicht.

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

6. Die resultierende `seed.iso`-Datei kann nun über ein virtuelles CD-ROM-Laufwerk an Ihre neue Amazon-Linux-2023-VM angehängt werden, damit `cloud-init` sie beim ersten Start finden und die Konfiguration auf das System anwenden kann.

VMware**`cloud-init`**guestinfo-Konfiguration für AL2023 auf VMware

VMwareUmgebungen verfügen nicht über den Amazon EC2 Instance Meta Data Service (IMDS), sodass eine alternative Methode zur Konfiguration von AL2023 erforderlich ist. In diesem Abschnitt wird beschrieben, wie Sie einen alternativen Konfigurationsmechanismus für das `seed.iso` virtuelle CD-ROM-Laufwerk verwenden, das in VMware vSphere verfügbar ist.

Diese Konfigurationsmethode verwendet den VMware `extraconfig` Mechanismus zur Bereitstellung von Konfigurationsdaten für. `cloud-init` Für jeden der folgenden Schlüssel muss eine entsprechende ***keyname*`.encoding`** Eigenschaft angegeben werden.

Die folgenden Schlüssel können für den VMware `extraconfig` Mechanismus bereitgestellt werden.

`guestinfo.metadata`

JSON oder YAML mit `cloud-init`-Metadaten

`guestinfo.userdata`

Ein YAML-Dokument mit `cloud-init`-Benutzerdaten im `cloud-config`-Format.

`guestinfo.vendordata` (optional)

YAMLenthält `cloud-init` Herstellerdaten

Die entsprechenden Verschlüsselungseigenschaften (`guestinfo.metadata.encoding`, `guestinfo.userdata.encoding` und`guestinfo.vendordata.encoding`) enthalten evtl.:

`base64`

Der Inhalt der Eigenschaft ist `base64`-verschlüsselt.

`gzip+base64`

Der Inhalt der Eigenschaft wird nach der `base64`-Verschlüsselung mit `gzip` komprimiert.

> **Note**
>
> Die `seed.iso` Methode unterstützt eine separate (optionale) `network-config` Konfigurationsdatei. VMware`guestinfo`unterscheidet sich darin, wie die Netzwerkkonfiguration bereitgestellt wird. Zusätzliche Informationen finden Sie im folgenden Abschnitt.

Wenn eine explizite Netzwerkkonfiguration gewünscht wird, sollte sie in Form von zwei YAML- oder JSON-Eigenschaften in `metadata` aufgenommen werden:

`network`

Enthält die kodierte Netzwerkkonfiguration in JSON- oder YAML-Form.

`network.encoding`

Enthält die Kodierung der obigen Netzwerkkonfigurationsdaten. Die `cloud-init`-unterstützten Verschlüsselungen sind dieselben wie für die `guestinfo`-Daten: `base64` und `gzip+base64`.

Example Verwenden des VMware**`govc`** vSphere-CLI-Tools zum Übergeben der Konfiguration mit **`guestinfo`**

1. Bereiten Sie die Konfigurationsdateien `meta-data`, `user-data` und die optionalen `network-config` Konfigurationsdateien wie unter beschrieben vorNoCloud (`seed.iso`) `cloud-init` Konfiguration für Amazon Linux 2023 auf KVM und VMWare.
2. Konvertieren Sie die Konfigurationsdateien in Formate, die von verwendet werden können VMware`guestinfo`.

```
# 'meta-data', `user-data` and `network-config` are the configuration
# files in the same format that would be used by a NoCloud (seed.iso)
# data source, read-them and convert them to VMware guestinfo
#
# The VM_NAME variable is assumed to be set to the name of the VM
# It is assumed that the necessary govc environment (credentials etc...) are
 already set

metadata=$(cat "meta-data")
userdata=$(cat "user-data")
if [ -e "network-config" ] ; then
    # We need to embed the network config inside the meta-data
    netconf=$(base64 -w0 "network-config")
    metadata=$(printf "%s\nnetwork: %s\nnetwork.encoding: base64" "$metadata"
 "$netconf")
fi
metadata=$(base64 -w0 <<< "$metadata")
govc vm.change -vm "$VM_NAME" \
    -e guestinfo.metadata="$metadata" \
    -e guestinfo.metadata.encoding="base64"
userdata=$(base64 -w0 <<< "$userdata")
govc vm.change -vm "$VM_NAME" \
    -e guestinfo.userdata="$userdata" \
```

```
    -e guestinfo.userdata.encoding="base64"
```

Vergleich von Paketen, die auf dem Amazon Linux 2023 Standard-AMI installiert sind, mit dem AL2023 KVM-Image

Ein Vergleich der auf dem AL203-Standard-AMI vorhandenen RPMs mit den auf dem AL2023-KVM-Image vorhandenen RPMs.

Paket	AMI	KVM
acl	2.3.1	2.3.1
acpid	2.0.32	
alternatives	1.15	1.15
amazon-chrony-config	4.3	
amazon-ec2-net-utils	2.4.1	
amazon-linux-onprem		1.2
amazon-linux-repo-cdn		2023,4.20240513
amazon-linux-repo-s3	2023,4.20240513	
amazon-linux-sb-keys	2023,1	2023,1
amazon-onprem-network		1.2
amazon-rpm-config	228	228
amazon-ssm-agent	3.3.380,0	3.3.380,0
at	3.1.23	3.1.23
attr	2.5.1	2.5.1

Paket	AMI	KVM
audit	3.0.6	3.0.6
audit-libs	3.0.6	3.0.6
aws-cfn-bootstrap	2.0	
awscli-2	2,15,30	2,15,30
basesystem	11	11
bash	5.2,15	5.2.15
bash-completion	2.11	2.11
bc	1,07,1	1,07,1
bind-libs	9,16,48	9,16,48
bind-license	9,16,48	9,16,48
bind-utils	9,16,48	9,16,48
binutils	2,39	2,39
boost-filesystem	1,75,0	1,75,0
boost-system	1,75,0	1,75,0
boost-thread	1,75,0	1,75,0
bzip2	1.0.8	1.0.8
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023,2,64	2023,2,64
c-ares	1.19.0	
checkpolicy	3.4	3.4

Paket	AMI	KVM
chkconfig	1.15	1.15
chrony	4.3	4.3
cloud-init	22,2,2	22.2.2
cloud-init-cfg-ec2	22.2.2	
cloud-init-cfg-onprem		22.2.2
cloud-utils-growpart	0,31	0,31
coreutils	8,32	8,32
coreutils-common	8,32	8,32
cpio	2,13	2,13
cracklib	2.9.6	2.9.6
cracklib-dicts	2.9.6	2.9.6
crontabs	1.11	1.11
crypto-policies	20220428	20220428
crypto-policies-scripts	20220428	20220428
cryptsetup	2.6.1	2.6.1
cryptsetup-libs	2.6.1	2.6.1
curl-minimal	8,5,0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27
cyrus-sasl-plain	2.1.27	2.1.27

Paket	AMI	KVM
dbus	1.12,28	1.12.28
dbus-broker	32	32
dbus-common	1.12.28	1.12.28
dbus-libs	1.12.28	1.12.28
device-mapper	1,02,185	1,02,185
device-mapper-libs	1,02,185	1,02,185
diffutils	3.8	3.8
dnf	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2
dnf-plugins-core	4.3.0	4.3.0
dnf-plugin-support-info	1.2	1.2
dnf-utils	4.3.0	4.3.0
dosfstools	4,2	4.2
dracut	055	055
dracut-config-ec2	3.0	
dracut-config-generic	055	055
dwz	0,14	0,14

Paket	AMI	KVM
dyninst	10.2.1	10.2.1
e2fsprogs	1,46,5	1,46,5
e2fsprogs-libs	1,46,5	1,46,5
ec2-hibinit-agent	1.0.8	
ec2-instance-connect	1.1	
ec2-instance-connect-selinux	1.1	
ec2-utils	2.2.0	
ed	1.14.2	1.14.2
efi-filesystem	5	5
efi-srpm-macros	5	5
efivar	38	38
efivar-libs	38	38
elfutils-debuginfod-client	0.188	0.188
elfutils-default-yama-scope	0.188	0.188
elfutils-libelf	0.188	0.188
elfutils-libs	0.188	0.188
ethtool	5,15	5,15
expat	2.5.0	2.5.0

Paket	AMI	KVM
file	5,39	5,39
file-libs	5,39	5,39
filesystem	3,14	3,14
findutils	4.8.0	4.8.0
fonts-srpm-macros	2.0.5	2.0.5
fstrm	0.6.1	0.6.1
fuse-libs	2.9.9	2.9.9
gawk	5.1.0	5.1.0
gdbm-libs	1,19	1,19
gdisk	1.0.8	1.0.8
gettext	0,21	0,21
gettext-libs	0,21	0,21
ghc-srpm-macros	1.5.0	1.5.0
glib2	2,74,7	2,74,7
glibc	2,34	2,34
glibc-all-langpacks	2,34	2,34
glibc-common	2,34	2,34
glibc-gconv-extra	2,34	2,34
glibc-locale-source	2,34	2,34
gmp	6.2.1	6.2.1

Paket	AMI	KVM
gnupg2-minimal	2.3.7	2.3.7
gnutls	3.8.0	3.8.0
go-srpm-macros	3.2.0	3.2.0
gpgme	1.15.1	1.15.1
gpm-libs	1.20.7	1,20,7
grep	3.8	3.8
groff-base	1,22,4	1.22,4
grub2-common	2,06	2,06
grub2-efi-aa64-ec2	2,06 (aarch64)	2,06 (aarch64)
grub2-efi-x64-ec2	2,06 (x86_64)	2,06 (x86_64)
grub2-pc		2,06 (x86_64)
grub2-pc-modules	2,06	2,06 (Noarch)
grub2-tools	2,06	2,06
grub2-tools-minimal	2,06	2,06
grubby	8,40	8,40
gssproxy	0.8.4	0.8.4
gzip	1.12	1.12
hostname	3,23	3,23
hunspell	1.7.0	1.7.0
hunspell-en	0,20140811,1	0,20140811,1

Paket	AMI	KVM
hunspell-en-GB	0,20140811,1	0,20140811,1
hunspell-en-US	0,20140811,1	0,20140811,1
hunspell-filesystem	1.7.0	1.7.0
hwdata	0,353	0,353
info	6.7	6.7
inih	49	49
initscripts	10,09	10,09
iproute	5.10.0	5.10.0
iputils	20210202	20210202
irqbalance	1.9.0	1.9.0
jansson	2.14	2.14
jitterentropy	3.4.1	3.4.1
jq	1.7.1	
json-c	0,14	0,14
kbd	2.4.0	2.4.0
kbd-misc	2.4.0	2.4.0
kernel	6.1,90	6.1,90
kernel-livepatch-repo-cdn		2023,4,20240513
kernel-livepatch-repo-s3	2023,4.20240513	

Paket	AMI	KVM
kernel-modules-extra		6.1,90
kernel-modules-extra-common		6.1,90
kernel-srpm-macros	1,0	1,0
kernel-tools	6.1,90	6.1,90
keyutils	1.6.3	1.6.3
keyutils-libs	1.6.3	1.6.3
kmod	29	29
kmod-libs	29	29
kpatch-runtime	0.9.7	0.9.7
krb5-libs	1,21	1,21
less	608	608
libacl	2.3.1	2.3.1
libaio	0,3,111	0,3.111
libarchive	3,5.3	3.5.3
libargon2	20171227	20171227
libassuan	2,5.5	2.5.5
libattr	2.5.1	2.5.1
libbasicobjects	0.1.1	0.1.1
libblkid	2,37,4	2,37,4
libcap	2,48	2,48

Paket	AMI	KVM
libcap-ng	0.8.2	0.8.2
libcbor	0.7.0	0.7.0
libcollection	0.7.0	0.7.0
libcom_err	1,46,5	1,46,5
libcomps	0,120	0,120
libconfig	1.7.2	1.7.2
libcurl-minimal	8.5.0	8.5.0
libdb	5.3.28	5.3.28
libdhash	0.5.0	
libdnf	0,69,0	0,69,0
libeconf	0,4,0	0,4,0
libedit	3.1	3.1
libev	4,33	4,33
libevent	2.1.12	2.1.12
libfdisk	2,37,4	2,37,4
libffi	3.4.4	3.4.4
libfido2	1.10.0	1.10.0
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	11.4.1

Paket	AMI	KVM
libgpg-error	1,42	1,42
libibverbs	48,0	48,0
libidn2	2.3.2	2.3.2
libini_config	1.3.1	1.3.1
libkcapi	1.4.0	1.4.0
libkcapi-hmaccalc	1.4.0	1.4.0
libldb	2.6.2	
libmaxminddb	1.5.2	1.5.2
libmetalink	0.1.3	0.1.3
libmnl	1.0.4	1.0.4
libmodulemd	2.13.0	2.13.0
libmount	2,37,4	2,37,4
libnfsidmap	2.5.4	2.5.4
libnghttp2	1,59,0	1,59,0
libnl3	3.5.0	3.5.0
libpath_utils	0,2.1	0.2.1
libpcap	1.10.1	1.10.1
libpipeline	1.5.3	1.5.3
libpkgconf	1.8.0	1.8.0
libpsl	0,21,1	0,21,1

Paket	AMI	KVM
libpwquality	1.4.4	1.4.4
libref_array	0,15	0.1.5
librepo	1,14,5	1.14,5
libreport-filesystem	2.15,2	2.15,2
libseccomp	2.5.3	2.5.3
libselinux	3.4	3.4
libselinux-utils	3.4	3.4
libsemanage	3.4	3.4
libsepol	3.4	3.4
libsigsegv	2,13	2,13
libsmartcols	2,37,4	2,37,4
libsolv	0,7.22	0.7.22
libss	1,46,5	1,46,5
libsss_certmap	2.9.4	
libsss_idmap	2.9.4	2.9.4
libsss_nss_idmap	2.9.4	2.9.4
libsss_sudo	2.9.4	
libstdc++	11.4.1	11.4.1
libstoragemgmt	1.9.4	1.9.4
libtalloc	2.3.4	

Paket	AMI	KVM
libtasn1	4.19.0	4.19,0
libtdb	1.4.7	
libtevent	0.13.0	
libtextstyle	0,21	0,21
libtirpc	1.3.3	1.3.3
libunistring	0,9,10	0.9,10
libuser	0,63	0,63
libutempter	1.2.1	1.2.1
libuuid	2,37,4	2,37,4
libuv	1.47.0	1.47.0
libverto	0.3.2	0.3.2
libverto-libev	0.3.2	0.3.2
libxcrypt	4.4.33	4.4.33
libxml2	2.10,4	2.10.4
libyaml	0.2.5	0,2,5
libzstd	1.5.5	1.5.5
lm_sensors-libs	3.6.0	3.6.0
lmdb-libs	0.9.29	0,9,29
logrotate	3,20,1	3,20,1
lsof	4,94,0	4,94,0

Paket	AMI	KVM
lua-libs	5.4.4	5.4.4
lua-srpm-macros	1	1
lz4-libs	1.9.4	1.9.4
man-db	2.9.3	2.9.3
man-pages	5,10	5,10
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)
mpfr	4.1.0	4.1.0
nano	5,8	5,8
ncurses	6.2	6.2
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
nettle	3.8	3.8
net-tools	2,0	2.0
newt	0,52,21	0,52,21
nfs-utils	2,5.4	2.5.4
npth	1,6	1,6
nspr	4,35,0	4,35,0
nss	3,90,0	3,90,0
nss-softokn	3,90,0	3,90,0
nss-softokn-freebl	3,90,0	3,90,0

Paket	AMI	KVM
nss-sysinit	3,90,0	3,90,0
nss-util	3,90,0	3,90,0
ntsysv	1.15	1.15
numactl-libs	2.0.14	2.0.14
ocaml-srpm-macros	6	6
oniguruma	6.9.7.1	
openblas-srpm-macros	2	2
openldap	2.4.57	2,4,57
openssh	8,7p1	8,7 p1
openssh-clients	8,7 p1	8,7 p1
openssh-server	8,7 p1	8,7 p1
openssl	3.0.8	3.0.8
openssl-libs	3.0.8	3.0.8
openssl-pkcs11	0.4.12	0.4.12
os-prober	1,77	1,77
p11-kit	0,24,1	0,24,1
p11-kit-trust	0,24,1	0,24,1
package-notes-srpm-macros	0.4	0.4
pam	1.5.1	1.5.1
parted	3.4	3.4

Paket	AMI	KVM
passwd	0,80	0,80
pciutils	3.7.0	3.7.0
pciutils-libs	3.7.0	3.7.0
pcre2	10,40	10,40
pcre2-syntax	10,40	10,40
perl-Carp	1,50	1,50
perl-Class-Struct	0,66	0,66
perl-constant	1,33	1,33
perl-DynaLoader	1,47	1,47
perl-Encode	3,15	3,15
perl-Errno	1,30	1,30
perl-Exporter	5,74	5,74
perl-Fcntl	1.13	1.13
perl-File-Basename	2,85	2,85
perl-File-Path	2,18	2,18
perl-File-stat	1,09	1,09
perl-File-Temp	0,231,100	0,231,100
perl-Getopt-Long	2,52	2,52
perl-Getopt-Std	1.12	1.12
perl-HTTP-Tiny	0,078	0,078

Paket	AMI	KVM
perl-if	0,60,800	0,60,800
perl-interpreter	5,32,1	5,32,1
perl-IO	1,43	1,43
perl-IPC-Open3	1,21	1,21
perl-libs	5,32,1	5,32,1
perl-MIME-Base64	3,16	3,16
perl-mro	1,23	1,23
perl-overload	1,31	1,31
perl-overloading	0,02	0,02
perl-parent	0,238	0,238
perl-PathTools	3,78	3,78
perl-Pod-Escapes	1,07	1,07
perl-podlators	4,14	4,14
perl-Pod-Perldoc	3,28,01	3,28,01
perl-Pod-Simple	3,42	3,42
perl-Pod-Usage	2,01	2,01
perl-POSIX	1,94	1,94
perl-Scalar-List-Utils	1,56	1,56
perl-SelectSaver	1.02	1.02
perl-Socket	2,032	2,032

Paket	AMI	KVM
perl-srpm-macros	1	1
perl-Storable	3,21	3,21
perl-subs	1,03	1,03
perl-Symbol	1,08	1,08
perl-Term-ANSIColor	5,01	5,01
perl-Term-Cap	1,17	1,17
perl-Text-ParseWords	3,30	3,30
perl-Text-Tabs+Wrap	2021,0726	2021,0726
perl-Time-Local	1,300	1,300
perl-vars	1,05	1,05
pkgconf	1.8.0	1.8.0
pkgconf-m4	1.8.0	1.8.0
pkgconf-pkg-config	1.8.0	1.8.0
policycoreutils	3.4	3.4
policycoreutils-python-utils	3.4	
popt	1,18	1,18
procps-ng	3.3,17	3.3.17
protobuf-c	1.4.1	1.4.1
psacct	6.6.4	6.6.4
psmisc	23,4	23,4

Paket	AMI	KVM
publicsuffix-list-dafsa	20240212	20240212
python3	3,9,16	3.9,16
python3-attrs	20,3,0	20.3,0
python3-audit	3.0.6	3.0.6
python3-awscrt	0.19,19	0,19,19
python3-babel	2.9.1	2.9.1
python3-cffi	1.14,5	1.14,5
python3-chardet	4.0.0	4.0.0
python3-colorama	0.4.4	0.4.4
python3-configobj	5.0.6	5.0.6
python3-cryptography	36,1	36,1
python3-daemon	2.3.0	
python3-dateutil	2.8.1	2.8.1
python3-dbus	1.2.18	1.2.18
python3-distro	1.5.0	1.5.0
python3-dnf	4.14.0	4.14.0
python3-dnf-plugins-core	4.3.0	4.3.0
python3-docutils	0,16	0,16
python3-gpg	1.15.1	1.15.1

Paket	AMI	KVM
python3-hawkey	0,69,0	0,69,0
python3-idna	(2.10)	(2.10)
python3-jinja2	2.11,3	2.11.3
python3-jmespath	0.10.0	0.10.0
python3-jsonpatch	1,21	1,21
python3-jsonpointer	2,0	2.0
python3-jsonschema	3.2.0	3.2.0
python3-libcomps	0,120	0,120
python3-libdnf	0,69,0	0,69,0
python3-libs	3,9,16	3.9,16
python3-libselinux	3.4	3.4
python3-libsemanage	3.4	3.4
python3-libstorage mgmt	1.9.4	1.9.4
python3-lockfile	0.12.2	
python3-markupsafe	1.1.1	1.1.1
python3-netifaces	0.10.6	0.10.6
python3-oauthlib	3.0.2	3.0.2
python3-pip-wheel	21.3.1	21.3.1
python3-ply	3.11	3,11

Paket	AMI	KVM
python3-policycoreutils	3.4	3.4
python3-prettytable	0.7.2	0.7.2
python3-prompt-toolkit	3.0,24	3,0,24
python3-pycparser	2,20	2,20
python3-pyrsistent	0,17,3	0,17,3
python3-pyserial	3.4	3.4
python3-pysocks	1.7.1	1.7.1
python3-pytz	2022.7.1	2022,7.1
python3-pyyaml	5.4.1	5.4.1
python3-requests	2,25,1	2,25,1
python3-rpm	4.16.1,3	4.16.1.3
python3-ruamel-yaml	0.16.6	0,16,6
python3-ruamel-yaml-clib	0.1.2	0.1.2
python3-setools	4.4.1	4.4.1
python3-setuptools	59,6,0	59,6,0
python3-setuptools-wheel	59,6,0	59,6,0
python3-six	1.15.0	1.15.0
python3-systemd	235	235

Paket	AMI	KVM
python3-urllib3	1,25,10	1,25,10
python3-wcwidth	0,2,5	0,2,5
python-chevron	0.13.1	
python-srpm-macros	3.9	3.9
quota	4,06	4,06
quota-nls	4,06	4,06
readline	8,1	8.1
rng-tools	6,14	6,14
rootfiles	8.1	8.1
rpcbind	1.2.6	1.2.6
rpm	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3
rpm-libs	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3	4.16.1.3
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3
rpm-sign-libs	4.16.1.3	4.16.1.3
rsync	3.2.6	3.2.6
rust-srpm-macros	21	21
sbsigntools	0.9.4	0.9.4
screen	4.8.0	4.8.0

Paket	AMI	KVM
sed	4.8	4,8
selinux-policy	37,22	37,22
selinux-policy-targeted	37,22	37,22
setup	2,13,7	2.13,7
shadow-utils	4,9 bis 4,9	4,9 bis 4,9
slang	2.3.2	2.3.2
sqlite-libs	3,40,0	3,40,0
sssd-client	2.9.4	2.9.4
sssd-common	2.9.4	
sssd-kcm	2.9.4	
sssd-nfs-idmap	2.9.4	
strace	6.8	6.8
sudo	1.9,15	1.9,15
sysctl-defaults	1,0	1,0
sysstat	12,5.6	12.5.6
systemd	252,16	252,16
systemd-libs	252,16	252,16
systemd-networkd	252,16	252,16
systemd-pam	252,16	252,16
systemd-resolved	252,16	252,16

Paket	AMI	KVM
systemd-udev	252,16	252,16
system-release	2023,4,20240513	2023,4.20240513
systemtap-runtime	4.8	4,8
tar	1,34	1,34
tbb	2020,3	2020,3
tcpdump	4,99,1	4,99,1
tcsh	6,24,07	6,24,07
time	1.9	1.9
traceroute	2.1.3	2.1.3
tzdata	2024a	2024a
unzip	6.0	6.0
update-motd	2.2	2.2
userspace-rcu	0.12.1	0.12.1
util-linux	2,37,4	2,37,4
util-linux-core	2,37,4	2,37,4
vim-common	9,0,2153	9,0,2153
vim-data	9,0,2153	9,0,2153
vim-enhanced	9,0,2153	9,0,2153
vim-filesystem	9,0,2153	9,0,2153
vim-minimal	9,0,2153	9,0,2153

Paket	AMI	KVM
wget	1,21,3	1.21.3
which	2,21	2,21
words	3.0	3.0
xfsdump	3.1.11	3.1.11
xfsprogs	5.18.0	5.18,0
xxd	9,0,2153	9,0,2153
xxhash-libs	0.8.0	0.8.0
xz	5.2,5	5.2.5
xz-libs	5.2.5	5.2.5
yum	4.14.0	4.14.0
zip	3.0	3.0
zlib	1.2.11	1.2.11
zram-generator	1.1.2	
zram-generator-defaults	1.1.2	
zstd	1.5.5	1.5.5

Vergleich von Paketen, die auf dem Amazon Linux 2023 Standard-AMI installiert sind, mit dem VMware-OVA-Image AL2023

Ein Vergleich der auf dem AL203-Standard-AMI vorhandenen RPMs mit den auf dem AL2023 VMware OVA-Image vorhandenen RPMs.

