aws

Benutzerhandbuch

Amazon Linux 2023

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Linux 2023: Benutzerhandbuch

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist Amazon Linux 2023? ... 1
Release-Taktfrequenz ... 1
Haupt- und Nebenversionen ... 2
Aufnahme neuer Versionen ... 2
Langfristige Support-Richtlinie ... 2
Benennung und Versionsverwaltung ... 3
Leistungs- und Betriebsoptimierungen ... 4
Beziehung zu Fedora ... 5
Maßgeschneidert cloud-init ... 6
Sicherheits-Updates und Features ... 7
Verwaltete Updates ... 8
Sicherheit in der Cloud ... 8
SELinux Modi ... 8
Compliance-Programm ... 8
SSH-Server-Standard ... 8
Hauptfunktionen von OpenSSL 3 ... 9
Netzwerkdienst ... 9
Core-Toolchain-Pakete glibc, gcc, binutils ... 10
Paketmanagement-Tool ... 11
Standard-SSH-Serverkonfiguration ... 11
Veraltete Funktionalität ... 14
`compat-`-Pakete ... 14
Veraltete Funktionen wurden eingestellt in AL1, entfernt in AL2 ... 14
32-Bit-x86 (i686) AMIs ... 15
`aws-apitools-*`ersetzt durch AWS CLI ... 15
`systemd`ersetzt in `upstart` AL2 ... 16
Funktionalität wurde in 023 veraltet AL2 und wurde in Version 023 entfernt AL2 ... 16
32-Bit-x86-Pakete (i686) ... 17
`aws-apitools-*`ersetzt durch AWS CLI ... 18
`amazon-cloudwatch-agent`ersetzt `awslogs` ... 18
`bzr`System zur Versionskontrolle ... 18
cgroup v1 ... 19
log4jHotpatch () `log4j-cve-2021-44228-hotpatch` ... 19
`lsb_release` und das `system-lsb-core`-Paket ... 19

mcrypt ... 20
OpenJDK (7) java-1.7.0-openjdk ... 20
Python 2.7 ... 20
rsyslog-opensslersetzt rsyslog-gnutls ... 21
Netzwerkinformationsdienst (NIS)/yp ... 21
Mehrere Domainnamen in Amazon VPC create-dhcp-options ... 21
Sun RPC in glibc ... 22
OpenSSH-Schlüssel-Fingerabdruck im Protokoll audit ... 22
ld.goldLinker ... 22
ping6 ... 23
ftpPackage ... 23
In 023 veraltet AL2 ... 25
32-Bit-x86-Laufzeitunterstützung (i686) ... 25
aspell ... 25
Datenbank Berkeley () libdb ... 26
cron ... 26
IMDSv1 ... 27
pcre Version 1 ... 27
System V init (sysvinit) ... 27
EOL-Pakete sind veraltet ... 28
Vergleich AL2 und 023 AL2 ... 29
Hinzugefügte, aktualisierte und entfernte Pakete ... 30
Support für die einzelnen Versionen ... 30
Änderungen bei der Benennung und Versionierung ... 30
Optimierungen ... 31
Aus mehreren Upstream-Quellen ... 31
Netzwerksystem-Dienst ... 31
Paketmanager ... 31
Verwenden von cloud-init ... 31
Grafische Desktop-Unterstützung ... 32
Compiler-Triplet ... 32
32-Bit x86-(i686)-Pakete ... 32
lsb_release und das system-lsb-core-Paket ... 33
EPEL ... 33
axel- HTTP/FTP-Client ... 35
brotliund libbrotli - Komprimierung ... 36

collectd- Daemon zur Erfassung von Statistiken 36
cpulimit 36
exim- Agent für die E-Mail-Übertragung 36
fuse3- Dateisystem im Userspace (FUSE) v3 37
ganglia- Verteiltes Überwachungssystem 37
git-lfs- Versionskontrolle großer Dateien mit Git 37
haveged- eine Entropiequelle unter Verwendung der HAVEGE Algorithmus 37
inotify-tools- inotify-Befehlszeilentools 37
iperf- TCP/UDP-Leistungs-Benchmark 38
jemalloc- alternative Implementierung malloc 38
libbsd- BSD-kompatible Funktionsbibliothek 39
libserf- HTTP-Clientbibliothek 39
libzstd- ZSTD-Kompressionsbibliothek 39
lighttpdWebserver 39
lshell- eine eingeschränkte Shell 39
monit- Prozess-, Datei-, Verzeichnis- und Geräteüberwachung 40
nodejs 40
perl-Config-General 41
python2-lockfile- Sperren von Dateien 41
python2-rsa- reines Python RSA 41
python2-simplejson- JSON-Routinen für Python 2 42
rkhunter- Rootkit Hunter 42
rssh- eine eingeschränkte Shell zur Verwendung mit OpenSSH 42
sscg- Generator für selbstsignierte SSL-Zertifikate 43
stress- Stresstest 43
stress-ng- Stresstest 43
tmpwatch- entfernt Dateien basierend auf dem Zeitpunkt des letzten Zugriffs 43
xmlstarlet- XML-Hilfsprogramme für die Befehlszeile 43
Python 2.7 wurden durch Python 3 ersetzt 44
Sicherheits-Updates 44
SELinux 45
OpenSSL 3 45
IMDSv2 45
Entfernen des log4j-Hotpatch (log4j-cve-2021-44228-hotpatch) 46
Deterministische Upgrades für Stabilität 46
gp3als standardmäßiger Amazon EBS-Volumetyp 47

Vereinheitlichte Kontrollgruppenhierarchie (cgroup v2) 47
systemdTimer ersetzen cron 48
Verbesserte Toolchain: gccbinutils, und glibc 48
systemdJournal ersetzt rsyslog 49
Minimierte Paketabhängigkeiten 49
Paketänderungen für curl und libcurl 50
GNU Privacy Guard (GNUPG) 50
Amazon Corretto als Standard-JVM 50
AWS CLI v2 51
UEFI Preferred und Secure Boot 51
Änderungen der Standardkonfiguration des SSH-Servers 51
Der Kernel ändert sich in AL2 023 von AL2 51
IPv4 TTL 52
Sicherheitsorientierte Änderungen an der Kernel-Konfiguration 52
Weitere Änderungen in der Kernelkonfiguration 56
Unterstützung für das Kernel-Dateisystem 58
/tmpÄnderungen 63
Änderungen an AMI und Container-Images 63
Vergleich von Amazon Linux 2 und AL2 023 AMI 64
Vergleich von Amazon Linux 2 und AL2 023 mit minimalem AMI 97
Vergleich von Amazon Linux 2 und AL2 023 Container 118
Vergleich AL1 und 023 AL2 126
Support für die einzelnen Versionen 126
systemd ersetzt upstart als init-System 127
Python 2.6 und 2.7 wurden durch Python 3 ersetzt 127
OpenJDK 8 als ältestes JDK 127
Kernel-Änderungen in AL2 023 von AL1 127
Kernel-Live-Patching 127
Unterstützung für das Kernel-Dateisystem 127
Sicherheitsorientierte Änderungen an der Kernel-Konfiguration 130
Weitere Änderungen in der Kernelkonfiguration 132
AL1 und AL2 023 AMI-Vergleich 133
AL1 und AL2 023 Minimaler AMI-Vergleich 167
AL1 und AL2 023 Container-Vergleich 188
Systemanforderungen 197
CPU-Anforderungen für die Ausführung von 023 AL2 197

ARM-CPU-Anforderungen für AL2 023 ... 197
x86-64-CPU-Anforderungen für 023 AL2 ... 198
Speicheranforderungen (RAM) für die Ausführung von 023 AL2 ... 199
Grafischer Desktop ... 200
Verwandte Themen ... 200
Anwendungen werden ausgeführt ... 201
Ressourcenkontrolle mit systemd ... 201
Ressourcensteuerung mit `systemd-run` zum Ausführen einmaliger Befehle ... 202
Ressourcenkontrolle in einem `systemd` Dienst ... 205
Die Verwendung von cgroups Dienstprogramme ... 209
Verwenden Sie AL2 023 auf AWS ... 212
Erste Schritte mit AWS ... 212
Melde dich an für ein AWS-Konto ... 212
Erstellen eines Benutzers mit Administratorzugriff ... 213
Erteilen programmgesteuerten Zugriffs ... 214
AL2023 bei Amazon EC2 ... 216
AL2023 über die EC2 Amazon-Konsole starten ... 217
AL2023 mit dem SSM-Parameter starten und AWS CLI ... 218
Starten des neuesten AL2 023-AMI mit AWS CloudFormation ... 219
AL2023 mit einer bestimmten AMI-ID starten ... 221
AL2023 AMI-Verfall und Lebenszyklus ... 221
Verbindung zu 023 Instances herstellen AL2 ... 222
Vergleich von AL2 023 Standard (Standard) und Minimal AMIs ... 222
AL2023 in Behältern ... 250
AL2023 Basis-Container-Image ... 251
AL2023 Minimales Container-Image ... 253
Erstellung von Bare-Bones AL2 (023) Container-Images ... 255
AL20.23: Vergleich der Liste der Container-Image-Pakete ... 259
AL2023 Minimales AMI im Vergleich zu Container-Images ... 265
AL2023 auf Elastic Beanstalk ... 282
AL2023 CloudShell ... 282
AL2023 für Amazon ECS-Container-Hosts ... 283
Amazon ECS relevante Änderungen seit AL2 ... 283
Benutzerdefinierte Amazon-ECS-optimierte AMIs ... 284
Amazon EFS auf AL2 023 ... 284
`amazon-efs-utils` ... 285

Ein Amazon-EFS-Dateisystems mounten ... 285
Amazon EMR auf 023 AL2 ... 286
AL2Amazon EMR-Versionen auf Basis von 2023 ... 286
AL2023 basiert auf Amazon EMR auf EKS ... 286
AL2023 ein AWS Lambda ... 286
`provided.al2023`-Lambda-Laufzeit ... 286
AL2023-basierte Laufzeiten ... 287
Tutorials ... 288
Installieren Sie LAMP auf 023 AL2 ... 288
Schritt 1: Vorbereiten des LAMP-Servers ... 289
Schritt 2: Testen Ihres Lamp-Servers ... 294
Schritt 3: Sichern des Datenbankservers ... 296
Schritt 4: (Optional) Installieren phpMyAdmin ... 297
Fehlerbehebung ... 300
Verwandte Themen ... 301
Konfigurieren Sie SSL/TLS auf 023 AL2 ... 302
Voraussetzungen ... 303
Schritt 1: Aktivieren von TLS auf dem Server ... 304
Schritt 2: Abrufen eines CA-signierten Zertifikats ... 307
Schritt 3: Testen und Verstärken der Sicherheitskonfiguration ... 316
Fehlerbehebung ... 320
Hosten Sie einen WordPress Blog auf 023 AL2 ... 321
Voraussetzungen ... 321
Installieren WordPress ... 322
Nächste Schritte ... 333
Hilfe! Mein öffentlicher DNS-Name hat sich geändert und jetzt funktioniert mein Blog nicht mehr. ... 334
Übergang von Redis 6 zu Valkey auf 023 AL2 ... 335
Supportzeitplan für Redis 6 ... 336
Einführung in Valkey ... 336
Migrationsplan und Zeitplan ... 336
Migrationsoptionen und -schritte ... 336
Verwandte Themen ... 340
Installieren Sie GNOME auf 023 AL2 ... 340
Voraussetzungen ... 340
Installation ... 341

Verwandte Themen ... 341
Konfigurieren Sie VNC auf 023 AL2 ... 341
Voraussetzungen ... 342
Schritt 1: Installation ... 342
Schritt 2: Konfiguration ... 342
Schritt 3: Connect mit einem VNC-Client her ... 344
(Optional) Starten Sie den Dienst beim Booten ... 345
(Optional) Deaktivieren Sie den Sperrbildschirm im Leerlauf ... 345
Verwandte Themen ... 346
AL2023 außerhalb von Amazon EC2 ... 347
Laden Sie 023 VM-Images herunter AL2 ... 347
Unterstützte Konfigurationen ... 347
KVM-Anforderungen ... 348
VMware Voraussetzungen ... 350
Hyper-V-Anforderungen ... 353
AL2023 VM-Konfiguration ... 355
NoCloud `seed.iso`-basierte Konfiguration ... 356
VMware Konfiguration auf Guestinfo-Basis ... 359
AL2023 Paketlistenvergleich für das Standard-AMI und das KVM-Image ... 362
AL2023 Paketlistenvergleich für das Standard-AMI- und VMware OVA-Image ... 387
AL2023 Paketlistenvergleich für das Standard-AMI- und Hyper-V-Image ... 412
Identifizieren von Amazon Linux-Versionen ... 438
`/etc/os-release` ... 438
Die wichtigsten Unterschiede: ... 439
Feldtypen ... 439
Beispiele für `/etc/os-release` ... 441
Vergleich mit anderen Distributionen ... 442
Amazon Linux-spezifisch ... 444
`/etc/sytem-release` ... 445
`/etc/image-id` ... 445
Spezifische Beispiele für Amazon Linux ... 446
Beispiel-Code ... 448
Aufbau des Dateisystems ... 461
`/` ... 461
`/boot` ... 462
`/boot/efi` ... 462

/etc ... 462
/home ... 462
/root ... 463
/srv ... 463
/tmp ... 464
/run ... 465
/usr ... 465
/usr/bin ... 466
/usr/include ... 466
/usr/lib und /usr/lib64 ... 466
/usr/local ... 466
/usr/share ... 467
/var ... 467
/var/cache ... 467
/var/lib ... 467
/var/log ... 467
/var/spool ... 468
/var/tmp ... 468
Aktualisierung 023 AL2 ... 470
Bewährte Methoden für die sichere Bereitstellung von Updates ... 470
Vorbereitung kleinerer Updates ... 474
Wir bereiten uns auf wichtige Updates vor ... 474
Erhalten Sie Benachrichtigungen über neue Updates ... 475
Deterministische Upgrades durch versionierte Repositorys ... 476
Kontrolle über die Updates, die Sie aus Haupt- und Nebenversionen erhalten ... 476
Unterschiede zwischen Haupt- und Nebenversions-Upgrades ... 477
Wissen, wann Updates verfügbar sind ... 477
Kontrollieren Sie die Paket-Updates, die in den AL2 023-Repositorys verfügbar sind ... 478
Ersatz von Instanzen ... 478
Deterministische Upgrades vor Ort ... 479
Verwalten von Aktualisierungen ... 487
Prüfen auf verfügbare Paket-Updates ... 488
Anwenden von Sicherheitsupdates mit DNF und Repository-Versionen ... 493
Automatischer Neustart des Dienstes nach (Sicherheits-) Updates ... 506
Wann ist ein Neustart erforderlich, um Sicherheitsupdates anzuwenden? ... 507
Starten einer Instance mit aktivierter neuester Repository-Version ... 508

Abrufen von Paketunterstützunginformationen 508
`dnf check-release-update` 509
Hinzufügen, aktivieren oder deaktivieren neuer Repositorys 513
Hinzufügen von Repositorys mit cloud-init 516
Kernel-Live-Patching 517
Einschränkungen 518
Unterstützte Konfigurationen und Voraussetzungen 518
Arbeiten mit Kernel-Live-Patching 519
Kernel-Aktualisierungen 524
Linux-Kernel-Versionen auf 023 AL2 525
Aktualisierung von 0.23 auf Kernel 6.12 AL2 525
AL2023 Kernel — Häufig gestellte Fragen 528
Programmiersprachen und Laufzeiten 529
C/C++ und Fortran 529
GCC14 530
Vergleich der Sprachstandardversionen 531
Go 533
AL2023 Lambda-Funktion: Go 533
Java 534
NodeJS 40
Perl 535
Perl modules 536
PHP 536
Migration zu einem neuen PHP versions 536
Migration von PHP 7.x 537
PHP modules 537
Python 537
Python modules 538
Rust 538
AL2023 Lambda-Funktion: Rust 539
AL2023 Reservierte Benutzer und Gruppen 540
Liste der AL2 023 reservierten Benutzer 540
Liste der AL2 023 reservierten Gruppen 548
Codecs sind in 023 verfügbar AL2 561
Sicherheit und Compliance 563
Sicherheitsempfehlungen 564

ALAS Announcements 564
LEIDER FAQs 565
ALAS-Empfehlungen 565
Hinweise und RPM-Repositorys 565
Hinweis IDs 566
Veröffentlichungsdatum der Empfehlung und Aktualisierungsdatum der Empfehlung 567
Arten von Hinweisen 567
Schweregrade der Empfehlung 567
Ratgeber und Pakete 568
Ratschläge und CVEs 569
Text des Hinweises 570
Kernel Live Patch Advisories 570
`updateinfo.xml`-Schema 571
Liste der zutreffenden Hinweise 572
Direkte Updates 576
Anwenden von Updates, die in einem Advisory erwähnt werden 576
SELinux Einstellmodi für AL2 023 580
SELinux Standardstatus und -modi für AL2 023 580
Wechseln in den `enforcing`-Modus 581
Option zum Deaktivieren SELinux 583
Aktivieren Sie den FIPS-Modus auf 023 AL2 584
Aktivieren Sie den FIPS-Modus in einem AL2 023-Container 586
Tauschen Sie OpenSSL FIPS-Anbieter auf 023 aus AL2 588
Kernel-Hardening 590
Kernel-Hardening-Optionen (architekturunabhängig) 590
Spezifische Kernel-Hardening-Optionen für x86-64 606
aarch64-spezifische Kernel-Hardening-Optionen 610
UEFI Secure Boot auf 023 AL2 612
Aktivieren Sie UEFI Secure Boot auf 023 AL2 612
Registrierung einer vorhandenen Instance 613
Image aus einem Snapshot registrieren 614
Widerruf-Updates 614
So funktioniert UEFI Secure Boot auf 023 AL2 615
Eigene Schlüssel registrieren 615
..... dcxvii

Was ist Amazon Linux 2023?

Amazon Linux 2023 (AL2023) ist die nächste Generation von Amazon Linux von Amazon Web Services (AWS). Mit AL2 023 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 Innovationen in Linux bietet. AL2023 wird ohne zusätzliche Kosten zur Verfügung gestellt.

AL2023 ist der Nachfolger von Amazon Linux 2 (AL2). Informationen zu den Unterschieden zwischen AL2 023 Vergleich AL2 und AL2 023 und AL2 finden Sie unter Paketänderungen in AL2 023.

Themen

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

Release-Taktfrequenz

Amazon Linux 2023 (AL2023) wurde im März 2023 veröffentlicht und wird bis zum 30. Juni 2029 unterstützt. Es gibt zwei Unterstützungsphasen:

- Standard-Support — Während dieser Phase erhält die Version vierteljährlich kleinere Versionsupdates. Die Standard-Supportphase endet am 30. Juni 2027.
- Wartung — Während dieser Phase erhält die Version nur Sicherheitsupdates und kritische Bugfixes. Diese Updates werden veröffentlicht, sobald sie verfügbar sind. Die Wartungsphase endet am 30. Juni 2029.

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 könnten größere Änderungen am Kernel, der Toolchain, beinhalten Glib C, OpenSSLund alle anderen Systembibliotheken und Dienstprogramme. Hauptversionen von Amazon Linux basieren teilweise auf der aktuellen Version der Fedora-Linux-Upstream-Distribution. AWS könnte spezifische 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 Releases können aktuelle Sprachlaufzeiten enthalten, wie zum Beispiel 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 EC2 Amazon-Instances jedoch im Laufe der Zeit auf neuere Repository-Versionen verweisen, um Updates auf die laufenden Instances anzuwenden. Sie können das Update auch durchführen, indem Sie neue Instances der neuesten Version starten AMIs.

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 glibc Bibliothek, OpenSSL, OpenSSH, und die DNF Paketmanager erhalten Support für die gesamte Lebensdauer der Hauptversion AL2 023. 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. Feedback zu AL2 023 erhalten Sie von Ihrem zuständigen AWS Vertreter oder indem Sie ein Problem im Amazon-Linux-2023-Repo unter einreichen. GitHub

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 die nächste Version von Amazon Linux veröffentlicht wird, wird AL2 023 in den erweiterten Support aufgenommen und erhält Updates für Sicherheitsupdates und kritische Bugfixes.

Kleinversionen von AL2 023 haben beispielsweise das folgende Format:

- `2023.`**`0`**`.20230301`
- `2023.`**`1`**`.20230601`
- `2023.`**`2`**`.20230901`

Die entsprechenden AL2 023 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 AL2 023-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.

AL2023 führt auch neue Dateien zur Plattformidentifikation ein:

- `/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 EC2 Amazon-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 von. AL2

Auswahl der Basis-Toolchain und Standard-Build-Flags

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

Paketauswahl und Versionen

- Ausgewählte Backports zu wichtigen Systemkomponenten beinhalten mehrere Leistungsverbesserungen für die Ausführung auf EC2 Amazon-Infrastrukturen, insbesondere Graviton-Instances.
- AL2023 ist in mehrere AWS-Services Funktionen integriert. Dazu gehören der SSM-Agent AWS CLI, der Amazon Kinesis Kinesis-Agent und. CloudFormation
- AL2023 verwendet Amazon Corretto als Java Development Kit (JDK).
- AL2023 bietet Datenbank-Engines und Runtime-Updates für Programmiersprachen für neuere 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 AL2 Base-023-AMI und die Container-Images werden häufig aktualisiert, um den Austausch von Patch-Instances zu unterstützen.
- Kernel-Updates sind in AL2 023 AMI-Updates enthalten. Das bedeutet, dass Sie keine Befehle wie `yum update` und `reboot` für Aktualisierungen Ihres Kernels verwenden müssen.
- Neben dem AL2 Standard-023-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 AL2 023 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 und häufige Veröffentlichungen. Dadurch bleiben die vertrauten RPM-basierten Betriebssysteme erhalten.

Die allgemein verfügbare (GA) Version von AL2 023 ist nicht direkt mit einer bestimmten Fedora-Version vergleichbar. Die AL2 023 GA-Version enthält Komponenten von Fedora 34, 35 und 36. Einige der Komponenten sind dieselben wie die Komponenten in Fedora, andere sind modifiziert.

Andere Komponenten ähneln eher den Komponenten in CentOS Stream 9 oder wurden unabhängig voneinander entwickelt. Der Amazon-Linux-Kernel basiert auf den langfristigen Support-Optionen auf kernel.org, die unabhängig von Fedora ausgewählt wurden.

Maßgeschneidert cloud-init

Das Tool cloud-init Package ist eine Open-Source-Anwendung, die Linux-Images in einer Cloud-Computing-Umgebung bootet. Weitere Informationen finden Sie in der Cloud-Init-Dokumentation.

AL20.23 enthält eine angepasste Version von cloud-init. Mit cloud-init, können Sie angeben, was beim Booten mit Ihrer Instance passiert.

Wenn Sie eine Instance starten, können Sie die Benutzerdatenfelder verwenden, um Aktionen an zu übergeben cloud-init. Das bedeutet, dass Sie gängige Amazon Machine Images (AMIs) für viele Anwendungsfälle verwenden und diese dynamisch konfigurieren können, wenn Sie eine Instance starten. AL2023 verwendet auch cloud-init um das `ec2-user` Konto zu konfigurieren.

AL2023 verwendet den cloud-init Aktionen in `/etc/cloud/cloud.cfg.d` und. `/etc/cloud/cloud.cfg` Sie können Ihre eigenen erstellen cloud-init Aktionsdateien im `/etc/cloud/cloud.cfg.d` Verzeichnis. Cloud-init liest alle Dateien in diesem Verzeichnis in lexikografischer Reihenfolge. Spätere Dateien überschreiben Werte in früher gelesenen Dateien. Wann cloud-init startet eine Instanz, die cloud-init Das Paket führt 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 den standardmäßigen Volume-Mount für den Instance-Speicher wie folgt überschreiben cloud-init Direktive:

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

  Weitere Informationen zur Steuerung von Halterungen finden Sie unter Halterungen im 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 EC2 Amazon-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 Festplatten-Setup im cloud-init -Dokumentation.

Informationen zur Verwendung von cloud-init mit SELinux, sieheVerwenden Sie cloud-init um den Modus zu aktivieren `enforcing`.

Für Informationen über cloud-init Benutzerdatenformate finden Sie unter Benutzerdatenformate im 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 an mit DNF und Repository-Versionen. Weitere Informationen finden Sie unter Paket- und Betriebssystemupdates in AL2 023 verwalten.

Sicherheit in der Cloud

Sicherheit ist eine gemeinsame Verantwortung von 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 2023.

SELinux Modi

SELinux Ist standardmäßig aktiviert und in 023 auf den permissiven Modus gesetzt. AL2 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 wählen. Bei den Richtlinien handelt es sich um gezielte oder mehrstufige Sicherheitsrichtlinien (MLS).

Weitere Informationen zu SELinux Modi und Richtlinien finden Sie unter SELinux Einstellmodi für AL2 023 und im SELinux Projekt-Wiki.

Compliance-Programm

Unabhängige Prüfer bewerten die Sicherheit und Konformität von AL2 023 zusammen mit vielen AWS Compliance-Programmen.

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).
- A HTTP or HTTPS Der Client in libcrypto unterstützt GET and POST Aktionen, Umleitung, Plain und ASN.1-kodierter Inhalt, Proxys und Timeouts.
- Das Tool EVP_KDF funktioniert mit wichtigen Ableitungsfunktionen.
- Das Tool EVP_MAC API funktioniert 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 beides IPv4 und die IPv6 Vernetzung von Schnittstellen, wenn sie an eine Instanz angeschlossen 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 EC2 Amazon-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, bekannt als altnames, damit die ENI-gestützten Schnittstellen die Eigenschaften verschiedener AWS Ressourcen widerspiegeln. 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.

AL2023 Instanzschnittstellennamen werden mithilfe des `systemd` Slot-Benennungsschemas generiert. Weitere Informationen finden Sie unter systemd.net-Namensschema.

Darüber hinaus verwendet AL2 023 standardmäßig den Netzwerkübertragungsplanungsalgorithmus für die `fq_codel` aktive Warteschlangenverwaltung. Weitere Informationen finden Sie in der CoDelÜbersicht.

Core-Toolchain-Pakete glibc, gcc, binutils

Eine Teilmenge der Pakete in Amazon Linux wird als Core-Toolchain-Pakete bezeichnet. Ein großer Teil von AL2 023 besteht aus Kernpaketen, die fünf Jahre lang Support erhalten. 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 (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++, Fortran.
binutils 2.35	Assembler und Linker sowie andere binäre Tools	Manipuliert oder untersucht Binärprogramme.

Wir empfehlen Updates für alle glibc Auf Bibliotheken folgt ein Neustart. 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 Standard-Tool zur Verwaltung von Softwarepaketen in AL2 023 ist DNF. DNF ist der Nachfolger von YUM, das Paketverwaltungstool in AL2.

DNF ist ähnlich wie YUM in seiner Verwendung. Viele DNF Befehle und Befehlsoptionen sind dieselben wie YUM Befehle. In einem Befehlszeilenschnittstellen-(CLI)-Befehl ersetzt `dnf` `yum` in den meisten Fällen.

Zum Beispiel für die folgenden AL2 `yum` Befehle:

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

In AL2 023 werden sie zu den folgenden Befehlen:

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

In AL2 023 ist der `yum` Befehl immer noch verfügbar, jedoch als Zeiger 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 DNF CLI. Für weitere Informationen zu den Unterschieden zwischen YUM CLI und die DNF CLI, siehe Änderungen in DNF CLI verglichen mit YUM.

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

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 AL2 023 Host-Schlüsseled25519. ECDSA 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 023 AL2

Funktionen, die in 023 veraltet sind AL2 und in 023 nicht vorhanden sind, sind hier dokumentiert AL2. Dabei handelt es sich um Funktionen wie Funktionen und Pakete, die zwar in 023 AL2, aber nicht in AL2 023 vorhanden sind und auch nicht zu 023 hinzugefügt werden. AL2 Weitere Informationen darüber, wie lange die Funktionalität unterstützt wird AL2, finden Sie unter Veraltete Funktionen in. AL2

Es gibt auch Funktionen in AL2 023, 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 AL2 023 bereitstellen und sich auf die nächste Hauptversion von Amazon Linux vorbereiten.

Themen

- compat--Pakete
- Veraltete Funktionen wurden eingestellt in AL1, entfernt in AL2
- Funktionalität wurde in 023 veraltet AL2 und wurde in Version 023 entfernt AL2
- In 023 veraltet AL2

`compat-`-Pakete

Alle Pakete 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 nicht vorhanden (z. B. AL2 023). Wir empfehlen dringend, die Software anhand der aktualisierten Versionen der Bibliotheken neu zu erstellen.

Veraltete Funktionen wurden eingestellt in AL1, entfernt in AL2

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

> **Note**
>
> Im Rahmen der Wartungsunterstützungsphase von hatten einige Pakete ein end-of-life (EOL-) Datum AL1, das vor dem EOL von lag. AL1 Weitere Informationen finden Sie unter Erklärungen zur AL1 Paketunterstützung.

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

Themen

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

32-Bit-x86 (i686) AMIs

Im Rahmen der Version 2014.09 von kündigte Amazon Linux an AL1, dass dies die letzte Version sein wird, die 32-Bit produziert. AMIs 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, die die Erstellung neuer 32-Bit-Binärdateien ermöglichen. AL2023 enthält keine 32-Bit-User-Space-Pakete mehr. Wir empfehlen Benutzern, die Umstellung auf 64-Bit-Code abzuschließen, bevor sie zu 023 migrieren. AL2

Wenn Sie 32-Bit-Binärdateien auf AL2 023 ausführen müssen, ist es möglich, den 32-Bit-Userspace AL2 innerhalb eines AL2 Containers zu verwenden, der auf 023 läuft. AL2

`aws-apitools-*`ersetzt durch AWS CLI

Vor der Veröffentlichung von AWS CLI im September 2013 wurde eine Reihe von Befehlszeilen-Hilfsprogrammen zur Verfügung gestellt, implementiert in AWS Java, wodurch Benutzer EC2 Amazon-API-Aufrufe tätigen konnten. Diese Tools wurden 2015 eingestellt und wurden zur AWS

CLI bevorzugten Art, über die EC2 APIs Befehlszeile mit Amazon zu interagieren. Das Set 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. Das 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 verwenden.

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 AL2 023 nicht vorhanden.

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

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

AL2 war die erste Amazon Linux-Version, die das `systemd` Init-System verwendete und `upstart` in AL1 ersetzte. Jede `upstart` spezifische Konfiguration muss im Rahmen der Migration von AL1 zu einer neueren Version von Amazon Linux geändert werden. Die Verwendung `systemd` auf ist nicht möglich AL1, daher `systemd` kann der Wechsel von `upstart` zu nur im Rahmen der Umstellung auf eine neuere Hauptversion von Amazon Linux wie AL2 or AL2 023 erfolgen.

Funktionalität wurde in 023 veraltet AL2 und wurde in Version 023 entfernt AL2

In diesem Abschnitt werden Funktionen beschrieben, die in 023 verfügbar und in AL2 Version 023 nicht mehr verfügbar sind. AL2

Themen

- 32-Bit-x86-Pakete (i686)
- aws-apitools-*ersetzt durch AWS CLI
- awslogsveraltet zugunsten eines vereinheitlichten Amazon CloudWatch Logs-Agenten
- bzrSystem zur Versionskontrolle
- 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
- Mehrere Domainnamen in Amazon VPC create-dhcp-options
- Sun RPC in glibc
- OpenSSH-Schlüssel-Fingerabdruck im Protokoll audit
- ld.goldLinker
- ping6
- ftpPackage

32-Bit-x86-Pakete (i686)

Im Rahmen der Version 2014.09 von haben wir angekündigt AL1, dass dies die letzte Version sein wird, die 32-Bit-Version unterstützt. AMIs 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 Kunden, die Umstellung auf 64-Bit-Code abzuschließen.

Wenn Sie 32-Bit-Binärdateien auf AL2 023 ausführen müssen, ist es möglich, den 32-Bit-Userspace AL2 innerhalb eines AL2 Containers zu verwenden, der auf 023 läuft. AL2

`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, EC2 Amazon-API-Aufrufe zu tätigen. Diese Tools wurden 2015 als veraltet eingestuft und wurden zur bevorzugten Art, über die Befehlszeile mit Amazon EC2 APIs zu interagieren. AWS CLI Dies beinhaltet 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 (wie`aws-apitools-ec2`) aus, um Kunden Abwärtskompatibilität zu bieten. Das 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 verwenden.

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 AL2 023 nicht vorhanden.

`awslogs`veraltet zugunsten eines vereinheitlichten Amazon CloudWatch Logs-Agenten

Das `awslogs`Paket ist in 023 veraltet AL2 und ist in Version 023 nicht mehr vorhanden. AL2 Es wird durch den vereinheitlichten CloudWatch Logs-Agenten ersetzt, der im Paket verfügbar ist. `amazon-cloudwatch-agent` Weitere Informationen finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.

`bzr`System zur Versionskontrolle

Das Revisionskontrollsystem GNU Bazaar(`bzr`) wurde in 023 eingestellt AL2 und ist seit AL2 023 nicht mehr verfügbar.

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

cgroup v1

AL2023 wechselt zur Unified Control Group-Hierarchie (cgroup v2), während cgroup v1 AL2 verwendet wird. Da cgroup v2 AL2 nicht unterstützt wird, muss diese Migration im Rahmen der Umstellung auf 023 abgeschlossen werden. AL2

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

> **Note**
>
> Das `log4j-cve-2021-44228-hotpatch` Paket ist in Version 023 veraltet AL2 und wurde in Version 023 entfernt. AL2

Als Reaktion auf CVE-2021-44228 veröffentlichte Amazon Linux eine RPM-Paketversion des Hotpatches für Apache Log4j für und. AL1 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 AL2 023 wurde 15 Monate nach CVE-2021-44228 veröffentlicht, sodass 023 nicht mit dem Hotpatch ausgeliefert wird (aktiviert oder nicht). AL2

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 rief manche Software den `lsb_release` Befehl (im `system-lsb-core` Paket enthalten) auf, um Informationen über die Linux-Distribution abzurufen, auf der sie ausgeführt AL2 wurde. 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 ist nicht im Lieferumfang des `lsb_release` Befehls enthalten und beinhaltet das `system-lsb-core` Paket nicht. 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 023 veraltet und sind in AL2 023 nicht mehr vorhanden. AL2

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 023 zu portierenOpenSSL, da dieser nicht zu AL2 023 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 023 veraltet und nicht mehr vorhanden. AL2 Das älteste JDK, das in AL2 023 verfügbar war, wird von Corretto 8 bereitgestellt.

Weitere Informationen zu Java auf Amazon Linux finden Sie unterJavaim Jahr AL2 023.

Python 2.7

> **Note**
>
> AL2023 hat Python 2.7 entfernt, sodass alle Betriebssystemkomponenten, die Python benötigen, so geschrieben sind, dass sie mit Python 3 funktionieren. 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 im Jahr AL2 023.

rsyslog-opensslersetzt **rsyslog-gnutls**

Das `rsyslog-gnutls` Paket ist in 023 veraltet und in AL2 Version 023 nicht mehr vorhanden. AL2 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 in 023 nicht mehr unterstützt AL2, YP ist aber in 023 nicht mehr verfügbar. AL2 Dazu gehören die folgenden Pakete:`ypbind`, und`ypserv`. `yp-tools` Bei anderen Paketen, die sich in integrieren NIS lassen, wurde diese Funktionalität in Version AL2 023 entfernt.

Mehrere Domainnamen in Amazon VPC **create-dhcp-options**

In Amazon Linux 2 war es möglich, mehrere Domainnamen im `domain-name` Parameter an zu übergeben `create-dhcp-options`, was dazu führen würde, dass sie etwas wie `/etc/resolv.conf` enthalten würden`search foo.example.com bar.example.com`. Der Amazon DHCP VPC-Server sendet die Liste der bereitgestellten Domainnamen mit DHCP Option 15, die nur einen einzelnen Domainnamen unterstützt (siehe RFC 2132, Abschnitt 3.17). Da AL2 023 `systemd-networkd` für die Netzwerkkonfiguration verwendet wird, was auf die folgtRFC, ist diese zufällige Funktion in AL2 023 nicht vorhanden AL2

In der AWS CLIDokumentation von Amazon VPC heißt es dazu: „Einige Linux-Betriebssysteme akzeptieren mehrere durch Leerzeichen getrennte Domainnamen. Andere Linux-Betriebssysteme behandeln den Wert jedoch als eine einzelne Domain, was zu unerwartetem Verhalten führt. Windows Wenn Ihr DHCP Optionssatz mit einer Amazon-VPC verknüpft ist, deren Instances Betriebssysteme ausführen, die den Wert als einzelne Domain behandeln, geben Sie nur einen Domainnamen an. “

Auf diesen Systemen, z. B. AL2 023, werden zwei Domänen mit der DHCP Option 15 angegeben (die nur eine erlaubt), und da das Leerzeichen in Domainnamen ungültig ist, führt dies dazu,

dass das Leerzeichen als codiert wird`032`, was zu enthält führt. `/etc/resolv.conf search foo.exmple.com032bar.example.com`

Um mehrere Domainnamen zu unterstützen, sollte ein DHCP Server DHCP Option 119 verwenden (siehe RFC 3397, Abschnitt 2). Im Amazon VPC-Benutzerhandbuch erfahren Sie, wann dies vom Amazon DHCP VPC-Server unterstützt wird.

Sun RPC in **`glibc`**

Die Implementierung von Sun RPC in `glibc` ist in Version 023 veraltet und wurde in AL2 Version 023 entfernt. AL2 Kunden wird empfohlen, zur Verwendung der `libtirpc` Bibliothek (verfügbar in AL2 und AL2 023) überzugehen, falls Sun RPC Funktionen erforderlich sind. Die Einführung ermöglicht `libtirpc` auch die Unterstützung IPv6 von Anwendungen.

Diese Änderung spiegelt die allgemeine Akzeptanz der Upstream-Version wider, diese Funktionalität zu `glibc` entfernen, zum Beispiel die Entfernung von Sun RPC Schnittstellen aus `glibc` Fedora und eine ähnliche Änderung in Gentoo.

OpenSSH-Schlüssel-Fingerabdruck im Protokoll **`audit`**

Später im Lebenszyklus von wurde dem OpenSSH-Paket ein Patch hinzugefügt AL2, der den zur Authentifizierung verwendeten Schlüsselfingerabdruck ausgibt. Diese Funktionalität ist in 023 nicht vorhanden. AL2

`ld.gold`Linker

Der `ld.gold` Linker ist in 023 verfügbar und wird in AL2 023 entfernt. AL2 Kunden, die Software entwickeln, die explizit auf den gold Linker verweist, sollten auf den regulären Linker () `ld.bfd` migrieren.

Die Upstream-Versionshinweise zu GNU Binutils für Version 2.44 (veröffentlicht im Februar 2025) dokumentieren die Entfernung von`ld.gold`: „In einer Änderung unserer bisherigen Praxis enthält der binutils-2.44.tar-Tarball in dieser Version nicht die Quellen für den Gold-Linker. Das liegt daran, dass der Gold-Linker nun veraltet ist und irgendwann entfernt werden wird, sofern sich nicht Freiwillige melden und anbieten, die Weiterentwicklung und Wartung fortzusetzen.“

ping6

In AL2 023 unterstützt das reguläre `ping` Hilfsprogramm nativ IPv6, und das separate Programm `/bin/ping6` ist nicht mehr erforderlich. In AL2 023 `/usr/sbin/ping6` ist es ein Symlink zur ausführbaren Datei. `/usr/bin/ping`

Diese Änderung folgt auf die Einführung neuerer `iputils` Versionen, die diese Funktionalität bieten, durch die breitere Community, beispielsweise die IPv6 Ping-Änderung in Fedora.

`ftp`Package

Das `ftp` Paket in AL2 ist ab AL2 023 nicht mehr in Amazon Linux verfügbar. Diese Entscheidung wurde im Rahmen unseres kontinuierlichen Engagements für Sicherheit, Wartbarkeit und moderne Softwareentwicklungspraktiken getroffen. Im Rahmen der (oder davor) Migration auf Version AL2 023 empfehlen wir, jegliche Verwendung des `ftp` Legacy-Pakets auf eine seiner Alternativen zu migrieren.

Hintergrund

Das `ftp` Legacy-Paket wurde im Upstream-Bereich seit vielen Jahren nicht mehr aktiv gepflegt. Die letzte bedeutende Aktualisierung des Quellcodes erfolgte Anfang der 2000er Jahre, und das ursprüngliche Quell-Repository ist nicht mehr verfügbar. Während einige Linux-Distributionen Patches für Sicherheitslücken enthielten, ist die Codebasis nach wie vor weitgehend unbewirtschaftet.

Empfohlene Alternativen

AL2023 bietet mehrere moderne, aktiv gepflegte Alternativen für FTP-Funktionen:

`lftp`(verfügbar in AL2 und AL2 023)

Ein ausgeklügeltes Dateiübertragungsprogramm, das FTP, HTTP, SFTP und andere Protokolle unterstützt. Es bietet mehr Funktionen als der herkömmliche `ftp` Client und wird aktiv gewartet.

Installiere mit: dnf install lftp

`curl`(verfügbar in AL2 und AL2 023)

Ein vielseitiges Befehlszeilentool für die Übertragung von Daten mit URLs Unterstützung von FTP, FTPS, HTTP, HTTPS und vielen anderen Protokollen.

Standardmäßig in AL2 023 über das Paket verfügbar. `curl-minimal` Für eine umfassendere Protokollunterstützung können Sie optional ein Upgrade auf die `curl-full` Verwendung vondnf swap curl-minimal curl-full.

`wget`(verfügbar in AL2 und AL2 023)

Ein nicht interaktives Befehlszeilenprogramm zum Herunterladen von Dateien aus dem Internet, das die Protokolle HTTP, HTTPS und FTP unterstützt.

Installation mit: dnf install wget (standardmäßig nicht in allen AL2 023 Images installiert)

`sftp`(verfügbar in AL2 und AL2 023)

Ein sicheres Dateiübertragungsprotokoll, das über SSH funktioniert und verschlüsselte Dateiübertragungen ermöglicht.

Standardmäßig als Teil des OpenSSH-Pakets verfügbar.

Überlegungen zur Migration

Wenn Ihre Anwendungen oder Skripts vom `ftp` Legacy-Client abhängen, sollten Sie die folgenden Migrationsansätze in Betracht ziehen:

1. Aktualisieren Sie Skripts, um moderne Alternativen zu verwenden: Ändern Sie Ihre Skripts `solftp`,,`curl`,`wget`, oder `sftp` anstelle des `ftp` Legacy-Clients zu verwenden.
2. Überprüfen Sie die Paketabhängigkeiten: Einige Anwendungen führen das `ftp` Paket möglicherweise als Abhängigkeit in ihren Paketmetadaten auf, obwohl sie seit langem intern auf die Verwendung moderner Protokolle umgestellt haben. In diesen Fällen kann es sein, dass die Anwendung auf AL2 023 korrekt funktioniert, obwohl sie nicht im Paket `/usr/bin/ftp` enthalten ist`ftp`. Prüfen Sie die tatsächlichen Anforderungen Ihrer Anwendung, anstatt sich ausschließlich auf die angegebenen Abhängigkeiten zu verlassen.
3. Aktualisieren Sie die Anwendungsabhängigkeiten: Bei Anwendungen, die Sie verwalten und die zwar immer noch eine Abhängigkeit vom `ftp` Paket deklarieren, es aber nicht wirklich verwenden, aktualisieren Sie die Paketmetadaten, um diese unnötige Abhängigkeit zu entfernen.

Sicherheitsüberlegungen

Das FTP-Protokoll überträgt Daten, einschließlich Anmeldeinformationen, im Klartext. Für sicherheitsrelevante Anwendungen empfehlen wir dringend, verschlüsselte Alternativen wie SFTP oder HTTPS zu verwenden, die von den empfohlenen alternativen Tools unterstützt werden.

In 023 veraltet AL2

In diesem Abschnitt werden Funktionen beschrieben, die in Version AL2 023 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)
- aspell
- Datenbank Berkeley () libdb
- cron
- IMDSv1
- pcre Version 1
- System V init (sysvinit)
- EOL-Pakete sind veraltet

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-Userspace-Binärdateien nicht mehr unterstützt.

`aspell`

AL2023 wird zwar mit dem `aspell` Paket ausgeliefert, ist jedoch veraltet und wird in der nächsten Hauptversion von Amazon Linux entfernt. Kunden wird empfohlen, auf moderne Ersatzprodukte wie oder umzusteigen. `hunspell` `enchant2`

Die Deprecation von Version AL2 023 folgt dem allgemeinen Wandel `aspell` in der Community, zum Beispiel der Abwertung in. `aspell`Fedora

Datenbank Berkeley () `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.

In AL2 023 gibt es nur wenige Pakete, 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 Version AL2 023 unterstützt weiterhin schreibgeschützt eine Datenbank im Berkeley DB (BDB) -Format. `rpm` Diese Unterstützung wird in der nächsten Hauptversion von Amazon Linux entfernt.

Die Abwertung von `libdb` folgt der Abkehr von der breiteren Community, zum Beispiel der `libdb`Abwertung von. Fedora

`cron`

Das `cronie` Paket wurde standardmäßig auf dem AL2 AMI installiert und bietet Unterstützung für die herkömmliche `crontab` Art der Planung periodischer Aufgaben. In AL2 023 `cronie` ist es standardmäßig nicht enthalten. Daher `crontab` wird die Unterstützung für standardmäßig nicht mehr bereitgestellt.

In AL2 023 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 AMIs sind AL2 023 so konfiguriert, dass sie nur im Modus „IMDSv2Nur“ gestartet werden, wodurch die Verwendung von deaktiviert wird. IMDSv1 Es besteht weiterhin die Möglichkeit, AL2 023 zu verwenden, wenn diese Option aktiviert ist. IMDSv1 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 EC2 Amazon-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 AL2 023 mit einer begrenzten Anzahl von Paketen ausgeliefert wurden, die darauf aufbauen`pcre`, werden diese Pakete innerhalb von 023 migriert. `pcre2` AL2 Die veraltete `pcre` Bibliothek wird auch in Version 023 verfügbar bleiben. AL2

> **Note**
>
> Die veraltete Version von `pcre` wird während der gesamten Lebensdauer von 023 keine Sicherheitsupdates erhalten. AL2 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`

Die Abwertung von `pcre` zugunsten von `pcre2` folgt dem allgemeinen Wandel der Community in diese Richtung, beispielsweise der `pcre`Abwertung in. Fedora

System V init (`sysvinit`)

Obwohl AL2 023 die Abwärtskompatibilität mit System V service (init) -Skripten beibehält, kündigte das `systemd` Upstream-Projekt im Rahmen seiner Version v254 die Einstellung der Unterstützung für System V-Dienstskripte an und gab an, dass die Unterstützung in einer future Version von entfernt wird. `systemd` Weitere Informationen finden Sie unter systemd.

AL2023 behält die Abwärtskompatibilität mit System V service (init) -Skripten bei, aber Benutzern wird empfohlen, auf native `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.

EOL-Pakete sind veraltet

Jedem in AL2 023 verfügbaren Paket ist eine Support-Erklärung zugeordnet, die Amazon Linux-spezifische Informationen enthält. Diese Aussagen beziehen sich auf den Kern des Betriebssystems und seine Lebensdauer sowie auf Pakete wie the section called “PHP” undthe section called “Python”, wobei AL2 023 mehrere Versionen ausliefert und jede für die Dauer unterstützt wird, für die das Open-Source-Projekt im Upstream-Format unterstützt wird.

In AL2 023 können Sie mithilfe des Paketmanagers Informationen zur `dnf` Paketunterstützung abrufen. Weitere Informationen finden Sie unter Abrufen von Paketunterstützunginformationen.

Wenn ein Paket vor dem Ende der Hauptversion von Amazon Linux nicht mehr unterstützt wird, sollte davon ausgegangen werden, dass dieses Paket veraltet ist und in der nächsten Hauptversion von Amazon Linux nicht mehr vorhanden sein wird.

Bei Paketen wie the section called “PHP” undthe section called “Python”, bei denen jede Hauptversion von Amazon Linux mehrere Versionen mit jeweils unterschiedlichem Support-Lebenszyklus ausgeliefert hat, ist es wahrscheinlich, dass sie weiterhin in neuen Hauptversionen von Amazon Linux enthalten sein werden, wenn auch mit geringen oder keinen Überschneidungen der Hauptversionen der Pakete. Es wird empfohlen, bei der Auswahl von Abhängigkeiten die Zeitpläne für die Unterstützung von Amazon Linux-Paketen zu berücksichtigen.

Vergleich AL2 und AL2 023

In den folgenden Themen werden die wichtigsten Unterschiede zwischen AL2 und AL2 023 beschrieben.

Weitere Informationen zu Funktionen, die in AL1, und AL2 023 veraltet sind AL2, finden Sie unter. Veraltete Funktionalität in 023 AL2

Themen

- Hinzugefügte, aktualisierte und entfernte Pakete
- Support für die einzelnen Versionen
- Änderungen bei der Benennung und Versionierung
- Optimierungen
- Aus mehreren Upstream-Quellen
- Netzwerksystem-Dienst
- Paketmanager
- Verwenden von cloud-init
- Grafische Desktop-Unterstützung
- Compiler-Triplet
- 32-Bit x86-(i686)-Pakete
- lsb_release und das system-lsb-core-Paket
- Extra Packages for Enterprise Linux (EPEL)
- Python 2.7 wurden durch Python 3 ersetzt
- Sicherheits-Updates
- Deterministische Upgrades für Stabilität
- gp3als standardmäßiger Amazon EBS-Volumetyp
- Vereinheitlichte Kontrollgruppenhierarchie (cgroup v2)
- systemdTimer ersetzen cron
- Verbesserte Toolchain: gccbinutils, und glibc
- systemdJournal ersetzt rsyslog

- Minimierte Paketabhängigkeiten
- Amazon Corretto als Standard-JVM
- AWS CLI v2
- UEFI Preferred und Secure Boot
- Änderungen der Standardkonfiguration des SSH-Servers
- AL2023 Kerneländerungen von AL2
- /tmp ist jetzt tmpfs
- Änderungen an AMI und Container-Images
- Vergleich der auf Amazon Linux 2 und Amazon Linux 2023 installierten Pakete AMIs
- Vergleich der auf Amazon Linux 2 und Amazon Linux 2023 Minimal installierten Pakete AMIs
- Vergleich der auf Amazon Linux 2 und Amazon Linux 2023 Basis-Container-Images installierten Pakete

Hinzugefügte, aktualisierte und entfernte Pakete

AL2023 enthält Tausende von Softwarepaketen, die verwendet werden können. Eine vollständige Liste aller Pakete, die im Jahr AL2 023 im Vergleich zu früheren Amazon Linux-Versionen hinzugefügt, aktualisiert oder entfernt wurden, finden Sie unter Paketänderungen in AL2 023.

Um zu beantragen, dass ein Paket in Version AL2 023 hinzugefügt oder geändert wird, melden Sie ein Problem im Amazon-Linux-2023-Repo unter. GitHub

Support für die einzelnen Versionen

Für AL2 023 bieten wir fünf Jahre Support an.

Weitere Informationen finden Sie unter Release-Taktfrequenz.

Änderungen bei der Benennung und Versionierung

AL2023 unterstützt dieselben Mechanismen wie die Identifizierung AL2 von Plattformen. AL2023 führt auch neue Dateien zur Plattformidentifikation ein.

Weitere Informationen finden Sie unter Benennung und Versionsverwaltung.

Optimierungen

AL2023 optimiert die Startzeit, um die Zeit vom Start der Instance bis zur Ausführung des Kunden-Workloads zu verkürzen. Diese Optimierungen umfassen die Kernelkonfiguration, `cloud-init` Konfigurationen und Funktionen der EC2 Amazon-Instanz, die in Pakete im Betriebssystem integriert sind, z. B. `kmod` und`systemd`.

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

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.

Netzwerksystem-Dienst

Der `systemd-networkd` Systemdienst verwaltet die Netzwerkschnittstellen in 023. AL2 Dies ist eine Änderung von AL2, die ISC dhclient oder `dhclient` verwendet.

Weitere Informationen finden Sie unter Netzwerkdienst.

Paketmanager

Das Standardtool zur Verwaltung von Softwarepaketen auf AL2 023 istDNF. DNFist der Nachfolger vonYUM, dem Paketverwaltungstool in AL2.

Weitere Informationen finden Sie unter Paketmanagement-Tool.

Verwenden von cloud-init

cloud-initVerwaltet in AL2 023 das Paket-Repository. In früheren Versionen von Amazon Linux wurden standardmäßig Sicherheitsupdates über cloud-init installiert. Dies ist nicht die Standardeinstellung für AL2 023. Die neuen deterministischen Upgrade-Funktionen für Updates

`releasever` beim Start beschreiben die AL2 023-Methode, Paketaktualisierungen beim Start zu aktivieren. Weitere Informationen erhalten Sie unter Paket- und Betriebssystemupdates in AL2 023 verwalten und Deterministische Upgrades für Stabilität.

Mit AL2 023 können Sie mit verwenden. cloud-init SELinux Weitere Informationen finden Sie unter Verwenden Sie cloud-init um den Modus zu aktivieren `enforcing`.

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 AL2 023 führen nicht verfügbare Remote-Ressourcen zu einem schwerwiegenden Fehler und die cloud-init Ausführung schlägt fehl. Diese Änderung des Verhaltens von AL2, bietet ein sichereres Standardverhalten beim Schließen eines Fehlers.

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

Grafische Desktop-Unterstützung

AL2023 bietet ab Version 2023.7 eine GNOME-basierte grafische Desktop-Umgebung, die den in verwendeten MATE-Desktop ersetzt. AL2 Diese Version bietet Benutzern ein anderes Desktop-Erlebnis und behält gleichzeitig die für die Cloud optimierte Leistung von 023 bei. AL2 Die GNOME-Desktop-Umgebung bietet verschiedene Anpassungsoptionen, Funktionen zur Systemintegration und ein eigenständiges Benutzeroberflächendesign, das Benutzern eine Alternative zur vorherigen MATE-Desktop-Umgebung bietet. Weitere Einzelheiten finden Sie AL2023 Grafischer Desktop auf der Seite.

Compiler-Triplet

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

Somit wird aus 023 AL2 `aarch64-redhat-linux-gcc`. `aarch64-amazon-linux-gcc` AL2

Dies sollte für die meisten Benutzer völlig transparent sein und betrifft möglicherweise nur diejenigen, die Compiler auf AL2 023 erstellen.

32-Bit x86-(i686)-Pakete

Im Rahmen der Veröffentlichung 2014.09 wurde angekündigt, dass AL1 es die letzte Version sein wird, die 32-Bit-Version produzieren wird. AMIs Daher unterstützt Amazon Linux ab Version

2015.03 die Ausführung des AL1 Systems im 32-Bit-Modus nicht mehr. AL2 bot eingeschränkte Laufzeitunterstützung für 32-Bit-Binärdateien auf x86-64-Hosts und stellte keine Entwicklungspakete zur Verfügung, um die Erstellung neuer 32-Bit-Binärdateien zu ermöglichen. AL2023 enthält keine 32-Bit-Userspace-Pakete mehr. Wir empfehlen Ihnen, die Umstellung auf 64-Bit-Code abzuschließen.

Wenn Sie 32-Bit-Binärdateien auf AL2 023 ausführen müssen, ist es möglich, den 32-Bit-Benutzerbereich AL2 innerhalb eines AL2 Containers zu verwenden, der auf 023 ausgeführt wird. AL2

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

In der Vergangenheit hat manche Software den `lsb_release` Befehl (im `system-lsb-core` Paket enthalten) aufgerufen, um Informationen über die Linux-Distribution zu erhalten, auf der sie ausgeführt AL2 wurde. 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 ist nicht im Lieferumfang des `lsb_release` Befehls enthalten und beinhaltet das `system-lsb-core` Paket nicht. Die Software sollte die Umstellung auf den `os-release`-Standard abschließen, um die Kompatibilität mit Amazon Linux und anderen wichtigen Linux-Distributionen aufrechtzuerhalten.

Extra Packages for Enterprise Linux (EPEL)

> **Warning**
>
> Das AL2 `epel` Extra hat den Drittanbieter aktiviert EPEL7 Endlager. Ab 2024-06-30 der Drittanbieter EPEL7 Das Repository wird nicht mehr verwaltet.
> Dieses Drittanbieter-Repository wird keine future Updates enthalten. Das bedeutet, dass es keine Sicherheitskorrekturen für Pakete im EPEL-Repository geben wird.
> In diesem Abschnitt werden die Optionen in AL2 0.23 für einige Pakete behandelt, die in EPEL.

Extra Packages for Enterprise Linux (EPEL) ist ein Projekt in der Fedora Community mit dem Ziel, eine große Auswahl an Paketen für Linux-Betriebssysteme auf Unternehmensebene zu erstellen. Das Projekt hat hauptsächlich Folgendes produziert RHEL and CentOS Pakete. AL2 bietet ein hohes Maß an Kompatibilität mit CentOS 7. Infolgedessen viele EPEL7 Pakete funktionieren weiter AL2.

Derzeit gibt es kein EPEL or EPELähnliches Repository für AL2 023. Es gibt jedoch eine Reihe von Paketen, die in EPEL7 die Kunden verwendet haben und AL2 die entweder in AL2 023 verfügbar sind oder Alternativen in AL2 023 haben. In diesem Abschnitt werden einige dieser Pakete und die Optionen in AL2 023 behandelt.

> ⚠ Warning
>
> Fügen Sie nur Repositorys hinzu, die für die Verwendung mit AL2 023 vorgesehen sind. Auch wenn Repositorys, die für andere Distributionen entworfen wurden, heute funktionieren, gibt es keine Garantie dafür, dass sie dies auch weiterhin tun werden, wenn ein Paket-Update in AL2 023 oder ein Projektarchiv nicht für die Verwendung mit 023 konzipiert ist. AL2

Es gibt auch Pakete, von denen aus installiert werden konnte EPEL AL2 welche nicht zu AL2 023 hinzugefügt werden. Häufige Gründe dafür sind Probleme wie das Upstream-Projekt, das nicht mehr gepflegt wird oder nicht repariert CVEs wird. In diesem Abschnitt werden auch einige dieser Pakete behandelt und erklärt, welche Alternativen es gibt.

Themen

- axel- HTTP/FTP-Client
- brotliund libbrotli - Komprimierung
- collectd- Daemon zur Erfassung von Statistiken
- cpulimit- Begrenzer für die CPU-Auslastung
- exim- Agent für die E-Mail-Übertragung
- fuse3- Dateisystem im Userspace (FUSE) v3
- ganglia- Verteiltes Überwachungssystem
- git-lfs- Versionskontrolle großer Dateien mit Git
- haveged- eine Entropiequelle unter Verwendung der HAVEGE Algorithmus
- inotify-tools- inotify-Befehlszeilentools
- iperf- TCP/UDP-Leistungs-Benchmark

- jemalloc- alternative Implementierung malloc
- libbsd- BSD-kompatible Funktionsbibliothek
- libserf- HTTP-Clientbibliothek
- libzstd- ZSTD-Kompressionsbibliothek
- lighttpdWebserver
- lshell- eine eingeschränkte Shell
- monit- Prozess-, Datei-, Verzeichnis- und Geräteüberwachung
- nodejs
- perl-Config-General
- python2-lockfile- Sperren von Dateien
- python2-rsa- reines Python RSA
- python2-simplejson- JSON-Routinen für Python 2
- rkhunter- Rootkit Hunter
- rssh- eine eingeschränkte Shell zur Verwendung mit OpenSSH
- sscg- Generator für selbstsignierte SSL-Zertifikate
- stress- Stresstest
- stress-ng- Stresstest
- tmpwatch- entfernt Dateien basierend auf dem Zeitpunkt des letzten Zugriffs
- xmlstarlet- XML-Hilfsprogramme für die Befehlszeile

`axel`- HTTP/FTP-Client

Das `axel` Paket war drin EPEL7, und wurde noch nie als Teil von Amazon Linux ausgeliefert. Die im Jahr AL2 2023 verfügbaren Alternativen sind `curl` und`wget`.

> **Warning**
>
> Die `-S` Option *`axel`* verwendet eine unverschlüsselte http Verbindung zum Auffinden von Spiegeln für eine Datei.

Es wird dringend empfohlen, jegliche Verwendung von `axel` Over auf entweder `curl` oder zu migrieren`wget`.

`brotli`und `libbrotli` - Komprimierung

Die `libbrotli` Pakete `brotli` und waren in EPEL7, während nur das `brotli` Paket im AL2 Core verfügbar war.

`brotli`Sowohl das als auch das `libbrotli` Paket sind in Version AL2 023 enthalten.

Das `brotli` Paket kann auf AL2 023 mit dem folgenden Befehl installiert werden:

```
[ec2-user ~]$ sudo dnf install brotli
```

Das `libbrotli` Paket kann auf AL2 023 mit dem folgenden Befehl installiert werden:

```
[ec2-user ~]$ sudo dnf install libbrotli
```

`collectd`- Daemon zur Erfassung von Statistiken

Das `collect` Paket war drin EPEL7, und war auch in den Optionen `collectd` und `collectd-python3` AL2 Extras erhältlich.

Das `collectd` Paket ist in AL2 0.23 enthalten und kann mit dem folgenden Befehl installiert werden:

```
[ec2-user ~]$ sudo dnf install collectd
```

`cpulimit`- Begrenzer für die CPU-Auslastung

`systemd`Bietet in Amazon Linux 2023 Funktionen zur Begrenzung der CPU-Auslastung von Prozessen oder Prozessgruppen. Diese Funktionalität ist auch für jeden `systemd` Service einfach zu verwenden.

Es stehen leistungsstarke Funktionen zur Ressourcenkontrolle zur Verfügung, mit `systemd` denen sichergestellt werden kann, dass jede Aufgabe oder Gruppe von Aufgaben nur begrenzt Ressourcen verbrauchen kann. Weitere Informationen finden Sie in der Upstream-Dokumentation zu systemd.resource-control, zusammen mit der. Begrenzung der Nutzung von Prozessressourcen in AL2 023 mithilfe von systemd

`exim`- Agent für die E-Mail-Übertragung

Das `exim` Paket war drin EPEL7, und war zuvor verfügbar in AL1. Amazon Linux 2023 bietet `postfix` sowohl die Mail Transfer Agents als auch `sendmail` Mail Transfer Agents (MTAs).

`fuse3`- Dateisystem im Userspace (FUSE) v3

Das `fuse3` Paket (einschließlich `fuse3-libs` und`fuse3-devel`) befand sich in EPEL7. Diese Pakete sind Teil von AL2 023 und jedes kann installiert werden, indem der entsprechende folgende Befehl ausgeführt wird:

```
[ec2-user ~]$ sudo dnf install fuse3
```

```
[ec2-user ~]$ sudo dnf install fuse3-libs
```

```
[ec2-user ~]$ sudo dnf install fuse3-devel
```

`ganglia`- Verteiltes Überwachungssystem

Das `ganglia` Paket war drin EPEL7, und war zuvor verfügbar in AL1. Es wurde nicht mit geliefert AL2.

Das vorgelagerte Projekt hatte eine Phase der Inaktivität, in der einige offene CVEs Fragen nicht bearbeitet wurden. Im Rahmen des vorgelagerten Projekts gab es zwar in letzter Zeit Aktivitäten, es ist jedoch nicht geplant, weitere AL2 023 `ganglia` zu erhöhen.

`git-lfs`- Versionskontrolle großer Dateien mit Git

Das `git-lfs` Paket war drin EPEL7. In Amazon Linux 2023 ist das `git-lfs` Paket im Core-Repository enthalten. Auf AL2 023 `git-lfs` kann es installiert werden, indem der folgende Befehl ausgeführt wird:

```
[ec2-user ~]$ sudo dnf install git-lfs
```

`haveged`- eine Entropiequelle unter Verwendung der HAVEGE Algorithmus

Das `haveged` Paket war drin EPEL7. Amazon Linux 2023 ist mit Entropiequellen vorkonfiguriert, sodass keine Verwendung von erforderlich ist. `haveged`

`inotify-tools`- inotify-Befehlszeilentools

Das `inotify-tools` Paket war drin EPEL7, und ist in AL2 023 enthalten.

> **Note**
>
> In AL2 023 wird eine pfadbasierte Aktivierung `systemd` unterstützt, mit der Maßnahmen ergriffen werden können, z. B. wenn ein Pfad existiert oder sich ändert.
> Vieles von dem, wofür `inotify-tools` es verwendet wird, kann jetzt mithilfe der `systemd` Pfadaktivierung besser und zuverlässiger durchgeführt werden. Weitere Informationen finden Sie unter systemd.path.

Das `inotify-tools` Paket ist in AL2 0.23 enthalten und kann mit dem folgenden Befehl installiert werden:

```
[ec2-user ~]$ sudo dnf install inotify-tools
```

`iperf`- TCP/UDP-Leistungs-Benchmark

Das Paket der `iperf` Version 2 war in EPEL7, und war auch im `testing` AL2 Extra verfügbar. und war auch verfügbar in AL1

> **Note**
>
> Das `iperf3` Paket ist ebenfalls verfügbar und bietet Version 3 von`iperf`.