Paket	AMI	VMware OVA
acl	2.3.1	2.3.1
acpid	2.0.32	
alternatives	1.15	1.15
amazon-chrony-config	4.3	
amazon-ec2-net-utils	2.4.1	
amazon-linux-onprem		1.2
amazon-linux-repo-cdn		2023,4.20240513
amazon-linux-repo-s3	2023,4.20240513	
amazon-linux-sb-keys	2023,1	2023,1
amazon-onprem-network		1.2
amazon-rpm-config	228	228
amazon-ssm-agent	3.3.380,0	3.3.380,0
at	3.1.23	3.1.23
attr	2.5.1	2.5.1
audit	3.0.6	3.0.6
audit-libs	3.0.6	3.0.6
aws-cfn-bootstrap	2.0	
awscli-2	2,15,30	2,15,30
basesystem	11	11

Paket	AMI	VMware OVA
bash	5.2,15	5.2.15
bash-completion	2.11	2.11
bc	1,07,1	1,07,1
bind-libs	9,16,48	9,16,48
bind-license	9,16,48	9,16,48
bind-utils	9,16,48	9,16,48
binutils	2,39	2,39
boost-filesystem	1,75,0	1,75,0
boost-system	1,75,0	1,75,0
boost-thread	1,75,0	1,75,0
bzip2	1.0.8	1.0.8
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023,2,64	2023,2,64
c-ares	1.19.0	
checkpolicy	3.4	3.4
chkconfig	1.15	1.15
chrony	4.3	4.3
cloud-init	22,2,2	22.2.2
cloud-init-cfg-ec2	22.2.2	
cloud-init-cfg-onprem		22.2.2

Paket	AMI	VMware OVA
cloud-utils-growpart	0,31	0,31
coreutils	8,32	8,32
coreutils-common	8,32	8,32
cpio	2,13	2,13
cracklib	2.9.6	2.9.6
cracklib-dicts	2.9.6	2.9.6
crontabs	1.11	1.11
crypto-policies	20220428	20220428
crypto-policies-scripts	20220428	20220428
cryptsetup	2.6.1	2.6.1
cryptsetup-libs	2.6.1	2.6.1
curl-minimal	8,5,0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27
cyrus-sasl-plain	2.1.27	2.1.27
dbus	1.12,28	1.12.28
dbus-broker	32	32
dbus-common	1.12.28	1.12.28
dbus-libs	1.12.28	1.12.28
device-mapper	1,02,185	1,02,185
device-mapper-libs	1,02,185	1,02,185

Paket	AMI	VMware OVA
diffutils	3.8	3.8
dnf	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2
dnf-plugins-core	4.3.0	4.3.0
dnf-plugin-support-info	1.2	1.2
dnf-utils	4.3.0	4.3.0
dosfstools	4,2	4.2
dracut	055	055
dracut-config-ec2	3.0	
dracut-config-generic	055	055
dwz	0,14	0,14
dyninst	10.2.1	10.2.1
e2fsprogs	1,46,5	1,46,5
e2fsprogs-libs	1,46,5	1,46,5
ec2-hibinit-agent	1.0.8	
ec2-instance-connect	1.1	
ec2-instance-connect-selinux	1.1	

Paket	AMI	VMware OVA
ec2-utils	2.2.0	
ed	1.14.2	1.14.2
efi-filesystem	5	5
efi-srpm-macros	5	5
efivar	38	38
efivar-libs	38	38
elfutils-debuginfod-client	0.188	0.188
elfutils-default-yama-scope	0.188	0.188
elfutils-libelf	0.188	0.188
elfutils-libs	0.188	0.188
ethtool	5,15	5,15
expat	2.5.0	2.5.0
file	5,39	5,39
file-libs	5,39	5,39
filesystem	3,14	3,14
findutils	4.8.0	4.8.0
fonts-srpm-macros	2.0.5	2.0.5
fstrm	0.6.1	0.6.1
fuse3		3.10,4

Paket	AMI	VMware OVA
fuse3-libs		3.10,4
fuse-common		3.10,4
fuse-libs	2.9.9	2.9.9
gawk	5.1.0	5.1.0
gdbm-libs	1,19	1,19
gdisk	1.0.8	1.0.8
gettext	0,21	0,21
gettext-libs	0,21	0,21
ghc-srpm-macros	1.5.0	1.5.0
glib2	2,74,7	2,74,7
glibc	2,34	2,34
glibc-all-langpacks	2,34	2,34
glibc-common	2,34	2,34
glibc-gconv-extra	2,34	2,34
glibc-locale-source	2,34	2,34
gmp	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7
gnutls	3.8.0	3.8.0
go-srpm-macros	3.2.0	3.2.0
gpgme	1.15.1	1.15.1

Paket	AMI	VMware OVA
gpm-libs	1.20.7	1,20,7
grep	3.8	3.8
groff-base	1,22,4	1.22,4
grub2-common	2,06	2,06
grub2-efi-x64-ec2	2,06	2,06
grub2-pc		2,06
grub2-pc-modules	2,06	2,06
grub2-tools	2,06	2,06
grub2-tools-minimal	2,06	2,06
grubby	8,40	8,40
gssproxy	0.8.4	0.8.4
gzip	1.12	1.12
hostname	3,23	3,23
hunspell	1.7.0	1.7.0
hunspell-en	0,20140811,1	0,20140811,1
hunspell-en-GB	0,20140811,1	0,20140811,1
hunspell-en-US	0,20140811,1	0,20140811,1
hunspell-filesystem	1.7.0	1.7.0
hwdata	0,353	0,353
info	6.7	6.7

Paket	AMI	VMware OVA
inih	49	49
initscripts	10,09	10,09
iproute	5.10.0	5.10.0
iputils	20210202	20210202
irqbalance	1.9.0	1.9.0
jansson	2.14	2.14
jitterentropy	3.4.1	3.4.1
jq	1.7.1	
json-c	0,14	0,14
kbd	2.4.0	2.4.0
kbd-misc	2.4.0	2.4.0
kernel	6.1,90	6.1,90
kernel-livepatch-repo-cdn		2023,4,20240513
kernel-livepatch-repo-s3	2023,4.20240513	
kernel-modules-extra		6.1,90
kernel-modules-extra-common		6.1,90
kernel-srpm-macros	1,0	1,0
kernel-tools	6.1,90	6.1,90

Paket	AMI	VMware OVA
keyutils	1.6.3	1.6.3
keyutils-libs	1.6.3	1.6.3
kmod	29	29
kmod-libs	29	29
kpatch-runtime	0.9.7	0.9.7
krb5-libs	1,21	1,21
less	608	608
libacl	2.3.1	2.3.1
libaio	0,3,111	0,3.111
libarchive	3,5.3	3.5.3
libargon2	20171227	20171227
libassuan	2,5.5	2.5.5
libattr	2.5.1	2.5.1
libbasicobjects	0.1.1	0.1.1
libblkid	2,37,4	2,37,4
libcap	2,48	2,48
libcap-ng	0.8.2	0.8.2
libcbor	0.7.0	0.7.0
libcollection	0.7.0	0.7.0
libcom_err	1,46,5	1,46,5

Paket	AMI	VMware OVA
libcomps	0,120	0,120
libconfig	1.7.2	1.7.2
libcurl-minimal	8.5.0	8.5.0
libdb	5.3.28	5.3.28
libdhash	0.5.0	
libdnf	0,69,0	0,69,0
libeconf	0,4,0	0,4,0
libedit	3.1	3.1
libev	4,33	4,33
libevent	2.1.12	2.1.12
libfdisk	2,37,4	2,37,4
libffi	3.4.4	3.4.4
libfido2	1.10.0	1.10.0
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	11.4.1
libgpg-error	1,42	1,42
libibverbs	48,0	48,0
libidn2	2.3.2	2.3.2
libini_config	1.3.1	1.3.1

Paket	AMI	VMware OVA
libkcapi	1.4.0	1.4.0
libkcapi-hmaccalc	1.4.0	1.4.0
libldb	2.6.2	
libmaxminddb	1.5.2	1.5.2
libmetalink	0.1.3	0.1.3
libmnl	1.0.4	1.0.4
libmodulemd	2.13.0	2.13.0
libmount	2,37,4	2,37,4
libmspack		0.10.1
libnfsidmap	2.5.4	2.5.4
libnghttp2	1,59,0	1,59,0
libnl3	3.5.0	3.5.0
libpath_utils	0,2.1	0.2.1
libpcap	1.10.1	1.10.1
libpipeline	1.5.3	1.5.3
libpkgconf	1.8.0	1.8.0
libpsl	0,21,1	0,21,1
libpwquality	1.4.4	1.4.4
libref_array	0,15	0.1.5
librepo	1,14,5	1.14,5

Paket	AMI	VMware OVA
libreport-filesystem	2.15,2	2.15,2
libseccomp	2.5.3	2.5.3
libselinux	3.4	3.4
libselinux-utils	3.4	3.4
libsemanage	3.4	3.4
libsepol	3.4	3.4
libsigsegv	2,13	2,13
libsmartcols	2,37,4	2,37,4
libsolv	0,7.22	0.7.22
libss	1,46,5	1,46,5
libsss_certmap	2.9.4	
libsss_idmap	2.9.4	2.9.4
libsss_nss_idmap	2.9.4	2.9.4
libsss_sudo	2.9.4	
libstdc++	11.4.1	11.4.1
libstoragemgmt	1.9.4	1.9.4
libtalloc	2.3.4	
libtasn1	4.19.0	4.19,0
libtdb	1.4.7	
libtevent	0.13.0	

Paket	AMI	VMware OVA
libtextstyle	0,21	0,21
libtirpc	1.3.3	1.3.3
libtool-ltdl		2.4.7
libunistring	0,9,10	0.9,10
libuser	0,63	0,63
libutempter	1.2.1	1.2.1
libuuid	2,37,4	2,37,4
libuv	1.47.0	1.47.0
libverto	0.3.2	0.3.2
libverto-libev	0.3.2	0.3.2
libxcrypt	4.4.33	4.4.33
libxml2	2.10,4	2.10.4
libxslt		1.1.34
libyaml	0,2,5	0,2,5
libzstd	1.5.5	1.5.5
lm_sensors-libs	3.6.0	3.6.0
lmdb-libs	0.9.29	0,9,29
logrotate	3,20,1	3,20,1
lsof	4,94,0	4,94,0
lua-libs	5.4.4	5.4.4

Paket	AMI	VMware OVA
lua-srpm-macros	1	1
lz4-libs	1.9.4	1.9.4
man-db	2.9.3	2.9.3
man-pages	5,10	5,10
microcode_ctl	2.1	2.1
mpfr	4.1.0	4.1.0
nano	5,8	5,8
ncurses	6.2	6.2
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
nettle	3.8	3.8
net-tools	2,0	2.0
newt	0,52,21	0,52,21
nfs-utils	2,5.4	2.5.4
npth	1,6	1,6
nspr	4,35,0	4,35,0
nss	3,90,0	3,90,0
nss-softokn	3,90,0	3,90,0
nss-softokn-freebl	3,90,0	3,90,0
nss-sysinit	3,90,0	3,90,0

Paket	AMI	VMware OVA
nss-util	3,90,0	3,90,0
ntsysv	1.15	1.15
numactl-libs	2.0.14	2.0.14
ocaml-srpm-macros	6	6
oniguruma	6.9.7.1	
openblas-srpm-macros	2	2
openldap	2.4.57	2,4,57
openssh	8,7p1	8,7 p1
openssh-clients	8,7 p1	8,7 p1
openssh-server	8,7 p1	8,7 p1
openssl	3.0.8	3.0.8
openssl-libs	3.0.8	3.0.8
openssl-pkcs11	0.4.12	0.4.12
open-vm-tools		12.3,0
os-prober	1,77	1,77
p11-kit	0,24,1	0,24,1
p11-kit-trust	0,24,1	0,24,1
package-notes-srpm-macros	0.4	0.4
pam	1.5.1	1.5.1
parted	3.4	3.4

Paket	AMI	VMware OVA
passwd	0,80	0,80
pciutils	3.7.0	3.7.0
pciutils-libs	3.7.0	3.7.0
pcre2	10,40	10,40
pcre2-syntax	10,40	10,40
perl-Carp	1,50	1,50
perl-Class-Struct	0,66	0,66
perl-constant	1,33	1,33
perl-DynaLoader	1,47	1,47
perl-Encode	3,15	3,15
perl-Errno	1,30	1,30
perl-Exporter	5,74	5,74
perl-Fcntl	1.13	1.13
perl-File-Basename	2,85	2,85
perl-File-Path	2,18	2,18
perl-File-stat	1,09	1,09
perl-File-Temp	0,231,100	0,231,100
perl-Getopt-Long	2,52	2,52
perl-Getopt-Std	1.12	1.12
perl-HTTP-Tiny	0,078	0,078

Paket	AMI	VMware OVA
perl-if	0,60,800	0,60,800
perl-interpreter	5,32,1	5,32,1
perl-IO	1,43	1,43
perl-IPC-Open3	1,21	1,21
perl-libs	5,32,1	5,32,1
perl-MIME-Base64	3,16	3,16
perl-mro	1,23	1,23
perl-overload	1,31	1,31
perl-overloading	0,02	0,02
perl-parent	0,238	0,238
perl-PathTools	3,78	3,78
perl-Pod-Escapes	1,07	1,07
perl-podlators	4,14	4,14
perl-Pod-Perldoc	3,28,01	3,28,01
perl-Pod-Simple	3,42	3,42
perl-Pod-Usage	2,01	2,01
perl-POSIX	1,94	1,94
perl-Scalar-List-Utils	1,56	1,56
perl-SelectSaver	1.02	1.02
perl-Socket	2,032	2,032

Paket	AMI	VMware OVA
perl-srpm-macros	1	1
perl-Storable	3,21	3,21
perl-subs	1,03	1,03
perl-Symbol	1,08	1,08
perl-Term-ANSIColor	5,01	5,01
perl-Term-Cap	1,17	1,17
perl-Text-ParseWords	3,30	3,30
perl-Text-Tabs+Wrap	2021,0726	2021,0726
perl-Time-Local	1,300	1,300
perl-vars	1,05	1,05
pkgconf	1.8.0	1.8.0
pkgconf-m4	1.8.0	1.8.0
pkgconf-pkg-config	1.8.0	1.8.0
policycoreutils	3.4	3.4
policycoreutils-python-utils	3.4	
popt	1,18	1,18
procps-ng	3.3,17	3.3.17
protobuf-c	1.4.1	1.4.1
psacct	6.6.4	6.6.4
psmisc	23,4	23,4

Paket	AMI	VMware OVA
publicsuffix-list-dafsa	20240212	20240212
python3	3,9,16	3.9,16
python3-attrs	20,3,0	20.3,0
python3-audit	3.0.6	3.0.6
python3-awscrt	0.19,19	0,19,19
python3-babel	2.9.1	2.9.1
python3-cffi	1.14,5	1.14,5
python3-chardet	4.0.0	4.0.0
python3-colorama	0.4.4	0.4.4
python3-configobj	5.0.6	5.0.6
python3-cryptography	36,1	36,1
python3-daemon	2.3.0	
python3-dateutil	2.8.1	2.8.1
python3-dbus	1.2.18	1.2.18
python3-distro	1.5.0	1.5.0
python3-dnf	4.14.0	4.14.0
python3-dnf-plugins-core	4.3.0	4.3.0
python3-docutils	0,16	0,16
python3-gpg	1.15.1	1.15.1

Paket	AMI	VMware OVA
python3-hawkey	0,69,0	0,69,0
python3-idna	(2.10)	(2.10)
python3-jinja2	2.11,3	2.11.3
python3-jmespath	0.10.0	0.10.0
python3-jsonpatch	1,21	1,21
python3-jsonpointer	2,0	2.0
python3-jsonschema	3.2.0	3.2.0
python3-libcomps	0,120	0,120
python3-libdnf	0,69,0	0,69,0
python3-libs	3,9,16	3.9,16
python3-libselinux	3.4	3.4
python3-libsemanage	3.4	3.4
python3-libstorage mgmt	1.9.4	1.9.4
python3-lockfile	0.12.2	
python3-markupsafe	1.1.1	1.1.1
python3-netifaces	0.10.6	0.10.6
python3-oauthlib	3.0.2	3.0.2
python3-pip-wheel	21.3.1	21.3.1
python3-ply	3.11	3,11

Paket	AMI	VMware OVA
python3-policycoreutils	3.4	3.4
python3-prettytable	0.7.2	0.7.2
python3-prompt-toolkit	3.0,24	3,0,24
python3-pycparser	2,20	2,20
python3-pyrsistent	0,17,3	0,17,3
python3-pyserial	3.4	3.4
python3-pysocks	1.7.1	1.7.1
python3-pytz	2022.7.1	2022,7.1
python3-pyyaml	5.4.1	5.4.1
python3-requests	2,25,1	2,25,1
python3-rpm	4.16.1,3	4.16.1.3
python3-ruamel-yaml	0.16.6	0,16,6
python3-ruamel-yaml-clib	0.1.2	0.1.2
python3-setools	4.4.1	4.4.1
python3-setuptools	59,6,0	59,6,0
python3-setuptools-wheel	59,6,0	59,6,0
python3-six	1.15.0	1.15.0
python3-systemd	235	235

Paket	AMI	VMware OVA
python3-urllib3	1,25,10	1,25,10
python3-wcwidth	0,2,5	0,2,5
python-chevron	0.13.1	
python-srpm-macros	3.9	3.9
quota	4,06	4,06
quota-nls	4,06	4,06
readline	8,1	8.1
rng-tools	6,14	6,14
rootfiles	8.1	8.1
rpcbind	1.2.6	1.2.6
rpm	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3
rpm-libs	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3	4.16.1.3
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3
rpm-sign-libs	4.16.1.3	4.16.1.3
rsync	3.2.6	3.2.6
rust-srpm-macros	21	21
sbsigntools	0.9.4	0.9.4
screen	4.8.0	4.8.0

Paket	AMI	VMware OVA
sed	4.8	4,8
selinux-policy	37,22	37,22
selinux-policy-targeted	37,22	37,22
setup	2,13,7	2.13,7
shadow-utils	4,9 bis 4,9	4,9 bis 4,9
slang	2.3.2	2.3.2
sqlite-libs	3,40,0	3,40,0
sssd-client	2.9.4	2.9.4
sssd-common	2.9.4	
sssd-kcm	2.9.4	
sssd-nfs-idmap	2.9.4	
strace	6.8	6.8
sudo	1.9,15	1.9,15
sysctl-defaults	1,0	1,0
sysstat	12,5.6	12.5.6
systemd	252,16	252,16
systemd-libs	252,16	252,16
systemd-networkd	252,16	252,16
systemd-pam	252,16	252,16
systemd-resolved	252,16	252,16

Paket	AMI	VMware OVA
systemd-udev	252,16	252,16
system-release	2023,4,20240513	2023,4.20240513
systemtap-runtime	4.8	4,8
tar	1,34	1,34
tbb	2020,3	2020,3
tcpdump	4,99,1	4,99,1
tcsh	6,24,07	6,24,07
time	1.9	1.9
traceroute	2.1.3	2.1.3
tzdata	2024a	2024a
unzip	6.0	6.0
update-motd	2.2	2.2
userspace-rcu	0.12.1	0.12.1
util-linux	2,37,4	2,37,4
util-linux-core	2,37,4	2,37,4
vim-common	9,0,2153	9,0,2153
vim-data	9,0,2153	9,0,2153
vim-enhanced	9,0,2153	9,0,2153
vim-filesystem	9,0,2153	9,0,2153
vim-minimal	9,0,2153	9,0,2153

Paket	AMI	VMware OVA
wget	1,21,3	1.21.3
which	2,21	2,21
words	3.0	3.0
xfsdump	3.1.11	3.1.11
xfsprogs	5.18.0	5.18,0
xmlsec1		1.2.33
xmlsec1-openssl		1.2.33
xxd	9,0,2153	9,0,2153
xxhash-libs	0.8.0	0.8.0
xz	5.2,5	5.2.5
xz-libs	5.2.5	5.2.5
yum	4.14.0	4.14.0
zip	3.0	3.0
zlib	1.2.11	1.2.11
zram-generator	1.1.2	
zram-generator-defaults	1.1.2	
zstd	1.5.5	1.5.5

Vergleich von Paketen, die auf dem Amazon Linux 2023 Standard-AMI installiert sind, mit dem AL2023 Hyper-V-Image

Ein Vergleich der auf dem AL203-Standard-AMI vorhandenen RPMs mit den auf dem AL2023 Hyper-V-Image vorhandenen RPMs.

Paket	AMI	Hyper-V VHDX
acl	2.3.1	2.3.1
acpid	2.0.32	
alternatives	1.15	1.15
amazon-chrony-config	4.3	
amazon-ec2-net-utils	2.4.1	
amazon-linux-onprem		1.2
amazon-linux-repo-cdn		2023,4.20240319
amazon-linux-repo-s3	2023,4.20240319	
amazon-linux-sb-keys	2023,1	2023,1
amazon-onprem-network		1.2
amazon-rpm-config	228	228
amazon-ssm-agent	3,2,2303,0	3,2,2303,0
at	3.1,23	3.1.23
attr	2.5.1	2.5.1
audit	3.0.6	3.0.6

Paket	AMI	Hyper-V VHDX
audit-libs	3.0.6	3.0.6
aws-cfn-bootstrap	2.0	
awscli-2	2.14.5	2.14.5
basesystem	11	11
bash	5.2.15	5.2.15
bash-completion	2.11	2.11
bc	1,07,1	1,07,1
bind-libs	9,16,48	9,16,48
bind-license	9,16,48	9,16,48
bind-utils	9,16,48	9,16,48
binutils	2,39	2,39
boost-filesystem	1,75,0	1,75,0
boost-system	1,75,0	1,75,0
boost-thread	1,75,0	1,75,0
bzip2	1.0.8	1.0.8
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023,2,64	2023,2,64
c-ares	1.19.0	
checkpolicy	3.4	3.4
chkconfig	1.15	1.15

Paket	AMI	Hyper-V VHDX
chrony	4.3	4.3
cloud-init	22,2,2	22.2.2
cloud-init-cfg-ec2	22.2.2	
cloud-init-cfg-onprem		22.2.2
cloud-utils-growpart	0,31	0,31
coreutils	8,32	8,32
coreutils-common	8,32	8,32
cpio	2,13	2,13
cracklib	2.9.6	2.9.6
cracklib-dicts	2.9.6	2.9.6
crontabs	1.11	1.11
crypto-policies	20220428	20220428
crypto-policies-scripts	20220428	20220428
cryptsetup	2.6.1	2.6.1
cryptsetup-libs	2.6.1	2.6.1
curl-minimal	8,5,0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27
cyrus-sasl-plain	2.1.27	2.1.27
dbus	1.12,28	1.12.28

Paket	AMI	Hyper-V VHDX
dbus-broker	32	32
dbus-common	1.12.28	1.12.28
dbus-libs	1.12.28	1.12.28
device-mapper	1,02,185	1,02,185
device-mapper-libs	1,02,185	1,02,185
diffutils	3.8	3.8
dnf	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2
dnf-plugins-core	4.3.0	4.3.0
dnf-plugin-support-info	1.2	1.2
dnf-utils	4.3.0	4.3.0
dosfstools	4,2	4.2
dracut	055	055
dracut-config-ec2	3.0	
dracut-config-generic	055	055
dwz	0,14	0,14
dyninst	10.2.1	10.2.1

Paket	AMI	Hyper-V VHDX
e2fsprogs	1,46,5	1,46,5
e2fsprogs-libs	1,46,5	1,46,5
ec2-hibinit-agent	1.0.8	
ec2-instance-connect	1.1	
ec2-instance-connect-selinux	1.1	
ec2-utils	2.2.0	
ed	1.14.2	1.14.2
efi-filesystem	5	5
efi-srpm-macros	5	5
efivar	38	38
efivar-libs	38	38
elfutils-debuginfod-client	0.188	0.188
elfutils-default-yama-scope	0.188	0.188
elfutils-libelf	0.188	0.188
elfutils-libs	0.188	0.188
ethtool	5,15	5,15
expat	2.5.0	2.5.0
file	5,39	5,39