Das `iperf` Paket ist in AL2 0.23 enthalten und kann mit dem folgenden Befehl installiert werden:

```
[ec2-user ~]$ sudo dnf install iperf
```

`jemalloc`- alternative Implementierung `malloc`

Das `jemalloc` Paket war drin EPEL7, und war in den `mariadb10.5` AL2 Extras `lamp-mariadb10.2-php7.2` und Extras erhältlich.

Das `jemalloc` Paket ist in AL2 0.23 enthalten und kann mit dem folgenden Befehl installiert werden:

```
[ec2-user ~]$ sudo dnf install jemalloc
```

`libbsd`- BSD-kompatible Funktionsbibliothek

Das `libbsd` Paket war in EPEL7, und war auch im `testing` AL2 Extra erhältlich.

Das `libbsd` Paket ist in Version AL2 023 enthalten und kann mit dem folgenden Befehl installiert werden:

```
[ec2-user ~]$ sudo dnf install libbsd
```

Die Entwicklungsdateien für `libbsd` können mit dem folgenden Befehl installiert werden.

```
[ec2-user ~]$ sudo dnf install libbsd-devel
```

`libserf`- HTTP-Clientbibliothek

Das `libserf` Paket war in EPEL7. Das `libserf` Paket wird in Amazon Linux 2023 bereitgestellt. Es kann installiert werden, indem Sie den folgenden Befehl ausführen:

```
[ec2-user ~]$ sudo dnf install libserf
```

`libzstd`- ZSTD-Kompressionsbibliothek

Das `libzstd` Paket war sowohl im AL2 Core als auch in EPEL7. Das `libzstd` Paket ist auch Teil von AL2 023.

```
[ec2-user ~]$ sudo dnf install libzstd
```

`lighttpd`Webserver

Das `lighttpd` Paket war drin EPEL7, und war zuvor verfügbar in AL1. Amazon Linux 2023 stellt sowohl den Apache als `httpd` auch den `nginx` Webserver bereit.

`lshell`- eine eingeschränkte Shell

Das `lshell` Paket wurde nie als Teil von Amazon Linux versendet. Es war verfügbar in EPEL6. Das Fedora-Packaging-Repository für `lshell` deckt ab, warum es nicht verpackt wurde EPEL7 oder Fedora 30. Es wurde auch aus Debian entfernt.

Das `lshell` Upstream-Projekt wird nicht mehr aktiv betreut und enthält bekannte kritische Dateien, die noch nicht gepatcht wurden CVEs: CVE-2016-6902 und CVE-2016-6903.

Die Alternative, die im Debian-Bug vorgeschlagen wurde, `rssh`ist im Upstream-Bereich ebenfalls nicht gepflegt, wobei der Autor unbehebbare Sicherheitsprobleme als Grund nennt.

Aus diesen Gründen ist eine Erweiterung der Version AL2 023 nicht `lshell` geplant.

`monit`- Prozess-, Datei-, Verzeichnis- und Geräteüberwachung

`systemd`Bietet in Amazon Linux 2023 eine Vielzahl von Funktionen zum Überwachen, Starten, Stoppen und Neustarten von Diensten. Dazu gehören die Ratenbegrenzung von Neustarts, das Warten zwischen Neustartversuchen und das Starten eines weiteren Dienstes bei einem Fehler. Weitere Informationen finden Sie in der systemd.service-Dokumentation.

In AL2 Version 023 wird `systemd` auch eine pfadbasierte Aktivierung unterstützt, mit der Maßnahmen ergriffen werden können, z. B. wenn ein Pfad existiert oder sich ändert. Weitere Informationen finden Sie unter systemd.path.

Es gibt allgemeine Konfigurationsoptionen für `systemd` Units, die es ermöglichen, Abhängigkeiten, Bedingungen und Aktionen anzugeben, die bei Erfolg oder Misserfolg ausgeführt werden sollen. Weitere Informationen finden Sie in der systemd.unit-Dokumentation.

Es stehen leistungsstarke Funktionen zur Ressourcenkontrolle zur Verfügung`systemd`, mit deren Hilfe sichergestellt werden kann, dass jede Überwachungsaufgabe nicht zu viel CPU oder Arbeitsspeicher beansprucht. Weitere Informationen finden Sie unter systemd.resource-control.

`nodejs`

Das Paket der `nodejs` Version 16 war in EPEL7, und `nodejs` ist jetzt in Version AL2 023 enthalten. Zum Zeitpunkt der Erstellung dieses Artikels waren sowohl `nodejs` Version 18 als auch Version 20 im Jahr AL2 023 verfügbar. Sie können `nodejs` 18 auf AL2 023 mit dem folgenden Befehl installieren:

```
[ec2-user ~]$ sudo dnf install nodejs
```

Sie können `nodejs` 20 auf AL2 023 mit dem folgenden Befehl installieren:

```
[ec2-user ~]$ sudo dnf install nodejs20
```

`perl-Config-General`

Das `perl-Config-General` Paket war in EPEL7, und ist jetzt in AL2 023 enthalten. Sie können das `perl-Config-General` Paket in AL2 023 mit dem folgenden Befehl installieren:

```
[ec2-user ~]$ sudo dnf install perl-Config-General
```

Perl-Module können auch installiert werden, indem Sie fragen DNF um das Paket zu installieren, das ein bestimmtes Perl-Modul bereitstellt. Mit dieser Methode können Sie den bekannteren Perl-Modulnamen anstelle des Betriebssystempaketnamens verwenden.

```
[ec2-user ~]$ sudo dnf install 'perl(Confg::General)'
```

`python2-lockfile`- Sperren von Dateien

Das `python2-lockfile` Paket war drin EPEL7, und AL2 beinhaltete ein `python-lockfile` Paket. In AL2 023 wird Python 2.7 wurden durch Python 3 ersetzt also eine Python-2-Variante dieses Pakets nicht zu AL2 023 hinzugefügt.

Die Python-3-Version dieses Pakets ist in AL2 0.23 enthalten. Sie können das `python3-lockfile` Paket in AL2 023 mit einem der folgenden Befehle installieren:

```
[ec2-user ~]$ sudo dnf install python3-lockfile
```

Python-Module können auch auf Anfrage installiert werden DNF um das Paket zu installieren, das ein bestimmtes Python-Modul bereitstellt.

```
[ec2-user ~]$ sudo dnf install 'python3dist(lockfile)'
```

`python2-rsa`- reines Python RSA

Das `python2-rsa` Paket war drin EPEL7, und AL2 beinhaltete ein `python2-rsa` Paket. In AL2 023 wird Python 2.7 wurden durch Python 3 ersetzt also eine Python-2-Variante dieses Pakets nicht zu AL2 023 hinzugefügt.

Die Python-3-Version dieses Pakets ist in AL2 0.23 enthalten. Sie können das `python3-rsa` Paket in AL2 023 mit einem der folgenden Befehle installieren:

```
[ec2-user ~]$ sudo dnf install python3-rsa
```

Python-Module können auch auf Anfrage installiert werden DNF um das Paket zu installieren, das ein bestimmtes Python-Modul bereitstellt.

```
[ec2-user ~]$ sudo dnf install 'python3dist(rsa)'
```

`python2-simplejson`- JSON-Routinen für Python 2

Das `python2-simplejson` Paket war drin EPEL7. In AL2 023 wird Python 2.7 wurden durch Python 3 ersetzt also eine Python-2-Variante dieses Pakets nicht zu AL2 023 hinzugefügt.

Die Python-3-Version dieses Pakets ist in AL2 0.23 enthalten. Sie können das `python3-simplejson` Paket in AL2 023 mit dem folgenden Befehl installieren:

```
[ec2-user ~]$ sudo dnf install python3-simplejson
```

Python-Module können auch auf Anfrage installiert werden DNF um das Paket zu installieren, das ein bestimmtes Python-Modul bereitstellt.

```
[ec2-user ~]$ sudo dnf install 'python3dist(simplejson)'
```

`rkhunter`- Rootkit Hunter

Das `rkhunter` Paket ist in AL2 023 zusammen mit enthalten. `chkrootkit`

```
[ec2-user ~]$ sudo dnf install rkhunter
```

```
[ec2-user ~]$ sudo dnf install chkrootkit
```

`rssh`- eine eingeschränkte Shell zur Verwendung mit OpenSSH

Das `rssh` Paket war in EPEL7. Das `rssh`Upstream-Paket wird nicht gewartet. Als Grund nennt der Autor unbehebbare Sicherheitsprobleme.

Da der Autor unbehebbare Sicherheitsprobleme anführt, ist eine Erweiterung der Version 023 nicht geplant. `rssh` AL2

`sscg`- Generator für selbstsignierte SSL-Zertifikate

Das `sscg` Paket befand sich sowohl im AL2 Core als auch in EPEL7. Das `sscg` Paket ist auch Teil von AL2 023.

```
[ec2-user ~]$ sudo dnf install sscg
```

`stress`- Stresstest

Das `stress` Paket war drin EPEL7, und war auch erhältlich in AL1

Das `stress` Paket ist in Version AL2 023 enthalten und kann mit dem folgenden Befehl installiert werden:

```
[ec2-user ~]$ sudo dnf install stress
```

`stress-ng`- Stresstest

Das `stress-ng` Paket war drin EPEL7, und war auch im `testing` AL2 Extra erhältlich.

Das `stress-ng` Paket ist in Version AL2 023 enthalten und kann mit dem folgenden Befehl installiert werden:

```
[ec2-user ~]$ sudo dnf install stress-ng
```

`tmpwatch`- entfernt Dateien basierend auf dem Zeitpunkt des letzten Zugriffs

In Amazon Linux 2023 wird diese Funktionalität von bereitgestellt `systemd-tmpfiles`.

`xmlstarlet`- XML-Hilfsprogramme für die Befehlszeile

Das `xmlstarlet` Paket war in EPEL7, und ist in AL2 023 nicht verfügbar.

Das Upstream-Paket wurde seit über 9 Jahren nicht mehr angefasst (zuletzt im August 2014). Weitere vier Jahre zuvor (mindestens seit Juli 2010) blieb eine Anfrage nach einem neuen Maintainer unbeantwortet. Aus diesem Grund ist eine Erweiterung um 023 nicht geplant. `xmlstarlet` AL2

Python 2.7 wurden durch Python 3 ersetzt

AL2 bietet bis Juni 2025 Support und Sicherheitspatches für Python 2.7 als Teil unseres langfristigen Supports (LTS) für AL2 Kernpakete. 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 AL2 023 wurde der `dnf` Paketmanager auf Python 3 migriert und benötigt Python 2.7 nicht mehr. AL2023 wurde komplett auf Python 3 umgestellt.

> **Note**
>
> AL2023 hat Python 2.7 entfernt, sodass alle Betriebssystemkomponenten, die Python benötigen, so geschrieben sind, dass sie mit Python 3 funktionieren. 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 im Jahr AL2 023.

Sicherheits-Updates

Amazon Linux 2023 verbessert die in vorhandene Härtung AL2. Weitere Informationen finden Sie unter Sicherheit und Compliance in Amazon Linux 2023. Weitere Informationen zu Änderungen an der Kernelhärtung von AL2 finden Sie unterSicherheitsorientierte Änderungen an der Kernel-Konfiguration.

Themen

- SELinux
- OpenSSL 3
- IMDSv2
- Entfernen des log4j-Hotpatch (log4j-cve-2021-44228-hotpatch)

SELinux

Standardmäßig ist Security Enhanced Linux (SELinux) für AL2 023 `enabled` und ist auf `permissive` Modus eingestellt. Im `permissive`-Modus werden Zugriffsverweigerungen protokolliert, aber nicht durchgesetzt.

SELinuxist eine Sicherheitsfunktion des Amazon Linux-Kernels, die enthalten war `disabled` AL2. SELinuxist eine Sammlung von Kernelfunktionen und Hilfsprogrammen, die eine Architektur mit obligatorischer Zugriffskontrolle (MAC) für die wichtigsten Subsysteme des Kernels bereitstellt.

Weitere Informationen finden Sie unter SELinux Einstellmodi für AL2 023.

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

OpenSSL 3

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

Standardmäßig wird AL2 OpenSSL 1.0.2 mitgeliefert. 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 AL2 023-AMI gestarteten Instances IMDSv2 nur -only und Ihr Standard-Hop-Limit ist auf 2 festgelegt, 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 EC2 Amazon-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 Option Instanz-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 EC2 Amazon-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 und. AL1 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 Version für allgemeine Verfügbarkeit (General Availability, GA) von AL2 023 wurde 15 Monate nach CVE-2021-44228 veröffentlicht, weshalb AL2 023 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, Aktualisierung AL2 023 damit Sie über Sicherheitspatches auf dem Laufenden bleiben können. Sicherheitsempfehlungen werden im Amazon Linux Security Center veröffentlicht.

Deterministische Upgrades für Stabilität

Mit der Funktion „Deterministische Upgrades durch versionierte Repositorys“ ist jedes AL2 023-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 AL2 023 ist das deterministische Upgrade standardmäßig aktiviert. Dies ist eine Verbesserung gegenüber der manuellen, inkrementellen Sperrmethode, die in AL2 und anderen früheren Versionen verwendet wurde.

Weitere Informationen finden Sie unter Deterministische Upgrades durch versionierte Repositorys auf 023 AL2.

gp3als standardmäßiger Amazon EBS-Volumetyp

Das AL2 023 AMI und AL2 beide verwenden das XFS Dateisystem auf dem Root-Dateisystem. Für AL2 023 sind die `mkfs` Optionen für das Root-Geräte-Dateisystem weiter für Amazon EC2 optimiert. AL2023 unterstützt auch eine Reihe anderer Dateisysteme, die Sie je nach Ihren spezifischen Anforderungen auf anderen Volumes verwenden können.

AL2023 AMIs verwendet standardmäßig Amazon `gp3` EBS-Volumes, wohingegen Amazon `gp2` EBS-Volumes standardmäßig AL2 AMIs verwendet werden. 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 EC2 Amazon-Instance finden Sie unter Starten einer Instance im EC2 Amazon-Benutzerhandbuch.

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 cgroupv1 und AL2 023 unterstütztcgroupv2. Dies ist gut zu wissen, wenn containerisierte Workloads ausgeführt werden, z. B. bei Verwendung von AL2 023-basiertem Amazon ECS AMIs zum Hosten von containerisierten Workloads.

Obwohl AL2 023 immer noch Code enthält, mit dem das System ausgeführt werden kanncgroupv1, ist dies 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.

`systemd`Timer ersetzen **`cron`**

Das `cronie` Paket wurde standardmäßig auf dem AL2 AMI installiert und bietet Unterstützung für die herkömmliche `crontab` Art der Planung periodischer Aufgaben. In AL2 023 `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.

Verbesserte Toolchain: gccbinutils, und glibc

AL2023 enthält viele der gleichen Kernpakete wie. AL2

Wir haben die folgenden drei Core-Toolchain-Pakete für AL2 023 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-Pakete glibc, gcc, binutils.

Weitere Informationen zuC, und Fortran SprachlaufzeitenC++, einschließlich aktualisierter Standardsprachenstandards, finden Sie unter. CC++, und Fortran im Jahr AL2 203

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

systemdJournal ersetzt **rsyslog**

Im AL2 Jahr 023 wurde das Logging-Systempaket von AL2 geändert. AL2023 `rsyslog` wird standardmäßig nicht installiert, daher sind die textbasierten Protokolldateien`/var/log/messages`, wie sie in verfügbar waren, standardmäßig AL2 nicht verfügbar. Die Standardkonfiguration für AL2 023 ist`systemd-journal`, was mit untersucht werden kann. `journalctl` Obwohl `rsyslog` es sich in AL2 023 um ein optionales Paket handelt, empfehlen wir die neue `systemd` `journalctl` Basisschnittstelle und zugehörige Pakete. Weitere Informationen finden Sie auf der `journalctl` Seite im Handbuch.

Das systmed journal Äquivalent zu einigen häufig verwendeten syslog Befehlen wird in der folgenden Tabelle behandelt.

AL2 syslog command	AL2023 entspricht systemd journal
`[ec2-user ~]$ cat /var/log/messages`	`[ec2-user ~]$ journalctl`
`[ec2-user ~]$ tail -f /var/log/messages`	`[ec2-user ~]$ journalctl -f`
`[ec2-user ~]$ grep foo /var/log/messages`	`[ec2-user ~]$ journalctl | grep foo`

Minimierte Paketabhängigkeiten

Amazon Linux 2023 minimiert das Abhängigkeitsdiagramm vieler Pakete, um den Platzbedarf für Anwendungen zu verringern. Bemerkenswerte Änderungen von AL2 Include the `curl-minimal` und `gnupg-minimal` Paketen, die die Anzahl der benötigten Pakete deutlich reduzieren und gleichzeitig häufig verwendete Funktionen beibehalten.

Themen

- Paketänderungen für curl und libcurl
- GNU Privacy Guard (GNUPG)

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

AL2023 unterteilt die gängigen Protokolle und Funktionen der Pakete und in `curl` und`libcurl`. `curl-minimal` `libcurl-minimal` Dies reduziert den Speicherplatz für Festplatten, Arbeitsspeicher und Abhängigkeiten für die meisten Benutzer und ist das Standardpaket für AL2 023 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 Pakete `gnupg2-minimal` und`gnupg2-full`. 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) ausgeliefert. Alle Java basierten Pakete in AL2 023 sind alle mit erstellt. Amazon Corretto 17

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

AWS CLI v2

AL2023 wird mit AWS CLI Version 2 ausgeliefert, wohingegen sie mit Version 1 von AL2 ausgeliefert wird AWS CLI.

UEFI Preferred und Secure Boot

Standardmäßig werden alle Instances, die mit dem AL2 023-AMI auf Instance-Typen gestartet werden, 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 EC2 Amazon-Benutzerhandbuch.

Bei EC2 Amazon-Instance-Typen, die UEFI Secure Boot unterstützen, ist es möglich, Secure Boot in Amazon Linux 2023 zu aktivieren. Weitere Informationen finden Sie unter UEFI Secure Boot auf 023 AL2.

Änderungen der Standardkonfiguration des SSH-Servers

Für das AL2 023-AMI haben wir die Typen von `sshd` Hostschlüsseln 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.

Darüber hinaus enthalten die AL2 023-Konfigurationseinstellungen in der `sshd_config` Standarddatei. `UseDNS=no` 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.

AL2023 Kerneländerungen von AL2

AL2023 bringt 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.

IPv4 TTL

Die TTL für IPv4 wird über konfiguriert`sysctl`, wobei die Standardwerte in vorhanden sind. `/etc/sysctl.d/00-defaults.conf` Dieser Wert kann mit den üblichen `sysctl` Methoden angepasst werden. Weitere Informationen finden Sie `sysctl` `man` auf der Seite.

AL2 setzt den `net.ipv4.ip_default_ttl` Wert auf 255, während AL2 023 ihn auf 127 festlegt. Dadurch werden die Standardeinstellungen von Amazon Linux mit denen anderer wichtiger Linux-Distributionen in Einklang gebracht. Es wird nicht empfohlen, diese Standardeinstellung ohne nachgewiesene Notwendigkeit zu ändern.

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	AL2023/61/aarch64	AL2023/61/x86_64	AL2023/612/aarch64	AL2023/612/x86_64
CONFIG_IG_ON_DA_CORRUPON	n	y	n	y	y	y	y	y
CONFIG_IFAULT_MIP_MIN_AIR	4096	4096	4096	4096	65536	65536	65536	65536
CONFIG_IVMEM	n	y	n	y	n	n	n	n
CONFIG_IVPORT	n	y	n	y	n	n	n	n
CONFIG_IRTIFY_SCRCE	n	y	n	y	y	y	y	y
CONFIG_IRDENED_U	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	AL2023/6. 12/ aarch64	AL2023/6, 12/ x86_64
ERCOPY_I LLBACK								
CONFIG_ IT_ON_AI OC_DEFAU T_ON	–	–	n	n	n	n	n	n
CONFIG_ IT_ON_FI E_DEFAUI _ON	–	–	n	n	n	n	n	n
CONFIG_ MMU_DEF/ LT_DMA_S RICT	–	–	–	–	n	n	n	n
CONFIG_I ISC_AUTC OAD	y	y	y	y	n	n	n	n
CONFIG_S HED_CORI	–	–	–	–	–	y	–	y
CONFIG_S HED_STAC _END_CHI K	n	y	n	y	y	y	y	y
CONFIG_S CURITY_I ESG_RES ICT	n	n	n	n	y	y	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	AL2023/6.12/aarch64	AL2023/6,12/x86_64
CONFIG_SECURITY_SELINUX_DISABLE	y	y	y	y	n	n	–	–
CONFIG_SLUB_PAGE_ALLOCATOR	–	N/A	y	y	y	y	y	y
CONFIG_SLAB_FREELIST_HARDENED	n	y	y	y	y	y	y	y
CONFIG_SLAB_FREELIST_RANDOM	n	n	y	y	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	AL2023/6,12/x86_64
CONFIG_AMD_IOMMU	y	y	y	y
CONFIG_AMD_IOMMU_V2	m	m	y	N/A

CONFIG-Option	AL2/4,14/x86_64	AL2/5.10/x86_64	AL2023/6,1/x86_64	AL2023/6,12/x86_64
CONFIG_RANDOMIZE_MEMORY	N/A	y	y	y

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

CONFIG-Option	AL2/4.14/aarch64	AL2/5.10/aarch64	AL2023/6.1/aarch64	AL2023/6,12/aarch64
CONFIG_ARM64_PTR_AUTH	N/A	y	y	y
CONFIG_ARM64_PTR_AUTH_KERNEL	–	N/A	y	y
CONFIG_ARM64_SW_TTBR0_PAN	y	y	y	y

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

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

Der `/dev/kmem` Code wurde im 5.13-Kernel vollständig aus Linux entfernt, und obwohl er in Version 0.23 deaktiviert wurde AL2, gilt er jetzt nicht mehr für 023. AL2

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

`FORTIFY_SOURCE`

AL2023 wird `CONFIG_FORTIFY_SOURCE` auf allen unterstützten Architekturen 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 AL2 023-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.

`dmesg`Zugriff für unprivilegierte Benutzer () `CONFIG_SECURITY_DMESG_RESTRICT`

Standardmäßig erlaubt AL2 023 unberechtigten Benutzern keinen Zugriff auf. `dmesg`

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

SELinux **`selinuxfs`**deaktivieren

AL2023 deaktiviert die veraltete `CONFIG_SECURITY_SELINUX_DISABLE` Kernel-Option, die eine Laufzeitmethode zur Deaktivierung 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	AL2023/61/aarch64	AL2023/61/x86_64	AL2023/612/aarch64	AL2023/612/x86_64
CONFIG_I	100	250	100	250	100	100	100	100
CONFIG_I_CPUS_	4096	8192	4096	8192	4096	8192	4096	8192

CONFIG-Option	AL2/4.14/aarch64	AL2/4,14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/61/aarch64	AL2023/61/x86_64	AL2023/612/aarch64	AL2023/612/x86_64
CONFIG_NIC_ON_CPS	y	n	y	n	y	y	y	y
CONFIG_NIC_ON_CPS_VALU	1	0	1	0	1	1	1	1
CONFIG_P	m	m	m	m	n	n	n	n
CONFIG_IP	m	m	m	m	n	n	n	n
CONFIG_N_PV	N/A	y	–	n	–	n	N/A	n

CONFIG_HZ

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

CONFIG_NR_CPUS

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

-Panic bei OOPS

Der AL2 023-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, in denen der Kernel das Einhängen unterstützt, zusammen mit Änderungen an den Partitionierungsschemata, die der Kernel analysiert. AL2

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	AL2023/6. 12/ aarch64	AL2023/6, 12/ x86_64
CONFIG_A S_FS	n	m	n	m	n	n	n	n
CONFIG_A _RXRPC	n	m	n	m	n	n	n	n
CONFIG_B D_DISKL EL	y	y	y	y	n	n	n	n
CONFIG_C AMFS	m	m	m	m	n	n	n	n
CONFIG_C AMFS_BL KDEV	N/A	–	y	n	–	–	–	–
CONFIG_E _CLONE	–	–	n	n	n	n	n	n
CONFIG_E _ERA	m	n	m	n	n	n	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	AL2023/6. 12/ aarch64	AL2023/6, 12/ x86_64
CONFIG_I _INTEGR Y	n	m	n	m	m	m	m	m
CONFIG_I _LOG_WR ES	n	n	m	m	m	m	m	m
CONFIG_I _SWITCH	m	n	m	n	n	n	n	n
CONFIG_I _VERITY	m	n	m	n	n	n	n	n
CONFIG_I RYPT_FS	n	m	n	m	n	n	n	n
CONFIG_I FAT_FS	–	N/A	m	m	m	m	m	m
CONFIG_I T2_FS	n	m	n	m	n	n	n	n
CONFIG_I T3_FS	n	m	n	m	n	n	n	n
CONFIG_(S2_FS	m	m	m	m	n	n	n	n
CONFIG_I SPLUS_F	n	m	n	m	n	n	n	n
CONFIG_I S_FS	n	m	n	m	n	n	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	AL2023/6. 12/ aarch64	AL2023/6, 12/ x86_64
CONFIG_ S_FS	n	n	n	n	n	n	n	n
CONFIG_ M_PARTI ON	n	y	n	y	n	n	n	n
CONFIG_ C_PARTI ON	n	y	n	y	n	n	n	n
CONFIG_ S_V2	n	m	n	m	n	n	n	n
CONFIG_ FS_FS	n	m	n	n	n	n	n	n
CONFIG_ MFS_FS	n	m	n	m	n	n	n	n
CONFIG_ LARIS_X _PARTIT N	n	y	n	y	n	n	n	n
CONFIG_ UASHFS_ TD	n	y	n	y	y	y	y	y
CONFIG_ N_PARTI ON	n	y	n	y	n	n	n	n

-Andrew-Dateisystem-Unterstützung (AFS)

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

-cramfs-Unterstützung

Der Kernel wird nicht mehr mit Unterstützung für das `cramfs`-Dateisystem erstellt. Der Nachfolger in AL2 023 ist das `squashfs` Dateisystem.

Unterstützung für BSD-disklabel

Der Kernel bietet keine Unterstützung für BSD-Disk-Labels mehr. Wenn das Lesen von Volumes mit BSD-Festplattenlabels erforderlich ist, BSDs können verschiedene gestartet werden.

-Device Mapper-Änderungen

Es wurden mehrere Änderungen an den im AL2 023-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 vorhanden AL1, wurden entfernt und AL2 023 baut den Kernel nicht mehr mit `ecryptfs` Unterstützung auf. AL2

exFAT

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

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__FS GFS2

Der Kernel wird nicht mehr mit GFS2 CONFIG_ _FS gebaut.

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. Im AL2 Jahr 023 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. Im AL2 Jahr 023 schließen wir die Einstellung der `hfs` Unterstützung in Amazon Linux ab.

Unterstützung für das JFS-Dateisystem

Ältere AL2 `x86-64` Kernel wurden mit `jfs` Dateisystemunterstützung erstellt. Der AL2 5.15-Kernel bietet keine `jfs` Unterstützung für jede Architektur. AL1 Weder noch im AL2 Lieferumfang des JFS-Benutzerbereichs enthalten. Im AL2 Jahr 023 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. Im Jahr 2024 `JFS` wurde es aus allen aktuellen Amazon Linux-Kerneln entfernt.

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.

NFSv2 Unterstützung

AL2023 unterstützt NFSv2,, NFSv4 .1 und NFSv4 .2 nicht mehr NFSv3 NFSv4, aber weiterhin. Wir empfehlen Ihnen, auf NFSv3 oder eine neuere Version 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 ersetzt. AL2 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 AL2 023-Kernel unterstützt nicht mehr. `romfs`

Solaris-x86-Festplattenpartitionsformat

AL2023 unterstützt das Solaris x86-Festplattenpartitionsformat nicht mehr.

squashfs-zstd-Komprimierung

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

Unterstützung für Sun-Partitionstabellen

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

/tmp ist jetzt **tmpfs**

Amazon Linux 2023 führt im Vergleich zu Amazon Linux 2 Änderungen am `/tmp` Verhalten ein. Die Standardkonfiguration für AL2 war, dass sich `/tmp` sowohl als auch im Root-Dateisystem `/var/tmp` befanden. Amazon Linux 2023 verwendet standardmäßig `tmpfs` for `/tmp` mit einem Limit von 50% RAM und maximal einer Millioninodes. Diese Änderungen bringen Amazon Linux in Einklang mit dem Verhalten anderer Linux-Distributionen.

Vollständige Informationen zum Dateisystem-Layout von AL2 023 finden Sie unter `/tmp` und `/var/tmp` im Dateisystem-Layout Abschnitt.

Änderungen an AMI und Container-Images

Es wurden einige Änderungen an den Paketen vorgenommen, die in AMIs Containern enthalten sind.

Amazon Linux 2023 führt eine the section called “AL2023 Minimales Container-Image” und Unterstützung für das Erstellen einthe section called “Erstellung von Bare-Bones AL2 (023) Container-Images”. Weitere Informationen finden Sie unter AL2023 in Containern verwenden.

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

Ein Vergleich der RPMs aktuellen Standards AMIs Amazon Linux 2 und AL2 023.

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`		2.5.1
`amazon-linux-extras`	2.0.3	
`amazon-linux-extras-yum-plugin`	2.0.3	
`amazon-linux-repo-s3`		2023,6.20241031
`amazon-linux-sb-keys`		2023,1
`amazon-rpm-config`		228
`amazon-ssm-agent`	3,3,987,0	3,3,987,0
`amd-ucode-firmware`	20200421 (Noarch)	20210208 (Noarch)
`at`	3.1.13	3.1.23
`attr`	2,4,46	2.5.1

Paket	AL2 AMI	AL2023 AMI
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	9,18,28
bind-libs-lite	9,11,4	
bind-license	9.11,4	9,18,28
bind-utils	9,11,4	9,18,28
binutils	2,29,1	2,39
blktrace	1.0.5	
boost-date-time	1,53,0 (x86_64)	
boost-filesystem		1,75,0
boost-system	1,53,0 (x86_64)	1,75,0

Paket	AL2 AMI	AL2023 AMI
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	2023,2,68	2023,2,68
c-ares		1.19.1
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	
cronie-anacron	1.4.11	
crontabs	1.11	1.11

Paket	AL2 AMI	AL2023 AMI
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
device-mapper-persistent-data	0.7.3	
dhclient	4.2,5	

Paket	AL2 AMI	AL2023 AMI
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
dwz		0,14
dyninst	9.3.1 (x86_64)	10.2.1

Paket	AL2 AMI	AL2023 AMI
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
elfutils-libs	0,176	0.188
ethtool	4,8	5,15

Paket	AL2 AMI	AL2023 AMI
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
gettext-libs	0.19.8.1	0,21
ghc-srpm-macros		1.5.0

Paket	AL2 AMI	AL2023 AMI
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
grub2-efi-aa64	2,06 (aarch64)	
grub2-efi-aa64-ec2	2,06 (aarch64)	2,06 (aarch64)

Paket	AL2 AMI	AL2023 AMI
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	0,384
info	5.1	6.7
inih		49

Paket	AL2 AMI	AL2023 AMI
initscripts	9,49,47	10,09
iproute	5.10.0	6.10.0
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	
jemalloc		5.2.1
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	5.10.228	6.1.112
kernel-libbpf		6.1.112
kernel-livepatch-repo-s3		2023,6,20241031
kernel-srpm-macros		1,0
kernel-tools	5,10.228	6.1.112

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	1.21.3
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		3.7.4
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
linux-firmware-whence		20210208 (Noarch)
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	

Paket	AL2 AMI	AL2023 AMI
mdadm	4,0	
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)
mlocate	0,26	
mpfr		4.1.0
mtr	0.92	
nano	2,9,8	5,8
ncurses	6.0	6.2
ncurses-base	6.0	6.2
ncurses-libs	6.0	6.2
nettle	2.7.1	3.8
net-tools	2,0	2.0
newt	0,52,15	0,52,21
newt-python	0,52,15	
nfs-utils	1.3.0	2,5.4
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

Paket	AL2 AMI	AL2023 AMI
nss-sysinit	3,90,0	3,90,0
nss-tools	3,90,0	
nss-util	3,90,0	3,90,0
ntsysv	1.7.4	1.15
numactl-libs	2.0.9	2.0.14
ocaml-srpm-macros		6
oniguruma		6.9.7.1
openblas-srpm-macros		2
openldap	2.4.44	2,4,57
openssh	7,4p1	8,7 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

Paket	AL2 AMI	AL2023 AMI
parted	3.1	3.4
passwd	0,79	0,80
pciutils	3.5.1	3.7.0
pciutils-libs	3.5.1	3.7.0
pcre	8,32	
pcre2	10,23	10,40
pcre2-syntax		10,40
perl	5,16.3	
perl-Carp	1,26	1,50
perl-Class-Struct		0,66
perl-constant	1,27	1,33
perl-DynaLoader		1,47
perl-Encode	2,51	3,15
perl-Errno		1,30
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

Paket	AL2 AMI	AL2023 AMI
perl-Filter	1,49	
perl-Getopt-Long	2,40	2,52
perl-Getopt-Std		1.12
perl-HTTP-Tiny	0,033	0,078
perl-if		0,60,800
perl-interpreter		5,32,1
perl-IO		1,43
perl-IPC-Open3		1,21
perl-libs	5.16,3	5.32,1
perl-macros	5.16,3	
perl-MIME-Base64		3,16
perl-mro		1,23
perl-overload		1,31
perl-overloading		0,02
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

Paket	AL2 AMI	AL2023 AMI
perl-Pod-Usage	1,63	2,01
perl-POSIX		1,94
perl-Scalar-List-Utils	1,27	1,56
perl-SelectSaver		1.02
perl-Socket	2,010	2,032
perl-srpm-macros		1
perl-Storable	2,45	3,21
perl-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

Paket	AL2 AMI	AL2023 AMI
pkgconfig	0,27,1	
pkgconf-m4		1.8.0
pkgconf-pkg-config		1.8.0
plymouth	0,8,9	
plymouth-core-libs	0,8,9	
plymouth-scripts	0,8,9	
pm-utils	1.4.1	
policycoreutils	2.5	3.4
policycoreutils-python-utils		3.4
popt	1.13	1,18
postfix	2.10.1	
procps-ng	3.3.10	3.3.17
protobuf-c		1.4.1
psacct	6.6.1	6.6.4
psmisc	22,20	23,4
pth	2.0.7	
publicsuffix-list-dafsa	20240208	20240212
pygpgme	0.3	
pyliblzma	0,5,3	

Paket	AL2 AMI	AL2023 AMI
pystache	0,5.3	
python	2.7.18	
python2-botocore	1.18.6	
python2-colorama	0.3.9	
python2-cryptography	1.7.2	
python2-dateutil	2.6.1	
python2-futures	3.0.5	
python2-jmespath	0.9.3	
python2-jsonschema	2.5.1	
python2-oauthlib	2.0.1	
python2-pyasn1	0.1.9	
python2-rpm	4.11.3	
python2-rsa	3.4.1	
python2-s3transfer	0.3.3	
python2-setuptools	41,2,0	
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

Paket	AL2 AMI	AL2023 AMI
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-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

Paket	AL2 AMI	AL2023 AMI
python3-jsonschema		3.2.0
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

Paket	AL2 AMI	AL2023 AMI
python3-pyrsistent		0,17,3
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pystache	0.5.4	
python3-pytz		2022.7.1
python3-pyyaml		5.4.1
python3-requests		2,25,1
python3-rpm		4.16.1,3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools	49.1,3	59,6,0
python3-setuptools-wheel		59,6,0
python3-simplejson	3.2.0	
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1,25,10
python3-wcwidth		0,2,5
python-babel	0.9.6	

Paket	AL2 AMI	AL2023 AMI
python-backports	1,0	
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	

Paket	AL2 AMI	AL2023 AMI
python-markupsafe	0,11	
python-pillow	2.0.0	
python-ply	3.4	
python-pycparser	2.14	
python-pycurl	7,19,0	
python-repoze-lru	0.4	
python-requests	2.6.0	
python-simplejson	3.2.0	
python-srpm-macros		3.9
python-urlgrabber	3,10	
python-urllib3	1,25,9	
pyxattr	0.5.1	
PyYAML	3,10	
qrencode-libs	3.4.1	
quota	4,01	4,06
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

Paket	AL2 AMI	AL2023 AMI
rpcbind	0.2.0	1.2.6
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
rsync	3.1.2	3.2.6
rsyslog	8.24,0	
rust-srpm-macros		21
sbsigntools		0.9.4
scl-utils	20130529	
screen	4.1.0	4.8.0
sed	4.2.2	4.8
selinux-policy	3.13,1	38,1,45
selinux-policy-targeted	3,13,1	38,1,45
setserial	2,17	
setup	2,8,71	2.13,7
setuptool	1.19,11	

Paket	AL2 AMI	AL2023 AMI
sgpio	1.2.0.10	
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	252,23
systemd-libs	219	252,23
systemd-networkd		252,23
systemd-pam		252,23
systemd-resolved		252,23
systemd-sysv	219	

Paket	AL2 AMI	AL2023 AMI
systemd-udev		252,23
system-release	2	2023,6,20241031
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

Paket	AL2 AMI	AL2023 AMI
util-linux-core		2,37,4
vim-common	9,0,2153	9,0,2153
vim-data	9,0,2153	9,0,2153
vim-enhanced	9,0,2153	9,0,2153
vim-filesystem	9,0,2153	9,0,2153
vim-minimal	9,0,2153	9,0,2153
virt-what	1,18	
wget	1.14	1,21,3
which	2,20	2,21
words	3.0	3.0
xfsdump	3.1.8	3.1.11
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	

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

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

Ein Vergleich der RPMs Gegenwart auf Amazon Linux 2 und AL2 023 Minimal AMIs.

Paket	AL2 Minimal	AL2023 Minimal
acl	2.2.51	
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.5.1
amazon-linux-extras	2.0.3	
amazon-linux-repo-s3		2023,6,20241031
amazon-linux-sb-keys		2023,1

Paket	AL2 Minimal	AL2023 Minimal
amd-ucode-firmware	20200421 (Noarch)	20210208 (Noarch)
audit	2.8.1	3.0.6
audit-libs	2.8.1	3.0.6
authconfig	6.2.8	
awscli-2		2,15,30
basesystem	10.0	11
bash	4.2,46	5.2.15
bind-export-libs	9,11,4	
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023,2,68	2023,2,68
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

Paket	AL2 Minimal	AL2023 Minimal
cracklib-dicts	2.9.0	2.9.6
cronie	1.4.11	
cronie-anacron	1.4.11	
crontabs	1.11	
crypto-policies		20220428
cryptsetup-libs	1,7.4	2.6.1
curl	8.3.0	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.26	2.1.27
dbus	1.10,24	1.12,28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.10.24	1.12,28
device-mapper	1,02,170	1,02,185
device-mapper-libs	1,02,170	1,02,185
dhclient	4.2,5	
dhcp-common	4.2.5	
dhcp-libs	4.2.5	
diffutils	3.3	3.8
dnf		4.14.0

Paket	AL2 Minimal	AL2023 Minimal
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0
dnf-plugin-support-info		1.2
dracut	033	055
dracut-config-ec2	2.0	3.0
dracut-config-generic	033	055
e2fsprogs	1,42,9	1,46,5
e2fsprogs-libs	1,42,9	1,46,5
ec2-utils	1.2	2.2.0
efibootmgr	15 (aarch64)	
efi-filesystem		5
efivar		38
efivar-libs	31 (aarch64)	38
elfutils-default-yama-scope	0,176	0.188
elfutils-libelf	0,176	0.188
elfutils-libs	0,176	0.188
expat	2.1.0	2.5.0

Paket	AL2 Minimal	AL2023 Minimal
file	5,11	5,39
file-libs	5,11	5,39
filesystem	3.2	3,14
findutils	4.5.11	4.8.0
fipscheck	1.4.1	
fipscheck-lib	1.4.1	
freetype	2.8	
fuse-libs	2.9.2	2.9.9
gawk	4.0.2	5.1.0
gdbm	1.13	
gdbm-libs		1,19
gdisk	0,8,10	1.0.8
gettext	0.19.8.1	0,21
gettext-libs	0.19.8.1	0,21
glib2	2,56,1	2,74,7
glibc	2,26	2,34
glibc-all-langpacks	2,26	2,34
glibc-common	2,26	2,34
glibc-locale-source	2,26	2,34
glibc-minimal-lang pack	2,26	

Paket	AL2 Minimal	AL2023 Minimal
gmp	6.0.0	6.2.1
gnupg2	2.0.22	
gnupg2-minimal		2.3.7
gnutls		3.8.0
gpgme	1.3.2	1.15.1
grep	2,20	3.8
groff-base	1,22,2	1.22,4
grub2	2,06	
grub2-common	2,06	2,06
grub2-efi-aa64	2,06 (aarch64)	
grub2-efi-aa64-ec2	2,06 (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	

Paket	AL2 Minimal	AL2023 Minimal
hostname	3.13	3,23
hwdata		0,384
info	5.1	
inih		49
initscripts	9,49,47	10,09
iproute	5.10.0	6.10.0
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	4,14,355	6.1.112
kernel-libbpf		6.1.112
kernel-livepatch-repo-s3		2023,6,20241031
keyutils-libs	1,5.8	1.6.3

Paket	AL2 Minimal	AL2023 Minimal
kmod	25	29
kmod-libs	25	29
kpartx	0.4.9	
krb5-libs	1.15.1	1.21.3
less	458	608
libacl	2,2,51	2.3.1
libarchive		3.7.4
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

Paket	AL2 Minimal	AL2023 Minimal
libdb	5.3.21	5.3.28
libdb-utils	5.3.21	
libdnf		0,69,0
libeconf		0,4,0
libedit	3.0	3.1
libestr	0.1.9	
libfastjson	0,99,4	
libfdisk	2,30,2	2,37,4
libffi	3.0,13	3.4.4
libfido2		1.10.0
libgcc	7.3.1	11.4.1
libgcrypt	1.5.3	1.10.2
libgomp	7.3.1	11.4.1
libgpg-error	1.12	1,42
libicu	50,2	
libidn	1,28	
libidn2	2.3.0	2.3.2
libkcapi		1.4.0
libkcapi-hmaccalc		1.4.0
libmetalink	0.1.3	

Paket	AL2 Minimal	AL2023 Minimal
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	
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

Paket	AL2 Minimal	AL2023 Minimal
libsolv		0,7.22
libss	1,42,9	1,46,5
libssh2	1.4.3	
libstdc++	7.3.1	11.4.1
libsysfs	2.1.0	
libtasn1	4,10	4.19,0
libtextstyle		0,21
libunistring	0.9.3	0,9,10
libuser	0,60	0,63
libutempter	1.1.6	1.2.1
libuuid	2,30,2	2,37,4
libverto	0,2,5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libyaml	0.1.4	0.2.5
libzstd		1.5.5
linux-firmware-whence		20210208 (Noarch)
logrotate	3.8.6	3,20,1
lua	5.1.4	
lua-libs		5.4.4

Paket	AL2 Minimal	AL2023 Minimal
lz4	1.7.5	
lz4-libs		1.9.4
make	3,82	
man-db	2.6.3	2.9.3
mariadb-libs	5.5,68	
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)
mpfr		4.1.0
ncurses	6.0	6.2
ncurses-base	6.0	6.2
ncurses-libs	6.0	6.2
nettle	2.7.1	3.8
net-tools	2,0	2.0
newt	0,52,15	
newt-python	0,52,15	
npth		1,6
nspr	4,35,0	
nss	3,90,0	
nss-pem	1.0.3	
nss-softokn	3,90,0	
nss-softokn-freebl	3,90,0	

Paket	AL2 Minimal	AL2023 Minimal
nss-sysinit	3,90,0	
nss-tools	3,90,0	
nss-util	3,90,0	
numactl-libs	2.0.9	2.0.14
oniguruma		6.9.7.1
openldap	2.4.44	2,4,57
openssh	7,4p1	8,7 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	

Paket	AL2 Minimal	AL2023 Minimal
pcre2	10,23	10,40
pcre2-syntax		10,40
pinentry	0.8.1	
pkgconfig	0,27,1	
policycoreutils	2.5	3.4
popt	1.13	1,18
postfix	2.10.1	
procps-ng	3.3.10	3.3.17
psmisc	22,20	23,4
pth	2.0.7	
publicsuffix-list-dafsa	20240208	20240212
pygpgme	0.3	
pyliblzma	0,5,3	
python	2.7.18	
python2-cryptography	1.7.2	
python2-jsonschema	2.5.1	
python2-oauthlib	2.0.1	
python2-pyasn1	0.1.9	
python2-rpm	4.11.3	
python2-setuptools	41,2,0	

Paket	AL2 Minimal	AL2023 Minimal
python2-six	1.11.0	
python3		3.9,16
python3-attrs		20,3,0
python3-audit		3.0.6
python3-awscrt		0.19,19
python3-babel		2.9.1
python3-cffi		1.14,5
python3-chardet		4.0.0
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36,1
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)

Paket	AL2 Minimal	AL2023 Minimal
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
python3-ply		3.11
python3-policycoreutils		3.4
python3-prettytable		0.7.2
python3-prompt-toolkit		3.0,24
python3-pycparser		2,20

Paket	AL2 Minimal	AL2023 Minimal
python3-pyrsistent		0,17,3
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pytz		2022.7.1
python3-pyyaml		5.4.1
python3-requests		2,25,1
python3-rpm		4.16.1,3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools		59,6,0
python3-setuptools-wheel		59,6,0
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1,25,10
python3-wcwidth		0,2,5
python-babel	0.9.6	
python-backports	1,0	

Paket	AL2 Minimal	AL2023 Minimal
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	
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	

Paket	AL2 Minimal	AL2023 Minimal
python-urlgrabber	3,10	
python-urllib3	1,25,9	
pyxattr	0.5.1	
PyYAML	3,10	
qrencode-libs	3.4.1	
readline	6.2	8.1
rng-tools	6.8	6,14
rootfiles	8.1	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
rsyslog	8.24,0	
sbsigntools		0.9.4
sed	4.2.2	4.8
selinux-policy	3.13,1	38,1,45
selinux-policy-targeted	3,13,1	38,1,45

Paket	AL2 Minimal	AL2023 Minimal
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	
sqlite-libs		3,40,0
sudo	1,8,23	1.9,15
sysctl-defaults	1,0	1,0
systemd	219	252,23
systemd-libs	219	252,23
systemd-networkd		252,23
systemd-pam		252,23
systemd-resolved		252,23
systemd-sysv	219	
systemd-udev		252,23
system-release	2	2023,6,20241031
sysvinit-tools	2,88	
tar	1,26	1,34
tcp_wrappers-libs	7.6	
tzdata	2024a	2024a

Paket	AL2 Minimal	AL2023 Minimal
update-motd	1.1.2	2.2
userspace-rcu		0.12.1
ustr	1.0.4	
util-linux	2,30,2	2,37,4
util-linux-core		2,37,4
vim-data	9,0,2153	9,0,2153
vim-minimal	9,0,2153	9,0,2153
which	2,20	2,21
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 aktuellen Container-Images auf Amazon Linux 2 und AL2 023.

Paket	AL2 Container	AL2023 Behälter
alternatives		1.15
amazon-linux-extras	2.0.3	
amazon-linux-repo-cdn		2023.6.20241031
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	2023,2,68	2023,2,68
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	

Paket	AL2 Container	AL2023 Behälter
diffutils	3.3	
dnf		4.14.0
dnf-data		4.14.0
elfutils-default-yama-scope		0.188
elfutils-libelf	0,176	0.188
elfutils-libs		0.188
expat	2.1.0	2.5.0
file-libs	5,11	5,39
filesystem	3.2	3,14
findutils	4.5.11	
gawk	4.0.2	5.1.0
gdbm	1.13	
gdbm-libs		1,19
glib2	2,56,1	2,74,7
glibc	2,26	2,34
glibc-common	2,26	2,34
glibc-langpack-en	2,26	
glibc-minimal-langpack	2,26	2,34
gmp	6.0.0	6.2.1

Paket	AL2 Container	AL2023 Behälter
gnupg2	2.0.22	
gnupg2-minimal		2.3.7
gpgme	1.3.2	1.15.1
grep	2,20	3.8
info	5.1	
json-c		0,14
keyutils-libs	1,5.8	1.6.3
krb5-libs	1.15.1	1.21.3
libacl	2,2,51	2.3.1
libarchive		3.7.4
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

Paket	AL2 Container	AL2023 Behälter
libdb	5.3.21	
libdb-utils	5.3.21	
libdnf		0,69,0
libffi	3.0,13	3.4.4
libgcc	7.3.1	11.4.1
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.12	1,42
libidn2	2.3.0	2.3.2
libmetalink	0.1.3	
libmodulemd		2.13.0
libmount	2,30,2	2,37,4
libnghttp2	1,41,0	1,59,0
libpsl	0,21,5	0,21,1
librepo		1,14,5
libreport-filesystem		2.15,2
libselinux	2.5	3.4
libsepol	2.5	3.4
libsigsegv		2,13
libsmartcols		2,37,4

Paket	AL2 Container	AL2023 Behälter
libsolv		0,7.22
libssh2	1.4.3	
libstdc++	7.3.1	11,4,1
libtasn1	4,10	4.19,0
libunistring	0.9.3	0.9.10
libuuid	2,30,2	2,37,4
libverto	0,2,5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libyaml		0.2.5
libzstd		1.5.5
lua	5.1.4	
lua-libs		5.4.4
lz4-libs		1.9.4
mpfr		4.1.0
ncurses	6.0	
ncurses-base	6.0	6.2
ncurses-libs	6.0	6.2
npth		1,6
nspr	4,35,0	

Paket	AL2 Container	AL2023 Behälter
nss	3,90,0	
nss-pem	1.0.3	
nss-softokn	3,90,0	
nss-softokn-freebl	3,90,0	
nss-sysinit	3,90,0	
nss-tools	3,90,0	
nss-util	3,90,0	
openldap	2,4,44	
openssl-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	

Paket	AL2 Container	AL2023 Behälter
python	2.7.18	
python2-rpm	4.11.3	
python3		3.9.16
python3-dnf		4.14.0
python3-gpg		1.15.1
python3-hawkey		0,69,0
python3-libcomps		0,120
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

Paket	AL2 Container	AL2023 Behälter
rpm-libs	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
sed	4.2.2	4.8
setup	2,8,71	2.13,7
shared-mime-info	1.8	
sqlite	3.7.17	
sqlite-libs		3,40,0
system-release	2	2023,6,20241031
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 AL1 und AL2 023

In den folgenden Themen werden die wichtigsten Unterschiede zwischen AL1 und AL2 023 beschrieben, die im Vergleich mit noch nicht behandelt wurden. AL2

> **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 Wartungssupport finden Sie im Blogbeitrag Update on Amazon Linux AMI end-of-life. Wir empfehlen, dass Sie Ihre Anwendungen auf Version AL2 023 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 installierten Pakete AMIs
- Vergleich der auf Amazon Linux 1 (AL1) und Amazon Linux 2023 Minimal installierten Pakete AMIs
- Vergleich von Paketen, die auf Amazon Linux 1 (AL1) und Amazon Linux 2023 Basiscontainer-Images installiert sind

Support für die einzelnen Versionen

Für AL2 Version 023 bieten wir fünf Jahre Support ab dem Veröffentlichungsdatum an. AL1 hat den Standardsupport am 31. Dezember 2020 und den Wartungssupport am 31. Dezember 2023 eingestellt.

Weitere Informationen finden Sie unter Release-Taktfrequenz.

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

In AL2 `upstart` wurde durch `systemd` as das `init` System ersetzt. AL2023 verwendet auch `systemd` als `init` System und übernimmt dabei weitere neue Merkmale und Funktionen von`systemd`.

Python 2.6 und 2.7 wurden durch Python 3 ersetzt

Obwohl Python 2.6 mit der Version 2018.03 als EOL AL1 markiert wurde, waren die Pakete weiterhin in den Repositorys zur Installation verfügbar. AL2 wurde mit Python 2.7 als frühester unterstützter Python-Version ausgeliefert, und AL2 023 schließt den Übergang zu Python 3 ab. In den AL2 023-Repositorien sind keine Python 2.x-Versionen enthalten.

Weitere Informationen zu Python auf Amazon Linux finden Sie unter Python im Jahr AL2 023.

OpenJDK 8 als ältestes JDK

AL2023 wird mit Amazon Corretto als standardmäßigem (und einzigen) Java Development Kit (JDK) ausgeliefert. Alle Java basierte Pakete in AL2 023 werden mit erstellt Amazon Corretto 17.

OpenJDK 1.6.0 (`java-1.6.0-openjdk`) wurde 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 Repositorys verfügbar waren. AL1 AL1 Die früheste OpenJDK-Version, die AL2 023 verfügbar war, ist OpenJDK 8, bereitgestellt von Amazon Corretto 8.

AL2023 Kernel-Änderungen gegenüber Amazon Linux (1) AL1

Kernel-Live-Patching

Sowohl AL2 023 als auch unterstützen die Kernel-Live-Patching-Funktionalität. AL2 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 023 AL2.

Unterstützung für das Kernel-Dateisystem

Es wurden mehrere Änderungen an den Dateisystemen vorgenommen, in AL1 denen der Kernel das Einhängen unterstützt, sowie an den Partitionierungsschemata, die der Kernel analysiert.

CONFIG-Option	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6,1/x86_64	AL2023/6.12/aarch64	AL2023/6,12/x86_64
CONFIG_AFS_FS	m	n	n	n	n
CONFIG_AF_RXRPC	m	n	n	n	n
CONFIG_BSD_DISKLABEL	y	n	n	n	n
CONFIG_CRAMFS	m	n	n	n	n
CONFIG_CRAMFS_BLOCKDEV	N/A	–	–	–	–
CONFIG_DM_CLONE	–	n	n	n	n
CONFIG_DM_ERA	n	n	n	n	n
CONFIG_DM_INTEGRITY	m	m	m	m	m
CONFIG_DM_LOG_WRITES	n	m	m	m	m
CONFIG_DM_SWITCH	n	n	n	n	n
CONFIG_DM_VERITY	n	n	n	n	n

CONFIG-Option	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6,1/x86_64	AL2023/6.12/aarch64	AL2023/6,12/x86_64
CONFIG_ECRYPT_FS	m	n	n	n	n
CONFIG_EXFAT_FS	N/A	m	m	m	m
CONFIG_EXT2_FS	m	n	n	n	n
CONFIG_EXT3_FS	m	n	n	n	n
CONFIG_GFS2_FS	n	n	n	n	n
CONFIG_HFSPLUS_FS	m	n	n	n	n
CONFIG_HFS_FS	m	n	n	n	n
CONFIG_JFS_FS	n	n	n	n	n
CONFIG_LDM_PARTITION	y	n	n	n	n
CONFIG_MAC_PARTITION	y	n	n	n	n
CONFIG_NFS_V2	m	n	n	n	n

CONFIG-Option	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6,1/x86_64	AL2023/6.12/aarch64	AL2023/6,12/x86_64
CONFIG_NTFS_FS	m	n	n	n	n
CONFIG_ROMFS_FS	m	n	n	n	n
CONFIG_SOLARIS_X86_PARTITION	y	n	n	n	n
CONFIG_SQUASHFS_ZSTD	y	y	y	y	y
CONFIG_SUN_PARTITION	y	n	n	n	n

Sicherheitsorientierte Änderungen an der Kernel-Konfiguration

CONFIG-Option	AL1/4,14/x86_64	AL2023/6.1/aarch64	AL2023/6,1/x86_64	AL2023/6.12/aarch64	AL2023/6,12/x86_64
CONFIG_BUG_ON_DATA_CORRUPTION	y	y	y	y	y
CONFIG_DEFAULT_MMAP_MIN_ADDR	4096	65536	65536	65536	65536

CONFIG-Option	AL1/4,14/ x86_64	AL2023/6.1/ aarch64	AL2023/6,1/ x86_64	AL2023/6.12/ aarch64	AL2023/6,12/ x86_64
CONFIG_DEVMEM	y	n	n	n	n
CONFIG_DEVPORT	y	n	n	n	n
CONFIG_FORTIFY_SOURCE	y	y	y	y	y
CONFIG_HARDENED_USERCOPY_FALLBACK	N/A	–	–	–	–
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	n	n
CONFIG_LDISC_AUTOLOAD	y	n	n	n	n

CONFIG-Option	AL1/4,14/ x86_64	AL2023/6.1/ aarch64	AL2023/6,1/ x86_64	AL2023/6.12/ aarch64	AL2023/6,12/ x86_64
CONFIG_SCHED_CORE	–	–	y	–	y
CONFIG_SCHED_STACK_END_CHECK	y	y	y	y	y
CONFIG_SECURITY_DMESG_RESTRICT	n	y	y	y	y
CONFIG_SECURITY_SELINUX_DISABLE	y	n	n	–	–
CONFIG_SHUFFLE_PAGE_ALLOCATOR	N/A	y	y	y	y
CONFIG_SLAB_FREELIST_HARDENED	y	y	y	y	y
CONFIG_SLAB_FREELIST_RANDOM	n	y	y	y	y

Weitere Änderungen in der Kernelkonfiguration

CONFIG-Option	AL1/4,14/x86_64	AL2023/6.1/aarch64	AL2023/6,1/x86_64	AL2023/6.12/aarch64	AL2023/6,12/x86_64
CONFIG_HZ	250	100	100	100	100
CONFIG_NR_CPUS	8192	4096	8192	4096	8192
CONFIG_PANIC_ON_OOPS	n	y	y	y	y
CONFIG_PANIC_ON_OOPS_VALUE	0	1	1	1	1
CONFIG_PPP	m	n	n	n	n
CONFIG_SLIP	m	n	n	n	n
CONFIG_XEN_PV	y	N/A	n	N/A	n

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

Ein Vergleich der RPMs aktuellen Standards On the AL1 und AL2 023. AMIs

Paket	AL1 AMI	AL2023 AMI
acl	2.2.49	2.3.1
acpid	2.0.19	2.0.32
alsa-lib	1.0.22	

Paket	AL1 AMI	AL2023 AMI
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.5.1
amazon-linux-repo-s3		2023,6,20241031
amazon-linux-sb-keys		2023,1
amazon-rpm-config		228
amazon-ssm-agent	3,2,2222,0	3,3,987,0
amd-ucode-firmware		20210208
at	3.1.10	3.1.23
attr	2,4,46	2.5.1
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

Paket	AL1 AMI	AL2023 AMI
bc	1,06,95	1,07,1
bind-libs	9.8.2	9,18,28
bind-license		9,18,28
bind-utils	9.8.2	9,18,28
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
ca-certificates	2023,2,62	2023,2,68
c-ares		1.19.1
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	

Paket	AL1 AMI	AL2023 AMI
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
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	

Paket	AL1 AMI	AL2023 AMI
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	
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

Paket	AL1 AMI	AL2023 AMI
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
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

Paket	AL1 AMI	AL2023 AMI
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
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	

Paket	AL1 AMI	AL2023 AMI
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
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

Paket	AL1 AMI	AL2023 AMI
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
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

Paket	AL1 AMI	AL2023 AMI
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
hunspell-filesystem		1.7.0
hwdata	0,233	0,384
info	5.1	6.7
inih		49
initscripts	9,03.58	10,09
iproute	4.4.0	6.10.0
iptables	1,4,21	
iputils	20121221	20210202
irqbalance	1.5.0	1.9.0
jansson		2.14

Paket	AL1 AMI	AL2023 AMI
java-1.7.0-openjdk	1.7.0.321	
javapackages-tools	0.9.1	
jemalloc		5.2.1
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	6.1.112
kernel-libbpf		6.1.112
kernel-livepatch-repo-s3		2023,6,20241031
kernel-srpm-macros		1,0
kernel-tools	4,14,336	6.1.112
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

Paket	AL1 AMI	AL2023 AMI
krb5-libs	1.15.1	1.21.3
lcms2	2.6	
less	436	608
libacl	2,2,49	2.3.1
libaio	0,3.109	0,3.111
libarchive		3.7.4
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
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

Paket	AL1 AMI	AL2023 AMI
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
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

Paket	AL1 AMI	AL2023 AMI
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
libmodulemd		2.13.0
libmount	2,23,2	2,37,4
libnetfilter_conntrack	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	

Paket	AL1 AMI	AL2023 AMI
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
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

Paket	AL1 AMI	AL2023 AMI
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
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

Paket	AL1 AMI	AL2023 AMI
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	
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

Paket	AL1 AMI	AL2023 AMI
linux-firmware-whence		20210208
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	
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	

Paket	AL1 AMI	AL2023 AMI
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
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

Paket	AL1 AMI	AL2023 AMI
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
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

Paket	AL1 AMI	AL2023 AMI
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
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

Paket	AL1 AMI	AL2023 AMI
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
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

Paket	AL1 AMI	AL2023 AMI
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
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

Paket	AL1 AMI	AL2023 AMI
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
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

Paket	AL1 AMI	AL2023 AMI
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	
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	

Paket	AL1 AMI	AL2023 AMI
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	
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	

Paket	AL1 AMI	AL2023 AMI
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
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

Paket	AL1 AMI	AL2023 AMI
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
python3-libs		3,9,16
python3-libselinux		3.4
python3-libsemanage		3.4
python3-libstorage mgmt		1.9.4
python3-lockfile		0.12.2

Paket	AL1 AMI	AL2023 AMI
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
python3-requests		2,25,1
python3-rpm		4.16.1,3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2

Paket	AL1 AMI	AL2023 AMI
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
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

Paket	AL1 AMI	AL2023 AMI
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	
rust-srpm-macros		21
sbsigntools		0.9.4
screen	4.0.3	4.8.0
sed	4.2.1	4.8
selinux-policy		38,1,45

Paket	AL1 AMI	AL2023 AMI
selinux-policy-targeted		38,1,45
sendmail	8,14,4	
setserial	2,17	
setup	2.8,14	2.13,7
sgpio	1.2.0.10	
shadow-utils	4.1.4.2	4,9 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
sudo	1.8.23	1.9,15
sysctl-defaults	1,0	1,0
sysfsutils	2.1.0	
sysstat		12,5.6
systemd		252,23

Paket	AL1 AMI	AL2023 AMI
systemd-libs		252,23
systemd-networkd		252,23
systemd-pam		252,23
systemd-resolved		252,23
systemd-udev		252,23
system-release	2018,03	2023,6.20241031
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
tmpwatch	2.9,16	
traceroute	2.0.14	2.1.3
ttmkfdir	3.0.9	
tzdata	2023c	2024a
tzdata-java	2023c	

Paket	AL1 AMI	AL2023 AMI
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
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	

Paket	AL1 AMI	AL2023 AMI
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
zstd		1.5.5

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

Ein Vergleich der RPMs Gegenwart auf dem AL1 und AL2 023 Minimal. AMIs

Paket	AL1 Minimal	AL2023 Minimal
acpid	2.0.19	
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.5.1
amazon-linux-repo-s3		2023,6.20241031
amazon-linux-sb-keys		2023,1
amd-ucode-firmware		20210208
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	
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023,2,62	2023,2,68
checkpolicy	2.1.10	3.4
chkconfig	1.3.49,3	
chrony		4.3

Paket	AL1 Minimal	AL2023 Minimal
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	
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	

Paket	AL1 Minimal	AL2023 Minimal
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
dnf-plugin-support-info		1.2
dracut	004	055
dracut-config-ec2		3.0
dracut-config-generic		055

Paket	AL1 Minimal	AL2023 Minimal
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
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	

Paket	AL1 Minimal	AL2023 Minimal
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
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

Paket	AL1 Minimal	AL2023 Minimal
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	0,384
info	5.1	
inih		49
initscripts	9,03.58	10,09
iproute	4.4.0	6.10.0
iptables	1,4,21	
iputils	20121221	20210202

Paket	AL1 Minimal	AL2023 Minimal
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	6.1.112
kernel-libbpf		6.1.112
kernel-livepatch-repo-s3		2023,6,20241031
keyutils-libs	1,5.8	1.6.3
kmod	14	29
kmod-libs	14	29
krb5-libs	1.15.1	1.21.3
less	436	608
libacl	2,2,49	2.3.1
libarchive		3.7.4
libargon2		20171227
libassuan	2.0.3	2,5.5
libattr	2,4,46	2.5.1

Paket	AL1 Minimal	AL2023 Minimal
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
libfido2		1.10.0
libgcc		11.4.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.10.2

Paket	AL1 Minimal	AL2023 Minimal
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
libpwquality	1.2.3	1.4.4
librepo		1,14,5
libreport-filesystem		2.15,2
libseccomp		2.5.3

Paket	AL1 Minimal	AL2023 Minimal
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
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

Paket	AL1 Minimal	AL2023 Minimal
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libyaml	0.1.6	0.2.5
libzstd		1.5.5
linux-firmware-whence		20210208
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
nettle		3.8
net-tools	1,60	2.0
newt	0,52,11	

Paket	AL1 Minimal	AL2023 Minimal
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
openssh-server	7,4p1	8,7 p1
openssl	1,2k	3.0.8
openssl-libs		3.0.8

Paket	AL1 Minimal	AL2023 Minimal
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	
procps-ng		3.3.17
psmisc	22,20	23,4
pth	2.0.7	

Paket	AL1 Minimal	AL2023 Minimal
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	
python27-pyxattr	0.5.0	
python27-PyYAML	3,10	
python27-requests	1.2.3	

Paket	AL1 Minimal	AL2023 Minimal
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
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils		0,16

Paket	AL1 Minimal	AL2023 Minimal
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
python3-ply		3.11
python3-policycoreutils		3.4
python3-prettytable		0.7.2

Paket	AL1 Minimal	AL2023 Minimal
python3-prompt-toolkit		3.0,24
python3-pycparser		2,20
python3-pyrsistent		0,17,3
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pytz		2022.7.1
python3-pyyaml		5.4.1
python3-requests		2,25,1
python3-rpm		4.16.1,3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools		59,6,0
python3-setuptools-wheel		59,6,0
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1,25,10
python3-wcwidth		0,2,5

Paket	AL1 Minimal	AL2023 Minimal
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		38,1,45
selinux-policy-targeted		38,1,45
sendmail	8,14,4	
setserial	2,17	
setup	2.8,14	2.13,7
shadow-utils	4.1.4.2	4,9 bis 4,9

Paket	AL1 Minimal	AL2023 Minimal
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		252,23
systemd-libs		252,23
systemd-networkd		252,23
systemd-pam		252,23
systemd-resolved		252,23
systemd-udev		252,23
system-release	2018,03	2023,6.20241031
sysvinit	2,87	
tar	1,26	1,34
tcp_wrappers-libs	7.6	
tzdata	2023c	2024a
udev	173	
update-motd	1.0.1	2.2

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

Paket	AL1 Minimal	AL2023 Minimal
zstd		1.5.5

Vergleich von Paketen, die auf Amazon Linux 1 (AL1) und Amazon Linux 2023 Basiscontainer-Images installiert sind

Ein Vergleich der RPMs aktuellen Container-Images on the AL1 und AL2 023.