Paket	AMI	Hyper-V VHDX
file-libs	5,39	5,39
filesystem	3,14	3,14
findutils	4.8.0	4.8.0
fonts-srpm-macros	2.0.5	2.0.5
fstrm	0.6.1	0.6.1
fuse-libs	2.9.9	2.9.9
gawk	5.1.0	5.1.0
gdbm-libs	1,19	1,19
gdisk	1.0.8	1.0.8
gettext	0,21	0,21
gettext-libs	0,21	0,21
ghc-srpm-macros	1.5.0	1.5.0
glib2	2,74,7	2,74,7
glibc	2,34	2,34
glibc-all-langpacks	2,34	2,34
glibc-common	2,34	2,34
glibc-gconv-extra	2,34	2,34
glibc-locale-source	2,34	2,34
gmp	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7

Paket	AMI	Hyper-V VHDX
gnutls	3.8.0	3.8.0
go-srpm-macros	3.2.0	3.2.0
gpgme	1.15.1	1.15.1
gpm-libs	1.20.7	1,20,7
grep	3.8	3.8
groff-base	1,22,4	1.22,4
grub2-common	2,06	2,06
grub2-efi-x64-ec2	2,06	2,06
grub2-pc		2,06
grub2-pc-modules	2,06	2,06
grub2-tools	2,06	2,06
grub2-tools-minimal	2,06	2,06
grubby	8,40	8,40
gssproxy	0.8.4	0.8.4
gzip	1.12	1.12
hostname	3,23	3,23
hunspell	1.7.0	1.7.0
hunspell-en	0,20140811,1	0,20140811,1
hunspell-en-GB	0,20140811,1	0,20140811,1
hunspell-en-US	0,20140811,1	0,20140811,1

Paket	AMI	Hyper-V VHDX
hunspell-filesystem	1.7.0	1.7.0
hwdata	0,353	0,353
hyperv-daemons		0
hyperv-daemons-license		0
hypervfcopyd		0
hypervkvpd		0
hyperv-tools		0
hypervvssd		0
info	6.7	6.7
inih	49	49
initscripts	10,09	10,09
iproute	5.10.0	5.10.0
iputils	20210202	20210202
irqbalance	1.9.0	1.9.0
jansson	2.14	2.14
jitterentropy	3.4.1	3.4.1
jq	1.7.1	1.7.1
json-c	0,14	0,14
kbd	2.4.0	2.4.0
kbd-misc	2.4.0	2.4.0

Paket	AMI	Hyper-V VHDX
kernel	6.1,79	6.1,79
kernel-livepatch-repo-cdn		2023,4,20240319
kernel-livepatch-repo-s3	2023,4.20240319	
kernel-modules-extra		6.1,79
kernel-modules-extra-common		6.1,79
kernel-srpm-macros	1,0	1,0
kernel-tools	6.1,79	6.1,79
keyutils	1.6.3	1.6.3
keyutils-libs	1.6.3	1.6.3
kmod	29	29
kmod-libs	29	29
kpatch-runtime	0.9.7	0.9.7
krb5-libs	1,21	1,21
less	608	608
libacl	2.3.1	2.3.1
libaio	0,3,111	0,3.111
libarchive	3,5.3	3.5.3
libargon2	20171227	20171227

Paket	AMI	Hyper-V VHDX
libassuan	2,5.5	2.5.5
libattr	2.5.1	2.5.1
libbasicobjects	0.1.1	0.1.1
libblkid	2,37,4	2,37,4
libcap	2,48	2,48
libcap-ng	0.8.2	0.8.2
libcbor	0.7.0	0.7.0
libcollection	0.7.0	0.7.0
libcom_err	1,46,5	1,46,5
libcomps	0,120	0,120
libconfig	1.7.2	1.7.2
libcurl-minimal	8.5.0	8.5.0
libdb	5.3.28	5.3.28
libdhash	0.5.0	
libdnf	0,69,0	0,69,0
libeconf	0,4,0	0,4,0
libedit	3.1	3.1
libev	4,33	4,33
libevent	2.1.12	2.1.12
libfdisk	2,37,4	2,37,4

Paket	AMI	Hyper-V VHDX
libffi	3.4.4	3.4.4
libfido2	1.10.0	1.10.0
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	11.4.1
libgpg-error	1,42	1,42
libibverbs	48,0	48,0
libidn2	2.3.2	2.3.2
libini_config	1.3.1	1.3.1
libkcapi	1.4.0	1.4.0
libkcapi-hmaccalc	1.4.0	1.4.0
libldb	2.6.2	
libmaxminddb	1.5.2	1.5.2
libmetalink	0.1.3	0.1.3
libmnl	1.0.4	1.0.4
libmodulemd	2.13.0	2.13.0
libmount	2,37,4	2,37,4
libnfsidmap	2.5.4	2.5.4
libnghttp2	1,57,0	1,57,0
libnl3	3.5.0	3.5.0

Paket	AMI	Hyper-V VHDX
libpath_utils	0.2.1	0.2.1
libpcap	1.10.1	1.10.1
libpipeline	1.5.3	1.5.3
libpkgconf	1.8.0	1.8.0
libpsl	0,21,1	0,21,1
libpwquality	1.4.4	1.4.4
libref_array	0,15	0.1.5
librepo	1,14,5	1.14,5
libreport-filesystem	2.15,2	2.15,2
libseccomp	2.5.3	2.5.3
libselinux	3.4	3.4
libselinux-utils	3.4	3.4
libsemanage	3.4	3.4
libsepol	3.4	3.4
libsigsegv	2,13	2,13
libsmartcols	2,37,4	2,37,4
libsolv	0,7.22	0.7.22
libss	1,46,5	1,46,5
libsss_certmap	2.9.4	
libsss_idmap	2.9.4	2.9.4

Paket	AMI	Hyper-V VHDX
libsss_nss_idmap	2.9.4	2.9.4
libsss_sudo	2.9.4	
libstdc++	11.4.1	11.4.1
libstoragemgmt	1.9.4	1.9.4
libtalloc	2.3.4	
libtasn1	4.19.0	4.19,0
libtdb	1.4.7	
libtevent	0.13.0	
libtextstyle	0,21	0,21
libtirpc	1.3.3	1.3.3
libunistring	0,9,10	0.9.10
libuser	0,63	0,63
libutempter	1.2.1	1.2.1
libuuid	2,37,4	2,37,4
libuv	1.47.0	1.47.0
libverto	0.3.2	0.3.2
libverto-libev	0.3.2	0.3.2
libxcrypt	4.4.33	4.4.33
libxml2	2.10,4	2.10.4
libyaml	0.2.5	0,2,5

Paket	AMI	Hyper-V VHDX
libzstd	1.5.5	1.5.5
lm_sensors-libs	3.6.0	3.6.0
lmdb-libs	0.9.29	0,9,29
logrotate	3,20,1	3,20,1
lsof	4,94,0	4,94,0
lua-libs	5.4.4	5.4.4
lua-srpm-macros	1	1
lz4-libs	1.9.4	1.9.4
man-db	2.9.3	2.9.3
man-pages	5,10	5,10
microcode_ctl	2.1	2.1
mpfr	4.1.0	4.1.0
nano	5,8	5,8
ncurses	6.2	6.2
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
nettle	3.8	3.8
net-tools	2,0	2.0
newt	0,52,21	0,52,21
nfs-utils	2,5.4	2.5.4

Paket	AMI	Hyper-V VHDX
npth	1,6	1,6
nspr	4,35,0	4,35,0
nss	3,90,0	3,90,0
nss-softokn	3,90,0	3,90,0
nss-softokn-freebl	3,90,0	3,90,0
nss-sysinit	3,90,0	3,90,0
nss-util	3,90,0	3,90,0
ntsysv	1.15	1.15
numactl-libs	2.0.14	2.0.14
ocaml-srpm-macros	6	6
oniguruma	6.9.7.1	6.9.7.1
openblas-srpm-macros	2	2
openldap	2.4.57	2,4,57
openssh	8,7p1	8,7 p1
openssh-clients	8,7 p1	8,7 p1
openssh-server	8,7 p1	8,7 p1
openssl	3.0.8	3.0.8
openssl-libs	3.0.8	3.0.8
openssl-pkcs11	0.4.12	0.4.12
os-prober	1,77	1,77

Paket	AMI	Hyper-V VHDX
p11-kit	0,24,1	0,24,1
p11-kit-trust	0,24,1	0,24,1
package-notes-srpm-macros	0.4	0.4
pam	1.5.1	1.5.1
parted	3.4	3.4
passwd	0,80	0,80
pciutils	3.7.0	3.7.0
pciutils-libs	3.7.0	3.7.0
pcre2	10,40	10,40
pcre2-syntax	10,40	10,40
perl-Carp	1,50	1,50
perl-Class-Struct	0,66	0,66
perl-constant	1,33	1,33
perl-DynaLoader	1,47	1,47
perl-Encode	3,15	3,15
perl-Errno	1,30	1,30
perl-Exporter	5,74	5,74
perl-Fcntl	1.13	1.13
perl-File-Basename	2,85	2,85
perl-File-Path	2,18	2,18

Paket	AMI	Hyper-V VHDX
perl-File-stat	1,09	1,09
perl-File-Temp	0,231,100	0,231,100
perl-Getopt-Long	2,52	2,52
perl-Getopt-Std	1.12	1.12
perl-HTTP-Tiny	0,078	0,078
perl-if	0,60,800	0,60,800
perl-interpreter	5,32,1	5,32,1
perl-IO	1,43	1,43
perl-IPC-Open3	1,21	1,21
perl-libs	5,32,1	5,32,1
perl-MIME-Base64	3,16	3,16
perl-mro	1,23	1,23
perl-overload	1,31	1,31
perl-overloading	0,02	0,02
perl-parent	0,238	0,238
perl-PathTools	3,78	3,78
perl-Pod-Escapes	1,07	1,07
perl-podlators	4,14	4,14
perl-Pod-Perldoc	3,28,01	3,28,01
perl-Pod-Simple	3,42	3,42

Paket	AMI	Hyper-V VHDX
perl-Pod-Usage	2,01	2,01
perl-POSIX	1,94	1,94
perl-Scalar-List-Utils	1,56	1,56
perl-SelectSaver	1.02	1.02
perl-Socket	2,032	2,032
perl-srpm-macros	1	1
perl-Storable	3,21	3,21
perl-subs	1,03	1,03
perl-Symbol	1,08	1,08
perl-Term-ANSIColor	5,01	5,01
perl-Term-Cap	1,17	1,17
perl-Text-ParseWords	3,30	3,30
perl-Text-Tabs+Wrap	2021,0726	2021,0726
perl-Time-Local	1,300	1,300
perl-vars	1,05	1,05
pkgconf	1.8.0	1.8.0
pkgconf-m4	1.8.0	1.8.0
pkgconf-pkg-config	1.8.0	1.8.0
policycoreutils	3.4	3.4

Paket	AMI	Hyper-V VHDX
policycoreutils-python-utils	3.4	
popt	1,18	1,18
procps-ng	3.3,17	3.3.17
protobuf-c	1.4.1	1.4.1
psacct	6.6.4	6.6.4
psmisc	23,4	23,4
publicsuffix-list-dafsa	20240212	20240212
python3	3,9,16	3.9,16
python3-attrs	20,3,0	20.3,0
python3-audit	3.0.6	3.0.6
python3-awscrt	0.19,19	0,19,19
python3-babel	2.9.1	2.9.1
python3-cffi	1.14,5	1.14,5
python3-chardet	4.0.0	4.0.0
python3-colorama	0.4.4	0.4.4
python3-configobj	5.0.6	5.0.6
python3-cryptography	36,1	36,1
python3-daemon	2.3.0	
python3-dateutil	2.8.1	2.8.1

Paket	AMI	Hyper-V VHDX
python3-dbus	1.2.18	1.2.18
python3-distro	1.5.0	1.5.0
python3-dnf	4.14.0	4.14.0
python3-dnf-plugins-core	4.3.0	4.3.0
python3-docutils	0,16	0,16
python3-gpg	1.15.1	1.15.1
python3-hawkey	0,69,0	0,69,0
python3-idna	(2.10)	(2.10)
python3-jinja2	2.11,3	2.11.3
python3-jmespath	0.10.0	0.10.0
python3-jsonpatch	1,21	1,21
python3-jsonpointer	2,0	2.0
python3-jsonschema	3.2.0	3.2.0
python3-libcomps	0,120	0,120
python3-libdnf	0,69,0	0,69,0
python3-libs	3,9,16	3.9,16
python3-libselinux	3.4	3.4
python3-libsemanage	3.4	3.4
python3-libstorage mgmt	1.9.4	1.9.4

Paket	AMI	Hyper-V VHDX
python3-lockfile	0.12.2	
python3-markupsafe	1.1.1	1.1.1
python3-netifaces	0.10.6	0.10.6
python3-oauthlib	3.0.2	3.0.2
python3-pip-wheel	21.3.1	21.3.1
python3-ply	3.11	3,11
python3-policycoreutils	3.4	3.4
python3-prettytable	0.7.2	0.7.2
python3-prompt-toolkit	3.0,24	3,0,24
python3-pycparser	2,20	2,20
python3-pyrsistent	0,17,3	0,17,3
python3-pyserial	3.4	3.4
python3-pysocks	1.7.1	1.7.1
python3-pytz	2022.7.1	2022,7.1
python3-pyyaml	5.4.1	5.4.1
python3-requests	2,25,1	2,25,1
python3-rpm	4.16.1,3	4.16.1.3
python3-ruamel-yaml	0.16.6	0,16,6

Paket	AMI	Hyper-V VHDX
python3-ruamel-yaml-clib	0.1.2	0.1.2
python3-setools	4.4.1	4.4.1
python3-setuptools	59,6,0	59,6,0
python3-setuptools-wheel	59,6,0	59,6,0
python3-six	1.15.0	1.15.0
python3-systemd	235	235
python3-urllib3	1,25,10	1,25,10
python3-wcwidth	0,2,5	0,2,5
python-chevron	0.13.1	
python-srpm-macros	3.9	3.9
quota	4,06	4,06
quota-nls	4,06	4,06
readline	8,1	8.1
rng-tools	6,14	6,14
rootfiles	8.1	8.1
rpcbind	1.2.6	1.2.6
rpm	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3
rpm-libs	4.16.1.3	4.16.1.3

Paket	AMI	Hyper-V VHDX
rpm-plugin-selinux	4.16.1.3	4.16.1.3
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3
rpm-sign-libs	4.16.1.3	4.16.1.3
rsync	3.2.6	3.2.6
rust-srpm-macros	21	21
sbsigntools	0.9.4	0.9.4
screen	4.8.0	4.8.0
sed	4.8	4,8
selinux-policy	37,22	37,22
selinux-policy-targeted	37,22	37,22
setup	2,13,7	2.13,7
shadow-utils	4,9 bis 4,9	4,9 bis 4,9
slang	2.3.2	2.3.2
sqlite-libs	3,40,0	3,40,0
sssd-client	2.9.4	2.9.4
sssd-common	2.9.4	
sssd-kcm	2.9.4	
sssd-nfs-idmap	2.9.4	
strace	5,16	5,16

Paket	AMI	Hyper-V VHDX
sudo	1.9,14	1.9.14
sysctl-defaults	1,0	1,0
sysstat	12.5.6	12.5.6
systemd	252,16	252,16
systemd-libs	252,16	252,16
systemd-networkd	252,16	252,16
systemd-pam	252,16	252,16
systemd-resolved	252,16	252,16
systemd-udev	252,16	252,16
system-release	2023,4,20240319	2023,4.20240319
systemtap-runtime	4.8	4,8
tar	1,34	1,34
tbb	2020,3	2020,3
tcpdump	4,99,1	4,99,1
tcsh	6,24,07	6,24,07
time	1.9	1.9
traceroute	2.1.3	2.1.3
tzdata	2024a	2024a
unzip	6.0	6.0
update-motd	2.2	2.2

Paket	AMI	Hyper-V VHDX
userspace-rcu	0.12.1	0.12.1
util-linux	2,37,4	2,37,4
util-linux-core	2,37,4	2,37,4
vim-common	9,0,2153	9,0,2153
vim-data	9,0,2153	9,0,2153
vim-enhanced	9,0,2153	9,0,2153
vim-filesystem	9,0,2153	9,0,2153
vim-minimal	9,0,2153	9,0,2153
wget	1,21,3	1.21.3
which	2,21	2,21
words	3.0	3.0
xfsdump	3.1.11	3.1.11
xfsprogs	5.18.0	5.18,0
xxd	9,0,2153	9,0,2153
xxhash-libs	0.8.0	0.8.0
xz	5.2,5	5.2.5
xz-libs	5.2.5	5.2.5
yum	4.14.0	4.14.0
zip	3.0	3.0
zlib	1.2.11	1.2.11

Paket	AMI	Hyper-V VHDX
`zram-generator`	1.1.2	
`zram-generator-defaults`	1.1.2	
`zstd`	1.5.5	1.5.5

Aktualisierung von AL2023

Es ist wichtig, über die AL2023-Versionen auf dem Laufenden zu bleiben, damit Sie von Sicherheitsupdates und neuen Funktionen profitieren können. Mit AL2023 können Sie die Konsistenz zwischen Paketversionen und Updates in Ihrer gesamten Umgebung mithilfe von Verwendung deterministischer Upgrades über ein versioniertes Repository auf AL2023 sicherstellen.

Themen

- Erhalten Sie Benachrichtigungen über neue Updates
- Paket- und Betriebssystemupdates in AL2023 verwalten
- Verwendung deterministischer Upgrades über ein versioniertes Repository auf AL2023
- Kernel-Live-Patching auf AL2023

Erhalten Sie Benachrichtigungen über neue Updates

Sie können Benachrichtigungen erhalten, wenn ein neues AL2023-AMI veröffentlicht wird. Benachrichtigungen werden mithilfe von Amazon SNS mit dem folgenden Thema veröffentlicht.

```
arn:aws:sns:us-east-1:137112412989:amazon-linux-2023-ami-updates
```

Nachrichten werden hier veröffentlicht, wenn ein neues AL2023-AMI veröffentlicht wurde. Die Version des AMI wird in der Nachricht angegeben.

Diese Nachrichten können mit verschiedenen Methoden empfangen werden. Wir empfehlen Ihnen, die folgende Methode zu verwenden.

1. Öffnen Sie die Amazon-SNS-Konsole.
2. Ändern Sie in der Navigationsleiste, falls erforderlich, AWS-Region den Wert auf USA Ost (Nord-Virginia). Sie müssen die Region auswählen, in der die von Ihnen abonnierte SNS-Benachrichtigung erstellt wird.
3. Wählen Sie im Navigationsbereich Abonnements und Abonnement erstellen aus.
4. Führen Sie im Dialogfeld Create subscription die folgenden Schritte aus:
   a. Kopieren Sie für das Thema ARN den folgenden Amazon-Ressourcennamen (ARN) und fügen Sie ihn ein:**`arn:aws:sns:us-east-1:137112412989:amazon-linux-2023-ami-updates`**.

b. Wählen Sie unter Protocol (Protokoll) die Option Email (E-Mail) aus.

c. Geben Sie unter Endpoint (Endpunkt) eine E-Mail-Adresse ein, um die Benachrichtigungen zu empfangen.

d. Wählen Sie Create subscription.

5. Sie erhalten eine Bestätigungs-E-Mail mit dem Betreff „AWS Benachrichtigung — Abonnementbestätigung“. Öffnen Sie die E-Mail und wählen Sie Confirm subscription aus, um Ihr Abonnement abzuschließen.

Paket- und Betriebssystemupdates in AL2023 verwalten

Im Gegensatz zu früheren Versionen von Amazon Linux sind AL2023-AMIs an eine bestimmte Version des Amazon Linux-Repositorys gebunden. Sie müssen die DNF-Konfiguration aktualisieren, um Sicherheits-Updates und Bugfixes auf eine AL2023-Instance anzuwenden. Alternativ können Sie eine neuere AL2023-Instance starten.

In diesem Abschnitt wird die Verwaltung von DNF-Paketen und Repositorys auf einer laufenden Instance beschrieben. Außerdem wird beschrieben, wie Sie DNF mithilfe eines Benutzerdatenskripts konfigurieren, um beim Start das neueste verfügbare Amazon-Linux-Repository zu aktivieren. Weitere Informationen finden Sie in der DNF-Befehlsreferenz.

Themen

- Prüfen auf verfügbare Paket-Updates
- Anwenden von Sicherheits-Updates mithilfe von DNF- und Repository-Versionen
- Automatischer Neustart des Dienstes nach (Sicherheits-) Updates
- Starten einer Instance mit aktivierter neuester Repository-Version
- Abrufen von Paketunterstützunginformationen
- Prüfen auf neuere Repository-Versionen
- Hinzufügen, aktivieren oder deaktivieren neuer Repositorys
- Hinzufügen von Repositorys mit cloud-init

Prüfen auf verfügbare Paket-Updates

Mit dem `dnf check-update`-Befehl können Sie jederzeit nach Updates für Ihr System suchen. Für AL2023 empfehlen wir, dass Sie die `--releasever=`*`version-number`*-Option zum Befehl hinzufügen.

Wenn Sie diese Option hinzufügen, wird DNF auch nach Updates für eine spätere Version des Repositorys suchen. Verwenden Sie beispielsweise nach der Ausführung des `dnf check-update`-Befehls die zuletzt zurückgegebene Version als Wert für *`version-number`*.

Wenn die Instance so aktualisiert wird, dass sie die neueste Version des Repositorys verwendet, enthält die Ausgabe eine Liste aller zu aktualisierenden Pakete.

> **Note**
>
> Wenn Sie die Release-Version nicht mit dem optionalen Flag für den `dnf check-update`-Befehl angeben, wird nur die aktuell konfigurierte Repository-Version überprüft. Das bedeutet, dass Pakete der neueren Repository-Version nicht überprüft werden.

```
$ sudo dnf check-update --releasever=2023.0.20230210
Last metadata expiration check: 0:06:13 ago on Mon 13 Feb 2023 10:39:32 PM UTC.

bind-libs.x86_64                        32:9.16.27-1.amzn2023          amazonlinux
bind-license.noarch                     32:9.16.27-1.amzn2023          amazonlinux
bind-utils.x86_64                       32:9.16.27-1.amzn2023          amazonlinux
cloud-init.noarch                       22.2.2-1.amzn2023.1.4          amazonlinux
dnf.noarch                              4.12.0-2.amzn2023.0.1          amazonlinux
dnf-data.noarch                         4.12.0-2.amzn2023.0.1          amazonlinux
dracut.x86_64                           055-6.amzn2023.0.4             amazonlinux
dracut-config-generic.x86_64            055-6.amzn2023.0.4             amazonlinux
glib2.x86_64                            2.73.2-678.amzn2023            amazonlinux
gmp.x86_64                              1:6.2.1-2.amzn2023             amazonlinux
grep.x86_64                             3.8-1.amzn2023.0.1             amazonlinux
kpatch-runtime.noarch                   0.9.4-7.amzn2023               amazonlinux
libgcc.x86_64                           11.3.1-2.amzn2023.0.6          amazonlinux
libgomp.x86_64                          11.3.1-2.amzn2023.0.6          amazonlinux
libpkgconf.x86_64                       1.7.3-7.amzn2023.0.1           amazonlinux
libstdc++.x86_64                        11.3.1-2.amzn2023.0.6          amazonlinux
lz4-libs.x86_64                         1.9.4-1.amzn2023               amazonlinux
pkgconf.x86_64                          1.7.3-7.amzn2023.0.1           amazonlinux
```

```
pkgconf-m4.noarch                    1.7.3-7.amzn2023.0.1          amazonlinux
pkgconf-pkg-config.x86_64            1.7.3-7.amzn2023.0.1          amazonlinux
python3-dnf.noarch                   4.12.0-2.amzn2023.0.1         amazonlinux
python3-rpm.x86_64                   4.16.1.3-12.amzn2023.0.2      amazonlinux
rpm.x86_64                           4.16.1.3-12.amzn2023.0.2      amazonlinux
rpm-build-libs.x86_64                4.16.1.3-12.amzn2023.0.2      amazonlinux
rpm-libs.x86_64                      4.16.1.3-12.amzn2023.0.2      amazonlinux
rpm-plugin-selinux.x86_64            4.16.1.3-12.amzn2023.0.2      amazonlinux
rpm-plugin-systemd-inhibit.x86_64    4.16.1.3-12.amzn2023.0.2      amazonlinux
rpm-sign-libs.x86_64                 4.16.1.3-12.amzn2023.0.2      amazonlinux
slang.x86_64                         2.3.2-9.amzn2023.0.1          amazonlinux
system-release.noarch                2023.0.20230210-0.amzn2023    amazonlinux
systemd.x86_64                       250.8-1.amzn2023.0.1          amazonlinux
systemd-libs.x86_64                  250.8-1.amzn2023.0.1          amazonlinux
systemd-networkd.x86_64              250.8-1.amzn2023.0.1          amazonlinux
systemd-pam.x86_64                   250.8-1.amzn2023.0.1          amazonlinux
systemd-resolved.x86_64              250.8-1.amzn2023.0.1          amazonlinux
systemd-udev.x86_64                  250.8-1.amzn2023.0.1          amazonlinux
vim-common.x86_64                    2:9.0.327-1.amzn2023.0.1      amazonlinux
vim-data.noarch                      2:9.0.327-1.amzn2023.0.1      amazonlinux
vim-enhanced.x86_64                  2:9.0.327-1.amzn2023.0.1      amazonlinux
vim-filesystem.noarch                2:9.0.327-1.amzn2023.0.1      amazonlinux
vim-minimal.x86_64                   2:9.0.327-1.amzn2023.0.1      amazonlinux
wget.x86_64                          1.21.3-1.amzn2023             amazonlinux
yum.noarch                           4.12.0-2.amzn2023.0.1         amazonlinux
```

Wenn neuere Pakete verfügbar sind, lautet der Rückgabecode für diesen Befehl „100“. Wenn keine neueren Pakete verfügbar sind, lautet der Rückgabecode für diesen Befehl „0“. Darüber hinaus wird auch eine Liste aller zu aktualisierenden Pakete ausgegeben.