Paket	AL1 Behälter	AL2023 Behälter
alternatives		1.15
amazon-linux-repo-cdn		2023.6.20241031
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	2023,2,68
chkconfig	1.3.49,3	
coreutils	8,22	
coreutils-single		8,32
crypto-policies		20220428
curl	7,61,1	
curl-minimal		8,5,0
cyrus-sasl-lib	2.1.23	

Paket	AL1 Behälter	AL2023 Behälter
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
glibc-minimal-langpack		2,34
gmp	6.0.0	6.2.1
gnupg2	2.0.28	

Paket	AL1 Behälter	AL2023 Behälter
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	1.21.3
libacl	2,2,49	2.3.1
libarchive		3.7.4
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
libcurl	7,61,1	
libcurl-minimal		8,5,0
libdnf		0,69,0

Paket	AL1 Behälter	AL2023 Behälter
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
libsmartcols		2,37,4
libsolv		0,7.22
libssh2	1.4.2	

Paket	AL1 Behälter	AL2023 Behälter
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	
ncurses-base	5,7	6.2
ncurses-libs	5,7	6.2
npth		1,6

Paket	AL1 Behälter	AL2023 Behälter
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	
pkgconfig	0,27,1	
popt	1.13	1,18
pth	2.0.7	

Paket	AL1 Behälter	AL2023 Behälter
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
python3-libs		3,9,16
python3-pip-wheel		21.3.1
python3-rpm		4.16.1.3

Paket	AL1 Behälter	AL2023 Behälter
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	2023,6.20241031
tar	1,26	
tzdata	2023c	2024a
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	

Paket	AL1 Behälter	AL2023 Behälter
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 AL2 023 beschrieben.

Themen

- CPU-Anforderungen für die Ausführung von 023 AL2
- Speicheranforderungen (RAM) für die Ausführung von 023 AL2

CPU-Anforderungen für die Ausführung von 023 AL2

Um einen beliebigen AL2 023-Code ausführen zu können, muss der verwendete Prozessor bestimmte Mindestanforderungen erfüllen. Versuche, AL2 023 auf einem System auszuführen CPUs , das diese Anforderungen nicht erfüllt, können bereits zu Beginn der Codeausführung zu illegalen Befehlsfehlern führen.

Die Mindestanforderungen gelten für AL2023 bei Amazon EC2AL2023 in Behältern, undAL2023 außerhalb von Amazon EC2.

ARM-CPU-Anforderungen für AL2 023

Alle AL2 `aarch64` 023 (ARM) Binärdateien sind für 64-Bit gebaut. Keine 32-Bit-Version ARM Binärdateien sind verfügbar, also eine 64-Bit-Version ARM CPU ist erforderlich.

> **Note**
>
> Für ARM-basierte Instances unterstützt AL2 023 nur Instance-Typen, die Graviton2-Prozessoren oder neuere Prozessoren verwenden. AL2023 unterstützt keine A1-Instances.

AL2023 erfordert einen ARMv8 2.2-kompatiblen Prozessor mit der Cryptography Extension (). `ARMv8.2+crypto` Alle AL2 023-Pakete für `aarch64` werden mit dem Compiler-Flag erstellt. `-march=armv8.2-a+crypto` Obwohl wir versuchen, ordentliche Fehlermeldungen auszugeben, wenn versucht wird, AL2 023-Code auf einer älteren Version auszuführen ARM Bei Prozessoren ist es möglich, dass es sich bei der ersten Fehlermeldung um einen unzulässigen Befehlsfehler handelt.

> **Note**
>
> Aufgrund der `aarch64` CPU-Basisanforderungen von AL2 023 sind alle Raspberry Pi Systeme vor dem Raspberry Pi 5 erfüllen nicht die Mindestanforderungen an die CPU.

x86-64-CPU-Anforderungen für 023 AL2

Alle AL2 `x86-64` 023-Binärdateien wurden für die `x86-64v2` Überarbeitung der `x86-64` Architektur erstellt, indem sie an den Compiler übergeben werden. `-march=x86-64-v2`

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. Zu den Beispielen gehören Intel Nehalem, AMD Jaguar, Atom Silvermont, zusammen mit dem VIA Nano and Eden C Mikroarchitekturen.

In Amazon EC2 unterstützen alle `x86-64` Instance-Typen`x86-64v2`, einschließlich `M1C1`, und `M2` Instance-Familien.

Kein 32-Bit-x86 (i686) AL2 023 Binärdateien wurden erstellt. Obwohl AL2 023 weiterhin die Ausführung von 32-Bit-Userspace-Binärdateien unterstützt, ist diese Funktionalität 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 023 AL2

Die EC2 `.nano` Amazon-Instance-Typen (`t2.nano`, `t3.nanot3a.nano`, und`t4g.nano`) verfügen über 512 MB RAM, was die Mindestanforderung für AL2 023 ist.

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

AL2023 Images wurden nicht auf Systemen mit weniger als 512 MB RAM getestet. Die Ausführung von AL2 023-basierten Container-Images in weniger als 512 MB RAM hängt von der containerisierten Arbeitslast ab.

Für einige Workloads, z. B. `dnf upgrade` zwischen einigen AL2 023-Versionen, können mehr als 512 MB RAM erforderlich sein. Aus diesem Grund wurde in Version AL2023.3 die standardmäßige Aktivierung `zram` für Instanzen mit weniger als 800 MB RAM eingeführt. Für containerisierte Workloads bedeutet dies, dass einige Workloads auf AL2 023-Instances mit dieser Speichermenge möglicherweise problemlos ausgeführt werden, aber fehlschlagen, wenn sie in einem Container ausgeführt werden, der auf diese Speicherbelegung beschränkt ist.

Bei Instance-Typen mit weniger als 800 MB RAM aktiviert AL2 023 (ab Version AL2023.3 oder neuer) standardmäßig den basierten Swap. `zram` Beispiele für EC2 Amazon-Instance-Typen mit weniger als 800 MB Arbeitsspeicher sind `t4g.nanot3a.nano`,`t3.nano`,`t2.nano`, und`t1.micro`. Dies bedeutet weniger Szenarien mit unzureichendem Arbeitsspeicher für diese Instance-Typen, da AL2 023 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.

AL2023 Grafischer Desktop

Amazon Linux 2023 bietet ab Version 2023.7 eine optionale, schlanke, Cloud-optimierte grafische Oberfläche, die auf GNOME basiert. Diese moderne Desktop-Umgebung bietet erweiterte Produktivitätsfunktionen mit integrierten Tools wie Firefox für sicheres Surfen und bietet gleichzeitig Amazon DCV- und VNC-Unterstützung für den Fernzugriff.

Verwandte Themen

Weitere Informationen zur Installation der grafischen Desktop-Umgebung finden Sie in der folgenden Dokumentation:

- Tutorial: Installieren Sie die GNOME-Desktop-Umgebung auf 023 AL2

Anwendungen auf AL2 023 ausführen

Dieser Abschnitt behandelt Methoden zum Ausführen von Anwendungen auf Amazon Linux 2023 (AL2023), einschließlich der Verwaltung, wann sie gestartet (und neu gestartet) werden, und zur Steuerung der Ressourcennutzung.

Themen

- Begrenzung der Nutzung von Prozessressourcen in AL2 023 mithilfe von systemd
- Begrenzung der Nutzung von Prozessressourcen in AL2 023 mithilfe cgroups

Begrenzung der Nutzung von Prozessressourcen in AL2 023 mithilfe von systemd

Für Amazon Linux 2023 (AL2023) empfehlen wir die Verwendung, `systemd` um zu kontrollieren, welche Ressourcen von Prozessen oder Prozessgruppen verwendet werden können. Using `systemd` ist ein leistungsfähiger und benutzerfreundlicher Ersatz für `cgroups` manuelles Bearbeiten oder für die Verwendung von Hilfsprogrammen wie`cpulimit`, die bisher nur für Amazon Linux im EPEL Drittanbieter-Repository verfügbar waren.

Umfassende Informationen finden Sie in der `systemd` Upstream-Dokumentation für systemd.resource-control oder im man Seite für auf einer `systemd.resource-control` 023-Instanz. AL2

In den folgenden Beispielen wird der `stress-ng` CPU-Stresstest (aus dem `stress-ng` Paket) verwendet, um eine CPU-intensive Anwendung und eine speicherintensive Anwendung `memcached` zu simulieren.

Die folgenden Beispiele behandeln die Festlegung eines CPU-Limits für einen einmaligen Befehl und eines Speicherlimits für einen Dienst. Die meisten der angebotenen `systemd` Ressourcenbeschränkungen können an jedem Ort verwendet `systemd` werden, an dem ein Prozess ausgeführt wird, und es können mehrere gleichzeitig verwendet werden. Die folgenden Beispiele beschränken sich zur Veranschaulichung auf eine einzige Einschränkung.

Ressourcensteuerung mit **`systemd-run`** zum Ausführen einmaliger Befehle

Wird häufig mit Systemdiensten in Verbindung gebracht, `systemd` kann aber auch von Benutzern ohne Root-Rechte verwendet werden, um Dienste auszuführen, Timer zu planen oder einmalige Prozesse auszuführen. Im folgenden Beispiel verwenden wir es `stress-ng` als Beispielanwendung. Im ersten Beispiel werden wir es mit `systemd-run` dem `ec2-user` Standardkonto ausführen, und im zweiten Beispiel werden wir die CPU-Auslastung einschränken.

Example Verwenden Sie **`systemd-run`** es in der Befehlszeile, um einen Prozess auszuführen, ohne die Ressourcennutzung einzuschränken

1. Stellen Sie sicher, dass das `stress-ng` Paket installiert ist, da wir es für unser Beispiel verwenden werden.

   ```
   [ec2-user ~]$ sudo dnf install -y stress-ng
   ```

2. Wird verwendet`systemd-run`, um einen 10-Sekunden-CPU-Stresstest durchzuführen, ohne die CPU-Auslastung einzuschränken.

   ```
   [ec2-user ~]$ systemd-run --user --tty --wait --property=CPUAccounting=1 stress-ng
    --cpu 1 --timeout 10
   Running as unit: run-u6.service
   Press ^] three times within 1s to disconnect TTY.
   stress-ng: info:  [339368] setting to a 10 second run per stressor
   stress-ng: info:  [339368] dispatching hogs: 1 cpu
   stress-ng: info:  [339368] successful run completed in 10.00s
   Finished with result: success
   Main processes terminated with: code=exited/status=0
   Service runtime: 10.068s
   CPU time consumed: 9.060s
   ```

   Die `--user` Option weist `systemd-run` an, den Befehl als der Benutzer auszuführen, als der wir angemeldet sind. Die `--tty` Option bedeutet TTY ist angehängt, `--wait` bedeutet, dass gewartet wird, bis der Dienst beendet ist, und die `--property=CPUAccounting=1` Option weist an, `systemd-run` aufzuzeichnen, wie viel CPU-Zeit für die Ausführung des Prozesses verwendet wurde. Die `--property` Befehlszeilenoption kann verwendet werden, um `systemd-`

`run` Einstellungen zu übergeben, die in einer `systemd.unit` Konfigurationsdatei konfiguriert werden könnten.

Wenn Sie angewiesen werden, die CPU zu belasten, verwendet das `stress-ng` Programm die gesamte verfügbare CPU-Zeit, um seinen Test für die Dauer durchzuführen, für die Sie es ausführen möchten. Für eine reale Anwendung kann es wünschenswert sein, die Gesamtlaufzeit eines Prozesses zu begrenzen. Im folgenden Beispiel bitten wir darum, dass `stress-ng` die Ausführung länger dauert als die maximale Dauer, die wir für den Vorgang festlegen. `systemd-run`

Example Verwenden Sie **`systemd-run`** es in der Befehlszeile, um einen Prozess auszuführen, wobei die CPU-Auslastung auf 1 Sekunde begrenzt wird

1. Stellen Sie sicher, dass der installiert `stress-ng` ist, um dieses Beispiel auszuführen.
2. `LimitCPU`Diese Eigenschaft entspricht dem Wert`ulimit -t`, der die maximale Zeit auf der CPU begrenzt, die dieser Prozess verwenden darf. In diesem Fall empfängt der Befehl ein `SIGXCPU` Signal und schlägt fehl, da wir einen 10-sekündigen Stresslauf anfordern und die CPU-Auslastung auf 1 Sekunde begrenzen.

```
[ec2-user ~]$ systemd-run --user --tty --wait --property=CPUAccounting=1 --
property=LimitCPU=1 stress-ng --cpu 1 --timeout 10
Running as unit: run-u12.service
Press ^] three times within 1s to disconnect TTY.
stress-ng: info:  [340349] setting to a 10 second run per stressor
stress-ng: info:  [340349] dispatching hogs: 1 cpu
stress-ng: fail:  [340349] cpu instance 0 corrupted bogo-ops counter, 1370 vs 0
stress-ng: fail:  [340349] cpu instance 0 hash error in bogo-ops counter and run
 flag, 3250129726 vs 0
stress-ng: fail:  [340349] metrics-check: stressor metrics corrupted, data is
 compromised
stress-ng: info:  [340349] unsuccessful run completed in 1.14s
Finished with result: exit-code
Main processes terminated with: code=exited/status=2
Service runtime: 1.201s
CPU time consumed: 1.008s
```

In der Regel möchten Sie möglicherweise den Prozentsatz der CPU-Zeit einschränken, der von einem bestimmten Prozess verbraucht werden kann. Im folgenden Beispiel beschränken wir den Prozentsatz der CPU-Zeit, der verbraucht werden kann`stress-ng`. Für einen echten

Dienst kann es wünschenswert sein, den maximalen Prozentsatz an CPU-Zeit zu begrenzen, den ein Hintergrundprozess verbrauchen kann, um Ressourcen für den Prozess freizugeben, der Benutzeranfragen bearbeitet.

Example Wird verwendet**`systemd-run`**, um einen Prozess auf 10% der CPU-Zeit auf einer CPU zu beschränken

1. Stellen Sie sicher, dass der installiert `stress-ng` ist, um dieses Beispiel ausführen zu können.
2. Wir werden die `CPUQuota` Eigenschaft verwenden, um anzugeben, dass `systemd-run` die CPU-Auslastung für den Befehl, den wir ausführen werden, eingeschränkt werden soll. Wir begrenzen nicht die Zeit, für die der Prozess ausgeführt werden kann, sondern nur, wie viel CPU er verwenden kann.

   ```
   [ec2-user ~]$ systemd-run --user --tty --wait --property=CPUAccounting=1 --
   property=CPUQuota=10% stress-ng --cpu 1 --timeout 10
   Running as unit: run-u13.service
   Press ^] three times within 1s to disconnect TTY.
   stress-ng: info:  [340664] setting to a 10 second run per stressor
   stress-ng: info:  [340664] dispatching hogs: 1 cpu
   stress-ng: info:  [340664] successful run completed in 10.08s
   Finished with result: success
   Main processes terminated with: code=exited/status=0
   Service runtime: 10.140s
   CPU time consumed: 1.014s
   ```

   Beachten Sie, wie CPU Aus der Buchhaltung geht hervor, dass der Dienst zwar 10 Sekunden lang lief, aber nur 1 Sekunde der tatsächlichen CPU-Zeit beanspruchte.

Es gibt viele Konfigurationsmöglichkeiten, `systemd` um die Ressourcennutzung für CPU, Arbeitsspeicher, Netzwerk und I/O zu begrenzen. Weitere Informationen finden Sie in der `systemd` Upstream-Dokumentation für systemd.resource-control oder im man Eine umfassende Dokumentation finden Sie `systemd.resource-control` auf der Seite für eine AL2 023-Instanz.

Hinter den Kulissen werden Funktionen des Linux-Kernels verwendet`cgroups`, um beispielsweise diese Grenzwerte zu implementieren, ohne dass Sie sie manuell konfigurieren müssen. `systemd` Die Linux-Kernel-Dokumentation für `cgroup-v2` enthält ausführliche Informationen zur `cgroups` Arbeit.

Ressourcenkontrolle in einem **`systemd`** Dienst

Es gibt mehrere Parameter, die dem `[Service]` Abschnitt `systemd` Dienste hinzugefügt werden können, um die Nutzung der Systemressourcen zu steuern. Dazu gehören sowohl harte als auch weiche Grenzwerte. Das genaue Verhalten der einzelnen Optionen finden Sie in der `systemd` Upstream-Dokumentation für systemd.resource-control oder in man Seite für auf einer `systemd.resource-control` 023-Instanz. AL2

Häufig verwendete Grenzwerte sind `MemoryHigh` die Angabe einer Drosselungsgrenze für die Speichernutzung, `MemoryMax` die Festlegung einer festen Obergrenze (bei deren Erreichen der OOM-Killer aufgerufen wird) und `CPUQuota` (wie im vorherigen Abschnitt dargestellt). Es ist auch möglich, Gewichte und Prioritäten anstelle von festen Zahlen zu konfigurieren.

Example Wird verwendet**`systemd`**, um Speichernutzungslimits für Dienste festzulegen

In diesem Beispiel legen wir ein Limit für die Festplattennutzung für `memcached` einen einfachen Schlüsselwertcache fest und zeigen, wie der OOM-Killer für diesen Dienst und nicht für das gesamte System aufgerufen wird.

1. Zuerst müssen wir die für dieses Beispiel benötigten Pakete installieren.

```
[ec2-user ~]$ sudo dnf install -y memcached libmemcached-awesome-tools
```

2. Aktivieren Sie den `memcached.service` und starten Sie dann den Dienst, damit er `memcached` läuft.

```
[ec2-user ~]$ sudo systemctl enable memcached.service
Created symlink /etc/systemd/system/multi-user.target.wants/memcached.service # /
usr/lib/systemd/system/memcached.service.
[ec2-user ~]$ sudo systemctl start memcached.service
```

3. Prüfen Sie, ob das `memcached.service` läuft.

```
[ec2-user ~]$ sudo systemctl status memcached.service
# memcached.service - memcached daemon
     Loaded: loaded (/usr/lib/systemd/system/memcached.service; enabled; preset:
 disabled)
     Active: active (running) since Fri 2025-01-31 22:36:42 UTC; 1s ago
   Main PID: 356294 (memcached)
      Tasks: 10 (limit: 18907)
     Memory: 1.8M
```

```
        CPU: 20ms
     CGroup: /system.slice/memcached.service
     ##356294 /usr/bin/memcached -p 11211 -u memcached -m 64 -c 1024 -l
 127.0.0.1,::1

Jan 31 22:35:36 ip-1-2-3-4.us-west-2.compute.internal systemd[1]: Started
 memcached.service - memcached daemon.
```

4. Jetzt, wo es installiert `memcached` ist und läuft, können wir beobachten, dass es funktioniert, indem wir einige zufällige Daten in den Cache einfügen

   In `/etc/sysconfig/memcached` der `CACHESIZE` Variablen ist standardmäßig 64 gesetzt, was 64 Megabyte bedeutet. Wenn wir mehr Daten in den Cache einfügen als die maximale Cachegröße, können wir sehen, dass wir den Cache füllen und einige Elemente gelöscht werden`memcached-tool`, und dass der etwa 64 MB Speicher `memcached.service` beansprucht.

```
[ec2-user ~]$ for i in $(seq 1 150); do dd if=/dev/random of=$i bs=512k count=1;
 memcp -s localhost $i; done
[ec2-user ~]$ memcached-tool localhost display
  #  Item_Size  Max_age   Pages   Count   Full?  Evicted Evict_Time OOM
  2     120B         0s       1       0      no        0        0    0
 39   512.0K         4s      63     126     yes       24        2    0
[ec2-user ~]$ sudo systemctl status memcached.service
# memcached.service - memcached daemon
     Loaded: loaded (/usr/lib/systemd/system/memcached.service; enabled; preset:
 disabled)
     Active: active (running) since Fri 2025-01-31 22:36:42 UTC; 7min ago
   Main PID: 356294 (memcached)
      Tasks: 10 (limit: 18907)
     Memory: 66.7M
        CPU: 203ms
     CGroup: /system.slice/memcached.service
             ##356294 /usr/bin/memcached -p 11211 -u memcached -m 64 -c 1024 -l
 127.0.0.1,::1

Jan 31 22:36:42 ip-1-2-3-4.us-west-2.compute.internal systemd[1]: Started
 memcached.service - memcached daemon.
```

5. Verwenden Sie die `MemoryMax` Eigenschaft, um ein festes Limit dafür festzulegen, `memcached.service` wo der OOM-Killer aufgerufen wird, wenn er erreicht wird. Zusätzliche Optionen können für den Dienst festgelegt werden, indem sie zu einer Override-Datei hinzugefügt werden. Dies kann entweder durch direktes Bearbeiten der `/etc/systemd/system/memcached.service.d/override.conf` Datei oder interaktiv mit dem `edit` Befehl von `systemctl` geschehen.

```
[ec2-user ~]$ sudo systemctl edit memcached.service
```

Fügen Sie der Überschreibung Folgendes hinzu, um eine feste Speicherbegrenzung von 32 MB für den Dienst festzulegen.

```
[Service]
MemoryMax=32M
```

6. Sagen Sie`systemd`, dass die Konfiguration neu geladen werden soll

```
[ec2-user ~]$ sudo systemctl daemon-reload
```

7. Beachten Sie, `memcached.service` dass der jetzt mit einem Speicherlimit von 32 MB läuft.

```
[ec2-user ~]$ sudo systemctl status memcached.service
# memcached.service - memcached daemon
     Loaded: loaded (/usr/lib/systemd/system/memcached.service; enabled; preset:
 disabled)
    Drop-In: /etc/systemd/system/memcached.service.d
             ##override.conf
     Active: active (running) since Fri 2025-01-31 23:09:13 UTC; 49s ago
   Main PID: 358423 (memcached)
      Tasks: 10 (limit: 18907)
     Memory: 1.8M (max: 32.0M available: 30.1M)
        CPU: 25ms
     CGroup: /system.slice/memcached.service
             ##358423 /usr/bin/memcached -p 11211 -u memcached -m 64 -c 1024 -l
 127.0.0.1,::1

Jan 31 23:09:13 ip-1-2-3-4.us-west-2.compute.internal systemd[1]: Started
 memcached.service - memcached daemon.
```

8. Der Dienst funktioniert normal, wenn er weniger als 32 MB Speicher verwendet. Dies können wir überprüfen, indem wir weniger als 32 MB zufälliger Daten in den Cache laden und dann den Status des Dienstes überprüfen.

```
[ec2-user ~]$ for i in $(seq 1 30); do dd if=/dev/random of=$i bs=512k count=1;
 memcp -s localhost $i; done
```

```
[ec2-user ~]$ sudo systemctl status memcached.service
# memcached.service - memcached daemon
     Loaded: loaded (/usr/lib/systemd/system/memcached.service; enabled; preset:
 disabled)
    Drop-In: /etc/systemd/system/memcached.service.d
             ##override.conf
     Active: active (running) since Fri 2025-01-31 23:14:48 UTC; 3s ago
   Main PID: 359492 (memcached)
      Tasks: 10 (limit: 18907)
     Memory: 18.2M (max: 32.0M available: 13.7M)
        CPU: 42ms
     CGroup: /system.slice/memcached.service
             ##359492 /usr/bin/memcached -p 11211 -u memcached -m 64 -c 1024 -l
 127.0.0.1,::1

Jan 31 23:14:48 ip-1-2-3-4.us-west-2.compute.internal systemd[1]: Started
 memcached.service - memcached daemon.
```

9. Wir können jetzt mehr als 32 MB Speicher verwenden, indem wir versuchen, die vollen 64 MB Cache zu verwenden, die in der Standardkonfiguration vorgesehen sind. `memcached` `memcached`

```
[ec2-user ~]$ for i in $(seq 1 150); do dd if=/dev/random of=$i bs=512k count=1;
 memcp -s localhost $i; done
```

Sie werden feststellen, dass irgendwann während des obigen Befehls Verbindungsfehler zum `memcached` Server auftreten. Dies liegt daran, dass der OOM-Killer den Prozess aufgrund der Einschränkung, die wir ihm auferlegt haben, beendet hat. Der Rest des Systems wird normal funktionieren, und der OOM-Killer berücksichtigt keine anderen Prozesse, da wir nur `memcached.service` diese eingeschränkt haben.

```
[ec2-user ~]$ sudo systemctl status memcached.service
# memcached.service - memcached daemon
```

```
     Loaded: loaded (/usr/lib/systemd/system/memcached.service; enabled; preset:
 disabled)
    Drop-In: /etc/systemd/system/memcached.service.d
             ##override.conf
     Active: failed (Result: oom-kill) since Fri 2025-01-31 23:20:28 UTC; 2s ago
   Duration: 2.901s
    Process: 360130 ExecStart=/usr/bin/memcached -p ${PORT} -u ${USER} -m
 ${CACHESIZE} -c ${MAXCONN} $OPTIONS (code=killed, signal=KILL)
   Main PID: 360130 (code=killed, signal=KILL)
        CPU: 94ms

Jan 31 23:20:25 ip-1-2-3-4.us-west-2.compute.internal systemd[1]: Started
 memcached.service - memcached daemon.
Jan 31 23:20:28 ip-1-2-3-4.us-west-2.compute.internal systemd[1]:
 memcached.service: A process of this unit has been killed by the OOM killer.
Jan 31 23:20:28 ip-1-2-3-4.us-west-2.compute.internal systemd[1]:
 memcached.service: Main process exited, code=killed, status=9/KILL
Jan 31 23:20:28 ip-1-2-3-4.us-west-2.compute.internal systemd[1]:
 memcached.service: Failed with result 'oom-kill'.
```

Begrenzung der Nutzung von Prozessressourcen in AL2 023 mithilfe cgroups

Die Verwendung wird zwar empfohlenRessourcenkontrolle mit systemd, in diesem Abschnitt wird jedoch die grundlegende Verwendung der `libcgroup-tools` Basisdienstprogramme behandelt, um die CPU- und Speicherauslastung von Prozessen zu begrenzen. Beide Methoden sind Alternativen zur Verwendung des `cpulimit` Hilfsprogramms, das zuvor unter beschrieben wurdeEPEL.

Das folgende Beispiel behandelt die Ausführung des `stress-ng` Stresstests (aus dem `stress-ng` Paket) bei gleichzeitiger Begrenzung der CPU- und Speicherauslastung mithilfe der Dienstprogramme aus dem `libcgroup-tools` Paket und den darin enthaltenen Tunables. `sysfs`

Verwenden Sie es **`libcgroup-tools`** in der Befehlszeile, um die Ressourcennutzung zu begrenzen

1. Installieren Sie das Paket `libcgroup-tools`.

```
[ec2-user ~]$ sudo dnf install libcgroup-tools
```

2. Erstellen Sie eine cgroup mit den cpu Controllern memory und und geben Sie ihr einen Namen (our-example-limits). Verwenden Sie die -t Optionen -a und, um es dem ec2-user Benutzer zu ermöglichen, die Tunables des cgroup

```
[ec2-user ~]$ sudo cgcreate -a ec2-user -t ec2-user -g memory,cpu:our-example-limits
```

Es gibt jetzt ein /sys/fs/cgroup/our-example-limits/ Verzeichnis, das Dateien enthält, die zur Steuerung der einzelnen Tunables verwendet werden können.

> **Note**
>
> Amazon Linux 2 verwendet cgroup-v1 Rathercgroup-v2, das auf AL2 023 verwendet wird. Auf AL2, die sysfs Pfade sind unterschiedlich, und es wird /sys/fs/cgroup/memory/our-example-limits /sys/fs/cgroup/cpu/our-example-limits Verzeichnisse gebenec2-user, deren Eigentümer Dateien sind, mit denen die Grenzen von kontrolliert werden können. cgroup

3. Beschränken Sie den Speicherverbrauch aller Prozesse in unserer cgroup Datenbank auf 100 Millionen Byte.

```
[ec2-user ~]$ echo 100000000 > /sys/fs/cgroup/our-example-limits/memory.max
```

> **Note**
>
> Amazon Linux 2 verwendet cgroup-v1 eher als cgroup-v2 das, was Amazon Linux 2023 verwendet. Das bedeutet, dass einige Tunables unterschiedlich sind. Um die Speichernutzung zu begrenzen AL2, wird stattdessen das unten stehende Tunable verwendet.
>
> ```
> [ec2-user ~]$ echo 10000000 > /sys/fs/cgroup/memory/our-example-limits/memory.limit_in_bytes
> ```

4. Beschränken Sie die CPU-Auslastung aller Prozesse in unserer Datenbank cgroup auf 10%. Das Format der cpu.max Datei beschränkt $MAX $PERIOD die Gruppe auf die Nutzung $MAX für jeden$PERIOD.

```
[ec2-user ~]$ echo 10000 100000 > /sys/fs/cgroup/our-example-limits/cpu.max
```

Amazon Linux 2 verwendet `cgroup-v1` eher als `cgroup-v2` das, was Amazon Linux 2023 verwendet. Das bedeutet, dass einige Tunables unterschiedlich sind, einschließlich der Frage, wie die CPU-Auslastung begrenzt werden kann.

5. Das folgende Beispiel läuft `stress-ng` (das durch Ausführen installiert werden kann`dnf install -y stress-ng`) im `our-example-limits` cgroup. Während der Ausführung des `stress-ng` Befehls können Sie beobachten, `top` dass er auf 10% von begrenzt ist CPU Zeit.

```
[ec2-user ~]$ sudo cgexec -g memory,cpu:our-example-limits stress-ng --cpu 1
```

6. Räumen Sie auf, indem Sie die Cgroup entfernen

```
[ec2-user ~]$ sudo cgdelete -g memory,cpu:our-example-limits
```

Die Linux-Kernel-Dokumentation für `cgroup-v2` enthält ausführliche Informationen darüber, wie sie funktionieren. In der Dokumentation der CPU - und Speichercontroller wird detailliert beschrieben, wie die einzelnen einstellbaren Optionen verwendet werden.

Verwenden Sie AL2 023 auf AWS

Sie können AL2 023 für die Verwendung mit anderen einrichten. AWS-Services Sie können beispielsweise ein AL2 023-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 bei Amazon EC2
- AL2023 in Containern verwenden
- AL2023 auf AWS Elastic Beanstalk
- Mit AL2 0,23 in AWS CloudShell
- Verwendung von AL2 023-basiertem Amazon ECS AMIs zum Hosten von containerisierten Workloads
- Verwenden von Amazon Elastic File System auf AL2 023
- Verwenden von Amazon EMR, das auf 023 basiert AL2
- Mit AL2 0,23 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/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen 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. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu https://aws.amazon.com/gehst und Mein Konto auswählst.

Erstellen eines Benutzers 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 eines Benutzers 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 Administratorbenutzer im IAM Identity Center Benutzerzugriff.

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

Anmelden als Administratorbenutzer

- 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 Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

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

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center Benutzerhandbuch.

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 das AWS CLI AWS SDKs, oder zu signieren. AWS APIs	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen zu den AWS CLI finden Sie unter Konfiguration der AWS CLI zur Verwendung AWS IAM Identity Center im AWS Command Line Interface Benutzerhandbuch. • Informationen zu AWS SDKs Tools und AWS APIs finden Sie unter IAM Identity Center-Authentifizierung im Referenzhandbuch AWS SDKs und im Tools-Ref erenzhandbuch.
IAM	Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an das AWS CLI AWS SDKs, oder zu signieren. AWS APIs	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 das AWS CLI AWS SDKs, oder zu signieren. AWS APIs	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen dazu AWS CLI finden Sie unter Authentifizierung mithilfe von IAM-Benutzeranmeld einformationen im AWS

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

AL2023 bei Amazon EC2

Verwenden Sie eines der folgenden Verfahren, um eine EC2 Amazon-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 AL2 023 Standard (Standard) und Minimal AMIs.

Themen

- AL2023 über die EC2 Amazon-Konsole starten
- AL2023 mit dem SSM-Parameter starten und AWS CLI
- Starten des neuesten AL2 023-AMI mit AWS CloudFormation
- AL2023 mit einer bestimmten AMI-ID starten
- AL2023 AMI-Verfall und Lebenszyklus
- Verbindung zu 023 Instances herstellen AL2
- Vergleich von AL2 023 Standard und Minimal AMIs

AL2023 über die EC2 Amazon-Konsole starten

Verwenden Sie die EC2 Amazon-Konsole, um ein AL2 023-AMI zu starten.

> **Note**
>
> Für ARM-basierte Instances unterstützt AL2 023 nur Instance-Typen, die Graviton2-Prozessoren oder neuere Prozessoren verwenden. AL2023 unterstützt keine A1-Instances.

Gehen Sie wie folgt vor, um eine EC2 Amazon-Instance mit einem AL2 023-AMI von der EC2 Amazon-Konsole aus zu starten.

So starten Sie eine EC2 Instance mit einem AL2 023-AMI

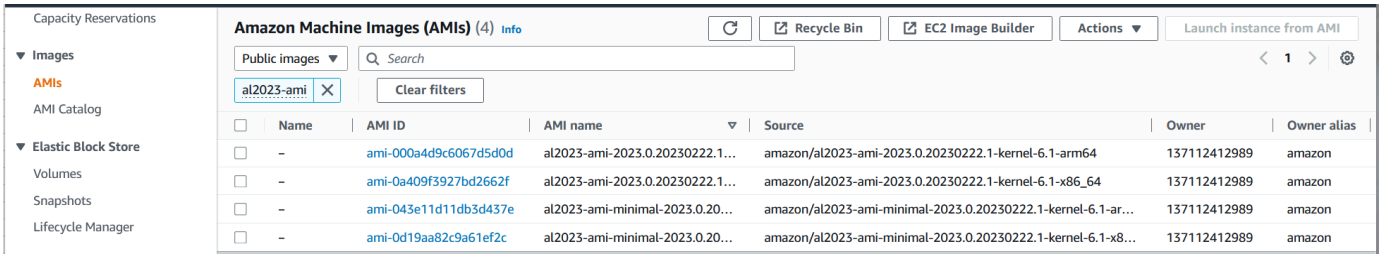
1. Öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich AMIs aus.
3. Wählen Sie im Dropdown-Menü Öffentliche Abbilder aus.
4. Geben Sie im Suchfeld **`al2023-ami`** ein.

   > **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 AL2 023-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. Die folgende Abbildung zeigt eine unvollständige Liste von AL2 AMIs 023.



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

AL2023 mit dem SSM-Parameter starten und AWS CLI

In der AWS CLI können Sie den SSM-Parameterwert eines AMI verwenden, um eine neue Instanz von 023 zu starten. AL2 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**
>
> Jedes der *`italic`* Elemente 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 Instanz 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 im AWS Command Line Interface Benutzerhandbuch unter Starten, Auflisten und Beenden von EC2 Amazon-Instances.

Starten des neuesten AL2 023-AMI mit AWS CloudFormation

Verwenden Sie eine der folgenden Vorlagen AWS CloudFormation, um ein AL2 023-AMI mit zu starten.

> **Note**
>
> Sie x86_64 und sie erfordern Arm64 AMIs jeweils unterschiedliche Instance-Typen. Weitere Informationen finden Sie unter EC2 Amazon-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

Sie können ein bestimmtes AL2 023-AMI mit der AMI-ID starten. Sie können anhand der AMI-Liste in der EC2 Amazon-Konsole ermitteln, welche AL2 023-AMI-ID benötigt wird. 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 des neuesten Amazon Linux AMI IDs mithilfe von AWS Systems Manager Parameter Store.

AL2023 AMI-Verfall und Lebenszyklus

Jede neue Version AL2 023 enthält ein neues AMI. Wenn das AMI registriert ist, ist es mit einem Beendigungsdatum gekennzeichnet. Das Verfallsdatum für jedes AL2 023-AMI liegt 90 Tage ab dem Zeitpunkt, an dem es veröffentlicht wurde, und entspricht somit dem Zeitraum, der für jede einzelne Kernel Live Patching auf 023 AL2 Kernel-Version angeboten wird.

> **Note**
>
> Das 90-Tage-Verfallsdatum bezieht sich auf ein einzelnes AMI und nicht auf den Zeitraum AL2 023 Release-Taktfrequenz oder den Produktsupport-Zeitraum.

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

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:

- AL2023 mit dem SSM-Parameter starten und AWS CLI
- Starten des neuesten AL2 023-AMI mit AWS CloudFormation

Verbindung zu 023 Instances herstellen AL2

Verwenden Sie SSH oder, AWS Systems Manager um eine Verbindung zu Ihrer AL2 023-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 EC2 Amazon-Benutzerhandbuch.

Connect zu Ihrer Instance her mit AWS Systems Manager

Anweisungen zum Herstellen einer Connect einer AL2 023-Instance finden Sie AWS Systems Manager unter Herstellen einer Verbindung zu Ihrer Linux-Instance mithilfe von Session Manager im EC2 Amazon-Benutzerhandbuch.

Verwenden von Amazon EC2 Instance Connect

Beim AL2 023 AMI, mit Ausnahme des Minimal-AMI, ist der EC2 Instance Connect-Agent standardmäßig installiert. Um EC2 Instance Connect mit einer AL2 023-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 Instance Connect eine Verbindung zu Ihrer EC2 Linux-Instance Connect im EC2 Amazon-Benutzerhandbuch.

Vergleich von AL2 023 Standard und Minimal AMIs

Sie können eine EC2 Amazon-Instance entweder mit einem Standard- (Standard) oder einem minimalen AL2 023-AMI starten. Anweisungen zum Starten einer EC2 Amazon-Instance mit dem Standard- oder Minimal-AMI-Typ finden Sie unterAL2023 bei Amazon EC2.

Im Standard AL2 023 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 AL2 023 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 entspricht eher dem minimalen AMI AL2 023 im Paketsatz.

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

Ein Vergleich der RPMs aktuellen Images auf dem AL2 023 AMI, Minimal AMI und Container.

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`	2.5.1	2.5.1	
`amazon-linux-repo-cdn`			2023,6.20241031
`amazon-linux-repo-s3`	2023,6.20241031	2023,6.20241031	
`amazon-linux-sb-keys`	2023,1	2023,1	
`amazon-rpm-config`	228		
`amazon-ssm-agent`	3,3,987,0		
`amd-ucode-firmware`	20210208 (Noarch)	20210208 (Noarch)	

Paket	AMI	Minimal-AMI	Container
at	3.1.23		
attr	2.5.1		
audit	3.0.6	3.0.6	
audit-libs	3.0.6	3.0.6	3.0.6
aws-cfn-bootstrap	2.0		
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	9,18,28		
bind-license	9,18,28		
bind-utils	9,18,28		
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

Paket	AMI	Minimal-AMI	Container
ca-certificates	2023,2,68	2023,2,68	2023,2,68
c-ares	1.19.1		
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		

Paket	AMI	Minimal-AMI	Container
cryptsetup	2.6.1		
cryptsetup-libs	2.6.1	2.6.1	
curl-minimal	8,5,0	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27	
cyrus-sasl-plain	2.1.27		
dbus	1.12,28	1.12.28	
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	

Paket	AMI	Minimal-AMI	Container
dnf-utils	4.3.0		
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	

Paket	AMI	Minimal-AMI	Container
efivar-libs	38	38	
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	

Paket	AMI	Minimal-AMI	Container
gettext	0,21	0,21	
gettext-libs	0,21	0,21	
ghc-srpm-macros	1.5.0		
glib2	2,74,7	2,74,7	2,74,7
glibc	2,34	2,34	2,34
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	

Paket	AMI	Minimal-AMI	Container
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-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	0,384	0,384	
info	6.7		

Paket	AMI	Minimal-AMI	Container
inih	49	49	
initscripts	10,09	10,09	
iproute	6.10.0	6.10.0	
iputils	20210202	20210202	
irqbalance	1.9.0	1.9.0	
jansson	2.14	2.14	
jemalloc	5.2.1		
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	6.1,112	6.1.112	
kernel-libbpf	6.1.112	6.1.112	
kernel-livepatch-repo-s3	2023,6,20241031	2023,6.20241031	
kernel-srpm-macros	1,0		
kernel-tools	6.1,112		
keyutils	1.6.3		

Paket	AMI	Minimal-AMI	Container
keyutils-libs	1.6.3	1.6.3	1.6.3
kmod	29	29	
kmod-libs	29	29	
kpatch-runtime	0.9.7		
krb5-libs	1.21.3	1.21.3	1.21.3
less	608	608	
libacl	2.3.1	2.3.1	2.3.1
libaio	0,3,111		
libarchive	3.7.4	3.7.4	3.7.4
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

Paket	AMI	Minimal-AMI	Container
libconfig	1.7.2		
libcurl-minimal	8.5.0	8.5.0	8.5.0
libdb	5.3.28	5.3.28	
libdhash	0.5.0		
libdnf	0,69,0	0,69,0	0,69,0
libeconf	0,4,0	0,4,0	
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	

Paket	AMI	Minimal-AMI	Container
libkcapi-hmaccalc	1.4.0	1.4.0	
libldb	2.6.2		
libmaxminddb	1.5.2		
libmetalink	0.1.3		
libmnl	1.0.4	1.0.4	
libmodulemd	2.13.0	2.13.0	2.13.0
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

Paket	AMI	Minimal-AMI	Container
libseccomp	2.5.3	2.5.3	
libselinux	3.4	3.4	3.4
libselinux-utils	3.4	3.4	
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		

Paket	AMI	Minimal-AMI	Container
libtextstyle	0,21	0,21	
libtirpc	1.3.3		
libunistring	0,9,10	0.9,10	0.9,10
libuser	0,63	0,63	
libutempter	1.2.1	1.2.1	
libuuid	2,37,4	2,37,4	2,37,4
libuv	1.47.0		
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
linux-firmware-whence	20210208 (Noarch)	20210208 (Noarch)	
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		

Paket	AMI	Minimal-AMI	Container
lz4-libs	1.9.4	1.9.4	1.9.4
man-db	2.9.3	2.9.3	
man-pages	5,10		
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)	
mpfr	4.1.0	4.1.0	4.1.0
nano	5,8		
ncurses	6.2	6.2	
ncurses-base	6.2	6.2	6.2
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		

Paket	AMI	Minimal-AMI	Container
ntsysv	1.15		
numactl-libs	2.0.14	2.0.14	
ocaml-srpm-macros	6		
oniguruma	6.9.7.1	6.9.7.1	
openblas-srpm-macros	2		
openldap	2.4.57	2,4,57	
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		

Paket	AMI	Minimal-AMI	Container
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
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		

Paket	AMI	Minimal-AMI	Container
perl-Getopt-Std	1.12		
perl-HTTP-Tiny	0,078		
perl-if	0,60,800		
perl-interpreter	5,32,1		
perl-IO	1,43		
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		

Paket	AMI	Minimal-AMI	Container
perl-Pod-Usage	2,01		
perl-POSIX	1,94		
perl-Scalar-List-Utils	1,56		
perl-SelectSaver	1.02		
perl-Socket	2,032		
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		

Paket	AMI	Minimal-AMI	Container
pkgconf-pkg-config	1.8.0		
policycoreutils	3.4	3.4	
policycoreutils-python-utils	3.4		
popt	1,18	1,18	1,18
procps-ng	3.3,17	3.3.17	
protobuf-c	1.4.1		
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	

Paket	AMI	Minimal-AMI	Container
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	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	

Paket	AMI	Minimal-AMI	Container
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
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	

Paket	AMI	Minimal-AMI	Container
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	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	

Paket	AMI	Minimal-AMI	Container
python3-setuptools	59,6,0	59,6,0	
python3-setuptools-wheel	59,6,0	59,6,0	59,6,0
python3-six	1.15.0	1.15.0	
python3-systemd	235	235	
python3-urllib3	1,25,10	1,25,10	
python3-wcwidth	0,2,5	0,2,5	
python-chevron	0.13.1		
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

Paket	AMI	Minimal-AMI	Container
rpm-plugin-selinux	4.16.1.3	4.16.1.3	
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3	
rpm-sign-libs	4.16.1.3	4.16.1.3	4.16.1.3
rsync	3.2.6		
rust-srpm-macros	21		
sbsigntools	0.9.4	0.9.4	
screen	4.8.0		
sed	4.8	4,8	4,8
selinux-policy	38,1,45	38,1,45	
selinux-policy-targeted	38,1,45	38,1,45	
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		

Paket	AMI	Minimal-AMI	Container
strace	6.8		
sudo	1.9,15	1.9,15	
sysctl-defaults	1,0	1,0	
sysstat	12,5.6		
systemd	252,23	252,23	
systemd-libs	252,23	252,23	
systemd-networkd	252,23	252,23	
systemd-pam	252,23	252,23	
systemd-resolved	252,23	252,23	
systemd-udev	252,23	252,23	
system-release	2023,6,20241031	2023,6.20241031	2023,6.20241031
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		

Paket	AMI	Minimal-AMI	Container
tzdata	2024a	2024a	2024a
unzip	6.0		
update-motd	2.2	2.2	
userspace-rcu	0.12.1	0.12.1	
util-linux	2,37,4	2,37,4	
util-linux-core	2,37,4	2,37,4	
vim-common	9,0,2153		
vim-data	9,0,2153	9,0,2153	
vim-enhanced	9,0,2153		
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

Paket	AMI	Minimal-AMI	Container
`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	

AL2023 in Containern verwenden

> **Note**
>
> Weitere Informationen zur Verwendung von AL2 023 zum Hosten von containerisierten Workloads auf Amazon ECS finden Sie unter. AL2023 für Amazon ECS-Container-Hosts

Je nach Anwendungsfall gibt es mehrere Möglichkeiten, wie AL2 023 in Containern verwendet werden kann. Das AL2023 Basis-Container-Image ist einem Amazon Linux 2-Container-Image und dem Minimal-AMI AL2 023 am ähnlichsten.

Für fortgeschrittene Benutzer bieten wir ein minimales Container-Image an, das in der Version AL2 023.2 eingeführt wurde, zusammen mit einer Dokumentation, in der beschrieben wird, wie Bare-Bone-Container erstellt werden.

AL2023 kann auch zum Hosten von containerisierten Workloads verwendet werden, entweder von AL2 023-basierten Container-Images oder von Containern, die auf anderen Linux-Distributionen basieren. Hierzu können Sie AL2023 für Amazon ECS-Container-Hosts verwenden oder die enthaltenen Container-Runtime-Pakete direkt verwenden. Die `nerdctl` Pakete `dockercontainerd`, und können auf 023 installiert und verwendet werden. AL2

Themen

- Verwenden Sie das AL2 023-Basiscontainer-Image
- AL2023 Minimales Container-Image
- Erstellung von Bare-Bones AL2 (023) 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 Sie das AL2 023-Basiscontainer-Image

Das AL2 023-Container-Image besteht aus denselben Softwarekomponenten, die im AL2 023-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 AL2 023 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 AL2 Docker 023-Container-Image abrufen, können Sie die Tags in einem der folgenden Formate verwenden:
>
> - Verwenden Sie das Tag, um die neueste Version des AL2 023-Container-Images zu erhalten. `:2023`
> - Um eine bestimmte Version von AL2 023 zu erhalten, können Sie das folgende Format verwenden:
>   - `:2023.`*`[0-7 release quarter]`*`.`*`[release date]`*`.`*`[build number]`*
>
> In den folgenden Beispielen wird das Tag verwendet `:2023` und das neueste verfügbare Container-Image von AL2 023 abgerufen.

```
$ 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
```

Um das AL2 023-Container-Image von Hub abzurufen Docker

1. Rufen Sie das AL2 023-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**
>
> Das Container-Image von AL2 023 verwendet nur den `dnf` Paketmanager zur Installation von Softwarepaketen. 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 AL2 023-Container-Images sind für die meisten Anwendungsfälle geeignet, und die Anpassung an das Minimal-Container-Image ist wahrscheinlich aufwändiger als die Anpassung an das AL2 023-Basiscontainer-Image.

Das in AL2 023.2 eingeführte Minimal-Container-Image AL2 023 unterscheidet sich vom Basis-Container-Image dadurch, dass es nur die Mindestmenge an Paketen enthält, die für die Installation anderer Pakete erforderlich sind. Das Minimal-Container-Image ist so konzipiert, dass es ein minimaler Satz von Paketen ist, kein praktischer Satz von Paketen.

Das Minimal-Container-Image AL2 023 basiert auf Softwarekomponenten, die bereits in AL2 023 verfügbar sind. Der wesentliche Unterschied beim Minimal-Container-Image besteht darin, den `dnf` Paketmanager zur Verfügung `microdnf` zu stellen, und nicht das Image mit vollem FunktionsumfangPython. `dnf` Dadurch kann das minimale Container-Image kleiner sein, mit dem Nachteil, dass nicht der gesamte Funktionsumfang des `dnf` Paketmanagers, der in der Version AL2 023 AMIs und im Basis-Container-Image enthalten ist, zur Verfügung steht.

Das minimale Container-Image AL2 023 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 AL2 023 weniger Pakete enthält als das AL2 023-Basis-Container-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 (1AL1)	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	

Verwenden des 023 Minimal Container-Images AL2

Das Minimal-Container-Image AL2 023 ist auf verfügbar ECR und das `2023-minimal` Tag zeigt immer auf das neueste AL2 023-basierte Minimal-Container-Image, obwohl das `minimal` Tag auf eine neuere Version von Amazon Linux als AL2 023 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 von Bare-Bones AL2 (023) Container-Images

Das AL2 023-Container-Image besteht aus denselben Softwarekomponenten, die im AL2 023-AMI enthalten sind. Es enthält eine Software, die es der Basis-Container-Ebene ermöglicht, sich ähnlich zu verhalten wie die Ausführung auf einer EC2 Amazon-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 standardmäßigen AL2 023-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. Es wird ein AL2 023-Container mit dem Namen gestartet. `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.

     Wenn wir die `--installroot` Option auf diese `dnf` Weise verwenden, erstellen wir die anderen AL2 023-Bilder. 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. 2023.8.20250721. Dies würde bedeuten, dass der `build` Container als

neueste Version 023 ausgeführt wird, der Barebones-Container jedoch unabhängig von der aktuellen Version AL2 023 ab 2023.8.20250721 erstellt wird. AL2

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 des RPMs aktuellen Container-Images auf dem AL2 023-Basis-Container-Image mit dem RPMs aktuellen auf dem AL2 023-Minimal-Container-Image.

Paket	Container	Minimaler Container
alternatives	1.15	1.15
amazon-linux-repo-cdn	2023.6.20241031	2023,6.20241031
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	2023,2,68	2023,2,68
coreutils-single	8,32	8,32
crypto-policies	20220428	20220428

Paket	Container	Minimaler Container
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	
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

Paket	Container	Minimaler Container
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	1,21,3	1.21.3
libacl	2.3.1	2.3.1
libarchive	3.7.4	3.7.4
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	

Paket	Container	Minimaler Container
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
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	

Paket	Container	Minimaler Container
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		3.10.0
microdnf-dnf		3.10.0
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	

Paket	Container	Minimaler Container
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	
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	2023,6,20241031	2023,6.20241031
tzdata	2024a	
xz-libs	5.2.5	5.2.5
yum	4.14.0	

Paket	Container	Minimaler Container
zlib	1.2.11	1.2.11

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

Ein Vergleich der RPMs Gegenwart auf dem AL2 023 Minimial AMI mit der RPMs Gegenwart auf der AL2 023 Basis- und 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	2.5.1		
amazon-linux-repo-cdn		2023.6.20241031	2023,6.20241031
amazon-linux-repo-s3	2023,6.20241031		
amazon-linux-sb-keys	2023,1		
amd-ucode-firmware	20210208 (Noarch)		
audit	3.0.6		
audit-libs	3.0.6	3.0.6	3.0.6
awscli-2	2,15,30		

Paket	Minimal-AMI	Container	Minimaler Container
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	2023,2,68	2023,2,68	2023,2,68
checkpolicy	3.4		
chrony	4.3		
cloud-init	22,2,2		
cloud-init-cfg-ec2	22.2.2		
cloud-utils-growpart	0,31		
coreutils	8,32		
coreutils-common	8,32		
coreutils-single		8,32	8,32
cpio	2,13		
cracklib	2.9.6		
cracklib-dicts	2.9.6		
crypto-policies	20220428	20220428	20220428
cryptsetup-libs	2.6.1		
curl-minimal	8,5,0	8.5.0	8.5.0

Paket	Minimal-AMI	Container	Minimaler Container
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		
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		

Paket	Minimal-AMI	Container	Minimaler Container
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	
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		

Paket	Minimal-AMI	Container	Minimaler Container
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		
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)		

Paket	Minimal-AMI	Container	Minimaler Container
grub2-pc-modules	2,06		
grub2-tools	2,06		
grub2-tools-minimal	2,06		
grubby	8,40		
gzip	1.12		
hostname	3,23		
hwdata	0,384		
inih	49		
initscripts	10,09		
iproute	6.10.0		
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	6.1,112		

Paket	Minimal-AMI	Container	Minimaler Container
kernel-libbpf	6.1.112		
kernel-livepatch-repo-s3	2023,6,20241031		
keyutils-libs	1.6.3	1.6.3	1.6.3
kmod	29		
kmod-libs	29		
krb5-libs	1,21,3	1.21.3	1.21.3
less	608		
libacl	2.3.1	2.3.1	2.3.1
libarchive	3,7.4	3.7.4	3.7.4
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

Paket	Minimal-AMI	Container	Minimaler Container
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
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		

Paket	Minimal-AMI	Container	Minimaler Container
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
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		

Paket	Minimal-AMI	Container	Minimaler Container
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
linux-firmware-whence	20210208 (Noarch)		
logrotate	3.20.1		
lua-libs	5.4.4	5.4.4	5.4.4
lz4-libs	1.9.4	1.9.4	1.9.4
man-db	2.9.3		
microcode_ctl	2.1 (x86_64)		
microdnf			3.10.0
microdnf-dnf			3.10.0
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		

Paket	Minimal-AMI	Container	Minimaler Container
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		
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		

Paket	Minimal-AMI	Container	Minimaler Container
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		
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	Minimal-AMI	Container	Minimaler Container
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		
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		

Paket	Minimal-AMI	Container	Minimaler Container
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		
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		

Paket	Minimal-AMI	Container	Minimaler Container
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		
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	

Paket	Minimal-AMI	Container	Minimaler Container
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
selinux-policy	38,1,45		
selinux-policy-targeted	38,1,45		
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	252,23		
systemd-libs	252,23		
systemd-networkd	252,23		
systemd-pam	252,23		

Paket	Minimal-AMI	Container	Minimaler Container
systemd-resolved	252,23		
systemd-udev	252,23		
system-release	2023,6,20241031	2023,6.20241031	2023,6.20241031
tar	1,34		
tzdata	2024a	2024a	
update-motd	2.2		
userspace-rcu	0.12.1		
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		

Paket	Minimal-AMI	Container	Minimaler Container
`zstd`	1.5.5		

AL2023 auf 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 EC2 Amazon-Instances, und auf diesen Instances wird Amazon Linux ausgeführt. 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 arbeitet daran, die Unterstützung für AL2 023 für weitere Elastic Beanstalk-Plattformen freizugeben.

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/WelcomeDie vollständige Liste der Elastic Beanstalk-Plattformunterstützung und der aktuellen Plattformen, die auf Version AL2 023 aufbauen, finden Sie im Elastic Beanstalk Developer Guide im Abschnitt Elastic Beanstalk Linux-Plattformen.

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

Mit AL2 0,23 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 AL2 023 migriert. Die Migration auf AL2 023 wird AWS-Regionen ab dem 4. Dezember 2023 insgesamt eingeführt. Weitere Informationen zur CloudShell Migration zu AL2 023 finden Sie unter AWS CloudShell Migration von Amazon Linux 2 zu Amazon Linux 2023.

Verwendung von AL2 023-basiertem Amazon ECS AMIs zum Hosten von containerisierten Workloads

> **Note**
>
> Weitere Informationen zur Verwendung von AL2 023 in einem Container finden Sie unter. AL2023 in Behältern

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.

Mit dem 023-basierten Amazon ECS-optimierten AMI können Sie containerisierte Workloads auf AL2 AL2 023 hosten. Weitere Informationen finden Sie unter Amazon ECS-optimiertes AMI

Änderungen im Jahr AL2 2023 für Amazon ECS im Vergleich zu AL2

Wie bei AL2 023 stellt 023 die Basispakete bereit AL2, die für die Ausführung als Amazon ECS-Linux-Instance erforderlich sind. In AL2 der Version `containerd` `docker`, und waren die `ecs-init` Pakete verfügbar `amazon-linux-extras`, wohingegen AL2 023 diese Pakete in den Kern-Repositorys enthält.

Mit der Funktion „Deterministische Upgrades durch versionierte Repositorys“ ist jedes AL2 023-AMI standardmäßig an eine bestimmte Repository-Version gebunden. Dies gilt auch für das für Amazon ECS optimierte AL2 023 AMI. 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 zu dieser AL2 023-Funktion finden Sie unter. Deterministische Upgrades durch versionierte Repositorys auf 023 AL2

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

> **Note**
>
> AL2023 Versionen vor 2023.2.20230920 (der ersten Version AL2 023.2) enthielten einen Fehler bei der (OOM-) Handhabung innerhalb einer Cgroup. `systemd` Out-of-Memory 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 ist seit Version 2023.2.20230920 von 023 behoben. AL2

Der Code zum Erstellen des Amazon ECS-optimierten AMI ist im amazon-ecs-ami GitHub Projekt verfügbar. In den Versionshinweisen wird beschrieben, welche Version AL2 023 welcher Amazon ECS AMI-Version zugeordnet ist.

Anpassen des AL2 023-basierten Amazon ECS-optimierten AMI

> **Important**
>
> Wir empfehlen die Verwendung des für Amazon ECS optimierten AL2 023 AMI. 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 von benutzerdefinierten Skripten verwendet AMIs. Weitere Informationen finden Sie unter Amazon ECS-optimiertes Linux-AMI-Build-Skript.

Verwenden von Amazon Elastic File System auf AL2 023

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 Network File System-Protokoll Version 4.1 (NFSv4.1 und NFSv4 .0), sodass die Anwendungen und Tools, die Sie heute verwenden, problemlos 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.

Installation **`amazon-efs-utils`** auf AL2 023

Das `amazon-efs-utils` Paket ist in den AL2 023-Repositorys verfügbar und kann installiert und für den Zugriff auf Amazon EFS-Dateisysteme verwendet werden.

Installieren Sie das **`amazon-efs-utils`** Paket auf 0.23 AL2

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

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

Mounten eines Amazon EFS-Dateisystems auf AL2 023

Nach `amazon-efs-utils` der Installation können Sie ein Amazon EFS-Dateisystem auf Ihrer AL2 023-Instance mounten.

Mounten Sie ein Amazon EFS-Dateisystem auf AL2 023

- 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 023 basiert AL2

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

AL2Amazon EMR-Versionen auf Basis von 2023

Amazon EMR Version 7.0.0 war die erste Version, die auf AL2 023 basiert. Mit dieser Version ist AL2 023 das Basisbetriebssystem für Amazon EMR und bietet Amazon EMR alle Vorteile von AL2 023. Weitere Informationen finden Sie in den Versionshinweisen zu Amazon EMR 7.0.0.

AL2023 basiert auf Amazon EMR auf EKS

Amazon EMR auf EKS 6.13 war die erste Version, in der AL2 023 als Option eingeführt wurde. Mit dieser Version können Sie Spark mit AL2 023 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.

Mit AL2 0,23 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.

AL2023 **`provided.al2023`** verwaltete Laufzeit und Container-Image

Die `provided.al2023` Basislaufzeit basiert auf dem minimalen Container-Image AL2 023 und bietet ein AL2 023-basiertes, von Lambda verwaltetes Runtime- und Container-Basis-Image. Da die `provided.al2023` Runtime auf dem minimalen Container-Image AL2 023 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.

AL2023-basierte Lambda-Laufzeiten

Zukünftige Versionen verwalteter Sprachlaufzeiten, wie Node.js 20, Python 3,12, Java 21, und .NET 8 basieren auf AL2 023 und werden als Basis-Image verwendet, `provided.al2023` wie in der Ankündigung von AL2 023-basierten Laufzeiten beschrieben.

AL2023 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 EC2 Amazon-Instances ausführen, auf denen Amazon Linux 2023 (AL2023) ausgeführt wird. Video-Tutorials finden Sie unter AWS Lehrvideos und Übungen.

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

Tutorials

- Tutorial: Installieren Sie einen LAMP-Server auf AL2 023
- Tutorial: SSL/TLS auf 023 konfigurieren AL2
- Tutorial: Hosten Sie einen WordPress Blog auf AL2 023
- Tutorial: Übergang von Redis 6 zu Valkey auf 023 AL2
- Tutorial: Installieren Sie die GNOME-Desktop-Umgebung auf 023 AL2
- Tutorial: TigerVNC-Server auf 023 konfigurieren AL2

Tutorial: Installieren Sie einen LAMP-Server auf AL2 023

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 AL2 023-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 023 vorgesehen. AL2 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 bereits eine neue Instanz mit AL2 023 gestartet haben, deren öffentlicher DNS-Name über das Internet erreichbar ist. Weitere Informationen finden Sie unter AL2023 bei 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 EC2 Amazon-Benutzerhandbuch.
- Mit dem folgenden Verfahren wird die neueste PHP-Version installiert, die am AL2 023 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 023 Instances herstellen AL2.
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 upgrade -y
   ```

3. Installieren Sie die neuesten Versionen des Apache-Webservers und der PHP-Pakete für AL2 0.23.

```
[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 beim Start eine ***N*** Launch-Wizard-Sicherheitsgruppe für Ihre Instance erstellt. Wenn Sie keine zusätzlichen Sicherheitsgruppenregeln hinzugefügt haben, enthält diese Gruppe nur eine einzige Regel, um SSH-Verbindungen zuzulassen.

   a. Öffnen Sie die EC2 Amazon-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**
>
> `0.0.0.0/0`Durch 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 das öffentliche DNS für Ihre Instance über die EC2 Amazon-Konsole abrufen (überprüfen Sie die Spalte Öffentliches IPv4 DNS; wenn diese Spalte ausgeblendet ist, wählen Sie Einstellungen (das zahnradförmige Symbol) und schalten Sie Öffentliches 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 eine Verbindung zu einem HTTP-Server herstellen, sind URLs die von Ihnen besuchten Daten, der Inhalt der Webseiten, die Sie erhalten, und die Inhalte (einschließlich Kennwörter) aller HTML-Formulare, die Sie einreichen, für Lauscher überall im Netzwerkpfad sichtbar. 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 023 konfigurieren AL2.