Anwenden von Sicherheits-Updates mithilfe von DNF- und Repository-Versionen

Neue Paket- und Sicherheits-Updates werden nur für neue Repository-Versionen bereitgestellt. Für Instances, die Sie aus früheren AL2023-AMI-Versionen gestartet haben, müssen Sie die Repository-Version aktualisieren, bevor Sie Sicherheits-Updates installieren können. Der `dnf check-release-update`-Befehl enthält ein Beispiel für einen Update-Befehl, der alle auf dem System installierten Pakete auf Versionen in einem neueren Repository aktualisiert.

```
$ sudo dnf update --releasever=2023.0.20230210
Last metadata expiration check: 0:01:40 ago on Mon 13 Feb 2023 10:39:32 PM UTC.
```

```
Dependencies resolved.
================================================================================
 Package                     Arch   Version                     Repository   Size
================================================================================
Upgrading:
 bind-libs                   x86_64 32:9.16.27-1.amzn2023      amazonlinux 1.2 M
 bind-license                noarch 32:9.16.27-1.amzn2023      amazonlinux  16 k
 bind-utils                  x86_64 32:9.16.27-1.amzn2023      amazonlinux 202 k
 cloud-init                  noarch 22.2.2-1.amzn2023.1.4      amazonlinux 1.1 M
 dnf                         noarch 4.12.0-2.amzn2023.0.1      amazonlinux 454 k
 dnf-data                    noarch 4.12.0-2.amzn2023.0.1      amazonlinux  42 k
 dracut                      x86_64 055-6.amzn2023.0.4         amazonlinux 345 k
 dracut-config-generic       x86_64 055-6.amzn2023.0.4         amazonlinux 8.5 k
 glib2                       x86_64 2.73.2-678.amzn2023        amazonlinux 2.7 M
 gmp                         x86_64 1:6.2.1-2.amzn2023         amazonlinux 324 k
 grep                        x86_64 3.8-1.amzn2023.0.1         amazonlinux 316 k
 kpatch-runtime              noarch 0.9.4-7.amzn2023           amazonlinux  30 k
 libgcc                      x86_64 11.3.1-2.amzn2023.0.6      amazonlinux 121 k
 libgomp                     x86_64 11.3.1-2.amzn2023.0.6      amazonlinux 296 k
 libpkgconf                  x86_64 1.7.3-7.amzn2023.0.1       amazonlinux  37 k
 libstdc++                   x86_64 11.3.1-2.amzn2023.0.6      amazonlinux 758 k
 lz4-libs                    x86_64 1.9.4-1.amzn2023           amazonlinux  81 k
 pkgconf                     x86_64 1.7.3-7.amzn2023.0.1       amazonlinux  41 k
 pkgconf-m4                  noarch 1.7.3-7.amzn2023.0.1       amazonlinux  15 k
 pkgconf-pkg-config          x86_64 1.7.3-7.amzn2023.0.1       amazonlinux  11 k
 python3-dnf                 noarch 4.12.0-2.amzn2023.0.1      amazonlinux 415 k
 python3-rpm                 x86_64 4.16.1.3-12.amzn2023.0.2   amazonlinux  89 k
 rpm                         x86_64 4.16.1.3-12.amzn2023.0.2   amazonlinux 487 k
 rpm-build-libs              x86_64 4.16.1.3-12.amzn2023.0.2   amazonlinux  92 k
 rpm-libs                    x86_64 4.16.1.3-12.amzn2023.0.2   amazonlinux 311 k
 rpm-plugin-selinux          x86_64 4.16.1.3-12.amzn2023.0.2   amazonlinux  18 k
 rpm-plugin-systemd-inhibit x86_64 4.16.1.3-12.amzn2023.0.2   amazonlinux  19 k
 rpm-sign-libs               x86_64 4.16.1.3-12.amzn2023.0.2   amazonlinux  22 k
 slang                       x86_64 2.3.2-9.amzn2023.0.1       amazonlinux 410 k
 system-release              noarch 2023.0.20230210-0.amzn2023 amazonlinux  25 k
 systemd                     x86_64 250.8-1.amzn2023.0.1       amazonlinux 4.2 M
 systemd-libs                x86_64 250.8-1.amzn2023.0.1       amazonlinux 615 k
 systemd-networkd            x86_64 250.8-1.amzn2023.0.1       amazonlinux 614 k
 systemd-pam                 x86_64 250.8-1.amzn2023.0.1       amazonlinux 335 k
 systemd-resolved            x86_64 250.8-1.amzn2023.0.1       amazonlinux 277 k
 systemd-udev                x86_64 250.8-1.amzn2023.0.1       amazonlinux 1.9 M
 vim-common                  x86_64 2:9.0.327-1.amzn2023.0.1   amazonlinux 7.2 M
 vim-data                    noarch 2:9.0.327-1.amzn2023.0.1   amazonlinux  27 k
 vim-enhanced                x86_64 2:9.0.327-1.amzn2023.0.1   amazonlinux 1.8 M
```

```
 vim-filesystem             noarch 2:9.0.327-1.amzn2023.0.1   amazonlinux  21 k
 vim-minimal                x86_64 2:9.0.327-1.amzn2023.0.1   amazonlinux 764 k
 wget                       x86_64 1.21.3-1.amzn2023          amazonlinux 813 k
 yum                        noarch 4.12.0-2.amzn2023.0.1      amazonlinux  39 k

Transaction Summary
================================================================================
Upgrade  43 Packages
...
```

Sie können die `--security`-Option hinzufügen, wenn die Pakete nur mit Sicherheitsfunktionen aktualisiert werden sollen.

```
$ sudo dnf update --releasever=2023.0.20230210 --security
Amazon Linux 2023 repository                    18 MB/s |  11 MB     00:00
Last metadata expiration check: 0:00:02 ago on Mon 13 Feb 2023 10:39:32 PM UTC.
Dependencies resolved.
================================================================================
 Package           Arch      Version                         Repository      Size
================================================================================
Upgrading:
 bind-libs         x86_64    32:9.16.27-1.amzn2023         amazonlinux    1.2 M
 bind-license      noarch    32:9.16.27-1.amzn2023         amazonlinux     16 k
 bind-utils        x86_64    32:9.16.27-1.amzn2023         amazonlinux    202 k
 gmp               x86_64    1:6.2.1-2.amzn2023            amazonlinux    324 k
 lz4-libs          x86_64    1.9.4-1.amzn2023              amazonlinux     81 k
 vim-common        x86_64    2:9.0.327-1.amzn2023.0.1      amazonlinux    7.2 M
 vim-data          noarch    2:9.0.327-1.amzn2023.0.1      amazonlinux     27 k
 vim-enhanced      x86_64    2:9.0.327-1.amzn2023.0.1      amazonlinux    1.8 M
 vim-filesystem    noarch    2:9.0.327-1.amzn2023.0.1      amazonlinux     21 k
 vim-minimal       x86_64    2:9.0.327-1.amzn2023.0.1      amazonlinux    764 k
 wget              x86_64    1.21.3-1.amzn2023             amazonlinux    813 k

Transaction Summary
================================================================================
Upgrade  11 Packages
...
```

Mit einer der folgenden Methoden können Sie Ihre aktuellen AL2023-Paketversionen feststellen:

- Führen Sie den Befehl `dnf check-update` aus.

- Abonnieren Sie das SNS-Thema zur Aktualisierung des Amazon-Linux-Repositorys (`arn:aws:sns:us-east-1:137112412989:amazon-linux-2023-ami-updates`). Weitere Informationen finden Sie unter Amazon SNS-Thema abonnieren im Amazon Simple Notification Service-Entwicklerhandbuch.
- Lesen Sie regelmäßig die Versionshinweise zu AL2023.

> **Important**
>
> Wenn Sie Sicherheitsupdates auf eine laufende Instance anwenden, stellen Sie sicher, dass DNF auf die neueste Repository-Version verweist.

Automatischer Neustart des Dienstes nach (Sicherheits-) Updates

Amazon Linux wird jetzt mit dem Smart-Restart-Paket ausgeliefert. `Smart-restart`startet die Systemd-Dienste bei Systemupdates jedes Mal neu, wenn ein Paket mit dem System-Paketmanager installiert oder gelöscht wird. Dies tritt jedes Mal auf, wenn es ausgeführt `dnf (update|upgrade|downgrade)` wird.

`Smart-restart`verwendet das `needs-restarting` Paket von `dnf-utils` und einen benutzerdefinierten Denylisting-Mechanismus, um festzustellen, welche Dienste neu gestartet werden müssen und ob ein Systemneustart empfohlen wird. Wenn ein Systemneustart empfohlen wird, wird eine Datei mit Hinweisen zum Neustart generiert (). `/run/smart-restart/reboot-hint-marker`

So installieren Sie **`smart-restart`**

Führen Sie den folgenden DNF Befehl aus (wie bei jedem anderen Paket).

```
$ sudo dnf install smart-restart
```

Nach der Installation lösen die nachfolgenden Transaktionen die `smart-restart` Logik aus.

Liste ablehnen

`Smart-restart`kann angewiesen werden, den Neustart bestimmter Dienste zu blockieren. Die blockierten Dienste tragen nicht zur Entscheidung bei, ob ein Neustart erforderlich ist. Um zusätzliche

Dienste zu blockieren, fügen Sie eine Datei mit dem Suffix `-denylist` in hinzu, `/etc/smart-restart-conf.d/` wie im folgenden Beispiel gezeigt.

```
$ cat /etc/smart-restart-conf.d/custom-denylist
# Some comments
myservice.service
```

> **Note**
>
> Bei der Entscheidung, ob ein Neustart erforderlich ist, werden alle `*-denylist` Dateien gelesen und ausgewertet.

Benutzerdefinierte Hooks

Zusätzlich zur Denylisting `smart-restart` bietet es einen Mechanismus zum Ausführen benutzerdefinierter Skripts vor und nach den Versuchen, den Dienst neu zu starten. Die benutzerdefinierten Skripts können verwendet werden, um Vorbereitungsschritte manuell durchzuführen oder um andere Komponenten über einen verbleibenden oder abgeschlossenen Neustart zu informieren.

Alle Skripten `/etc/smart-restart-conf.d/` mit dem Suffix `-pre-restart` oder `-post-restart` werden ausgeführt. Wenn die Reihenfolge wichtig ist, stellen Sie allen Skripten eine Zahl voran, um die Ausführungsreihenfolge sicherzustellen, wie im folgenden Beispiel gezeigt.

```
$ ls /etc/smart-restart-conf.d/*-pre-restart
001-my-script-pre-restart
002-some-other-script-pre-restart
```

Starten einer Instance mit aktivierter neuester Repository-Version

Sie können DNF-Befehle zu einem Benutzerdatenskript hinzufügen, um zu steuern, welche RPM-Pakete beim Start auf einem Amazon-Linux-AMI installiert werden sollen. Im folgenden Beispiel wird ein Benutzerdatenskript verwendet, um sicherzustellen, dass auf jeder mit dem Benutzerdatenskript gestarteten Instance dieselben Paket-Updates installiert werden.

```
#!/bin/bash
dnf update --releasever=2023.0.20230210
# Additional setup and install commands below
```

```
dnf install httpd php7.4 mysql80
```

Dieses Skript muss vom Superuser (Root) ausgeführt werden. Führen Sie dazu den folgenden Befehl aus.

```
$ sudo sh -c "bash nameofscript.sh"
```

Weitere Informationen finden Sie unter Benutzerdaten und Shell-Skripts im Amazon EC2 EC2-Benutzerhandbuch.

> **Note**
>
> Anstatt ein Benutzerdatenskript zu verwenden, starten Sie das neueste Amazon-Linux-AMI oder ein benutzerdefiniertes AMI, das auf dem Amazon-Linux-AMI basiert. Im neuesten Amazon-Linux-AMI sind alle erforderlichen Updates installiert und es ist so konfiguriert, dass es auf eine bestimmte Repository-Version verweist.

Abrufen von Paketunterstützunginformationen

AL2023 beinhaltet viele verschiedene Open-Source-Softwareprojekte. Jedes dieser Projekte wird unabhängig von Amazon Linux verwaltet und hat unterschiedliche Versionen und end-of-support Zeitpläne. Das DNF-`supportinfo`-Plugin stellt Ihnen Amazon-Linux-spezifische Informationen in Form von Metadaten zu einem Paket bereit. Im folgenden Beispiel gibt der **`dnf supportinfo`**-Befehl Metadaten für das `glibc`-Paket zurück.

```
$ sudo dnf supportinfo --pkg glibc
Last metadata expiration check: 0:07:56 ago on Wed Mar  1 23:21:49 2023.
Name                 : glibc
Version              : 2.34-52.amzn2023.0.2
State                : installed
Support Status       : supported
Support Periods      : from 2023-03-15      : supported
                     : from 2028-03-15      : unsupported
Support Statement    : Amazon Linux 2023 End Of Life
Link                 : https://aws.amazon.com/amazon-linux-ami/faqs/
Other Info           : This is the support statement for AL2023. The
     ...: end of life of Amazon Linux 2023 would be March 2028.
     ...: From this point, the Amazon Linux 2023 packages (listed
     ...: below) will no longer, receive any updates from AWS.
```

Prüfen auf neuere Repository-Versionen

In einer AL2023-Instance können Sie mit dem DNF-Hilfsprogramm Repositorys verwalten und aktualisierte RPM-Pakete anwenden. Diese Pakete sind in den Amazon-Linux-Repositorys verfügbar. Mit dem DNF-Befehl `dnf check-release-update` können Sie nach neuen Versionen des DNF-Repositorys suchen.

```
$ sudo dnf check-release-update
WARNING:
  A newer release of "Amazon Linux" is available.

  Available Versions:

  Version 2023.0.20230210:
    Run the following command to update to 2023.0.20230210:

      dnf update --releasever=2023.0.20230210

    Release notes:
     https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes.html
```

Es wird eine vollständige Liste aller neueren Versionen der verfügbaren DNF-Repositorys zurückgegeben. Wenn nichts zurückgegeben wird, bedeutet dies, dass DNF aktuell für die Nutzung der neuesten verfügbaren Version konfiguriert ist. Die Version des aktuell installierten `system-release`-Pakets legt die `releasever`-DNF-Variable fest. Mit folgendem Befehl können Sie die aktuelle Repository-Version abfragen.

```
$ rpm -q system-release --qf "%{VERSION}\n"
```

Wenn Sie DNF-Pakettransaktionen ausführen (z. B. Installieren, Aktualisieren oder Entfernen), werden Sie mit einer Warnmeldung über neue Repository-Versionen informiert. Wenn Sie beispielsweise das `httpd`-Paket auf einer Instance installieren, die aus einer älteren Version von AL2023 heraus gestartet wurde, wird Folgendes zurückgegeben.

```
$ sudo dnf install httpd -y
Last metadata expiration check: 0:16:52 ago on Wed Mar  1 23:21:49 2023.
Dependencies resolved.
================================================================
 Package            Arch   Version                Repository   Size
================================================================
```

```
Installing:
 httpd              x86_64 2.4.54-3.amzn2023.0.4  amazonlinux  46 k
Installing dependencies:
 apr                x86_64 1.7.2-2.amzn2023.0.2   amazonlinux 129 k
 apr-util           x86_64 1.6.3-1.amzn2023.0.1   amazonlinux  98 k
 generic-logos-httpd
                    noarch 18.0.0-12.amzn2023.0.3 amazonlinux  19 k
 httpd-core         x86_64 2.4.54-3.amzn2023.0.4  amazonlinux 1.3 M
 httpd-filesystem   noarch 2.4.54-3.amzn2023.0.4  amazonlinux  13 k
 httpd-tools        x86_64 2.4.54-3.amzn2023.0.4  amazonlinux  80 k
 libbrotli          x86_64 1.0.9-4.amzn2023.0.2   amazonlinux 315 k
 mailcap            noarch 2.1.49-3.amzn2023.0.3  amazonlinux  33 k
Installing weak dependencies:
 apr-util-openssl   x86_64 1.6.3-1.amzn2023.0.1   amazonlinux  17 k
 mod_http2          x86_64 1.15.24-1.amzn2023.0.3 amazonlinux 152 k
 mod_lua            x86_64 2.4.54-3.amzn2023.0.4  amazonlinux  60 k

Transaction Summary
================================================================================
Install  12 Packages

Total download size: 2.3 M
Installed size: 6.8 M
Downloading Packages:
(1/12): apr-util-openssl-1.6.3-1.am 212 kB/s |  17 kB     00:00
(2/12): apr-1.7.2-2.amzn2023.0.2.x8 1.1 MB/s | 129 kB     00:00
(3/12): httpd-core-2.4.54-3.amzn202 8.9 MB/s | 1.3 MB     00:00
(4/12): mod_http2-1.15.24-1.amzn202 1.9 MB/s | 152 kB     00:00
(5/12): apr-util-1.6.3-1.amzn2023.0 1.7 MB/s |  98 kB     00:00
(6/12): mod_lua-2.4.54-3.amzn2023.0 1.4 MB/s |  60 kB     00:00
(7/12): httpd-2.4.54-3.amzn2023.0.4 1.5 MB/s |  46 kB     00:00
(8/12): libbrotli-1.0.9-4.amzn2023. 4.4 MB/s | 315 kB     00:00
(9/12): mailcap-2.1.49-3.amzn2023.0 753 kB/s |  33 kB     00:00
(10/12): httpd-tools-2.4.54-3.amzn2 978 kB/s |  80 kB     00:00
(11/12): httpd-filesystem-2.4.54-3. 210 kB/s |  13 kB     00:00
(12/12): generic-logos-httpd-18.0.0 439 kB/s |  19 kB     00:00
--------------------------------------------------------------------------------
Total                               6.6 MB/s | 2.3 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                              1/1
```

```
  Installing       : apr-1.7.2-2.amzn2023.0.2.x86_64           1/12
  Installing       : apr-util-openssl-1.6.3-1.amzn2023.0.1.    2/12
  Installing       : apr-util-1.6.3-1.amzn2023.0.1.x86_64      3/12
  Installing       : mailcap-2.1.49-3.amzn2023.0.3.noarch      4/12
  Installing       : httpd-tools-2.4.54-3.amzn2023.0.4.x86_    5/12
  Installing       : generic-logos-httpd-18.0.0-12.amzn2023    6/12
  Running scriptlet: httpd-filesystem-2.4.54-3.amzn2023.0.4    7/12
  Installing       : httpd-filesystem-2.4.54-3.amzn2023.0.4    7/12
  Installing       : httpd-core-2.4.54-3.amzn2023.0.4.x86_6    8/12
  Installing       : mod_http2-1.15.24-1.amzn2023.0.3.x86_6    9/12
  Installing       : libbrotli-1.0.9-4.amzn2023.0.2.x86_64    10/12
  Installing       : mod_lua-2.4.54-3.amzn2023.0.4.x86_64     11/12
  Installing       : httpd-2.4.54-3.amzn2023.0.4.x86_64       12/12
  Running scriptlet: httpd-2.4.54-3.amzn2023.0.4.x86_64       12/12
  Verifying        : apr-1.7.2-2.amzn2023.0.2.x86_64           1/12
  Verifying        : apr-util-openssl-1.6.3-1.amzn2023.0.1.    2/12
  Verifying        : httpd-core-2.4.54-3.amzn2023.0.4.x86_6    3/12
  Verifying        : mod_http2-1.15.24-1.amzn2023.0.3.x86_6    4/12
  Verifying        : apr-util-1.6.3-1.amzn2023.0.1.x86_64      5/12
  Verifying        : mod_lua-2.4.54-3.amzn2023.0.4.x86_64      6/12
  Verifying        : libbrotli-1.0.9-4.amzn2023.0.2.x86_64     7/12
  Verifying        : httpd-2.4.54-3.amzn2023.0.4.x86_64        8/12
  Verifying        : httpd-tools-2.4.54-3.amzn2023.0.4.x86_    9/12
  Verifying        : mailcap-2.1.49-3.amzn2023.0.3.noarch     10/12
  Verifying        : httpd-filesystem-2.4.54-3.amzn2023.0.4   11/12
  Verifying        : generic-logos-httpd-18.0.0-12.amzn2023   12/12

Installed:
  apr-1.7.2-2.amzn2023.0.2.x86_64
  apr-util-1.6.3-1.amzn2023.0.1.x86_64
  apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
  httpd-2.4.54-3.amzn2023.0.4.x86_64
  httpd-core-2.4.54-3.amzn2023.0.4.x86_64
  httpd-filesystem-2.4.54-3.amzn2023.0.4.noarch
  httpd-tools-2.4.54-3.amzn2023.0.4.x86_64
  libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  mailcap-2.1.49-3.amzn2023.0.3.noarch
  mod_http2-1.15.24-1.amzn2023.0.3.x86_64
  mod_lua-2.4.54-3.amzn2023.0.4.x86_64

Complete!
```

Hinzufügen, aktivieren oder deaktivieren neuer Repositorys

Wenn Sie ein Paket aus einem anderen Repository mithilfe des DNF-Paketmanagementsystems installieren möchten, fügen Sie die Repository-Angaben zur `/etc/dnf/dnf.conf`-Datei oder zu der *`repository.repo`*-Datei im Verzeichnis `/etc/yum.repos.d` hinzu. Sie können dies manuell durchführen. Die meisten DNF-Repositorys stellen jedoch ihre eigene *`repository.repo`*-Datei unter ihrer Repository-URL bereit.

> **Note**
>
> Derzeit gibt es keine zusätzlichen Repositorys, die zu AL2023 hinzugefügt werden können. Dies kann sich jedoch in Zukunft ändern. Sie könnten auch Ihre eigenen Pakete schreiben und diese Pakete für Ihre AL203-Enterprise-Umgebung verfügbar machen. Sie müssen dann das Repository, in dem die Pakete gespeichert sind, hinzufügen und aktivieren, bevor diese Pakete verwendet werden können.

Mit folgendem Befehl prüfen Sie, welche Repositorys derzeit aktiviert sind:

```
$ dnf repolist all --verbose
Loaded plugins: builddep, changelog, config-manager, copr, debug, debuginfo-install,
 download, generate_completion_cache, groups-manager, needs-restarting, playground,
 release-notification, repoclosure, repodiff, repograph, repomanage, reposync,
 supportinfo
DNF version: 4.12.0
cachedir: /var/cache/dnf
Last metadata expiration check: 0:00:02 ago on Wed Mar  1 23:40:15 2023.
Repo-id            : amazonlinux
Repo-name          : Amazon Linux 2023 repository
Repo-status        : enabled
Repo-revision      : 1677203368
Repo-updated       : Fri Feb 24 01:49:28 2023
Repo-pkgs          : 12632
Repo-available-pkgs: 12632
Repo-size          : 12 G
Repo-mirrors       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/mirrors/2023.0.20230222/x86_64/mirror.list
Repo-baseurl       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/guids/
cf9296325a6c46ff40c775a8e2d632c4c3fd9d9164014ce3304715d61b90ca8e/x86_64/
                   : (0 more)
```

```
Repo-expire        : 172800 second(s) (last: Wed Mar  1 23:40:15
                   : 2023)
Repo-filename      : /etc/yum.repos.d/amazonlinux.repo

Repo-id            : amazonlinux-debuginfo
Repo-name          : Amazon Linux 2023 repository - Debug
Repo-status        : disabled
Repo-mirrors       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/mirrors/2023.0.20230222/debuginfo/x86_64/mirror.list
Repo-expire        : 21600 second(s) (last: unknown)
Repo-filename      : /etc/yum.repos.d/amazonlinux.repo

Repo-id            : amazonlinux-source
Repo-name          : Amazon Linux 2023 repository - Source packages
Repo-status        : disabled
Repo-mirrors       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/mirrors/2023.0.20230222/SRPMS/mirror.list
Repo-expire        : 21600 second(s) (last: unknown)
Repo-filename      : /etc/yum.repos.d/amazonlinux.repo

Repo-id            : kernel-livepatch
Repo-name          : Amazon Linux 2023 Kernel Livepatch repository
Repo-status        : disabled
Repo-mirrors       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/kernel-livepatch/mirrors/al2023/x86_64/mirror.list
Repo-expire        : 172800 second(s) (last: unknown)
Repo-filename      : /etc/yum.repos.d/kernel-livepatch.repo

Repo-id            : kernel-livepatch-source
Repo-name          : Amazon Linux 2023 Kernel Livepatch repository -
                   : Source packages
Repo-status        : disabled
Repo-mirrors       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/kernel-livepatch/mirrors/al2023/SRPMS/mirror.list
Repo-expire        : 21600 second(s) (last: unknown)
Repo-filename      : /etc/yum.repos.d/kernel-livepatch.repo
Total packages: 12632
```

> **Note**
>
> Wenn Sie das `--verbose`-Options-Flag nicht hinzufügen, werden lediglich die `Repo-id`-, `Repo-name`- und `Repo-status`-Informationen zurückgegeben.

So fügen Sie ein **yum**-Repository zu einem **/etc/yum.repos.d**-Verzeichnis hinzu:

1. Suchen Sie den Speicherort der Datei `.repo`. In diesem Beispiel befindet sich die Datei `.repo` unter *https://www.example.com/repository.repo*.
2. Erstellen Sie ein Repository mit dem Befehl `dnf config-manager`.

```
$ sudo dnf config-manager --add-repo https://www.example.com/repository.repo
Loaded plugins: priorities, update-motd, upgrade-helper
adding repo from: https://www.example.com/repository.repo
grabbing file https://www.example.com/repository.repo to /etc/
yum.repos.d/repository.repo
repository.repo                                          | 4.0 kB     00:00
repo saved to /etc/yum.repos.d/repository.repo
```

Nach der Installation von Repositorys müssen diese wie folgt aktiviert werden.

Wenn Sie ein yum-Repository in `/etc/yum.repos.d` aktivieren möchten, verwenden Sie den `dnf config-manager`-Befehl mit dem `--enable`-Flag und dem *Repository*-Namen.

```
$ sudo dnf config-manager --enable repository
```

> **Note**
>
> Wenn Sie ein Repository deaktivieren möchten, verwenden Sie dieselbe Befehlssyntax, ersetzen jedoch `--enable` mit `--disable` im Befehl.

Hinzufügen von Repositorys mit cloud-init

Zusätzlich zur obigen Methode können Sie mithilfe des `cloud-init`-Frameworks ein neues Repository hinzufügen.

Wenn Sie ein neues Paket-Repository hinzufügen möchten, empfehlen wir die Verwendung der folgenden Vorlage. Wir empfehlen, diese Datei lokal zu speichern.

```
#cloud-config
yum_repos:
  repository.repo:
    baseurl: https://www.example.com/
```

```
    enabled: true
    gpgcheck: true
    gpgkey: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EXAMPLE
    name: Example Repository
```

> **Note**
>
> Ein Vorteil von `cloud-init` ist, dass Sie Ihrer Konfigurationsdatei einen `packages:`-Abschnitt hinzufügen können. In diesem Abschnitt können Sie die Namen der zu installierenden Pakete angeben. Sie können Pakete entweder aus dem Standard-Repository oder dem neuen Repository installieren, das Sie der `cloud-config`-Datei hinzugefügt haben.
> Detailliere Informationen zur Struktur der YAML-Datei finden Sie unter Hinzufügen eines YUM-Repositorys in the *`cloud-init`*-Dokumentation.

Nachdem Sie die Datei im YAML-Format eingerichtet haben, können Sie sie im `cloud-init`-Framework unter AWS CLI ausführen. Stellen Sie für den Aufruf der gewünschten Operation sicher, dass die `--userdata`-Option und der Name der `.yml`-Datei angegeben wurden.

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

Verwendung deterministischer Upgrades über ein versioniertes Repository auf AL2023

> **Note**
>
> Standardmäßig erhält Ihre AL203-Instance beim Start nicht automatisch zusätzliche kritische und wichtige Sicherheitsupdates. Ihre Instance enthält zunächst nur die Updates, die in der entsprechenden AL2023-Version und dem ausgewählten AMI verfügbar waren.

Kontrolle über die Updates, die Sie aus Haupt- und Nebenversionen erhalten

Mit AL2023 können Sie die Konsistenz zwischen Paketversionen und Updates in Ihrer gesamten Umgebung sicherstellen. Sie können die Konsistenz auch für mehrere Instances desselben Amazon Machine Image (AMI) sicherstellen. Mit dem Feature für deterministische Upgrades durch versionierte Repositorys, das standardmäßig aktiviert ist, können Sie stattdessen Aktualisierungen nach einem Zeitplan durchführen, der Ihren Anforderungen entspricht.

Mit jedem von uns veröffentlichten neuen Paket-Update gibt es eine neue Versionsanbindung sowie neue AMIs die an diese Version gebunden sind.

AL2023 bindet sich an eine bestimmte Version Ihres Repositorys. Dieses Verhalten wird für Haupt- und Nebenversionen unterstützt. Das AL2023-AMI, das Sie über unsere SSM-Parameter anzeigen können, ist immer die neueste Version. Es enthält die meisten up-to-date Pakete und Updates, einschließlich kritischer und wichtiger Sicherheitsupdates.

Wenn Sie eine Instance über ein vorhandenes AMI starten, werden Updates nicht automatisch angewendet. Alle zusätzlichen Pakete, die als Teil Ihres Bereitstellung installiert werden, werden der Repository-Version des vorhandenen AMI zugeordnet.

Bei Verwendung dieses Features sind Sie dafür verantwortlich, die Konsistenz zwischen den Paketversionen und Updates in Ihrer gesamten Umgebung sicherzustellen. Insbesondere dann, wenn Sie mehrere Instances von demselben AMI aus starten. Sie können Updates nach einem Zeitplan anwenden, der Ihren Anforderungen entspricht. Sie können auch einen bestimmten Satz von Updates beim Start anwenden, da diese auch an eine bestimmte Repository-Version gebunden werden können.

Unterschiede zwischen Haupt- und Nebenversions-Upgrades

Hauptversionen von AL2023 beinhalten umfangreiche Updates und können ggf. Pakete hinzufügen, löschen oder aktualisieren. Aktualisieren Sie Ihre Instance erst auf eine neue Hauptversion, nachdem Sie Ihre Anwendung mit dieser Version getestet haben, um Kompatibilität sicherzustellen.

Nebenversionen von AL2023 enthalten Funktions- und Sicherheits-Updates, jedoch keine Paketänderungen. Dadurch wird sichergestellt, dass die Linux-Funktionen und die Systembibliothek-API auch in neuen Versionen verfügbar bleiben. Das Testen Ihrer Anwendung vor einem Update ist nicht unbedingt notwendig.

Kontrollieren Sie die Paket-Updates, die in den AL2023-Repositorien verfügbar sind

Wenn wir eine neue Version der AL2023-Repositorien veröffentlichen, sind alle vorherigen Versionen weiterhin verfügbar. Standardmäßig ist das Plugin für die Verwaltung von Repository-Versionen an die Version gebunden, mit der das AMI erstellt wurde. Gehen Sie wie folgt vor, wenn Sie Kontrolle über Paketaktualisierungen haben möchten.

1. Mit dem folgenden Befehl können Sie die verfügbaren Repository-Versionen einsehen.

```
$ sudo dnf check-release-update
```

2. Mit dem folgenden Befehl wählen Sie eine Version aus.

```
$ sudo dnf --releasever=version update
```

Dieser Befehl startet mithilfe von `dnf` ein Update Ihrer aktuellen Amazon-Linux-Version auf die Version, die in der Befehlszeile angegeben ist. Eine Liste der Paket-Updates wird unter `dnf` angezeigt. Sie müssen die Aktualisierung bestätigen, bevor das Update verarbeitet wird. Nach Abschluss des Updates wird die neue Version zur Standard-Version, die `dnf` für alle zukünftigen Aktivitäten verwendet.

Weitere Informationen finden Sie unter Paket- und Betriebssystemupdates in AL2023 verwalten.

Deterministische Upgrades durch Nutzung versionierter Repositorys

Themen

- Verwendung eines aktualisierten deterministischen Systems
- Selektives Update eines deterministisch aktualisierten Systems
- Persistente Überschreibung bei einem deterministischen Upgrade

Verwendung eines aktualisierten deterministischen Systems

Wenn Sie den `dnf upgrade`-Befehl ausführen, sucht das System im von der `releasever`-Variablen angegebenen Repository nach Upgrades. *Eine gültige Version `releasever` ist entweder die neueste Version oder eine mit einem Datumsstempel versehene Version wie `2023.4.20240513`.*

Sie können den Wert `releasever` mit einer der folgenden Methoden ändern. Diese Methoden sind in absteigender Systempriorität aufgeführt. Das bedeutet, dass Methode 1 die Methoden 2 und 3 überschreibt und Methode 2 Methode 3 überschreibt.

1. Der Wert im Befehlszeilen-Flag, `--releasever=`*`latest`*, falls dieser verwendet wird.
2. Der Wert, der in der Override-Variablendatei angegeben ist, `/etc/dnf/vars/releasever`, falls dieser gesetzt ist.
3. Die aktuell installierte Version des `system-release`-Pakets.

Im folgenden Beispiel ist die Version *`2023.0.20230210`*:

```
$ rpm -q system-release
system-release-2023.0.20230210-0.amzn2023.noarch
```

In einem neu installierten System ist die Override-Variable nicht vorhanden. Es sind keine Upgrades verfügbar, da das System an die installierte `system-release`-Version gebunden ist.

```
$ cat /etc/dnf/vars/releasever
cat: /etc/dnf/vars/releasever: No such file or directory
```

```
$ sudo dnf upgrade
Last metadata expiration check: 0:00:02 ago on Wed 15 Feb 2023 06:14:12 PM UTC.
Dependencies resolved.
Nothing to do.
Complete!
```

Sie können Pakete einer bestimmten Version abrufen, indem Sie das `releasever`-Flag verwenden, um die gewünschte Version anzugeben.

```
$ rpm -q system-release
system-release-2023.0.20230222-0.amzn2023.noarch
```

```
$ sudo dnf upgrade --releasever=2023.0.20230329
Amazon Linux 2023 repository                    26 MB/s |  12 MB     00:00
Dependencies resolved.
================================================================================
 Package                 Arch    Version                      Repository   Size
================================================================================
Installing:
```

```
 kernel                  aarch64 6.1.21-1.45.amzn2023        amazonlinux  26 M
Upgrading:
 amazon-linux-repo-s3    noarch  2023.0.20230329-0.amzn2023  amazonlinux  18 k
 ca-certificates         noarch  2023.2.60-1.0.amzn2023.0.1  amazonlinux 828 k
 cloud-init              noarch  22.2.2-1.amzn2023.1.7       amazonlinux 1.1 M

         ... [ list edited for clarity ]

 system-release          noarch  2023.0.20230329-0.amzn2023  amazonlinux  29 k

         ... [ list edited for clarity ]

 vim-data                noarch  2:9.0.1403-1.amzn2023.0.1   amazonlinux  25 k
 vim-minimal             aarch64 2:9.0.1403-1.amzn2023.0.1   amazonlinux 753 k

Transaction Summary
================================================================================
Install    1 Package
Upgrade   42 Packages

Total download size: 56 M
```

Da die `--releasever`-Option sowohl `system-release` als auch `/etc/dnf/vars/releasever` überschreibt, ist das Ergebnis dieses Upgrades wie folgt:

1. Das Upgrade ersetzt alle installierten Pakete, die zwischen der vorherigen und der neuen Version geändert wurden.
2. Das Upgrade bindet das System für die neue Version von `system-release` an das Repository.

Selektives Update eines deterministisch aktualisierten Systems

Sie können beispielsweise ausgewählte Pakete aus einer neu veröffentlichten Version installieren, während das System auf die ursprüngliche Release-Version beschränkt bleibt.

Mithilfe von `dnf check-update` können Sie die zu aktualisierenden Pakete identifizieren.

```
$ sudo dnf check-update --releasever=latest --security
Amazon Linux 2023 repository                    13 MB/s |  10 MB     00:00
Last metadata expiration check: 0:00:02 ago on Wed 15 Feb 2023 02:52:21 AM UTC.

bind-libs.aarch64                32:9.16.27-1.amzn2023.0.1         amazonlinux
```

```
bind-license.noarch                32:9.16.27-1.amzn2023.0.1          amazonlinux
bind-utils.aarch64                 32:9.16.27-1.amzn2023.0.1          amazonlinux
cryptsetup.aarch64                 2.4.3-2.amzn2023.0.1               amazonlinux
cryptsetup-libs.aarch64            2.4.3-2.amzn2023.0.1               amazonlinux
curl-minimal.aarch64               7.85.0-1.amzn2023.0.1              amazonlinux
glibc.aarch64                      2.34-40.amzn2023.0.2               amazonlinux
glibc-all-langpacks.aarch64        2.34-40.amzn2023.0.2               amazonlinux
glibc-common.aarch64               2.34-40.amzn2023.0.2               amazonlinux
glibc-locale-source.aarch64        2.34-40.amzn2023.0.2               amazonlinux
gmp.aarch64                        1:6.2.1-2.amzn2023.0.1             amazonlinux
gnupg2-minimal.aarch64             2.3.7-1.amzn2023.0.2               amazonlinux
gzip.aarch64                       1.10-5.amzn2023.0.1                amazonlinux
kernel.aarch64                     6.1.12-17.42.amzn2023              amazonlinux
kernel-tools.aarch64               6.1.12-17.42.amzn2023              amazonlinux
libarchive.aarch64                 3.5.3-2.amzn2023.0.1               amazonlinux
libcurl-minimal.aarch64            7.85.0-1.amzn2023.0.1              amazonlinux
libsepol.aarch64                   3.4-3.amzn2023.0.2                 amazonlinux
libsolv.aarch64                    0.7.22-1.amzn2023.0.1              amazonlinux
libxml2.aarch64                    2.9.14-1.amzn2023.0.1              amazonlinux
logrotate.aarch64                  3.20.1-2.amzn2023.0.2              amazonlinux
lua-libs.aarch64                   5.4.4-3.amzn2023.0.1               amazonlinux
lz4-libs.aarch64                   1.9.4-1.amzn2023.0.1               amazonlinux
openssl.aarch64                    1:3.0.5-1.amzn2023.0.3             amazonlinux
openssl-libs.aarch64               1:3.0.5-1.amzn2023.0.3             amazonlinux
pcre2.aarch64                      10.40-1.amzn2023.0.1               amazonlinux
pcre2-syntax.noarch                10.40-1.amzn2023.0.1               amazonlinux
rsync.aarch64                      3.2.6-1.amzn2023.0.2               amazonlinux
vim-common.aarch64                 2:9.0.475-1.amzn2023.0.1           amazonlinux
vim-data.noarch                    2:9.0.475-1.amzn2023.0.1           amazonlinux
vim-enhanced.aarch64               2:9.0.475-1.amzn2023.0.1           amazonlinux
vim-filesystem.noarch              2:9.0.475-1.amzn2023.0.1           amazonlinux
vim-minimal.aarch64                2:9.0.475-1.amzn2023.0.1           amazonlinux
xz.aarch64                         5.2.5-9.amzn2023.0.1               amazonlinux
xz-libs.aarch64                    5.2.5-9.amzn2023.0.1               amazonlinux
zlib.aarch64                       1.2.11-32.amzn2023.0.3             amazonlinux
```

Installieren Sie die Pakete, die Sie aktualisieren möchten. Verwenden Sie `sudo dnf upgrade --releasever=latest` und die Paketnamen, um sicherzustellen, dass das `system-release`-Paket unverändert bleibt.

```
$ sudo dnf upgrade --releasever=latest openssl openssl-libs
Last metadata expiration check: 0:01:28 ago on Wed 15 Feb 2023 02:52:21 AM UTC.
Dependencies resolved.
```

```
================================================================================
 Package          Arch        Version                       Repository       Size
================================================================================
Upgrading:
 openssl          aarch64     1:3.0.5-1.amzn2023.0.3        amazonlinux     1.1 M
 openssl-libs     aarch64     1:3.0.5-1.amzn2023.0.3        amazonlinux     2.1 M

Transaction Summary
================================================================================
Upgrade  2 Packages

Total download size: 3.2 M
```

> **Note**
>
> Mit `sudo dnf upgrade --releasever=latest` werden alle Pakete aktualisiert, einschließlich `system-release`. Die Version bleibt an das neue `system-release` gebunden, es sei denn, Sie legen eine persistente Überschreibung fest.

Persistente Überschreibung bei einem deterministischen Upgrade

Anstatt `--releasever=latest` hinzuzufügen, können Sie mithilfe der persistenten Überschreibung das System entsperren. Hierzu setzen Sie den Variablenwert auf *`am neuesten`*.

```
$ echo latest | sudo tee /etc/dnf/vars/releasever
latest
```

```
$ sudo dnf upgrade
Last metadata expiration check: 0:03:36 ago on Wed 15 Feb 2023 02:52:21 AM UTC.
Dependencies resolved.
================================================================================
 Package                 Arch    Version                       Repository   Size
================================================================================
Installing:
 kernel                  aarch64 6.1.73-45.135.amzn2023        amazonlinux  24 M
Upgrading:
 acl                     aarch64 2.3.1-2.amzn2023.0.1          amazonlinux  72 k
 alternatives            aarch64 1.15-2.amzn2023.0.1           amazonlinux  36 k
 amazon-ec2-net-utils    noarch  2.3.0-1.amzn2023.0.1          amazonlinux  16 k
 at                      aarch64 3.1.23-6.amzn2023.0.1         amazonlinux  60 k
```

```
attr                    aarch64 2.5.1-3.amzn2023.0.1         amazonlinux  59 k
audit                   aarch64 3.0.6-1.amzn2023.0.1         amazonlinux 249 k
audit-libs              aarch64 3.0.6-1.amzn2023.0.1         amazonlinux 116 k
aws-c-auth-libs         aarch64 0.6.5-6.amzn2023.0.2         amazonlinux  79 k
aws-c-cal-libs          aarch64 0.5.12-7.amzn2023.0.2        amazonlinux  34 k
aws-c-common-libs       aarch64 0.6.14-6.amzn2023.0.2        amazonlinux 119 k
aws-c-compression-libs  aarch64 0.2.14-5.amzn2023.0.2        amazonlinux  22 k
aws-c-event-stream-libs aarch64 0.2.7-5.amzn2023.0.2         amazonlinux  47 k
aws-c-http-libs         aarch64 0.6.8-6.amzn2023.0.2         amazonlinux 147 k
aws-c-io-libs           aarch64 0.10.12-5.amzn2023.0.6       amazonlinux 109 k
aws-c-mqtt-libs         aarch64 0.7.8-7.amzn2023.0.2         amazonlinux  61 k
aws-c-s3-libs           aarch64 0.1.27-5.amzn2023.0.3        amazonlinux  54 k
aws-c-sdkutils-libs     aarch64 0.1.1-5.amzn2023.0.2         amazonlinux  26 k
aws-checksums-libs      aarch64 0.1.12-5.amzn2023.0.2        amazonlinux  50 k
awscli-2                noarch  2.7.8-1.amzn2023.0.4         amazonlinux 7.3 M
basesystem              noarch  11-11.amzn2023.0.1           amazonlinux 7.8 k
bash                    aarch64 5.1.8-2.amzn2023.0.1         amazonlinux 1.6 M
bash-completion         noarch  1:2.11-2.amzn2023.0.1        amazonlinux 292 k
bc                      aarch64 1.07.1-14.amzn2023.0.1       amazonlinux 120 k
bind-libs               aarch64 32:9.16.27-1.amzn2023.0.1    amazonlinux 1.2 M
bind-license            noarch  32:9.16.27-1.amzn2023.0.1    amazonlinux  14 k
bind-utils              aarch64 32:9.16.27-1.amzn2023.0.1    amazonlinux 206 k
binutils                aarch64 2.38-20.amzn2023.0.3         amazonlinux 4.6 M
boost-filesystem        aarch64 1.75.0-4.amzn2023.0.1        amazonlinux  55 k
boost-system            aarch64 1.75.0-4.amzn2023.0.1        amazonlinux  14 k
boost-thread            aarch64 1.75.0-4.amzn2023.0.1        amazonlinux  54 k
bzip2                   aarch64 1.0.8-6.amzn2023.0.1         amazonlinux  53 k
bzip2-libs              aarch64 1.0.8-6.amzn2023.0.1         amazonlinux  44 k
c-ares                  aarch64 1.17.2-1.amzn2023.0.1        amazonlinux 107 k
ca-certificates         noarch  2021.2.50-1.0.amzn2023.0.3   amazonlinux 343 k
checkpolicy             aarch64 3.4-3.amzn2023.0.1           amazonlinux 345 k
chkconfig               aarch64 1.15-2.amzn2023.0.1          amazonlinux 162 k
chrony                  aarch64 4.2-7.amzn2023.0.4           amazonlinux 314 k
cloud-init              noarch  22.2.2-1.amzn2023.1.7        amazonlinux 1.1 M
cloud-utils-growpart    aarch64 0.31-8.amzn2023.0.2          amazonlinux  31 k
coreutils               aarch64 8.32-30.amzn2023.0.2         amazonlinux 1.1 M
coreutils-common        aarch64 8.32-30.amzn2023.0.2         amazonlinux 2.0 M
cpio                    aarch64 2.13-10.amzn2023.0.1         amazonlinux 269 k
cracklib                aarch64 2.9.6-27.amzn2023.0.1        amazonlinux  83 k
cracklib-dicts          aarch64 2.9.6-27.amzn2023.0.1        amazonlinux 3.6 M
crontabs                noarch  1.11-24.20190603git.amzn2023.0.1
                                                             amazonlinux  19 k
crypto-policies         noarch  20230128-1.gitdfb10ea.amzn2023.0.1
                                                             amazonlinux  61 k
```

```
 crypto-policies-scripts noarch  20230128-1.gitdfb10ea.amzn2023.0.1
                                                              amazonlinux  81 k
...
Installing dependencies:
 amazon-linux-repo-cdn   noarch  2023.0.20230210-0.amzn2023   amazonlinux  16 k
 xxhash-libs             aarch64 0.8.0-3.amzn2023.0.1         amazonlinux  32 k
Installing weak dependencies:
 amazon-chrony-config    noarch  4.2-7.amzn2023.0.4           amazonlinux  14 k
 gawk-all-langpacks      aarch64 5.1.0-3.amzn2023.0.1         amazonlinux 207 k

Transaction Summary
================================================================================
Install    5 Packages
Upgrade  413 Packages

Total download size: 199 M
```

> **Note**
>
> Wenn Sie die Überschreibungsvariable `/etc/dnf/vars/releasever` verwendet haben, können Sie mithilfe des folgenden Befehls den Überschreibungswert löschen und so das Standardsperrverhalten wiederherstellen.
>
> ```
> $ sudo rm /etc/dnf/vars/releasever
> ```

Kernel-Live-Patching auf AL2023

Sie können Kernel Live Patching für AL2023 verwenden, um Sicherheitslücken und kritische Bug-Patches auf einen laufenden Linux-Kernel anzuwenden, ohne laufende Anwendungen neu zu starten oder zu unterbrechen. Darüber hinaus kann Kernel-Live-Patching dazu beitragen, die Verfügbarkeit Ihres Systems zu verbessern und gleichzeitig Ihre Infrastruktur sicher und auf dem neuesten Stand zu halten.