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

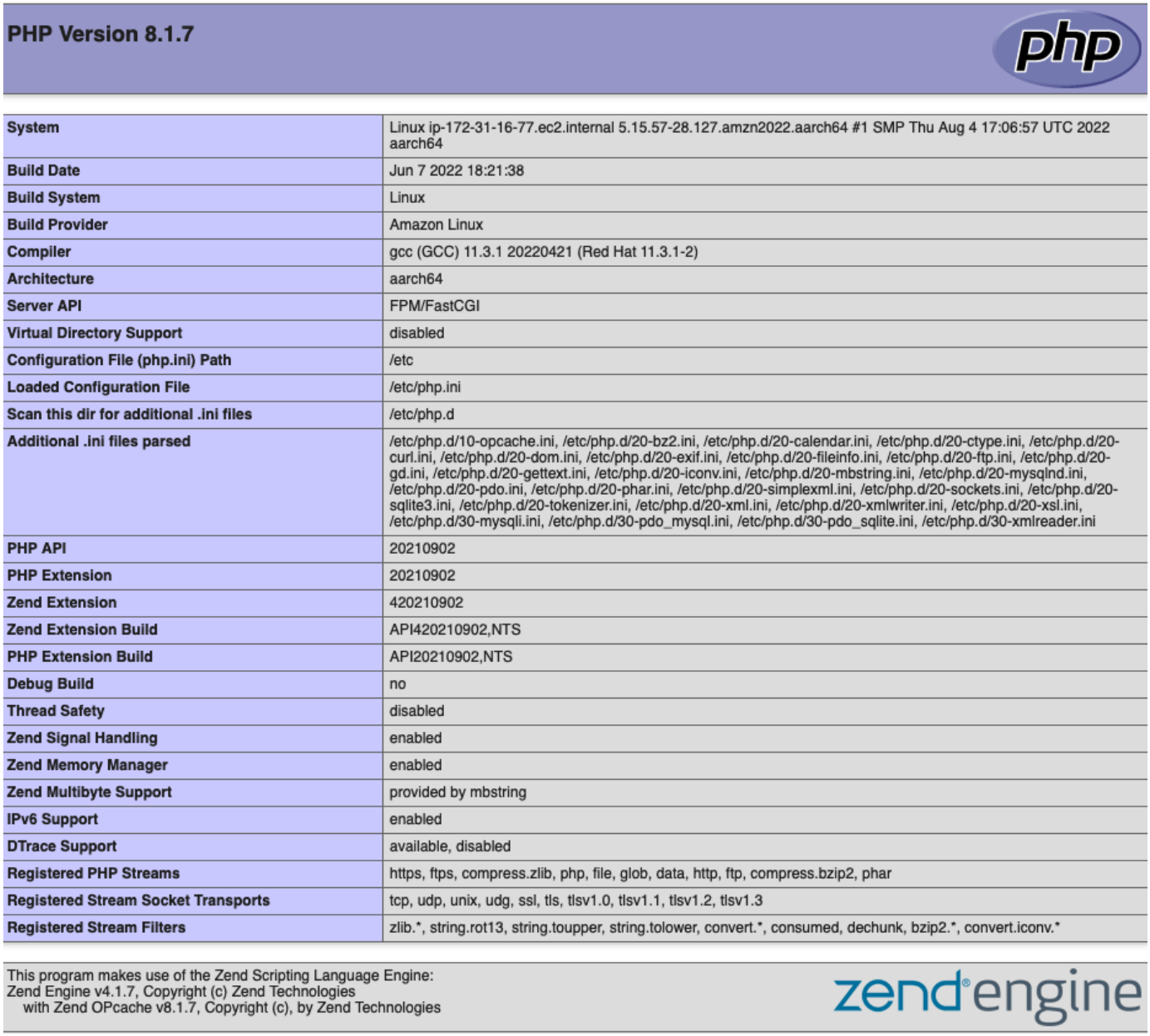
1. Erstellen Sie eine PHP-Datei im Dokumenten-Stammverzeichnis von Apache.

   ```
   [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 Hinweise zur Erstellung 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 Instanz 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 zum Sichern eines Webservers auf einer EC2 Instance finden Sie unterTutorial: SSL/TLS auf 023 konfigurieren AL2.

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 *`phpMyAdmin-latest-all-languages.tar.gz`* Tarball.

```
[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 phpMyAdmin Installation ein. 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: Zum Beispiel:

   ```
   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 023 konfigurieren AL2.

- 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 EC2 Amazon-Benutzerhandbuch.
- Übertragen Sie Dateien mithilfe eines SCP-Clients auf Linux-Instances im EC2 Amazon-Benutzerhandbuch.
- Tutorial: Hosten Sie einen WordPress Blog auf AL2 023

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 023 konfigurieren AL2

Secure Sockets Layer/Transport Layer Security (SSL/TLS) creates an encrypted channel between a web server and web client that protects data in transit from being eavesdropped on. This tutorial explains how to add support manually for SSL/TLS auf einer EC2 Instanz mit AL2 023 und Apache Webserver. 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. Webbrowser unterstützen zwar weiterhin SSL, sein Nachfolgeprotokoll TLS ist jedoch 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 verfügbar, AL2 solange 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 AL2 0.23 vorgesehen. Wenn Sie versuchen, eine EC2 Instance einzurichten, auf der eine andere Distribution ausgeführt wird, oder eine Instance, auf der eine alte Version von Amazon Linux ausgeführt wird, funktionieren einige Verfahren in diesem Tutorial möglicherweise 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-Instances mit Nitro Enclaves ausgeführt werden. EC2 AWS Nitro Enclaves ist eine EC2 Amazon-Funktion, die die Schaffung isolierter Computerumgebungen ermöglicht, um hochsensible Daten wie SSL/TLS-Zertifikate und private Schlüssel zu schützen und sicher zu verarbeiten.
> ACM for Nitro Enclaves arbeitet mit Nginx, das auf Ihrer Amazon EC2 Linux-Instance ausgeführt wird, um private Schlüssel zu erstellen, Zertifikate und private Schlüssel zu verteilen und Zertifikatserneuerungen zu verwalten.
> Um ACM for Nitro Enclaves verwenden zu können, müssen Sie eine Enclave-fähige Linux-Instance nutzen.
> Weitere Informationen finden Sie unter Was ist Nitro Enclaves? AWS 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 023-Instance. AL2 Weitere Informationen finden Sie unter AL2023 bei 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 EC2 Amazon-Benutzerhandbuch.

- Installieren Sie den Apache-Webserver. step-by-stepAnweisungen finden Sie unter. Tutorial: Installieren Sie einen LAMP-Server auf AL2 023 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). Um Ihre EC2 Instance zum Hosten einer öffentlichen Website zu verwenden, müssen Sie einen Domainnamen für Ihren Webserver registrieren oder einen vorhandenen Domainnamen auf Ihren EC2 Amazon-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 AL2 023 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 023 Instances herstellen AL2.

   ```
   [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 sicher, dass der TCP-Port 443 auf Ihrer EC2 Instanz zugänglich ist, wie zuvor beschrieben.

5. Ihr Apache-Webserver sollte jetzt HTTPS (sicheres HTTP) über Port 443 unterstützen. Testen Sie es, indem Sie die IP-Adresse oder den vollqualifizierten Domainnamen Ihrer EC2 Instanz in eine Browser-URL-Leiste mit dem Präfix eingeben**`https://`**.

   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 der vom Browserhersteller zu diesem Zweck CAs vertrauenswürdigen Websites. 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 über HTTPS eine Verbindung zu einem Webserver herstellt, präsentiert der Server dem Browser ein Zertifikat, das er anhand seiner Liste vertrauenswürdiger Server überprüfen kann CAs. 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 kostenlose Basiszertifikate an. Das bemerkenswerteste davon 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 AL2 023 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. Connect zu Ihrer Instance her und navigieren Sie zu to /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 023 Instances herstellen AL2
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 CAs 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 der allgemeine Name nur aus dem Hostnamen bestehen. CAs bieten auch teurere Zertifikate an, die Platzhalternamen akzeptieren, wie z. **`*.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 Namen (SANs) für das Zertifikat anzugeben. 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 Ihr CA-Webformular dies zulässt, nehmen Sie den allgemeinen Namen in die Liste der auf SANs. Manche CAs schließen ihn 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, Ihr neues Zertifikat auf Ihre EC2 Instanz hochzuladen. Die einfachste und informativste Methode besteht jedoch darin, einen Texteditor (z. B. vi, nano oder Notepad) sowohl auf Ihrem lokalen Computer als auch auf Ihrer Instanz zu öffnen und dann den Dateiinhalt zwischen den Instanzen zu kopieren und einzufügen. Sie benötigen Root-Rechte [sudo], wenn Sie diese Operationen auf der EC2 Instanz ausführen möchten. 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 stark restriktiven AL2 023-Standardwerten entsprechen (owner=root, group=root, read/write nur für Besitzer). 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, Ihren benutzerdefinierten Schlüssel auf Ihre EC2 Instanz hochzuladen. Die einfachste und informativste Methode besteht jedoch darin, einen Texteditor (z. B. vi, nano oder Notepad) 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-Rechte [sudo], wenn Sie diese Operationen auf der EC2 Instanz ausführen möchten. 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 stark restriktiven AL2 023-Standardeinstellungen entsprechen (owner=root, group=root, read/write nur für Besitzer).

```
[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**
   >
   > Manche CAs kombinieren das Host-Zertifikat und die Zwischenzertifikate in einer einzigen Datei, sodass die Direktive überflüssig wird. `SSLCACertificateFile` 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 AL2 023 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 wird für die Verwendung durch bestimmte ältere Browser unterstützt. Eine Chiffre ist der mathematische Kern eines Verschlüsselungsalgorithmus. RC4, eine schnelle Chiffre, die zur Verschlüsselung von TLS-Datenströmen verwendet wird, weist bekanntermaßen mehrere schwerwiegende Schwächen auf. 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. 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 werden diese Chiffren nicht unterstützt. RC4

   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 erneut auf Qualys SSL Labs testen, sollten Sie feststellen, dass die RC4 Sicherheitslücke und andere Warnungen behoben sind und die Zusammenfassung 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 AL2 023-Instanz bei 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, der `custom.key` im Standardverzeichnis aufgerufen wird, und das zugehörige Passwort lautet, führen Sie die folgenden Befehle auf Ihrer EC2 Instance aus**abcde12345**, um eine unverschlüsselte Version des Schlüssels zu generieren.

  ```
  [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
  ```

  Das bedeutet in der Regel, dass Ihre EC2 Instance nicht 023 ausführt AL2. Dieses Tutorial unterstützt nur Instances, die frisch von einem offiziellen AL2 023-AMI erstellt wurden.

Tutorial: Hosten Sie einen WordPress Blog auf AL2 023

Die folgenden Verfahren helfen Ihnen bei der Installation, Konfiguration und Sicherung eines WordPress Blogs auf Ihrer AL2 023-Instance. Dieses Tutorial ist eine gute Einführung EC2 in die Verwendung von Amazon, da Sie die volle Kontrolle über einen Webserver haben, auf dem Ihr WordPress Blog gehostet wird, was bei einem herkömmlichen Hosting-Dienst 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 eine Hochverfügbarkeitslösung mit einer entkoppelten Datenbank benötigen, finden Sie weitere Informationen unter Deployment a High Availability WordPress Website im Developer Guide.AWS Elastic Beanstalk

> **Important**
>
> Diese Verfahren sind für die Verwendung mit 023 vorgesehen. AL2 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 Elastic IP-Adressen im EC2 Amazon-Benutzerhandbuch. Das Tutorial: Installieren Sie einen LAMP-Server auf AL2 023-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 023 Instances herstellen AL2.

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 upgrade --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 upgrade --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 mit dem Namen`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. nanooder 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. Besuchen Sie https://api.wordpress. org/secret-key/1.1/salt/, um nach dem Zufallsprinzip eine Reihe von Schlüsselwerten zu generieren, die Sie kopieren und in Ihre `wp-config.php` Datei 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 von dem Verzeichnis aus 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 0.23 zu installieren AL2

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 AL2 023 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 AL2 023 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 AL2 023 vorgesehen. Verwenden Sie das Verfahren, das diesem folgt, mit AL2 023 AMI.

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 AL2 023 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 der EIP-Adresse Ihrer EC2 Instanz ein Domainname zugeordnet ist, können Sie Ihren Blog so konfigurieren, dass dieser Name anstelle der EC2 öffentlichen DNS-Adresse verwendet wird. 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 EC2 Amazon-Benutzerhandbuch.

Erhöhen der Kapazität

Wenn Ihr WordPress Blog immer beliebter 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 Problembehandlung bei Ihrer Installation finden Sie unter Häufige Installationsprobleme.
- Informationen dazu, wie Sie Ihr WordPress Blog sicherer machen können, finden Sie unter Hardening WordPress.
- Informationen dazu, wie Sie Ihr WordPress Blog behalten können 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 beenden und neu starten, ändert sich die öffentliche DNS-Adresse (sofern sie nicht mit einer Elastic IP-Adresse verknüpft ist) und Ihr Blog funktioniert nicht mehr, da er auf Ressourcen verweist, die nicht mehr existieren (oder einer anderen EC2 Instance zugewiesen sind). 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 mit SSH eine Connect zu Ihrer EC2 Instance her.

2. Notieren Sie die alte und die neue Website-URL für Ihre Instance. Bei der alten Site-URL handelt es sich wahrscheinlich um den öffentlichen DNS-Namen für Ihre EC2 Instance, als Sie sie installiert haben WordPress. Die neue Site-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 Ihre EC2 Instanz und den Pfad URLs zu Ihrer WordPress Installation durch die alte und die neue Site (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.

Tutorial: Übergang von Redis 6 zu Valkey auf 023 AL2

In der folgenden Dokumentation werden die wichtigsten Aspekte der Umstellung von Redis 6 auf Valkey auf Version 023 beschrieben. AL2

Supportzeitplan für Redis 6

Redis 6 erreicht am 31. August 2025 sein Lebensende (EOL). Nach diesem Datum erhält Redis 6 keine Updates oder Sicherheitspatches mehr vom Redis-Projekt. Wir empfehlen Benutzern dringend, vor August 2025 zu Valkey zu migrieren, um weiterhin Support und Sicherheitsupdates zu gewährleisten.

Weitere Informationen zu den Zeitplänen für die Unterstützung von Redis-Versionen finden Sie in der Dokumentation zu Redis Schedule. End-Of-Life

Einführung in Valkey

Valkey ist ein Open-Source-Fork von Redis 7, der von der Linux Foundation verwaltet wird. Es ist vollständig kompatibel mit den Versionen 2.x bis 7.2.x der Redis Open Source Software (OSS). Valkey behält die vertraute Redis-API und -Funktionalität bei und bietet gleichzeitig mehrere Verbesserungen:

- Verbesserte Leistung durch Multithreading.
- Verbesserte Speichereffizienz, insbesondere im Clustermodus.
- Zweikanal-Replikation für bessere Datenkonsistenz.

Migrationsplan und Zeitplan

Benutzern wird dringend empfohlen, vor dem 31. August 2025, wenn Redis 6 sein Lebensende (EOL) erreicht, von Redis 6 auf Valkey zu migrieren. Diese Migration erfordert manuelles Eingreifen und erfolgt nicht automatisch.

Amazon Linux empfiehlt diese Migration, um die kontinuierliche Funktionalität, den Support und die Sicherheitsupdates für Ihre von Redis abhängigen Anwendungen sicherzustellen.

Migrationsoptionen und -schritte

Wir schlagen drei Migrationspfade zu Valkey vor, die auf Ihren Bereitstellungsanforderungen und betrieblichen Anforderungen basieren.

Option 1: Installation einer neuen Instanz

Für neue Bereitstellungen oder wenn keine Datenmigration erforderlich ist:

1. Installieren Sie Valkey:

```
[ec2-user ~]$ sudo dnf install valkey
```

2. Starten Sie Valkey:

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

3. (Optional) Aktivieren Sie Valkey beim Booten:

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

4. Überprüfen Sie die Installation:

```
[ec2-user ~]$ valkey-cli info server
[ec2-user ~]$ valkey-cli ping
```

Option 2: Austausch vor Ort

Für bestehende Instanzen, in denen keine Datenpersistenz erforderlich ist:

1. Stoppen Sie Redis 6:

```
[ec2-user ~]$ sudo systemctl stop redis6
```

2. Installieren Sie Valkey:

```
[ec2-user ~]$ sudo dnf install valkey
```

3. (Optional) Verwenden Sie die Redis 6-Konfiguration in Valkey:

```
[ec2-user ~]$ sudo cp /etc/redis6/redis6.conf /etc/valkey/valkey.conf
[ec2-user ~]$ sudo cp /etc/valkey/valkey.conf /etc/valkey/valkey.conf.backup
[ec2-user ~]$ sudo chown valkey:root /etc/valkey/valkey.conf
[ec2-user ~]$ sudo sed -i 's|^dir\s.*|dir /var/lib/valkey|g' /etc/valkey/
valkey.conf
```

4. (Optional) Verwenden Sie die Redis 6-Sentinel-Konfigurationsdatei in Valkey:

```
[ec2-user ~]$ sudo cp /etc/redis6/sentinel.conf /etc/valkey/sentinel.conf
```

```
[ec2-user ~]$ sudo chown valkey:root /etc/valkey/sentinel.conf
```

5. Starten Sie Valkey:

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

6. (Optional) Aktivieren Sie Valkey beim Booten:

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

7. Überprüfen Sie die Valkey-Installation:

```
[ec2-user ~]$ valkey-cli info server
[ec2-user ~]$ valkey-cli ping
```

8. Redis 6 entfernen:

```
[ec2-user ~]$ sudo dnf remove redis6
```

Option 3: Datenmigration

Mit dieser Option können Sie sowohl Redis 6 als auch Valkey gleichzeitig ausführen.

1. Installieren Sie Valkey, ohne Redis 6 zu entfernen:

```
[ec2-user ~]$ sudo dnf install valkey
```

2. (Optional) Verwenden Sie die Redis 6-Konfiguration in Valkey:

```
[ec2-user ~]$ sudo cp /etc/redis6/redis6.conf /etc/valkey/valkey.conf
[ec2-user ~]$ sudo cp /etc/valkey/valkey.conf /etc/valkey/valkey.conf.backup
[ec2-user ~]$ sudo chown valkey:root /etc/valkey/valkey.conf
[ec2-user ~]$ sudo sed -i 's|^dir\s.*|dir /var/lib/valkey|g' /etc/valkey/
valkey.conf
```

3. (Optional) Verwenden Sie die Redis 6-Sentinel-Konfigurationsdatei in Valkey:

```
[ec2-user ~]$ sudo cp /etc/redis6/sentinel.conf /etc/valkey/sentinel.conf
[ec2-user ~]$ sudo chown valkey:root /etc/valkey/sentinel.conf
```

4. Ändern Sie die Valkey-Konfiguration:

Bearbeiten Sie die 'port'-Direktive `/etc/valkey/valkey.conf` und setzen Sie sie auf einen anderen Wert (z. B. 6380), um Konflikte mit Redis 6 zu vermeiden.

5. Starten Sie Valkey:

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

6. (Optional) Aktivieren Sie Valkey beim Booten:

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

7. Überprüfen Sie die Valkey-Installation:

```
[ec2-user ~]$ valkey-cli -p port info server
[ec2-user ~]$ valkey-cli -p port ping
```

> **Note**
>
> Ersetzen Sie *port* durch die konfigurierte Portnummer.

8. Daten migrieren:

   Sie können jetzt Daten mithilfe von Replikations- oder manuellen Datenübertragungsmethoden von Redis 6 nach Valkey migrieren.

9. Anwendungskonfigurationen aktualisieren:

   Aktualisieren Sie Ihre Anwendungen schrittweise, um den Valkey-Port zu verwenden.

10. Entfernen Sie Redis 6:

    Sobald alle Daten und Anwendungen migriert wurden, können Sie Redis 6 beenden und entfernen.

```
[ec2-user ~]$ sudo systemctl stop redis6
[ec2-user ~]$ sudo dnf remove redis6
```

> **Note**
>
> Es wird dringend empfohlen, den Migrationsprozess in einer Testumgebung zu validieren, bevor Änderungen in Produktionssystemen implementiert werden.

Verwandte Themen

Weitere Informationen zu Valkey finden Sie unter:

- Valkey: https://valkey.io/
- Valkey-Migration: https://valkey. io/topics/migration/

Tutorial: Installieren Sie die GNOME-Desktop-Umgebung auf 023 AL2

Die GNOME-Desktop-Umgebung ist ab Version 2023.7 oder höher als optionale grafische Benutzeroberfläche für Version AL2 023 verfügbar.

Die folgenden Verfahren helfen Ihnen bei der Installation der GNOME-Desktop-Umgebung auf Ihrer 023-Instanz. AL2 Sie können diese grafische Oberfläche verwenden, um mit Ihrem Linux-System über eine vertraute Desktop-Umgebung zu interagieren, anstatt nur über die Befehlszeilenschnittstelle.

Inhalt

- Voraussetzungen
- Installation
- Verwandte Themen

Voraussetzungen

- Die Desktop-Umgebung benötigt mindestens 2,4 GB Arbeitsspeicher. Daher wird empfohlen, eine Instanz vom Typ `t2.medium` oder besser zu verwenden, um eine angemessene Leistung zu gewährleisten. Beispiele für Instance-Typen mit unzureichendem Arbeitsspeicher sind `t2.nanot2.micro`, und`t2.small`. Diese Einschränkung gilt `t3` auch für `t4` Instances dieser Größe sowie für alle anderen Instance-Typen, die die Speicheranforderungen nicht erfüllen.

- In diesem Tutorial wird davon ausgegangen, dass Sie bereits eine Instance mit Version AL2 023 mit Version 2023.7 oder höher gestartet haben. Weitere Informationen finden Sie auf den AL2023 bei Amazon EC2 Seiten und. Aktualisierung AL2 023

Installation

- Installieren Sie die GNOME-Desktop-Umgebung und die zugehörigen Pakete.

```
[ec2-user ~]$ sudo dnf groupinstall "Desktop" -y
```

> **Note**
>
> Um auf die grafische Desktop-Umgebung zuzugreifen, müssen Sie zusätzliche Software wie Amazon DCV oder VNC installieren und konfigurieren. Mit diesen Tools können Sie über das Netzwerk eine Verbindung zur grafischen Benutzeroberfläche herstellen und mit ihr interagieren.

Verwandte Themen

Weitere Informationen zur grafischen Desktop-Umgebung finden Sie in der folgenden Dokumentation:

- Was ist Amazon DCV? im Amazon DCV-Administratorhandbuch
- Tutorial: TigerVNC-Server auf 023 konfigurieren AL2

Tutorial: TigerVNC-Server auf 023 konfigurieren AL2

Die folgenden Verfahren helfen Ihnen beim Einrichten des VNC-Servers auf Ihrer 023-Instanz. AL2 Mit VNC können Sie über eine sichere Netzwerkverbindung remote auf die grafische Desktop-Umgebung zugreifen und mit ihr interagieren.

Inhalt

- Voraussetzungen
- Schritt 1: Installation
- Schritt 2: Konfiguration
- Schritt 3: Connect mit einem VNC-Client her

- (Optional) Starten Sie den Dienst beim Booten
- (Optional) Deaktivieren Sie den Sperrbildschirm im Leerlauf
- Verwandte Themen

Voraussetzungen

- In diesem Tutorial wird davon ausgegangen, dass Sie die GNOME-Desktop-Umgebung bereits auf Ihrer AL2 023-Instanz installiert haben. Weitere Informationen finden Sie auf der Seite Tutorial: Installieren Sie die GNOME-Desktop-Umgebung auf 023 AL2.
- Dieses Tutorial verwendet SSH-Portweiterleitung für den Zugriff auf den VNC-Server. Weitere Informationen zur Einrichtung Ihres key pair finden Sie unter Connect zu Ihrer Linux-Instance mithilfe von SSH herstellen im EC2 Amazon-Benutzerhandbuch.
- Das folgende Verfahren führt Sie nicht durch den Prozess der Installation eines VNC-Clients. Sie müssen einen VNC-Client auf Ihrem lokalen Computer installiert haben, um eine Verbindung zur Desktop-Umgebung herstellen und mit ihr interagieren zu können.

Schritt 1: Installation

1. Verbinden Sie sich mit der Instance. Weitere Informationen finden Sie unter Verbindung zu 023 Instances herstellen AL2.
2. Installieren Sie das TigerVNC-Serverpaket für 023. AL2

   Die `-y` Option installiert das Paket, ohne nach einer Bestätigung zu fragen. Wenn Sie das Paket vor der Installation überprüfen möchten, können Sie diese Option weglassen.

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

Schritt 2: Konfiguration

1. Stellen Sie sicher, dass der Benutzer ein VNC-Passwort konfiguriert hat.

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

2. Weisen Sie dem Benutzer eine Anzeigenummer zu.

```
[ec2-user ~]$ sudo vi /etc/tigervnc/vncserver.users
```

Fügen Sie die folgende Konfiguration hinzu:

```
:1=ec2-user
```

> **Note**
>
> Sie können dem Benutzer eine beliebige Display-Nummer zuweisen. `:1`Für dieses Beispiel verwenden wir Display.

3. Bearbeiten Sie die VNC-Serverkonfigurationsdatei.

```
[ec2-user ~]$ sudo vi /etc/tigervnc/vncserver-config-defaults
```

Fügen Sie die folgende Konfiguration hinzu:

```
session=gnome
securitytypes=vncauth,tlsvnc
geometry=1920x1080
localhost
alwaysshared
```

> **Note**
>
> Sie können die Auflösung des Displays mithilfe des `geometry` Parameters ändern. Wir verwenden `1920x1080` für dieses Beispiel.

4. Starten Sie den VNC-Server. Dieser Vorgang muss jedes Mal wiederholt werden, wenn Sie Ihre Instance neu starten. Wenn Sie den Prozess des Startens dieses Dienstes automatisieren möchten, lesen Sie den optionalen Abschnitt weiter unten.

```
[ec2-user ~]$ sudo systemctl start vncserver@:1
```

> **Important**
>
> Beim Starten des `vncserver` Dienstes `@` muss der Teil hinter dem mit der für den Benutzer in der `/etc/tigervnc/vncserver.users` Datei festgelegten Anzeigenummer übereinstimmen.

Nachdem Sie diesen Schritt ausgeführt haben, können Sie den SSH-Tunnel von Ihrem lokalen Computer aus erstellen und mit Ihrem VNC-Client eine Verbindung herstellen.

Schritt 3: Connect mit einem VNC-Client her

Der VNC-Server stellt einen TCP-Socket für Client-Verbindungen zur Verfügung. Sie könnten den VNC-Port zwar direkt über Ihre Sicherheitsgruppe verfügbar machen, aber dieses Tutorial zeigt, wie Sie SSH-Tunneling als sichereren Ansatz verwenden können, indem die Verbindung zwischen Ihrem lokalen Computer und der Instanz verschlüsselt wird. EC2 Sobald Sie die Verbindung über den Tunnel hergestellt haben, authentifizieren Sie sich beim VNC-Server mit dem Passwort, das Sie im vorherigen Schritt konfiguriert haben. Weitere Informationen zu Sicherheitsgruppen finden Sie unter Ändern der Sicherheitsgruppen für Ihre EC2 Amazon-Instance im EC2 Amazon-Benutzerhandbuch.

1. Erstellen Sie einen SSH-Tunnel von Ihrem lokalen Computer aus.

```
$ ssh -i <keypair> -L 5901:localhost:5901 ec2-user@<address>
```

> **Note**
>
> `<keypair>`Ersetzen Sie ihn durch den Pfad zu Ihrem SSH-Schlüssel und `<address>` durch die öffentliche IP oder den DNS-Namen Ihrer Instanz. Der Port ändert sich je nach der Anzeigenummer, die zum Starten von verwendet wurde. `vncserver` Beispielsweise `:1` verwendet das Display den Port`5901`, `:2` das Display den Port `5902` usw.

2. Verwenden Sie Ihren VNC-Client, um eine Verbindung zu `localhost:5901` oder `127.0.0.1:5901` mit dem zuvor festgelegten VNC-Passwort herzustellen.

> **Important**
>
> Halten Sie den SSH-Tunnel geöffnet, während Sie VNC verwenden. Wenn der SSH-Tunnel nicht geöffnet ist, können Sie Ihren VNC-Client nicht verwenden, um die Desktop-Umgebung anzusehen und mit ihr zu interagieren.

(Optional) Starten Sie den Dienst beim Booten

Wenn Sie VNC regelmäßig verwenden möchten, sollten Sie den VNC-Server so konfigurieren, dass er beim Booten Ihrer Instanz automatisch gestartet wird. Dadurch entfällt die Notwendigkeit, den VNC-Server bei jedem Neustart Ihrer Instanz manuell zu starten. Diese Konfiguration stellt sicher, dass Ihre grafische Desktop-Umgebung bereit und zugänglich ist, sobald Ihre Instanz ihren Startvorgang abgeschlossen hat.

- Konfigurieren Sie den Dienst so, dass er beim Booten gestartet wird.

```
[ec2-user ~]$ sudo systemctl enable vncserver@:1
```

> **Important**
>
> Bei der Aktivierung des `vncserver` Dienstes `@` muss der Teil hinter dem mit der für den Benutzer in der `/etc/tigervnc/vncserver.users` Datei eingestellten Anzeigenummer übereinstimmen. Darüber hinaus können Sie das `--now` Argument after übergebene`enable`, um den Dienst sofort zu starten.

Nachdem Sie diesen Schritt ausgeführt haben, müssen Sie nicht mehr `vncserver` jedes Mal neu starten, wenn Sie Ihre Instanz neu starten.

(Optional) Deaktivieren Sie den Sperrbildschirm im Leerlauf

- Setzen Sie die Leerlaufverzögerung auf Null, um den Sperrbildschirm zu deaktivieren, wenn der Benutzer für einen längeren Zeitraum inaktiv war.