AWS veröffentlicht zwei Arten von Kernel-Live-Patches für AL2023:

- Sicherheitsupdates – Enthält Updates für die häufigsten Schwachstellen und Risiken von Linux (Common Vulnerabilities and Exposures, CVE). Diese Updates werden typischerweise als wichtig oder kritisch eingestuft, wobei die Amazon Linux Security Advisory-Bewertungen verwendet werden. Sie entsprechen im Allgemeinen einem CVSS-Score (Common Vulnerability Scoring

System) von 7 und höher. In einigen Fällen AWS kann es Updates bereitstellen, bevor ein CVE zugewiesen wird. In diesen Fällen erscheinen die Patches möglicherweise als Bugfixes.

- Bugfixes – Enthält Fixes für kritische Bugs und Stabilitätsprobleme, die nichts mit CVEs zu tun haben.

AWS stellt Kernel-Live-Patches für eine AL2023-Kernelversion für bis zu 3 Monate nach ihrer Veröffentlichung bereit. Nach Ablauf der Frist müssen Sie auf eine spätere Kernel-Version aktualisieren, um weiterhin Live-Kernel-Patches zu erhalten.

AL2023-Kernel-Live-Patches werden als signierte RPM-Pakete in den vorhandenen AL2023-Repositorys bereitgestellt. Die Patches können mithilfe vorhandener DNF-Paketmanager-Workflows auf einzelnen Instances installiert werden. Oder sie können mithilfe von AWS Systems Manager auf einer Gruppe verwalteter Instanzen installiert werden.

Kernel-Live-Patching auf AL2023 wird ohne zusätzliche Kosten zur Verfügung gestellt.

Themen

- Einschränkungen
- Unterstützte Konfigurationen und Voraussetzungen
- Arbeiten mit Kernel-Live-Patching

Einschränkungen

Während der Anwendung eines Kernel-Live-Patches können folgende Aktionen nicht ausgeführt werden: System-Hibernation, erweiterte Debugging-Tools verwenden (z. B. `SystemTap`, `kprobes` und `eBPF`-basierte Tools), auf `ftrace`-Ausgabedateien zugreifen, die von der Kernel-Live-Patching-Infrastruktur genutzt werden.

Unterstützte Konfigurationen und Voraussetzungen

Kernel-Live-Patching wird auf Amazon-EC2-Instances und On-Premises-VMs, die AL2023 ausführen unterstützt.

Gehen Sie wie folgt vor, um Kernel-Live-Patching auf AL2023 zu verwenden:

- Eine 64-Bit `x86_64`- oder `ARM64`-Architektur
- Kernel-Version `6.1`

Richtlinienanforderungen

Um Pakete aus AL2023-Repositorys herunterzuladen, benötigt Amazon EC2 Zugriff auf serviceeigene Amazon S3 S3-Buckets. Wenn Sie in Ihrer Umgebung einen Amazon Virtual Private Cloud (VPC) -Endpunkt für Amazon S3 verwenden, stellen Sie sicher, dass Ihre VPC-Endpunktrichtlinie den Zugriff auf diese öffentlichen Buckets zulässt. In der folgenden Tabelle wird der Amazon S3 S3-Bucket beschrieben, auf den Amazon EC2 möglicherweise für Kernel Live Patching zugreifen muss.

S3 Bucket-ARN	Beschreibung
*arn:aws:s3: ::al2023-repos-region -de612dc2/**	Amazon S3 S3-Bucket mit AL2023-Repositorys

Arbeiten mit Kernel-Live-Patching

Sie können Kernel-Live-Patching für einzelne Instances über die Befehlszeile auf der Instance selbst aktivieren und anwenden. Alternativ können Sie Kernel-Live-Patching auf eine Gruppe verwalteter Instances mithilfe des AWS -Systems-Manager aktivieren und anwenden.

In den folgenden Abschnitten wird erläutert, wie Sie Kernel-Live-Patching auf einzelnen Instances über die Befehlszeile aktivieren und verwenden.

Weitere Informationen zur Aktivierung und Anwendung von Kernel-Live-Patching für eine Gruppe verwalteter Instances finden Sie unter Verwenden von Kernel-Live-Patching auf AL2023-Instancesim AWS Systems Manager -Benutzerhandbuch.

Themen

- Aktivieren des Kernel-Live-Patching
- Anzeigen der verfügbaren Kernel-Live-Patches
- Anwenden von Kernel-Live-Patches
- Anzeigen der angewendeten Kernel-Live-Patches
- Deaktivieren des Kernel-Live-Patching

Aktivieren des Kernel-Live-Patching

Kernel-Live-Patching ist für AL2023 standardmäßig deaktiviert. Um Live-Patching zu verwenden, müssen Sie das DNF-Plugin für Kernel-Live-Patching installieren und die Live-Patching-Funktionalität aktivieren.

So aktivieren Sie das Kernel-Live-Patching:

1. Kernel-Live-Patches sind für AL2023 mit Kernel-Version `6.1` oder höher verfügbar. Um Ihre Kernel-Version zu überprüfen, führen Sie den folgenden Befehl aus.

   ```
   $ sudo dnf list kernel
   ```

2. Installieren Sie das DNF-Plugin für Kernel-Live-Patching.

   ```
   $ sudo dnf install -y kpatch-dnf
   ```

3. Aktivieren Sie das DNF-Plugin für Kernel-Live-Patching.

   ```
   $ sudo dnf kernel-livepatch -y auto
   ```

   Mit diesem Befehl wird auch die neueste Version des Kernel-Live-Patch-RPM aus den konfigurierten Repositorys installiert.

4. Führen Sie den folgenden Befehl aus, um zu prüfen, ob das DNF-Plugin für das Kernel-Live-Patching korrekt installiert wurde.

   Wenn Sie Kernel-Live-Patching aktivieren, wird automatisch ein leeres Kernel-Live-Patch-RPM angewendet. Wenn Kernel-Live-Patching erfolgreich aktiviert wurde, gibt dieser Befehl eine Liste zurück, die das anfänglich leere Kernel-Live-Patch-RPM enthält.

   ```
   $ sudo rpm -qa | grep kernel-livepatch
   dnf-plugin-kernel-livepatch-1.0-0.11.amzn2023.noarch
   kernel-livepatch-6.1.12-17.42-1.0-0.amzn2023.x86_64
   ```

5. Installieren Sie das kpatch-Paket.

   ```
   $ sudo dnf install -y kpatch-runtime
   ```

6. Aktualisieren Sie den kpatch-Service, falls er zuvor installiert wurde.

```
$ sudo dnf update kpatch-runtime
```

7. Starten Sie den kpatch-Service. Dieser Service lädt alle Live-Patches des Kernels bei der Initialisierung oder beim Booten.

```
$ sudo systemctl enable kpatch.service && sudo systemctl start kpatch.service
```

Anzeigen der verfügbaren Kernel-Live-Patches

Amazon Linux-Sicherheitswarnungen werden über das Amazon Linux-Sicherheitszentrum veröffentlich. Weitere Informationen über AL2023-Sicherheitswarnungen, die auch Warnungen für Kernel-Live-Patches enthalten, finden Sie im Amazon Linux Security Center. Kernel-Live-Patches wird das Präfix `ALASLIVEPATCH` vorangestellt. Das Amazon Linux-Sicherheitszentrum listet möglicherweise keine Live-Kernel-Patches auf, die Fehler beheben.

Sie können auch die verfügbaren Kernel-Live-Patches für Advisories und CVEs über die Befehlszeile ermitteln.

So listen Sie alle verfügbaren Kernel-Live-Patches für Advisories auf:

Verwenden Sie den folgenden Befehl.

```
$ sudo dnf updateinfo list
Last metadata expiration check: 1:06:23 ago on Mon 13 Feb 2023 09:28:19 PM UTC.
ALAS2LIVEPATCH-2021-123   important/Sec. kernel-
livepatch-6.1.12-17.42-1.0-4.amzn2023.x86_64
ALAS2LIVEPATCH-2022-124   important/Sec. kernel-
livepatch-6.1.12-17.42-1.0-3.amzn2023.x86_64
```

So listen Sie alle verfügbaren Kernel-Live-Patches für CVEs auf:

Verwenden Sie den folgenden -Befehl.

```
$ sudo dnf updateinfo list cves
Last metadata expiration check: 1:07:26 ago on Mon 13 Feb 2023 09:28:19 PM UTC.
CVE-2022-0123    important/Sec. kernel-livepatch-6.1.12-17.42-1.0-4.amzn2023.x86_64
CVE-2022-3210    important/Sec. kernel-livepatch-6.1.12-17.42-1.0-3.amzn2023.x86_64
```

Anwenden von Kernel-Live-Patches

Sie wenden Kernel-Live-Patches mithilfe des DNF-Paketmanagers auf dieselbe Weise wie regelmäßige Updates an. Das DNF-Plugin für Kernel-Live-Patching verwaltet die anzuwendenden Kernel-Live-Patches und macht einen Neustart überflüssig.

> **Tip**
>
> Wir empfehlen, dass Sie Ihren Kernel regelmäßig mit Kernel-Live-Patching aktualisieren, um dessen Sicherheit und Aktualität sicherzustellen.

Sie können wählen, ob Sie einen bestimmten Kernel-Live-Patch oder alle verfügbaren Kernel-Live-Patches zusammen mit Ihren regelmäßigen Sicherheitsupdates anwenden wollen.

So wenden Sie einen bestimmten Kernel-Live-Patch an:

1. Holen Sie sich die Kernel-Live-Patch-Version mit einem der in Anzeigen der verfügbaren Kernel-Live-Patches beschriebenen Befehle.
2. Wenden Sie den Kernel-Live-Patch für Ihren AL2023-Kernel an.

   ```
   $ sudo dnf install kernel-livepatch-kernel_version-package_version.amzn2023.x86_64
   ```

   Der folgende Befehl wendet beispielsweise einen Kernel-Live-Patch für die AL2023-Kernel-Version `6.1.12-17.42` an.

   ```
   $ sudo dnf install kernel-livepatch-6.1.12-17.42-1.0-4.amzn2023.x86_64
   ```

So wenden Sie alle verfügbaren Kernel-Live-Patches zusammen mit Ihren regelmäßigen Sicherheitsupdates an:

Verwenden Sie den folgenden Befehl.

```
$ sudo dnf update --security
```

Lassen Sie die Option `--security` weg, um Bugfixes einzuschließen.

Important

- Die Kernel-Version wird nach der Anwendung von Kernel-Live-Patches nicht aktualisiert. Die Version wird erst nach einem Neustart der Instance auf die neue Version aktualisiert.
- Ein AL2023-Kernel erhält Kernel-Live-Patches für einen Zeitraum von drei Monaten. Nach Ablauf der drei Monate werden für diese Kernel-Version keine neuen Kernel-Live-Patches mehr veröffentlicht.
- Wenn Sie nach 3 Monaten weiterhin Kernel-Live-Patches erhalten möchten, müssen Sie die Instance neu starten, um auf die neue Kernel-Version zu aktualisieren. Die Instance wird nach dem Update wieder für 3 Monate Kernel-Live-Patches erhalten.
- Führen Sie den folgenden Befehl aus, um das verbleibende Zeitfenster Ihrer Kernel-Version zu prüfen:

```
$ sudo dnf kernel-livepatch support
```

Anzeigen der angewendeten Kernel-Live-Patches

So zeigen Sie die angewendeten Kernel-Live-Patches an:

Verwenden Sie den folgenden Befehl.

```
$ sudo kpatch list
Loaded patch modules:
livepatch_CVE_2022_36946 [enabled]

Installed patch modules:
livepatch_CVE_2022_36946 (6.1.57-29.131.amzn2023.x86_64)
livepatch_CVE_2022_36946 (6.1.57-30.131.amzn2023.x86_64)
```

Der Befehl gibt eine Liste der geladenen und installierten Sicherheitsupdate-Kernel-Live-Patches zurück. Es folgt eine Beispielausgabe.

Note

Ein einziger Kernel-Live-Patch kann mehrere Live-Patches enthalten und installieren.

Deaktivieren des Kernel-Live-Patching

Wenn Sie das Kernel-Live-Patching nicht mehr verwenden möchten, können Sie es jederzeit deaktivieren.

- Deaktivieren Sie die Verwendung von livepatches:
  1. Das Plugin deaktivieren:

     ```
     $ sudo dnf kernel-livepatch manual
     ```

  2. Den kpatch-Dienst deaktivieren:

     ```
     $ sudo systemctl disable --now kpatch.service
     ```

- Die livepatch-Tools vollständig entfernen:
  1. Das Plugin entfernen:

     ```
     $ sudo dnf remove kpatch-dnf
     ```

  2. kpatch-runtime entfernen:

     ```
     $ sudo dnf remove kpatch-runtime
     ```

  3. Alle installierten livepatches entfernen:

     ```
     $ sudo dnf remove kernel-livepatch\*
     ```

Erste Schritte mit der Programmierung von Laufzeiten auf AL2023

AL2023 bietet verschiedene Versionen einiger Sprachlaufzeiten. Wir arbeiten mit Upstream-Projekten, die mehrere Versionen gleichzeitig unterstützen. Hier finden Sie Informationen zur Installation und Verwaltung dieser Pakete mit Namensversionen mithilfe des `dnf`-Befehls für die Suche nach und Installation von diesen Paketen.

In den folgenden Themen wird beschrieben, wie die einzelnen Sprachökosysteme in AL2023 existieren.

Themen

- C, C++ und Fortran in AL2023
- Go in AL2023
- Java in AL2023
- Perl in AL2023
- PHP in AL2023
- Python in AL2023
- Rust in AL2023

C, C++ und Fortran in AL2023

AL2023 beinhaltet sowohl die GNU Compiler Collection (GCC) als auch das Clang Frontend für LLVM (Low Level Virtual Machine).

Die Hauptversion von GCC wird während der gesamten Lebensdauer von AL2023 beibehalten. Nebenversionen enthalten Fehlerkorrekturen und sind möglicherweise in AL2023-Versionen enthalten. Andere Fehler-, Leistungs- und Sicherheitskorrekturen werden möglicherweise auf die GCC-Hauptversion zurückportiert, die in AL2023 enthalten ist.

AL2023 enthält Version 11 von GCC mit den Frontends C (`gcc`), C++ () und Fortran (`g++`). `gfortran`

AL2023 aktiviert die Frontends (), Ada (`gnat`), Objective-C Go oder `gcc-go` Objective-C++ nicht.

Zu den Standard-Compiler-Flags, mit denen AL203-RPMs erstellt wurden, gehören Optimierungs- und Hardening-Flags. Um Ihren eigenen Code mit GCC zu erstellen, empfehlen wir Ihnen, Optimierungs- und Hardening-Flags hinzuzufügen.

> **Note**
>
> Bei einem Aufruf von `gcc --version` wird eine Versionszeichenfolge wie `gcc (GCC) 11.3.1 20221121 (Red Hat 11.3.1-4)` angezeigt. `Red Hat` bezieht sich auf den GCC-Herstellerzweig, auf dem das Amazon Linux GCC-Paket basiert. Gemäß der von `gcc --help` angezeigten Fehlerbericht-URL sollten alle Fehlerberichte und Support-Anfragen an Amazon Linux gerichtet werden.
> Weitere Informationen über einige der langfristigen Änderungen in diesem Herstellerbereich, wie zum Beispiel das `__GNUC_RH_RELEASE__` Makro, finden Sie unter Fedora-Paketquellen.

Weitere Informationen zur Kern-Toolchain finden Sie unter. Core-Toolchain-Paketeglibc, gcc und binutils

Weitere Informationen zu AL2023 und seiner Beziehung zu anderen Linux-Distributionen finden Sie unter. Beziehung zu Fedora

Weitere Informationen zur Änderung der Compiler-Tripletts in AL2023 im Vergleich zu AL2 finden Sie unter. Compiler-Triplet

Go in AL2023

Möglicherweise möchten Sie Ihren eigenen Code erstellen, der Goauf Amazon Linux geschrieben ist, und vielleicht möchten Sie eine mit AL2023 bereitgestellte Toolchain verwenden. Ähnlich wie AL2 aktualisiert AL2023 die Go Toolchain während der gesamten Lebensdauer des Betriebssystems. Dies kann als Reaktion auf ein beliebiges CVE in der von uns gelieferten Toolchain oder als Teil einer vierteljährlichen Nebenversion geschehen.

Goist eine Sprache, die sich relativ schnell bewegt. Es kann vorkommen, dass bestehende Anwendungen, in die geschrieben Go wurden, an neue Versionen der Go Toolchain angepasst werden müssen. Weitere Informationen dazu finden Sie Go unter Go1 und die Zukunft der Go Programme.

Obwohl AL2023 im Laufe seiner Laufzeit neue Versionen der Go Toolchain enthalten wird, wird dies nicht im Gleichschritt mit den Upstream-Versionen geschehen. Go Daher ist die Verwendung der in

AL2023 bereitgestellten Go Toolchain möglicherweise nicht geeignet, wenn Sie Go Code mit den neuesten Funktionen der Sprache und der Go Standardbibliothek erstellen möchten.

Während der Lebensdauer von AL2023 werden frühere Paketversionen nicht aus den Repositorys entfernt. Wenn eine frühere Go Toolchain erforderlich ist, können Sie auf die Fehler- und Sicherheitskorrekturen neuerer Go Toolchains verzichten und eine frühere Version aus den Repositorys installieren, indem Sie dieselben Mechanismen verwenden, die für jedes RPM verfügbar sind.

Wenn Sie Ihren eigenen Go Code auf AL2023 erstellen möchten, können Sie die in AL2023 enthaltene Go Toolchain mit dem Wissen verwenden, dass diese Toolchain während der gesamten Lebensdauer von AL2023 weiterentwickelt werden könnte.

AL2023 Lambda-Funktionen, geschrieben in Go

Bei der Go Kompilierung zu nativem Code wird Lambda Go wie eine benutzerdefinierte Laufzeit behandelt. Sie können die `provided.al2023` Laufzeit verwenden, um Go Funktionen auf AL2023 für Lambda bereitzustellen.

Weitere Informationen finden Sie unter Erstellen von Lambda-Funktionen mit Go im AWS Lambda Entwicklerhandbuch.

Java in AL2023

AL2023 bietet mehrere Versionen von Amazon Corretto zur Unterstützung Java von basierten Workloads. Alle in AL2023 enthaltenen Java basierten Pakete werden mit erstellt. Amazon Corretto 17 17

Corretto ist ein Build des Open Java Development Kit (OpenJDK) mit langfristiger Unterstützung von. Amazon Corretto ist mit dem Java Technical Compatibility Kit (TCK) zertifiziert, um sicherzustellen, dass es dem Java SE-Standard entspricht und unter Linux,, Windows und verfügbar ist. macOS

Für Corretto 1.8.0, Corretto 11 und Corretto 17 ist jeweils ein Amazon Corretto-Paket verfügbar.

Jede Corretto-Version in AL2023 wird für den gleichen Zeitraum wie die Corretto-Version unterstützt oder bis zum Ende der Lebensdauer von AL2023 unterstützt, je nachdem, was früher eintritt. Weitere Informationen finden Sie in den Support-Erklärungen für Amazon Linux-Pakete und in den häufig gestellten Fragen zu Amazon Corretto.

Perl in AL2023

AL2023 stellt Version 5.32 der PerlProgrammiersprache bereit.

Obwohl Amazon Linux in den letzten Jahrzehnten im Rahmen von Perl 5 Releases ein hohes Maß an Sprachkompatibilität geboten Perl hat, ist nicht damit zu rechnen, dass Amazon Linux in der Version AL2023 auf Perl Version 5.32 umsteigen wird. Amazon Linux wird Perl für die gesamte Lebensdauer von AL2023 weiterhin Sicherheitspatches gemäß unseren Paket-Unterstützungserklärungen verwenden.

Perl-Module in AL2023

Verschiedene Perl Module sind in AL2023 als RPMs verpackt. Obwohl viele Perl Module als RPMs verfügbar sind, ist Amazon Linux nicht bestrebt, jedes mögliche Perl Modul zu paketieren. Module, die als RPMs verpackt sind, können von anderen RPM-Paketen für Betriebssysteme verwendet werden, sodass Amazon Linux diesen Sicherheitspatches Vorrang vor reinen Funktionsupdates einräumt.

AL2023 beinhaltet auch, `CPAN` dass Perl Entwickler den idiomatischen Paketmanager für Module verwenden können. Perl

PHP in AL2023

AL2023 bietet derzeit zwei Versionen der PHPProgrammiersprache, die jeweils für den gleichen Zeitraum wie die PHP Upstream-Version unterstützt werden. Weitere Informationen finden Sie unter Erklärungen zur Paketunterstützung.

Mit AL2023 können Sie die neuen Funktionen von PHP 8.2 verwenden und gleichzeitig Anwendungen unterstützen, für die PHP 8.1 erforderlich ist.

Migration aus älteren PHP-Versionen

Die PHP Upstream-Community hat eine umfassende Migrationsdokumentation für die Umstellung von PHP 8.1 auf PHP 8.2 zusammengestellt. Es gibt ebenfalls Dokumentation für die Migration von PHP 8.0 auf 8.1.

AL2 umfasst PHP 8.0, 8.1 und 8.2, um einen einfachen Upgrade-Pfad auf AL2023 zu `amazon-linux-extras` ermöglichen.

Migration aus PHP 7.x-Versionen

> **Note**
>
> Das PHPProjekt führt eine Liste und einen Zeitplan unterstützter Versionen sowie eine Liste der nicht unterstützten Zweige.
> Als AL2023 veröffentlicht wurde, wurden alle 7.x- und 5.x-Versionen von von der PHP Community nicht unterstützt und PHPwaren nicht als Optionen in AL2023 enthalten.

Die PHP Upstream-Community hat eine umfassende Migrationsdokumentation für die Umstellung von 7.4 auf 8.0 zusammengestellt. PHP PHP In Kombination mit der Dokumentation, auf die im vorherigen Abschnitt zur Migration auf PHP 8.1 und PHP 8.2 verwiesen wurde, können Sie Ihre Basisanwendung PHP auf die moderne PHP Version migrieren.

> **Note**
>
> AL2 umfasst PHP 7.1, 7.2, 7.3 und 7.4 Zoll. `amazon-linux-extras` Es ist wichtig zu beachten, dass für all diese Extras garantiert weitere Sicherheitsupdates verfügbar sind end-of-life und nicht garantiert werden.

PHP-Module in AL2023

AL2023 enthält viele PHP Module, die in PHP Core enthalten sind. AL2023 zielt nicht darauf ab, alle Pakete in die PHPExtension Community Library (PECL) aufzunehmen.

Python in AL2023

AL2023 hat Python 2.7 entfernt und alle Komponenten, die dies erfordern, Python sind jetzt so geschrieben, dass sie mit Python 3 funktionieren.

AL2023 stellt Python 3 zur Verfügung, `/usr/bin/python3` um die Kompatibilität mit Kundencode zu gewährleisten, sowie Python-Code, der mit AL2023 ausgeliefert wurde. Dieser Wert bleibt für die gesamte Lebensdauer von AL2023 bei Python 3.9.

Die Version von Python, auf die `/usr/bin/python3` verwiesen wird, wird als das System Python betrachtet und für AL2023 ist dies Python 3.9.

Neuere Versionen vonPython, wie z. B. Python 3.11, werden in AL2023 als Pakete zur Verfügung gestellt und für die gesamte Lebensdauer der Upstream-Versionen unterstützt. Informationen darüber, wie lange Python 3.11 unterstützt wird, finden Sie unter Python 3.11.

Auf AL2023 können mehrere Versionen von Python gleichzeitig installiert sein. Obwohl es immer Python 3.9 sein `/usr/bin/python3` wird, hat jede Version von Python einen Namespace und kann anhand ihrer Versionsnummer gefunden werden. Wenn beispielsweise `python3.11` installiert ist, dann kann `/usr/bin/python3.11` parallel zu `/usr/bin/python3.9` und dem `/usr/bin/python3`-Symlink zu `/usr/bin/python3.9` existieren.

> **Note**
>
> Ändern Sie nicht, worauf der `/usr/bin/python3` Symlink verweist, da dies die Kernfunktionalität von AL2023 beeinträchtigen könnte.

Python-Module in AL2023

Verschiedene Python Module sind in AL2023 als RPMs verpackt. In der Regel werden RPMs für Python-Module nur für die Python-Systemversion erstellt.

Rust in AL2023

Möglicherweise möchten Sie Ihren in RustAmazon Linux geschriebenen Code erstellen und möglicherweise eine mit AL2023 bereitgestellte Toolchain verwenden.

Ähnlich wie AL2 aktualisiert AL2023 die Rust Toolchain während der gesamten Lebensdauer des Betriebssystems. Dies kann als Reaktion auf ein beliebiges CVE in der von uns gelieferten Toolchain oder als Teil einer vierteljährlichen Nebenversion geschehen.

Rust ist eine relativ schnelllebige Sprache, mit neuen Veröffentlichungen in einem Rhythmus von etwa sechs Wochen. In diese Versionen werden neue Sprach- oder Standardbibliotheksfunktionen hinzugefügt. AL2023 wird zwar im Laufe seiner Laufzeit neue Versionen der Rust Toolchain integrieren, dies wird jedoch nicht im Gleichschritt mit den Upstream-Versionen erfolgen. Rust Daher ist die Verwendung der in AL2023 bereitgestellten Rust Toolchain möglicherweise nicht geeignet, wenn Sie Rust Code mit den neuesten Funktionen der Sprache erstellen möchten. Rust

Während der Lebensdauer von AL2023 werden alte Paketversionen nicht aus den Repositorys entfernt. Wenn eine ältere Rust Toolchain erforderlich ist, können Sie auf Fehler- und

Sicherheitskorrekturen neuerer Rust Toolchains verzichten und eine ältere Version aus den Repositorys installieren, indem Sie dieselben Mechanismen verwenden, die für jedes RPM verfügbar sind.

Wenn Sie Ihren eigenen Rust Code auf AL2023 erstellen möchten, können Sie die in AL2023 enthaltene Rust Toolchain mit dem Wissen verwenden, dass diese Toolchain während der gesamten Lebensdauer von AL2023 weiterentwickelt werden könnte.

AL2023 Lambda-Funktionen, geschrieben in Rust

Da zu nativem Code Rust kompiliert wird, behandelt Lambda es Rust wie eine benutzerdefinierte Laufzeit. Sie können die `provided.al2023` Laufzeit verwenden, um Rust Funktionen auf AL2023 für Lambda bereitzustellen.

Weitere Informationen finden Sie unter Erstellen von Lambda-Funktionen mit Rust im AWS Lambda Entwicklerhandbuch.

Sicherheit und Compliance in Amazon Linux 2

> **Important**
>
> Wenn Sie eine Sicherheitslücke melden möchten oder Sicherheitsbedenken in Bezug auf AWS Cloud-Dienste oder Open-Source-Projekte haben, wenden Sie sich über die Seite zur Meldung von AWS Sicherheitslücken an die Sicherheitsabteilung

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für AL2 gelten, finden Sie unter AWS Services im Umfang nach Compliance-Programmen AWS .
- Sicherheit in der Cloud: Ihr Verantwortungsumfang wird durch den AWS -Service bestimmt, den Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, einschließlich der Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Themen

- Amazon Linux-Sicherheitsempfehlungen für AL2023
- Einstellung der SELinux-Modi für AL2023
- Aktivieren Sie den FIPS-Modus auf AL2023
- AL2023 Kernhärtung
- UEFI Secure Boot auf AL2023

Amazon Linux-Sicherheitsempfehlungen für AL2023

Wir arbeiten kontinuierlich an der Sicherheit von Amazon Linux, dennoch werden von Zeit zu Zeit Sicherheitsprobleme auftauchen, die behoben werden müssen. Ein Hinweis wird herausgegeben, wenn ein Update verfügbar ist. Der Hauptort, an dem wir unsere Empfehlungen veröffentlichen, ist das Amazon Linux Security Center (ALAS). Weitere Informationen erhalten Sie im Amazon-Linux-Sicherheitszentrum.

> **Important**
>
> Wenn Sie eine Sicherheitslücke melden möchten oder Sicherheitsbedenken in Bezug auf AWS Cloud-Dienste oder Open-Source-Projekte haben, wenden Sie sich über die Seite zur Meldung von AWS Sicherheitslücken an die Sicherheitsabteilung

Informationen zu Problemen und den relevanten Updates, die AL2023 betreffen, werden vom Amazon Linux-Team an verschiedenen Orten veröffentlicht. Normalerweise rufen Sicherheitstools Informationen aus diesen Primärquellen ab und präsentieren Ihnen die Ergebnisse. Daher interagieren Sie möglicherweise nicht direkt mit den primären Quellen, die Amazon Linux veröffentlicht, sondern mit der Schnittstelle, die von Ihren bevorzugten Tools wie Amazon Inspector bereitgestellt wird.

Ankündigungen des Amazon Linux Security Center

Ankündigungen von Amazon Linux beziehen sich auf Artikel, die nicht in eine Empfehlung passen. Dieser Abschnitt enthält Ankündigungen über ALAS selbst sowie Informationen, die nicht in eine Empfehlung passen. Weitere Informationen finden Sie unter Ankündigungen des Amazon Linux Security Center (ALAS).

Zum Beispiel passt die Amazon Linux Hotpatch-Ankündigung 2021-001 für Apache Log4j eher in eine Ankündigung als in eine Empfehlung. In dieser Ankündigung hat Amazon Linux ein Paket hinzugefügt, das Kunden dabei unterstützt, ein Sicherheitsproblem in Software zu beheben, die nicht Teil von Amazon Linux war.

Der Amazon Linux Security Center CVE Explorer wurde ebenfalls in den ALAS-Ankündigungen angekündigt. Weitere Informationen finden Sie unter Neue Website für CVEs.

Häufig gestellte Fragen zum Amazon Linux Security Center

Antworten auf einige häufig gestellte Fragen zu ALAS und zur Bewertung von CVEs durch Amazon Linux finden Sie unter Häufig gestellte Fragen (FAQs) im Amazon Linux Security Center (ALAS).

Einstellung der SELinux-Modi für AL2023

Standardmäßig ist Security Enhanced Linux (SELinux) aktiviert `enabled` und auf den Modus für AL2023 eingestellt. `permissive` Im permissiven Modus werden Zugriffsverweigerungen protokolliert, aber nicht durchgesetzt. SELinux ist eine Sammlung von Kernel-Features und Hilfsprogrammen, die eine starke, flexible MAC-Architektur (Mandatory Access Control) für die wichtigsten Subsysteme des Kernels bereitstellen.

SELinux bietet einen erweiterten Mechanismus zur Durchsetzung der Trennung von Informationen auf der Grundlage von Vertraulichkeits- und Integritätsanforderungen. Diese Trennung von Informationen reduziert die Gefahr, dass die Sicherheitsmechanismen von Anwendungen manipuliert und umgangen werden. Sie begrenzt auch Schäden, die durch bösartige oder fehlerhafte Anwendungen verursacht werden können.

SELinux enthält eine Reihe von Beispielkonfigurationsdateien für Sicherheitsrichtlinien, die darauf ausgelegt sind, alltägliche Sicherheitsziele zu erreichen.

Weitere Informationen zu den Merkmalen und Funktionen von SELinux finden Sie unter SELinux Notebook und Policy Languages.

Themen

- Standard-SELinux-Status und -Modi für AL2023
- Wechseln in den enforcing-Modus
- Option zum Deaktivieren von SELinux für AL2023

Standard-SELinux-Status und -Modi für AL2023

Für AL2023 ist SELinux standardmäßig auf Modus eingestellt. `enabled` `permissive` Im `permissive`-Modus werden Zugriffsverweigerungen protokolliert, aber nicht durchgesetzt.

Mit den Befehlen **`sestatus`** oder **`getenforce`** können Sie sich den aktuellen Status, die Richtlinie und den Modus von SELinux anzeigen lassen.

Wenn der Standardstatus auf `enabled` und `permissive` gesetzt ist, dann gibt der **`getenforce`**-Befehl `permissive` zurück.

Der **`sestatus`** Befehl gibt den SELinux-Status und die aktuelle SELinux-Richtlinie zurück, wie im folgenden Beispiel gezeigt:

```
$ sestatus
SELinux status:                 enabled
  SELinuxfs mount:                /sys/fs/selinux
  SELinux root directory:         /etc/selinux
  Loaded policy name:             targeted
  Current mode:                   permissive
  Mode from config file:          permissive
  Policy MLS status:              enabled
  Policy deny_unknown status:     allowed
  Memory protection checking:     actual (secure)
  Max kernel policy version:      33
```

Wenn Sie SELinux im `permissive` Modus ausführen, beschriften Benutzer Dateien möglicherweise falsch. Wenn Sie SELinux im `disabled`-Status ausführen, werden Dateien nicht gekennzeichnet. Falsch benannte oder unbenannte Dateien können Probleme verursachen, wenn Sie in den `enforcing`-Modus wechseln.

Zur Vermeidung dieses Problems benennt SELinux Dateien automatisch um. SELinux verhindert Kennzeichnungsprobleme durch eine automatisches Umbenennung, wenn Sie den Status auf `enabled` setzen.

Wechseln in den **`enforcing`**-Modus

Wenn Sie SELinux im `enforcing` Modus ausführen, ist das SELinux Hilfsprogramm `enforcing` die konfigurierte Richtlinie. SELinuxsteuert die Funktionen ausgewählter Anwendungen, indem der Zugriff auf der Grundlage der Richtlinienregeln zugelassen oder verweigert wird.

Führen Sie den Befehl aus, um SELinux den aktuellen Modus zu finden. `getenforce`

```
getenforce
Permissive
```

Bearbeiten der Konfigurationsdateizur **`enforcing`**-Modusaktivierung

Gehen Sie wie folgt vor`enforcing`, um den Modus zu ändern.

1. Bearbeiten Sie die `/etc/selinux/config`-Datei, um in den `enforcing`-Modus zu wechseln. Die `SELINUX` Einstellung sollte wie im folgenden Beispiel aussehen.

```
SELINUX=enforcing
```

2. Starten Sie Ihr System neu, um den `enforcing`-Moduswechsel abzuschließen.

```
$ sudo reboot
```

Beim nächsten Start werden alle SELinux Dateien und Verzeichnisse im System neu beschriftet. SELinuxfügt außerdem den SELinux Kontext für Dateien und Verzeichnisse hinzu, die zu dem Zeitpunkt SELinux erstellt wurden. `disabled`

Nach dem `enforcing` Moduswechsel werden SELinux möglicherweise einige Aktionen aufgrund falscher oder fehlender SELinux Richtlinienregeln verweigert. Sie können die Aktionen, die SELinux abgelehnt wurden, mit dem folgenden Befehl anzeigen.

```
$ sudo ausearch -m AVC,USER_AVC,SELINUX_ERR,USER_SELINUX_ERR -ts recent
```

Den cloud-init-Modus mit **`enforcing`** aktivieren

Alternativ können Sie beim Starten Ihrer Instance Folgende `cloud-config` als Benutzerdaten übergeben, um den `enforcing`-Modus zu aktivieren.

```
#cloud-config
selinux:
  mode: enforcing
```

Standardmäßig führt diese Einstellung dazu, dass die Instance neu gestartet wird. Für mehr Stabilität empfehlen wir, Ihre Instance neu zu starten. Sie können den Neustart auch überspringen, wenn Sie möchten. Geben Sie dazu Folgendes an: `cloud-config`.

```
#cloud-config
selinux:
  mode: enforcing
  selinux_no_reboot: 1
```

Option zum Deaktivieren von SELinux für AL2023

Wenn Sie die Option deaktivierenSELinux, wird SELinux die Richtlinie weder geladen noch durchgesetzt, und Access Vector Cache (AVC) -Meldungen werden nicht protokolliert. Sie verlieren alle Vorteile des Laufens. SELinux

Anstatt den Modus zu deaktivierenSELinux, empfehlen wir, den `permissive` Modus zu verwenden. Die Ausführung im `permissive` Modus kostet nur wenig mehr als die SELinux vollständige Deaktivierung. Der Übergang von `permissive` Modus zu `enforcing` Modus erfordert viel weniger Konfigurationsanpassungen als der Übergang zurück in den `enforcing` Modus nach der Deaktivierung. SELinux Sie können Dateien kennzeichnen, und das System kann Aktionen verfolgen und protokollieren, die die aktive Richtlinie möglicherweise verweigert hat.

In den Modus wechseln SELinux**`permissive`**

Wenn Sie SELinux im `permissive` Modus ausführen, wird SELinux die Richtlinie nicht durchgesetzt. Im `permissive` Modus werden AVC-Meldungen SELinux protokolliert, Operationen werden jedoch nicht verweigert. Sie können diese AVC-Meldungen zur Problembehandlung, zum Debuggen und SELinux zur Verbesserung der Richtlinien verwenden.

Gehen Sie wie folgt vorSELinux, um in den permissiven Modus zu wechseln.

1. Bearbeiten Sie die `/etc/selinux/config`-Datei, um in den `permissive`-Modus zu wechseln. Der `SELINUX` Wert sollte wie im folgenden Beispiel aussehen.

   ```
   SELINUX=permissive
   ```

2. Starten Sie Ihr System neu, um den `permissive`-Moduswechsel abzuschließen.

   ```
   sudo reboot
   ```

Deaktivieren von SELinux

Wenn Sie die Option deaktivierenSELinux, wird SELinux die Richtlinie weder geladen noch durchgesetzt, und AVC-Meldungen werden nicht protokolliert. Sie verlieren alle Vorteile des Laufens. SELinux

Gehen Sie SELinux zum Deaktivieren wie folgt vor.

1. Stellen Sie sicher, dass das `grubby` Paket installiert ist.

```
rpm -q grubby
grubby-version
```

2. Konfigurieren Sie Ihren Bootloader so, dass er `selinux=0` zur Kernel-Befehlszeile hinzufügt.

```
sudo grubby --update-kernel ALL --args selinux=0
```

3. Starten Sie Ihr System neu.

```
sudo reboot
```

4. Führen Sie den `getenforce` Befehl aus, um zu bestätigen, dass dies SELinux der Fall ist`Disabled`.

```
$ getenforce
Disabled
```

Weitere Informationen zu SELinux finden Sie im SELinuxNotizbuch und unter SELinuxKonfiguration.

Aktivieren Sie den FIPS-Modus auf AL2023

In diesem Abschnitt wird erklärt, wie die Federal Information Processing Standards (FIPS) auf AL2023 aktiviert werden. Weitere Informationen über FIPS finden Sie unter:

- Federal Information Processing Standard (FIPS)
- Compliance-FAQs: Federal Information Processing Standards

> **Note**
>
> In diesem Abschnitt wird beschrieben, wie der FIPS-Modus in AL2023 aktiviert wird. Der Zertifizierungsstatus der kryptografischen Module von AL2023 wird nicht behandelt.

Voraussetzungen

- Eine vorhandene AL2023-Amazon-EC2-Instance (AL2023.2 oder höher) mit Internetzugang zum Herunterladen der erforderlichen Pakete. Weitere Informationen zum Starten einer AL2023-mazon-EC2-Instance finden Sie unter AL2023 mit der Amazon EC2 EC2-Konsole starten.
- Ihre Amazon-EC2-Instance muss über SSH oder AWS Systems Manager verbunden werden. Weitere Informationen finden Sie unter Verbindung zu AL203-Instances herstellen.

> **Important**
>
> ED25519-SSH-Benutzerschlüssel werden im FIPS-Modus nicht unterstützt. Wenn Sie Ihre Amazon-EC2-Instance mit einem ED25519-SSH-Schlüsselpaar gestartet haben, müssen Sie neue Schlüssel mit einem anderen Algorithmus (z. B. RSA) generieren. Andernfalls verlieren Sie möglicherweise den Zugriff auf Ihre Instance, nachdem Sie den FIPS-Modus aktivieren. Weitere Informationen finden Sie unter Create key pairs im Amazon EC2 EC2-Benutzerhandbuch.

Aktivieren des FIPS-Modus

1. Stellen Sie per SSH oder AWS Systems Manager eine Verbindung zu Ihrer AL2023-Instance her.
2. Stellen Sie sicher, dass das System auf dem neuesten Stand ist. Weitere Informationen finden Sie unter Paket- und Betriebssystemupdates in AL2023 verwalten.
3. Stellen Sie sicher, dass die `crypto-policies` Dienstprogramme installiert sind und up-to-date.

   ```
   sudo dnf -y install crypto-policies crypto-policies-scripts
   ```

4. Aktivieren Sie den FIPS-Modus mit folgendem Befehl.

   ```
   sudo fips-mode-setup --enable
   ```

5. Starten Sie die Instance mit dem folgenden Befehl neu.

   ```
   sudo reboot
   ```

6. Stellen Sie erneut eine Verbindung mit Ihrer Instance her und führen Sie den folgenden Befehl aus, um zu prüfen, ob der FIPS-Modus aktiviert ist.

```
sudo fips-mode-setup --check
```

In der folgenden Beispielausgabe sehen Sie, dass der FIPS-Modus aktiviert ist:

```
FIPS mode is enabled.
Initramfs fips module is enabled.
The current crypto policy (FIPS) is based on the FIPS policy.
```

AL2023 Kernhärtung

Der 6.1-Linux-Kernel in AL2023 ist mit verschiedenen Härtungsoptionen und -funktionen konfiguriert und gebaut.

Kernel-Hardening-Optionen (architekturunabhängig)

CONFIG-Option	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_ACPI_CUSTOM_METHOD	n	n
CONFIG_BINFMT_MISC	m	m
CONFIG_BUG	y	y
CONFIG_BUG_ON_DATA_CORRUPTION	y	y
CONFIG_CFI_CLANG	N/A	–
CONFIG_CFI_PERMISSIVE	–	–
CONFIG_COMPAT	y	y
CONFIG_COMPAT_BRK	n	n

CONFIG-Option	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_COMPAT_VDSO	–	n
CONFIG_DEBUG_CREDENTIALS	n	n
CONFIG_DEBUG_LIST	y	y
CONFIG_DEBUG_NOTIFIERS	n	n
CONFIG_DEBUG_SG	n	n
CONFIG_DEBUG_VIRTUAL	n	n
CONFIG_DEBUG_WX	n	n
CONFIG_DEFAULT_MMAP_MIN_ADDR	65536	65536
CONFIG_DEVKMEM	–	–
CONFIG_DEVMEM	n	n
CONFIG_EFI_DISABLE_PCI_DMA	n	n
CONFIG_FORTIFY_SOURCE	y	y
CONFIG_HARDENED_USERCOPY	y	y
CONFIG_HARDENED_USERCOPY_FALLBACK	–	–
CONFIG_HARDENED_USERCOPY_PAGESPAN	–	–

CONFIG-Option	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_HIBERNATION	y	y
CONFIG_HW_RANDOM_TPM	–	–
CONFIG_INET_DIAG	m	m
CONFIG_INIT_ON_ALLOC_DEFAULT_ON	n	n
CONFIG_INIT_ON_FREE_DEFAULT_ON	n	n
CONFIG_INIT_STACK_ALL_ZERO	–	–
CONFIG_IOMMU_DEFAULT_DMA_STRICT	n	n
CONFIG_IOMMU_SUPPORT	y	y
CONFIG_IO_STRICT_DEVMEM	–	–
CONFIG_KEXEC	y	y
CONFIG_KFENCE	n	n
CONFIG_LDISC_AUTOLOAD	n	n
CONFIG_LEGACY_PTYS	n	n
CONFIG_LOCK_DOWN_KERNEL_FORCE_CONFIDENTIALITY	n	n
CONFIG_MODULES	y	y

CONFIG-Option	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_MODULE_SIG	y	y
CONFIG_MODULE_SIG_ALL	y	y
CONFIG_MODULE_SIG_FORCE	n	n
CONFIG_MODULE_SIG_HASH	sha512	sha512
CONFIG_MODULE_SIG_KEY	certs/signing_key.pem	certs/signing_key.pem
CONFIG_MODULE_SIG_SHA512	y	y
CONFIG_PAGE_POISONING	n	n
CONFIG_PAGE_POISONING_NO_SANITY	–	–
CONFIG_PAGE_POISONING_ZERO	–	–
CONFIG_PANIC_ON_OOPS	y	y
CONFIG_PANIC_TIMEOUT	0	0
CONFIG_PROC_KCORE	y	y
CONFIG_RANDOMIZE_KSTACK_OFFSET_DEFAULT	n	n
CONFIG_RANDOM_TRUST_BOOTLOADER	y	y

CONFIG-Option	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_RANDOM_TRUST_CPU	y	y
CONFIG_REFCOUNT_FULL	–	–
CONFIG_SCHED_CORE	–	y
CONFIG_SCHED_STACK_END_CHECK	y	y
CONFIG_SECCOMP	y	y
CONFIG_SECCOMP_FILTER	y	y
CONFIG_SECURITY	y	y
CONFIG_SECURITY_DMESG_RESTRICT	y	y
CONFIG_SECURITY_LANDLOCK	n	n
CONFIG_SECURITY_LOCKDOWN_LSM	y	y
CONFIG_SECURITY_LOCKDOWN_LSM_EARLY	y	y
CONFIG_SECURITY_SELINUX_BOOTPARAM	y	y
CONFIG_SECURITY_SELINUX_DEVELOP	y	y
CONFIG_SECURITY_SELINUX_DISABLE	n	n

CONFIG-Option	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_SECURITY_WRITABLE_HOOKS	–	N/A
CONFIG_SECURITY_YAMA	y	y
CONFIG_SHUFFLE_PAGE_ALLOCATOR	y	y
CONFIG_SLAB_FREELIST_HARDENED	y	y
CONFIG_SLAB_FREELIST_RANDOM	y	y
CONFIG_SLUB_DEBUG	y	y
CONFIG_STACKPROTECTOR	y	y
CONFIG_STACKPROTECTOR_STRONG	y	y
CONFIG_STATIC_USERMODEHELPER	n	n
CONFIG_STRICT_DEVMEM	n	n
CONFIG_STRICT_KERNEL_RWX	y	y
CONFIG_STRICT_MODULE_RWX	y	y
CONFIG_SYN_COOKIES	y	y
CONFIG_VMAP_STACK	y	y
CONFIG_WERROR	n	n

CONFIG-Option	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_ZERO_CALL_USED_REGS	n	n

Erlaubt das Einfügen/Ersetzen von ACPI-Methoden zur Laufzeit (CONFIG_ACPI_CUSTOM_METHOD)

Diese Option ist in Amazon Linux deaktiviert, da sie `root`-Benutzern erlaubt, in beliebigen Kernelspeicher zu schreiben.

Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Verschiedene Binärformate (**`binfmt_misc`**)

Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt. In AL2023 ist diese Funktion optional und wurde als Kernelmodul erstellt.

`BUG()`-Support

Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

`BUG()` wenn der Kernel bei der Gültigkeitspprüfung der Kernel-Speicherstrukturen auf beschädigte Daten stößt

Einige Teile des Linux-Kernels prüfen die interne Konsistenz von Datenstrukturen und können `BUG()`, falls beschädigte Daten gefunden werden.

Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

`COMPAT_BRK`

Wenn diese Option deaktiviert ist (dies ist die Kernelkonfiguration in Amazon Linux), wird die `randomize_va_space-sysctl`-Einstellung standardmäßig auf 2 gesetzt, wodurch zusätzlich eine Randomisierungen der `mmap`-Basis-, -Stack- und -VDSO-Seite aktiviert wird.

Der Kernel bietet diese Option, um Kompatibilität mit einigen alten `libc.so.5`-Binärdateien aus 1996 und früher zu gewährleisten.

Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

`COMPAT_VDSO`

Diese Konfigurationsoption ist relevant für `x86-64` und nicht `aarch64`. Wenn Sie diese Option auf n setzen, macht der Amazon Linux-Kernel kein virtuelles dynamisches Shared Object (VDSO) mit 32 Bit an einer vorhersehbaren Adresse sichtbar. Die jüngste `glibc`, von der bekannt ist, dass sie durch die Festlegung dieser Option auf n beschädigt wurde, ist `glibc` 2.3.3 aus dem Jahr 2004.

Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

`CONFIG_DEBUG` Gated Hardening

Die Konfigurationsoptionen des Linux-Kernels, die von `CONFIG_DEBUG` gesteuert werden, sind normalerweise für die Verwendung in Kerneln konzipiert, die für Debugging-Probleme gebaut wurden. Hier haben Dinge wie Leistung keine Priorität. AL2023 `CONFIG_DEBUG_LIST` aktiviert die Härtungsoption.

Deaktivieren von DMA für PCI-Geräte im EFI-Stub vor der IOMMU-Konfigurierung

Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt.

-Hardening für Speicherkopierung zwischen Kernel und Userspace

Wenn der Kernel Speicher in den oder aus dem Userspace kopieren muss, aktiviert diese Option einige Prüfungen, die vor einigen Arten von Heap-Overflow-Problemen schützen können.

Die `CONFIG_HARDENED_USERCOPY_FALLBACK`-Option war bereits in den Kerneln 4.16 bis 5.15 vorhanden, um Kernel-Entwicklern zu helfen, fehlende Allowlist-Einträge mithilfe von `WARN()` aufzufinden. Da AL2023 einen 6.1-Kernel ausliefert, ist diese Option für AL2023 nicht mehr relevant.

Die `CONFIG_HARDENED_USERCOPY_PAGESPAN` Option existierte in Kerneln hauptsächlich als Debugging-Option für Entwickler und gilt nicht mehr für den 6.1-Kernel in AL2023.

Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Hibernation-Unterstützung

Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene

Einstellung gesetzt. Diese Option muss aktiviert sein, um Hibernation der On-Demand Instance und Hibernation für unterbrochene Spot Instances zu unterstützen.

Zufallszahlengenerierung

Der AL2023-Kernel ist so konfiguriert, dass sichergestellt ist, dass eine angemessene Entropie für die Verwendung in EC2 verfügbar ist.

`CONFIG_INET_DIAG`

Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt. In AL2023 ist diese Funktion optional und wurde als Kernelmodul erstellt.

Der gesamte Kernel-Page- und Slab-Allocator-Speicher wird bei Zuweisung und Freigabe auf Null gesetzt

Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt. Diese Optionen sind in AL2023 deaktiviert, da sich eine standardmäßige Aktivierung dieser Funktion möglicherweise auf die Leistung auswirken würde. Das `CONFIG_INIT_ON_ALLOC_DEFAULT_ON`-Verhalten kann durch Hinzufügen von `init_on_alloc=1` zur Kernel-Befehlszeile aktiviert werden, und das `CONFIG_INIT_ON_FREE_DEFAULT_ON`-Verhalten kann durch Hinzufügen von `init_on_free=1` aktiviert werden.

Alle Stack-Variablen als Null (`CONFIG_INIT_STACK_ALL_ZERO`) initialisieren

Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt. Für diese Option ist GCC 12 oder höher erforderlich, während AL2023 mit GCC 11 geliefert wird.

Signieren des Kernel-Moduls

AL2023 signiert und validiert die Signaturen von Kernelmodulen. Die `CONFIG_MODULE_SIG_FORCE`-Option, nach der Module über eine gültige Signatur verfügen müssten, ist nicht aktiviert, um die Kompatibilität für Benutzer zu gewährleisten, die Module von Drittanbietern erstellen. Für Benutzer, die sicherstellen möchten, dass alle Kernelmodule signiert sind, kann Lockdown Linux Security Module (LSM) so konfiguriert werden, dass eine Verifizierung erzwungen wird.

`kexec`

Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt. Diese Option ist aktiviert, sodass die `kdump`-Funktionalität genutzt werden kann.

`IOMMU`-Unterstützung

AL2023 aktiviert die IOMMU-Unterstützung. Die `CONFIG_IOMMU_DEFAULT_DMA_STRICT`-Option ist standardmäßig nicht aktiviert, aber diese Funktionalität kann durch eine Hinzufügung von `iommu.passthrough=0 iommu.strict=1` zur Kernel-Befehlszeile konfiguriert werden.

`kfence`

Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt.

Legacy-`pty`-Unterstützung

AL2023 verwendet die moderne PTY Schnittstelle (). `devpts`

Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Lockdown Linux Security Module (LSM)

AL2023 erstellt das `lockdown` LSM, das den Kernel automatisch sperrt, wenn Secure Boot verwendet wird.

Die `CONFIG_LOCK_DOWN_KERNEL_FORCE_CONFIDENTIALITY`-Option ist nicht aktiviert. Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt. Wenn Secure Boot nicht verwendet wird, kann das Lockdown-LSM aktiviert und nach Bedarf konfiguriert werden.

Page Poisoning

Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt. In ähnlicher Weise ist dies im AL2023-Kernel aufgrund möglicher Auswirkungen

auf die Leistung deaktiviert. Der gesamte Kernel-Page- und Slab-Allocator-Speicher wird bei Zuweisung und Freigabe auf Null gesetzt

Stack Protector

Der AL2023-Kernel wurde so gebaut, dass die Stack-Protector-Funktion mit der GCC Option aktiviert wurde. `-fstack-protector-strong`

Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

seccomp BPF-API

Das seccomp-Hardening-Feature wird von Software wie `systemd` und Container-Runtimes verwendet, um Userspace-Anwendungen zu schützen.

Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

`panic()` timeout

Der AL2023-Kernel ist so konfiguriert, dass dieser Wert auf gesetzt ist`0`, was bedeutet, dass der Kernel nicht neu gestartet wird, wenn er in Panik gerät. Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt. Dies kann über `sysctl /proc/sys/kernel/panic` oder über die Kernel-Befehlszeile konfiguriert werden.

Sicherheitsmodelle

AL2023 aktiviert SELinux standardmäßig im permissiven Modus. Weitere Informationen finden Sie unter Einstellung der SELinux-Modi für AL2023.

Die `yama`- und Lockdown Linux Security Module (LSM) -Module sind ebenfalls aktiviert.

`/proc/kcore`

Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt.

Randomisierung des Kernel-Stack-Offsets bei Eingabe von „syscall“

Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene

Einstellung gesetzt. Dies kann durch eine Einstellung `randomize_kstack_offset=on` in der Kernel-Befehlszeile aktiviert werden.

Prüfungen zur Referenzzählung (**`CONFIG_REFCOUNT_FULL`**)

Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt. Diese Option ist derzeit aufgrund möglicher Auswirkungen auf die Leistung nicht aktiviert.

Scheduler-Kenntnisnahme der SMT-Cores (**`CONFIG_SCHED_CORE`**)

Der AL2023-Kernel ist mit gebaut`CONFIG_SCHED_CORE`, was die Verwendung von Userspace-Anwendungen ermöglicht. `prctl(PR_SCHED_CORE)` Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Stack-Fehlerprüfung bei Aufrufen von **`schedule()`** (**`CONFIG_SCHED_STACK_END_CHECK`**)

Der AL2023-Kernel wurde mit aktivierter Option erstellt. `CONFIG_SCHED_STACK_END_CHECK` Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Memory Allocator Hardening

Der AL2023-Kernel ermöglicht das Härten des Kernel-Speicherzuweisers mit den Optionen`CONFIG_SHUFFLE_PAGE_ALLOCATOR`, und`CONFIG_SLAB_FREELIST_HARDENED`. `CONFIG_SLAB_FREELIST_RANDOM` Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

SLUB Debugging-Support

Der AL2023-Kernel aktiviert diese Option, `CONFIG_SLUB_DEBUG` da diese Option optionale Debugging-Funktionen für den Allocator aktiviert, die über die Kernel-Befehlszeile aktiviert werden können. Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

CONFIG_STATIC_USERMODEHELPER

Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene

Einstellung gesetzt. Dies liegt daran, dass `CONFIG_STATIC_USERMODEHELPER` besonderen Support von der Distribution benötigt, der derzeit in Amazon Linux nicht verfügbar ist.

Schreibgeschützter Kerneltext und rodata (**`CONFIG_STRICT_KERNEL_RWX`** und **`CONFIG_STRICT_MODULE_RWX`**)

Der AL2023-Kernel ist so konfiguriert, dass er Text und Speicher des Kernels und des Kernelmoduls als schreibgeschützt markiert und rodata Nicht-Text-Speicher als nicht ausführbar markiert. Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

TCP Syncookie-Support (**`CONFIG_SYN_COOKIES`**)

Der AL2023-Kernel wurde mit Unterstützung für TCP-Syncookies entwickelt. Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Virtuell zugeordneter Stack mit Guard-Seiten (**`CONFIG_VMAP_STACK`**)

Der AL2023-Kernel ist so gebaut`CONFIG_VMAP_STACK`, dass er virtuell zugeordnete Kernel-Stacks mit Schutzseiten ermöglicht. Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Build mit Compiler-Warnungen als Fehler angezeigt (**`CONFIG_WERROR`**)

Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt.

Nullstellung des Registers bei Funktionsbeendigung (**`CONFIG_ZERO_CALL_USED_REGS`**)

Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt.

Mindestadresse für Userspace-Zuweisungen

Diese Hardening-Option kann die Auswirkungen von Kernel-NULL-Pointer-Bugs reduzieren. Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

`clang` spezifische Hardening-Optionen

Der AL2023-Kernel ist mit und GCC nicht gebautclang, sodass die `CONFIG_CFI_CLANG` Hardening-Option nicht aktiviert werden kann, was ebenfalls keine Gültigkeit hat. `CONFIG_CFI_PERMISSIVE` Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt.

Spezifische Kernel-Hardening-Optionen für x86-64

CONFIG-Option	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_AMD_IOMMU	N/A	y
CONFIG_AMD_IOMMU_V2	–	y
CONFIG_IA32_EMULATION	–	y
CONFIG_INTEL_IOMMU	–	y
CONFIG_INTEL_IOMMU_DEFAULT_ON	–	n
CONFIG_INTEL_IOMMU_SVM	–	n
CONFIG_LEGACY_VSYSCALL_NONE	–	n
CONFIG_MODIFY_LDT_SYSCALL	–	n
CONFIG_PAGE_TABLE_ISOLATION	–	y
CONFIG_RANDOMIZE_MEMORY	–	y
CONFIG_X86_64	–	y

CONFIG-Option	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_X86_MSR	–	y
CONFIG_X86_VSYSCALL_EMULATION	–	y
CONFIG_X86_X32	–	–
CONFIG_X86_X32_ABI	N/A	n

x86-64-Unterstützung

Basis-Support für x86-64 umfasst die Unterstützung für Physical Address Extension (PAE) und No-Execute-Bits (NX). Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

AMD- und Intel-IOMMU-Support

Der AL2023-Kernel wird mit Unterstützung für AMD und Intel gebaut. IOMMUs Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Die `CONFIG_INTEL_IOMMU_DEFAULT_ON`-Option ist nicht gesetzt, kann aber durch Übergabe von `intel_iommu=on` an die Kernel-Befehlszeile aktiviert werden. Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt.

Die `CONFIG_INTEL_IOMMU_SVM` Option ist derzeit in AL2023 nicht aktiviert. Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt.

Support für 32-Bit-Userspace

> **Important**
>
> Support für 32-Bit-x86-Userspace ist veraltet und Unterstützung für die Ausführung von 32-Bit-Userspace-Binärdateien wird möglicherweise in einer zukünftigen Hauptversion von Amazon Linux entfernt.

> **Note**
>
> AL2023 enthält zwar keine 32-Bit-Pakete mehr, aber der Kernel unterstützt weiterhin die Ausführung von 32-Bit-Benutzerbereichen. Weitere Informationen finden Sie unter 32-Bit x86-(i686)-Pakete.

Um die Ausführung von 32-Bit-Userspace-Anwendungen zu unterstützen, aktiviert AL2023 die `CONFIG_X86_VSYSCALL_EMULATION` Option nicht und aktiviert stattdessen die Optionen, und. `CONFIG_IA32_EMULATION` `CONFIG_COMPAT` `CONFIG_X86_VSYSCALL_EMULATION` Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt.

Die x32-native 32-Bit-ABI für 64-Bit-Prozessoren ist nicht aktiviert (`CONFIG_X86_X32` und `CONFIG_X86_X32_ABI`). Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

x86 Model Specific Register (MSR)-Support

Die `CONFIG_X86_MSR`-Option ist aktiviert, um Unterstützung für `turbostat` zuzulassen. Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt.

`modify_ldt`-Syscall

AL2023 erlaubt Benutzerprogrammen nicht, die x86-Local Descriptor Table (LDT) mit dem Syscall zu ändern. `modify_ldt` Dieser Aufruf ist erforderlich, um 16-Bit-Code oder segmentierten Code auszuführen, und sein Fehlen kann dazu führen, dass Software wie `dosemu`, das Ausführen einiger Programme unter WINE und einige sehr alte Threading-Bibliotheken beschädigt werden könnte. Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Entfernen der Kernelzuordnung im Benutzermodus

AL2023 konfiguriert den Kernel so, dass die Mehrheit der Kerneladressen nicht dem Benutzerbereich zugeordnet wird. Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Randomisieren von Kernel-Speicherbereichen

AL2023 konfiguriert den Kernel so, dass die virtuellen Basisadressen der Kernel-Speicherbereiche nach dem Zufallsprinzip sortiert werden. Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

aarch64-spezifische Kernel-Hardening-Optionen

CONFIG-Option	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_ARM64_BTI	y	N/A
CONFIG_ARM64_BTI_KERNEL	–	–
CONFIG_ARM64_PTR_AUTH	y	–
CONFIG_ARM64_PTR_AUTH_KERNEL	y	–
CONFIG_ARM64_SW_TTBR0_PAN	y	–
CONFIG_UNMAP_KERNEL_AT_EL0	y	N/A

Identifizierung des -Abzweigungsziels

Der AL2023-Kernel ermöglicht die Unterstützung von Branch Target Identification (). `CONFIG_ARM64_BTI` Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Die `CONFIG_ARM64_BTI_KERNEL`-Option ist in AL2023 nicht aktiviert, da sie mit GCC konzipiert wurde, und da Unterstützung für Kernelerstellungen mit dieser Option derzeit im Upstream-Kernel deaktiviert ist (Grund: ein Gcc-Bug). Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, jedoch wird diese Konfigurationsoption in AL2023 nicht auf die von KSPP empfohlene Einstellung gesetzt.

Pointer-Authentifizierung (**CONFIG_ARM64_PTR_AUTH**)

Der AL2023-Kernel wurde mit Unterstützung für die Pointer Authentication-Erweiterung (Teil der ARMv8.3-Erweiterungen) entwickelt, die zur Abschwächung von ROP-Techniken (Return Oriented Programming) verwendet werden kann. Die erforderliche Hardwareunterstützung für Pointer-Authentifizierung unter Graviton wurde mit Graviton 3 eingeführt.

Die `CONFIG_ARM64_PTR_AUTH`-Option ist aktiviert und unterstützt die Pointer-Authentifizierung für den Userspace. Da die `CONFIG_ARM64_PTR_AUTH_KERNEL` Option ebenfalls aktiviert ist, kann der AL2023-Kernel den Schutz der Absenderadresse für sich selbst nutzen.

Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Privilegierten Zugriff emulieren – Niemals **TTBR0_EL1**-Switching verwenden

Diese Option verhindert direkten Kernel-Zugriff auf den Userspace-Speicher. `TTBR0_EL1` wird von den Benutzerzugriffsroutinen nur vorübergehend auf einen gültigen Wert gesetzt.

Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Entfernen der Kernel-Zuweisung bei Ausführung im Userspace

Der AL2023-Kernel ist so konfiguriert, dass er die Zuordnung des Kernels aufhebt, wenn er im Userspace () ausgeführt wird. `CONFIG_UNMAP_KERNEL_AT_EL0` Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

UEFI Secure Boot auf AL2023

AL2023 unterstützt UEFI Secure Boot ab Version 2023.1. AL2023 muss mit Amazon-EC2-Instances verwendet werden, die sowohl UEFI als auch UEFI Secure Boot unterstützen. Weitere Informationen finden Sie unter Launch an Instance im Amazon EC2 EC2-Benutzerhandbuch.

AL203-Instances mit aktiviertem UEFI Secure Boot akzeptieren nur Code auf Kernelebene, einschließlich des Linux-Kernels sowie Module, die signiert sind, Amazon sodass Sie sicherstellen können, dass Ihre Instance nur Codes auf Kernelebene ausführt, die von signiert wurden. AWS

Weitere Informationen zu Amazon EC2 EC2-Instances und UEFI Secure Boot finden Sie unter UEFI Secure Boot im Amazon EC2 EC2-Benutzerhandbuch.

Voraussetzungen

- Mit AL2023 Version 2023.1 oder höher muss ein AMI verwendet werden.

- Der Instance-Typ muss UEFI Secure Boot unterstützen. Weitere Informationen finden Sie unter Launch an Instance im Amazon EC2 EC2-Benutzerhandbuch.

Aktivieren Sie UEFI Secure Boot auf AL2023

Standard-AL2023-AMIs enthalten einen Bootloader und einen Kernel, der mit unseren Schlüsseln signiert ist. Sie können UEFI Secure Boot aktivieren, indem Sie entweder vorhandene Instances registrieren oder AMIs mit vorab aktiviertem UEFI Secure Boot erstellen, für die Sie ein Image aus einem Snapshot registrieren. UEFI Secure Boot ist standardmäßig in den Standard-AL2023-AMIs aktiviert.

Der Startmodus von AL2023-AMIs ist auf `uefi-preferred` gesetzt, was sicherstellt dass mit diesen AMIs gestartete Instances die UEFI-Firmware verwenden, sofern der Instance-Typ UEFI unterstützt. Sollte der Instance-Typ UEFI nicht unterstützen, dann wird die Instance mit Legacy-BIOS gestartet. Wird eine Instance im Legacy-BIOS-Modus gestartet, so wird UEFI Secure Boot nicht durchgesetzt.

Weitere Informationen zu AMI-Startmodi auf Amazon EC2 EC2-Instances finden Sie unter Startmodi im Amazon EC2 EC2-Benutzerhandbuch.

Themen

- Registrierung einer vorhandenen Instance
- Image aus einem Snapshot registrieren
- Widerruf-Updates
- Wie funktioniert UEFI Secure Boot auf AL2023
- Eigene Schlüssel registrieren

Registrierung einer vorhandenen Instance

Wenn Sie eine vorhandene Instanz registrieren möchten, befüllen Sie die spezifischen UEFI-Firmware-Variablen mit einem Schlüsselsatz, die der Firmware erlaubt, beim nächsten Start den Bootloader zu verifizieren und dem Bootloader erlaubt, den Kernel zu verifizieren.

1. Amazon Linux bietet ein Tool zur Vereinfachung des Registrierungsprozesses. Mit dem folgenden Befehl stellen Sie der Instance den erforderlichen Satz von Schlüsseln und Zertifikaten bereit.

```
sudo amazon-linux-sb enroll
```

2. Führen Sie den folgenden Befehl aus, um die -Instance neu zu starten. UEFI Secure Boot wird nach dem Neustart der Instanz aktiviert.

```
sudo reboot
```

> **Note**
>
> Amazon-Linux-AMIs unterstützt das Nitro Trusted Platform Module (NitroTPM) nicht. Wenn Sie NitroTPM zusätzlich zu UEFI Secure Boot benötigen, helfen Ihnen die Informationen im folgenden Abschnitt weiter.

Image aus einem Snapshot registrieren

Wenn Sie mithilfe der Amazon-`register-image`-API ein AMI aus einem Snapshot eines Amazon EBS-Root-Volumes registrieren, können Sie das AMI mit einem binären Blob bereitstellen, der den Status des UEFI-Variablenspeichers enthält. Durch die Bereitstellung von AL2023-`UefiData` aktivieren Sie UEFI Secure Boot und können die Anleitungen im obigen Abschnitt ignorieren.

Weitere Informationen zum Erstellen und Verwenden eines binären Blobs finden Sie unter Option B: Erstellen Sie ein binäres Blob mit einem vorausgefüllten Variablenspeicher im Amazon EC2 EC2-Benutzerhandbuch.

AL2023 bietet einen vorkonfigurierten binären Blob, der direkt auf Amazon-EC2-Instances verwendet werden kann. Der binäre Blob befindet sich auf einer laufenden Instance in `/usr/share/amazon-linux-sb-keys/uefi.vars`. Dieser Blob wird durch das `amazon-linux-sb-keys`-RPM-Paket bereitgestellt, das ab Version 2023.1 standardmäßig in AL2023-AMIs installiert ist.

> **Note**
>
> Wenn Sie sicherstellen wollen, dass Sie die neueste Version von Schlüsseln und Widerrufungen nutzen, verwenden Sie den Blob aus derselben Version von AL2023, mit der Sie das AMI erstellt haben.

Wir empfehlen, bei der Registrierung eines Images den `BootMode`-Parameter der `RegisterImage`-API auf `uefi` zu setzen. Das widerum erlaubt Ihnen, NitroTPM zu aktivieren, indem Sie den

`TpmSupport`-Parameter auf `v2.0` setzen. Die Einstellung des Parameters `BootMode` auf `uefi` stellt außerdem sicher, dass UEFI Secure Boot aktiviert ist und nicht versehentlich deaktiviert werden kann, wenn zu einem Instance-Typ gewechselt wird, der UEFI nicht unterstützt.

Weitere Informationen zu NitroTPM finden Sie unter NitroTPM im Amazon EC2 EC2-Benutzerhandbuch.

Widerruf-Updates

Möglicherweise muss Amazon Linux eine neue Version des Bootloaders `grub2` oder des Linux-Kernels verteilen, die mit aktualisierten Schlüsseln signiert ist. In diesem Fall muss der alte Schlüssel möglicherweise widerrufen werden, um zu verhindern, dass ausnutzbare Bugs aus früheren Versionen des Bootloaders den UEFI-Secure-Boot-Verifizierungsprozess umgehen können.

Paketaktualisierungen der `kernel`- oder `grub2`-Pakete aktualisieren die Liste der Widerrufe immer automatisch im UEFI-Variablenspeicher der laufenden Instanz. Das bedeutet, dass wenn UEFI Secure Boot aktiviert ist, die alte Version eines Pakets nicht mehr ausgeführt werden kann, nachdem ein Sicherheits-Update für das Paket installiert wurde.

Wie funktioniert UEFI Secure Boot auf AL2023

Im Gegensatz zu anderen Linux-Distributionen bietet Amazon Linux keine zusätzliche Komponente (Shim), die als Bootloader der ersten Stufe fungiert. Ein Shim wird in der Regel mit Microsoft-Schlüsseln signiert. Bei Linux-Distributionen mit dem Shim lädt der Shim beispielsweise den `grub2`-Bootloader, der den eigenen Code des Shims verwendet, um den Linux-Kernel zu verifizieren. Außerdem verwaltet der Shim seinen eigenen Satz von Schlüsseln und Widerrufen in der MOK-Datenbank (Machine Owner Key), die sich im UEFI-Variablenspeicher befindet und mit dem `mokutil`-Tool gesteuert wird.

Amazon Linux stellt keinen Shim zur Verfügung. Da der AMI-Besitzer Kontrolle über die UEFI-Variablen hat, ist dieser Zwischenschritt nicht erforderlich und würde sich negativ auf die Start- und Boot-Zeiten auswirken. Wir haben uns dafür entschieden, standardmäßig keinen Herstellerschlüsseln zu vertrauen, um die Wahrscheinlichkeit zu verringern, dass unerwünschte Binärdateien ausgeführt werden könnten. Wie immer können Kunden natürlich eigene Binärdateien hinzufügen, wenn sie dies wünschen.

Bei Amazon Linux lädt und verifiziert UEFI unseren `grub2`-Bootloader direkt. Der `grub2`-Bootloader wurde so geändert, dass er UEFI zur Verifizierung des Linux-Kernels nach dem Laden verwendet. Der Linux-Kernel wird also mit denselben Zertifikaten verifiziert, die in der normalen UEFI-db-

Variable (autorisierte Schlüsseldatenbank) gespeichert sind, und anhand derselben `dbx`-Variable (Sperrdatenbank) wie der Bootloader und andere UEFI-Binärdateien getestet. Da wir unsere eigenen PK- und KEK-Schlüssel bereitstellen, die den Zugriff auf die DB-Datenbank und die DBX-Datenbank steuern, können wir signierte Updates und Widerrufe nach Bedarf ohne Zwischenhändler wie den Shim verteilen.

Weitere Informationen zu UEFI Secure Boot finden Sie unter So funktioniert UEFI Secure Boot im Amazon EC2 EC2-Benutzerhandbuch.

Eigene Schlüssel registrieren

Wie im vorherigen Abschnitt dokumentiert, benötigt Amazon Linux keinen `shim` für UEFI Secure Boot auf Amazon EC2. Wenn Sie die Dokumentation für andere Linux-Distributionen lesen, finden Sie möglicherweise Informationen zur Verwaltung der MOK-Datenbank (Machine Owner Key) mithilfe von `mokutil`, was unter AL2023 nicht vorhanden ist. Die `shim`- und MOK-Umgebungen umgehen einige Einschränkungen der Schlüsselregistrierung in der UEFI-Firmware, die nicht auf die Implementierung von UEFI Secure Boot durch Amazon EC2 zutreffen. Bei Amazon EC2 gibt es Mechanismen, um die Schlüssel im UEFI-Variablenspeicher ganz einfach direkt zu manipulieren.

Wenn Sie Ihre eigenen Schlüssel registrieren möchten, können Sie entweder den Variablenspeicher innerhalb einer vorhandenen Instance bearbeiten (siehe Schlüssel aus der Instance zum Variablenspeicher hinzufügen) oder einen vorausgefüllten binären Blob erstellen, der vorgefüllt ist (siehe Erstellen eines binären Blobs mit vorgefülltem Variablenspeicher).

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.