```
[ec2-user ~]$ gsettings set org.gnome.desktop.session idle-delay 0
```

Verwandte Themen

Weitere Informationen zur grafischen Desktop-Umgebung finden Sie in der folgenden Dokumentation:

- Tutorial: Installieren Sie die GNOME-Desktop-Umgebung auf 023 AL2
- Was ist Amazon DCV? im Amazon DCV-Administratorhandbuch

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 Behältern.

Amazon Linux 2023 (AL2023) kann auch als virtualisierter Gast ausgeführt werden, wenn es nicht direkt auf Amazon ausgeführt wird. EC2 Derzeit gibt es KVM (`qcow2`), VMware (`OVA`) und Hyper-V (`vhdx`) -Images sind verfügbar.

> **Note**
>
> Die Konfiguration der Amazon-Linux-2023-Images ist anders als bei Amazon Linux 2. Wenn Sie Amazon Linux 2 als virtuelle Maschine vor Ort ausführen möchten, müssen Sie Ihre Konfiguration anpassen, damit sie mit AL2 023 kompatibel ist.

Laden Sie Amazon Linux 2023-Images zur Verwendung mit KVM und VMware Hyper-V herunter

Amazon Linux 2023-Festplatten-Images zur Verwendung mit KVM und Hyper-V können von cdn.amazonlinux.com heruntergeladen werden. VMware

Unterstützte Konfigurationen von Amazon Linux 2023 zur Verwendung in EC2 virtualisierten Umgebungen außerhalb von Amazon

In diesem Abschnitt werden die Anforderungen für die Ausführung von Amazon Linux 2023 in EC2 virtualisierten Umgebungen behandelt, die nicht von Amazon stammen, z. B. auf KVM VMware, oder und Hyper-V.

Die Basis AL2023 Systemanforderungen gilt für alle EC2 virtualisierten Umgebungen, die nicht von Amazon stammen. Eine Liste der unterstützten Gerätemodelle für jede Hypervisor-Umgebung finden Sie in den folgenden Themen.

KVM und Hyper-V bieten viele Konfigurationsoptionen, und es muss sorgfältig darauf geachtet werden, sie für Ihre Sicherheits-, Leistungs- und Zuverlässigkeitsanforderungen zu konfigurieren. VMware Weitere Informationen finden Sie in der Dokumentation Ihres Hypervisors.

Themen

- Voraussetzungen für die Ausführung von AL2 023 auf KVM
- Voraussetzungen für die Ausführung von AL2 023 auf VMware
- Anforderungen für die Ausführung von Amazon Linux 2023 auf Hyper-V

Voraussetzungen für die Ausführung von AL2 023 auf KVM

In diesem Abschnitt werden die Anforderungen für die Ausführung von AL2 023 auf KVM beschrieben. Die KVM-Images von AL2 023 sind für beide Architekturen verfügbar. `aarch64` `x86-64` Diese Anforderungen gelten zusätzlich zur Grundlage AL2023 Systemanforderungen für die KVM-Images.

Themen

- KVM-Hostanforderungen für die Ausführung von AL2 023 auf KVM
- Geräteunterstützung für AL2 0.23 auf KVM
- Startmodus (UEFI and BIOS) Unterstützung für AL2 023 auf KVM
- Einschränkungen bei der Ausführung von AL2 023 auf KVM

KVM-Hostanforderungen für die Ausführung von AL2 023 auf KVM

Die KVM-Images sind derzeit auf einem Host qualifiziert, auf dem Ubuntu 22.04.3 LTS mit Version `qemu` 6.2+dfsg-2ubuntu6.15, bereitgestellt von dieser Ubuntu-Version, wobei ein `q35` Maschinentyp für `x86-64` und ein `virt` Maschinentyp für verwendet werden. `aarch64`

Geräteunterstützung für AL2 0.23 auf KVM

Folgende **`qemu`** Gerätemodelle wurden für die Verwendung mit AL2 023 KVM-Images (sowohl als auch**`aarch64`**) getestet: **`x86-64`**

- `virtio-blk` (`virtio`-Blockgerät)
- `virtio-scsi` (`virtio` SCSI Controller mit Festplatte (Gerät)
- `virtio-net` (`virtio`-Netzwerkgerät)

- `ahci` (zur Verwendung mit dem virtuellen CD-ROM-Laufwerk)
- `usb-storage` (über `xhci`)

Weitere **`qemu`** Gerätemodelle, die in AL2 023 KVM-Image-Qualifizierung aktiviert wurden, aber nicht intensiv genutzt wurden, sind:

- VGA (`qemu` VGA) nur ein `x86-64`
- `virtio-rng` (virtueller Zufallszahlengenerator)
- Legacy AT Tastatur und PS/2 Mausgeräte
- Serielles Legacy-Gerät

Startmodus (UEFI and BIOS) Unterstützung für AL2 023 auf KVM

Das `x86-64` Image wurde mit beiden älteren Versionen getestet BIOS and UEFI Startmodi. Die `aarch64` Bilder werden getestet mit UEFI Boot-Modus.

> **Note**
>
> Standardmäßig bei der Verwendung UEFI Im Startmodus stellen einige Manager virtueller Maschinen der VM Microsoft Secure Boot-Schlüssel zur Verfügung, wodurch Secure Boot aktiviert wird. Mit dieser Konfiguration wird AL2 023 nicht gestartet.
> Da der AL2 023-Bootloader nicht von Microsoft signiert ist, muss die VM entweder ohne UEFI-Schlüssel oder mit den AL2 023-Schlüsseln für Secure Boot bereitgestellt werden.

> **Important**
>
> Secure Boot-Unterstützung für KVMImages wurde noch nicht validiert.

Einschränkungen bei der Ausführung von AL2 023 auf KVM

Es gibt einige bekannte Einschränkungen bei der Ausführung von AL2 023 auf KVM.

> **Note**
>
> Code, der einige der aufgelisteten, nicht unterstützten Funktionen implementiert, ist möglicherweise in AL2 023 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 beim Ausführen von AL2 023 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 die Ausführung von AL2 023 auf VMware

In diesem Abschnitt werden die Anforderungen für die Ausführung von AL2 023 auf beschrieben VMware. Das VMware Bilder von AL2 023 sind nur für die `x86-64` Architektur verfügbar. VMware Bilder für `aarch64` sind nicht verfügbar oder werden nicht unterstützt. Diese Anforderungen gelten zusätzlich zur Grundlage AL2023 Systemanforderungen für VMware Bilder.

Themen

- VMware Hostanforderungen für die Ausführung von AL2 023 auf VMware
- Geräteunterstützung für AL2 023 auf VMware
- Startmodus (UEFI and BIOS) Unterstützung für AL2 023 auf VMware
- Einschränkungen bei Ausführung von AL2 023 auf VMware

VMware Hostanforderungen für die Ausführung von AL2 023 auf VMware

Die AL2 023 VMware OVA-Bilder sind derzeit für Folgendes qualifiziert:

- VMware Workstation 17.5.0 läuft auf Hosts mit einem Intel (R) Xeon (R) Platinum 8124M Prozessor
- VMware vSphere 8.0 mit einem Intel (R) Xeon (R) Platinum 8275CL Prozessor

Das 023 AL2 VMware OVA-Images spezifizieren eine Maschinenhardwareversion von 13.

VMware Maschinenhardwareversion 13 wird unterstützt von:

- ESXi 6.5 oder höher
- VMware Workstation 14 oder höher

Geräteunterstützung für AL2 023 auf VMware

Folgendes VMware Gerätemodelle wurden für die Verwendung mit AL2 023 getestet VMware OVA-Bilder (**`x86-64`**nur):

- `vmw_pvscsi` (VMware paravirtualisiert SCSI Steuerung)
- `vmxnet3` (VMware paravirtualisiertes Netzwerkgerät)
- `ata_piix`(Legacy) IDE nur zur Verwendung mit dem virtuellen CD-ROM-Laufwerk)

Zusätzlich VMware Gerätemodelle wurden im Jahr AL2 023 aktiviert VMware Bildqualifikation, aber nicht stark beansprucht:

- `vmw_vmci`und die zugehörige `vsock` Schnittstelle (virtueller Socket-Transport für VMware Gastagent)
- `vmw_balloon`-Memory-Ballooning-Gerät
- VMware `SVGA`Kontrolleur
- Legacy AT Tastatur und PS/2 Mausgeräte

Das Tool VMware Das Gast-Agent-Paket (`open-vm-tools`) ist verfügbar und standardmäßig im AL2 023 installiert VMware OVA-Bilder.

Startmodus (UEFI and BIOS) Unterstützung für AL2 023 auf VMware

Ab der Version 2023.3.20231211, die 023 AL2 VMware Das OVA-Image wurde in beiden älteren Versionen validiert BIOS and UEFI Startmodi. Die OVA-Standardkonfiguration ist immer noch veraltet BIOS kann aber vom Benutzer geändert werden.

> **Important**
>
> Secure Boot-Unterstützung erfordert UEFI, das nicht für AL2 023 validiert wurde, läuft auf VMware.

Einschränkungen bei Ausführung von AL2 023 auf VMware

Es gibt einige bekannte Einschränkungen bei der Ausführung von AL2 023 auf VMware.

> **Note**
>
> Code, der einige der aufgelisteten, nicht unterstützten Funktionen implementiert, ist möglicherweise in AL2 023 vorhanden und funktioniert ordnungsgemäß. 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 AL2 von 023 auf VMware

- UEFI Secure Boot ist derzeit nicht validiert, wenn AL2 023 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 AL2 023 sind nur für die Architektur verfügbar. `x86-64` Hyper-V-Bilder 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 EC2 `c5.metal` 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 AL2 023 Hyper-V-Images getestet:
  - Virtueller Hyper-V-Speicher *`hv_storvsc`* für die Stammfestplatte und das emulierte CD-ROM-Laufwerk der Generation 2 VMs
  - Emulierte PIIX-IDE *`ata_piix`* für das virtuelle CD-ROM-Laufwerk der Generation 1 VMs
  - Virtuelles Hyper-V-Ethernet `hv_netvsc`
- Die folgenden Gerätemodelle sind aktiviert, wurden jedoch nur geringfügig getestet:
  - Legacy-VGA-Textmodus auf Generation 1 VMs
  - Framebuffer auf UEFI-Firmware-Basis *`simpledrmfb`* auf Generation 2 VMs
  - Hyper-V-Ballon `hv_balloon`
  - Hyper-V-Ballon `hv_balloon`

  - Hyper-V HID/Maus `hid_hyperv`
- Die folgenden Gerätemodi sind in AL2023 derzeit 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**
>
> Code, der einige der aufgelisteten, nicht unterstützten Funktionen implementiert, ist möglicherweise in Version AL2 023 vorhanden und funktioniert ordnungsgemäß. 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 beim Ausführen von AL2 023 auf Hyper-V

- Der UEFI Secure Boot-Modus wird derzeit nicht unterstützt und funktioniert auch nicht mit 023 auf Hyper-V AL2
- 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.

Amazon Linux 2023 Einrichtung und `cloud-init` Konfiguration 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 ausgeführt wird EC2, z. B. wenn sie auf KVM oder Hyper-V läuft. VMware

Standardmäßig werden Images einer virtuellen Amazon Linux 2023-Maschine nicht mit einem Benutzerkennwort oder SSH-Schlüssel bereitgestellt und erhalten ihre Netzwerkkonfiguration über DHCP auf der ersten entdeckten Netzwerkschnittstelle. 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 herkömmliche Methode zur Konfiguration von lokalen Images über eine virtuelle CD-ROM, die einen Seed enthält ISO9660 Image mit `cloud-init` Konfigurationsdateien.

VMware

Amazon Linux 2023 unterstützt zusätzlich die Konfiguration von VMware Images, die auf vSphere ausgeführt werden, über die VMware spezifische Datenquelle über VMware `guestinfo.userdata` und`guestinfo.metadata`.

> **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 or VMware. Weil KVM and VMware Umgebungen haben keinen Amazon EC2 Instance Meta Data Service (IMDS), 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, ihre Passwörter, ssh Schlüsselpaare und/oder Zugriffsmechanismen. Standardmäßig erstellen Amazon Linux 2023 KVM und VMware Images 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 ist die Verwendung 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 Editor und fügen Sie Folgendes hinzu und *`vm-hostname`* ersetzen Sie es durch den Hostnamen für die VM:

   ```
   #cloud-config
   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.

```
#cloud-config
version: 2
ethernets:
  enp1s0:
    addresses:
      - 192.168.122.161/24
    gateway4: 192.168.122.1
    nameservers:
      addresses: 192.168.122.1
```

> **Note**
>
> `cloud-init`Die Netzwerkkonfiguration bietet Mechanismen zum Abgleich mit MAC Adresse der Schnittstelle statt Angabe des Schnittstellennamens, 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 mithilfe eines virtuellen CD-ROM-Laufwerks an Ihre neue virtuelle Amazon Linux 2023-Maschine angehängt werden`cloud-init`, um die Konfiguration beim ersten Start zu finden und auf das System anzuwenden.

VMware**`cloud-init`**guestinfo-Konfiguration für AL2 023 auf VMware

VMware Umgebungen verfügen nicht über den Amazon EC2 Instance Meta Data Service (IMDS), sodass eine alternative Methode zur Konfiguration von AL2 023 erforderlich ist. In diesem Abschnitt wird beschrieben, wie Sie einen alternativen Konfigurationsmechanismus zum `seed.iso` virtuellen CD-ROM-Laufwerk verwenden, der unter verfügbar ist VMware vSphere.

Diese Konfigurationsmethode verwendet die 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 dem zur Verfügung gestellt werden VMware `extraconfig`Mechanismus.

`guestinfo.metadata`

JSON or YAML enthält `cloud-init` Metadaten

`guestinfo.userdata`

A YAML Dokument, das `cloud-init` Benutzerdaten im Format enthält. `cloud-config`

`guestinfo.vendordata` (optional)

YAML enthält Herstellerdaten `cloud-init`

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 eingebettet werden `metadata` YAML or JSON Eigenschaften:

`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 Verwendung der VMware vSphere **`govc`** CLI-Tool zum Übergeben der Konfiguration mit **`guestinfo`**

1. Bereiten Sie die optionalen `network-config` Konfigurationsdateien `meta-data` `user-data`, und wie unter beschrieben vorNoCloud (`seed.iso`) `cloud-init` Konfiguration für Amazon Linux 2023 auf KVM und VMware.
2. Konvertiert die Konfigurationsdateien in Formate, die verwendet werden können von 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 AL2 023 KVM-Image

Ein Vergleich des RPMs aktuellen auf dem AL2 023 Standard AMI mit dem RPMs aktuellen auf dem AL2 023 KVM-Image.

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	2.5.1	
amazon-linux-onprem		1.2
amazon-linux-repo-cdn		2023,6.20241031
amazon-linux-repo-s3	2023,6.20241031	
amazon-linux-sb-keys	2023,1	2023,1
amazon-onprem-network		1.2
amazon-rpm-config	228	228
amazon-ssm-agent	3,3,987,0	3,3,987,0
amd-ucode-firmware	20210208 (Noarch)	20210208 (Noarch)
at	3.1.23	3.1.23

Paket	AMI	KVM
attr	2.5.1	2.5.1
audit	3.0.6	3.0.6
audit-libs	3.0.6	3.0.6
aws-cfn-bootstrap	2.0	
awscli-2	2,15,30	2,15,30
basesystem	11	11
bash	5.2,15	5.2.15
bash-completion	2.11	2.11
bc	1,07,1	1,07,1
bind-libs	9,18,28	9,18,28
bind-license	9,18,28	9,18,28
bind-utils	9,18,28	9,18,28
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	2023,2,68	2023,2,68
c-ares	1.19.1	

Paket	AMI	KVM
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
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

Paket	AMI	KVM
cyrus-sasl-plain	2.1.27	2.1.27
dbus	1.12,28	1.12.28
dbus-broker	32	32
dbus-common	1.12.28	1.12.28
dbus-libs	1.12.28	1.12.28
device-mapper	1,02,185	1,02,185
device-mapper-libs	1,02,185	1,02,185
diffutils	3.8	3.8
dnf	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2
dnf-plugins-core	4.3.0	4.3.0
dnf-plugin-support-info	1.2	1.2
dnf-utils	4.3.0	4.3.0
dosfstools	4,2	4.2
dracut	055	055
dracut-config-ec2	3.0	
dracut-config-generic	055	055

Paket	AMI	KVM
dwz	0,14	0,14
dyninst	10.2.1	10.2.1
e2fsprogs	1,46,5	1,46,5
e2fsprogs-libs	1,46,5	1,46,5
ec2-hibinit-agent	1.0.8	
ec2-instance-connect	1.1	
ec2-instance-connect-selinux	1.1	
ec2-utils	2.2.0	
ed	1.14.2	1.14.2
efi-filesystem	5	5
efi-srpm-macros	5	5
efivar	38	38
efivar-libs	38	38
elfutils-debuginfod-client	0.188	0.188
elfutils-default-yama-scope	0.188	0.188
elfutils-libelf	0.188	0.188
elfutils-libs	0.188	0.188
ethtool	5,15	5,15

Paket	AMI	KVM
expat	2.5.0	2.5.0
file	5,39	5,39
file-libs	5,39	5,39
filesystem	3,14	3,14
findutils	4.8.0	4.8.0
fonts-srpm-macros	2.0.5	2.0.5
fstrm	0.6.1	0.6.1
fuse-libs	2.9.9	2.9.9
gawk	5.1.0	5.1.0
gdbm-libs	1,19	1,19
gdisk	1.0.8	1.0.8
gettext	0,21	0,21
gettext-libs	0,21	0,21
ghc-srpm-macros	1.5.0	1.5.0
glib2	2,74,7	2,74,7
glibc	2,34	2,34
glibc-all-langpacks	2,34	2,34
glibc-common	2,34	2,34
glibc-gconv-extra	2,34	2,34
glibc-locale-source	2,34	2,34

Paket	AMI	KVM
gmp	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7
gnutls	3.8.0	3.8.0
go-srpm-macros	3.2.0	3.2.0
gpgme	1.15.1	1.15.1
gpm-libs	1.20.7	1,20,7
grep	3.8	3.8
groff-base	1,22,4	1.22,4
grub2-common	2,06	2,06
grub2-efi-aa64-ec2	2,06 (aarch64)	2,06 (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

Paket	AMI	KVM
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	0,384	0,384
info	6.7	6.7
inih	49	49
initscripts	10,09	10,09
iproute	6.10.0	6.10.0
iputils	20210202	20210202
irqbalance	1.9.0	1.9.0
jansson	2.14	2.14
jemalloc	5.2.1	5.2.1
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	6.1,112	6.1.112
kernel-libbpf	6.1.112	6.1.112

Paket	AMI	KVM
kernel-livepatch-repo-cdn		2023,6,20241031
kernel-livepatch-repo-s3	2023,6.20241031	
kernel-modules-extra		6.1,112
kernel-modules-extra-common		6.1.112
kernel-srpm-macros	1,0	1,0
kernel-tools	6.1.112	6.1.112
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	1.21.3	1.21.3
less	608	608
libacl	2.3.1	2.3.1
libaio	0,3,111	0,3.111
libarchive	3.7.4	3.7.4
libargon2	20171227	20171227
libassuan	2,5.5	2.5.5

Paket	AMI	KVM
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
libffi	3.4.4	3.4.4

Paket	AMI	KVM
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	1,59,0	1,59,0
libnl3	3.5.0	3.5.0
libpath_utils	0,2.1	0.2.1

Paket	AMI	KVM
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
libsss_nss_idmap	2.9.4	2.9.4

Paket	AMI	KVM
libsss_sudo	2.9.4	
libstdc++	11.4.1	11.4.1
libstoragemgmt	1.9.4	1.9.4
libtalloc	2.3.4	
libtasn1	4.19.0	4.19,0
libtdb	1.4.7	
libtevent	0.13.0	
libtextstyle	0,21	0,21
libtirpc	1.3.3	1.3.3
libunistring	0,9,10	0.9,10
libuser	0,63	0,63
libutempter	1.2.1	1.2.1
libuuid	2,37,4	2,37,4
libuv	1.47.0	1.47.0
libverto	0.3.2	0.3.2
libverto-libev	0.3.2	0.3.2
libxcrypt	4.4.33	4.4.33
libxml2	2.10,4	2.10.4
libyaml	0.2.5	0,2,5
libzstd	1.5.5	1.5.5

Paket	AMI	KVM
linux-firmware-whence	20210208 (Noarch)	20210208 (Noarch)
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 (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

Paket	AMI	KVM
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	
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	KVM
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	KVM
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	KVM
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	KVM
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	KVM
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	KVM
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	KVM
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	KVM
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	38,1,45	38,1,45
selinux-policy-targeted	38,1,45	38,1,45
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

Paket	AMI	KVM
sudo	1.9,15	1.9,15
sysctl-defaults	1,0	1,0
sysstat	12,5.6	12.5.6
systemd	252,23	252,23
systemd-libs	252,23	252,23
systemd-networkd	252,23	252,23
systemd-pam	252,23	252,23
systemd-resolved	252,23	252,23
systemd-udev	252,23	252,23
system-release	2023,6,20241031	2023,6.20241031
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	KVM
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	KVM
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 AL2 023 VMware OVA-Image

Ein Vergleich der RPMs Gegenwart auf dem AL2 023-Standard-AMI mit der RPMs Gegenwart auf dem AL2 VMware 023-OVA-Bild.

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	2.5.1	
amazon-linux-onprem		1.2
amazon-linux-repo-cdn		2023,6.20241031
amazon-linux-repo-s3	2023,6.20241031	
amazon-linux-sb-keys	2023,1	2023,1
amazon-onprem-network		1.2

Paket	AMI	VMware OVA
amazon-rpm-config	228	228
amazon-ssm-agent	3,3,987,0	3,3,987,0
amd-ucode-firmware	20210208	20210208
at	3.1,23	3.1.23
attr	2.5.1	2.5.1
audit	3.0.6	3.0.6
audit-libs	3.0.6	3.0.6
aws-cfn-bootstrap	2.0	
awscli-2	2,15,30	2,15,30
basesystem	11	11
bash	5.2,15	5.2.15
bash-completion	2.11	2.11
bc	1,07,1	1,07,1
bind-libs	9,18,28	9,18,28
bind-license	9,18,28	9,18,28
bind-utils	9,18,28	9,18,28
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

Paket	AMI	VMware OVA
bzip2	1.0.8	1.0.8
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023,2,68	2023,2,68
c-ares	1.19.1	
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
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

Paket	AMI	VMware OVA
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
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

Paket	AMI	VMware OVA
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	
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

Paket	AMI	VMware OVA
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
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

Paket	AMI	VMware OVA
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
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

Paket	AMI	VMware OVA
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	0,384	0,384
info	6.7	6.7
inih	49	49
initscripts	10,09	10,09
iproute	6.10.0	6.10.0
iputils	20210202	20210202
irqbalance	1.9.0	1.9.0
jansson	2.14	2.14
jemalloc	5.2.1	5.2.1
jitterentropy	3.4.1	3.4.1
jq	1.7.1	

Paket	AMI	VMware OVA
json-c	0,14	0,14
kbd	2.4.0	2.4.0
kbd-misc	2.4.0	2.4.0
kernel	6.1,112	6.1.112
kernel-libbpf	6.1.112	6.1.112
kernel-livepatch-repo-cdn		2023,6,20241031
kernel-livepatch-repo-s3	2023,6.20241031	
kernel-modules-extra		6.1,112
kernel-modules-extra-common		6.1.112
kernel-srpm-macros	1,0	1,0
kernel-tools	6.1.112	6.1.112
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	1.21.3	1.21.3
less	608	608

Paket	AMI	VMware OVA
libacl	2.3.1	2.3.1
libaio	0,3,111	0,3.111
libarchive	3.7.4	3.7.4
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
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

Paket	AMI	VMware OVA
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
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

Paket	AMI	VMware OVA
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
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

Paket	AMI	VMware OVA
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	
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

Paket	AMI	VMware OVA
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
linux-firmware-whence	20210208	20210208
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

Paket	AMI	VMware OVA
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
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

Paket	AMI	VMware OVA
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
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

Paket	AMI	VMware OVA
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
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

Paket	AMI	VMware OVA
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
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

Paket	AMI	VMware OVA
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
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

Paket	AMI	VMware OVA
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
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

Paket	AMI	VMware OVA
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
python3-policycore utils	3.4	3.4
python3-prettytable	0.7.2	0.7.2
python3-prompt-too lkit	3.0,24	3,0,24
python3-pycparser	2,20	2,20
python3-pyrsistent	0,17,3	0,17,3

Paket	AMI	VMware OVA
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
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

Paket	AMI	VMware OVA
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
sed	4.8	4,8
selinux-policy	38,1,45	38,1,45
selinux-policy-targeted	38,1,45	38,1,45
setup	2,13,7	2.13,7
shadow-utils	4,9 bis 4,9	4,9 bis 4,9

Paket	AMI	VMware OVA
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	252,23	252,23
systemd-libs	252,23	252,23
systemd-networkd	252,23	252,23
systemd-pam	252,23	252,23
systemd-resolved	252,23	252,23
systemd-udev	252,23	252,23
system-release	2023,6,20241031	2023,6.20241031
systemtap-runtime	4.8	4,8
tar	1,34	1,34
tbb	2020,3	2020,3

Paket	AMI	VMware OVA
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
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

Paket	AMI	VMware OVA
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 AL2 023 Hyper-V-Image

Ein Vergleich der RPMs Gegenwart auf dem AL2 023-Standard-AMI mit der RPMs Gegenwart auf dem AL2 023-Hyper-V-Bild.

Paket	AMI	Hyper-V VHDX
acl	2.3.1	2.3.1
acpid	2.0.32	

Paket	AMI	Hyper-V VHDX
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
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

Paket	AMI	Hyper-V VHDX
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
cloud-utils-growpart	0,31	0,31
coreutils	8,32	8,32

Paket	AMI	Hyper-V VHDX
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
diffutils	3.8	3.8
dnf	4.14.0	4.14.0

Paket	AMI	Hyper-V VHDX
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	
ec2-utils	2.2.0	
ed	1.14.2	1.14.2

Paket	AMI	Hyper-V VHDX
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
fuse-libs	2.9.9	2.9.9
gawk	5.1.0	5.1.0
gdbm-libs	1,19	1,19

Paket	AMI	Hyper-V VHDX
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
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

Paket	AMI	Hyper-V VHDX
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
hyperv-daemons		0
hyperv-daemons-license		0
hypervfcopyd		0
hypervkvpd		0
hyperv-tools		0
hypervvssd		0

Paket	AMI	Hyper-V VHDX
info	6.7	6.7
inih	49	49
initscripts	10,09	10,09
iproute	5.10.0	5.10.0
iputils	20210202	20210202
irqbalance	1.9.0	1.9.0
jansson	2.14	2.14
jitterentropy	3.4.1	3.4.1
jq	1.7.1	1.7.1
json-c	0,14	0,14
kbd	2.4.0	2.4.0
kbd-misc	2.4.0	2.4.0
kernel	6.1,79	6.1,79
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

Paket	AMI	Hyper-V VHDX
kernel-tools	6.1,79	6.1,79
keyutils	1.6.3	1.6.3
keyutils-libs	1.6.3	1.6.3
kmod	29	29
kmod-libs	29	29
kpatch-runtime	0.9.7	0.9.7
krb5-libs	1,21	1,21
less	608	608
libacl	2.3.1	2.3.1
libaio	0,3,111	0,3.111
libarchive	3,5.3	3.5.3
libargon2	20171227	20171227
libassuan	2,5.5	2.5.5
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

Paket	AMI	Hyper-V VHDX
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
libgpg-error	1,42	1,42
libibverbs	48,0	48,0
libidn2	2.3.2	2.3.2

Paket	AMI	Hyper-V VHDX
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
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	Hyper-V VHDX
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	Hyper-V VHDX
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
lua-libs	5.4.4	5.4.4
lua-srpm-macros	1	1
lz4-libs	1.9.4	1.9.4

Paket	AMI	Hyper-V VHDX
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
nss-util	3,90,0	3,90,0
ntsysv	1.15	1.15

Paket	AMI	Hyper-V VHDX
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
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

Paket	AMI	Hyper-V VHDX
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
perl-if	0,60,800	0,60,800
perl-interpreter	5,32,1	5,32,1
perl-IO	1,43	1,43

Paket	AMI	Hyper-V VHDX
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
perl-srpm-macros	1	1
perl-Storable	3,21	3,21
perl-subs	1,03	1,03

Paket	AMI	Hyper-V VHDX
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
publicsuffix-list-dafsa	20240212	20240212
python3	3,9,16	3.9,16

Paket	AMI	Hyper-V VHDX
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
python3-hawkey	0,69,0	0,69,0
python3-idna	(2.10)	(2.10)
python3-jinja2	2.11,3	2.11.3

Paket	AMI	Hyper-V VHDX
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
python3-policycore utils	3.4	3.4
python3-prettytable	0.7.2	0.7.2

Paket	AMI	Hyper-V VHDX
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
python3-urllib3	1,25,10	1,25,10
python3-wcwidth	0,2,5	0,2,5

Paket	AMI	Hyper-V VHDX
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
sed	4.8	4,8
selinux-policy	37,22	37,22

Paket	AMI	Hyper-V VHDX
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
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

Paket	AMI	Hyper-V VHDX
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
wget	1,21,3	1.21.3
which	2,21	2,21

Paket	AMI	Hyper-V VHDX
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

Identifizieren von Amazon Linux-Instances und -Versionen

Es kann wichtig sein, feststellen zu können, um welche Linux-Distribution es sich handelt und um welche Version dieser Distribution es sich bei einem Betriebssystem-Image oder einer Instance handelt. Amazon Linux bietet Mechanismen, um Amazon Linux von anderen Linux-Distributionen zu unterscheiden und um zu ermitteln, um welche Version von Amazon Linux es sich bei dem Image handelt.

In diesem Abschnitt werden die verschiedenen Methoden, die verwendet werden können, und ihre Einschränkungen sowie einige Anwendungsbeispiele behandelt.

Themen

- Unter Verwendung des os-release Standards
- Amazon Linux-spezifisch
- Beispielcode für die Betriebssystemerkennung

Unter Verwendung des `os-release` Standards

Amazon Linux entspricht dem `os-release`Standard zur Identifizierung von Linux-Distributionen. Diese Datei enthält maschinenlesbare Informationen zur Betriebssystemidentifikation und Versionsinformationen.

> **Note**
>
> Der Standard schreibt vor, dass zuerst versucht `/etc/os-release` wird, analysiert zu werden, gefolgt von. `/usr/lib/os-release` Es sollte darauf geachtet werden, dass der Standard in Bezug auf Dateinamen und Pfade eingehalten wird.

Themen

- Die wichtigsten Unterschiede bei der Identifizierung
- Feldtypen: Maschinenlesbar oder menschenlesbar
- Beispiele für /etc/os-release
- Vergleich mit anderen Distributionen

Die wichtigsten Unterschiede bei der Identifizierung

`os-release`Das finden Sie unter`/etc/os-release`, und falls das nicht vorhanden ist, unter`/usr/lib/os-release`. Vollständige Informationen finden Sie in der `os-release`Norm.

Die zuverlässigste Methode, um festzustellen, ob auf einer Instance Amazon Linux ausgeführt wird, besteht darin, das `ID` Feld einzucheckenos-release.

Die zuverlässigste Methode, um zwischen Versionen zu unterscheiden, besteht darin, das `VERSION_ID` Feld wie folgt einzucheckenos-release:

- Amazon Linux AMI: `VERSION_ID` enthält eine datumsbasierte Version (z. B.) `2018.03`
- AL2: `VERSION_ID="2"`
- AL2023: `VERSION_ID="2023"`

> **Note**
>
> Denken Sie daran, dass `VERSION_ID` es sich um ein maschinenlesbares Feld handelt, das für den programmatischen Gebrauch bestimmt ist, während `PRETTY_NAME` es für die Anzeige durch Benutzer konzipiert ist. Weitere Informationen the section called “Feldtypen” zu Feldtypen finden Sie unter.

Feldtypen: Maschinenlesbar oder menschenlesbar

Die `/etc/os-release` Datei (oder `/usr/lib/os-release` falls `/etc/os-release` nicht vorhanden) enthält zwei Arten von Feldern: maschinenlesbare Felder für den programmatischen Gebrauch und menschenlesbare Felder, die Benutzern präsentiert werden sollen.

Maschinenlesbare Felder

Diese Felder verwenden standardisierte Formate und sind für die Verarbeitung durch Skripte, Paketmanager und andere automatisierte Tools vorgesehen. Sie enthalten nur Kleinbuchstaben, Zahlen und begrenzte Satzzeichen (Punkte, Unterstriche und Bindestriche).

- `ID`— Betriebssystem-ID. Amazon Linux verwendet es `amzn` in allen Versionen und unterscheidet es von anderen Distributionen wie Debian (`debian`), Ubuntu (`ubuntu`) oder Fedora () `fedora`
- `VERSION_ID`— Betriebssystemversion für programmatische Zwecke (z. B.) `2023`

- `ID_LIKE`— Durch Leerzeichen getrennte Liste verwandter Distributionen (z. B.) `fedora`
- `VERSION_CODENAME`— Geben Sie den Codenamen für Skripte frei (z. B.) `karoo`
- `VARIANT_ID`— Varianten-ID für programmatische Entscheidungen
- `BUILD_ID`— Build-ID für Systemabbilder
- `IMAGE_ID`— Image-ID für containerisierte Umgebungen
- `PLATFORM_ID`— Plattform-ID (z. B.) `platform:al2023`

Für Menschen lesbare Felder

Diese Felder sind für die Anzeige durch Benutzer vorgesehen und können Leerzeichen, Groß- und Kleinschreibung sowie beschreibenden Text enthalten. Sie sollten verwendet werden, wenn Betriebssysteminformationen in Benutzeroberflächen dargestellt werden.

- `NAME`— Name des Betriebssystems für die Anzeige (z. B.`Amazon Linux`)
- `PRETTY_NAME`— Vollständiger Betriebssystemname mit Version für die Anzeige (z. B.`Amazon Linux 2023.8.20250721`)
- `VERSION`— Versionsinformationen, die für die Benutzerpräsentation geeignet sind
- `VARIANT`— Name der Variante oder Edition für die Anzeige `Server Edition` (z. B.

Andere Informationsfelder

Diese Felder enthalten zusätzliche Metadaten zum Betriebssystem:

- `HOME_URL`— URL der Projekt-Homepage
- `DOCUMENTATION_URL`— URL der Dokumentation
- `SUPPORT_URL`— URL Support Support-Informationen
- `BUG_REPORT_URL`— URL zur Fehlerberichterstattung
- `VENDOR_NAME`— Name des Anbieters
- `VENDOR_URL`— URL des Anbieters
- `SUPPORT_END`— End-of-support Datum im YYYY-MM-DD Format
- `CPE_NAME`— Gemeinsamer Bezeichner für die Plattformaufzählung
- `ANSI_COLOR`— ANSI-Farbcode für die Terminalanzeige

Verwenden Sie beim Schreiben von Skripten oder Anwendungen, die Amazon Linux programmgesteuert identifizieren müssen, maschinenlesbare Felder wie und. `ID` `VERSION_ID` Wenn Sie Benutzern Betriebssysteminformationen anzeigen, verwenden Sie menschenlesbare Felder wie. `PRETTY_NAME`

Beispiele für **`/etc/os-release`**

Der `/etc/os-release` Dateiinhalt variiert zwischen den Amazon Linux-Versionen:

AL2023

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

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

AL2

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

```
NAME="Amazon Linux"
VERSION="2"
ID="amzn"
ID_LIKE="centos rhel fedora"
VERSION_ID="2"
PRETTY_NAME="Amazon Linux 2"
ANSI_COLOR="0;33"
```

```
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2"
HOME_URL="https://amazonlinux.com/"
SUPPORT_END="2026-06-30"
```

Amazon Linux AMI

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

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

Vergleich mit anderen Distributionen

Um zu verstehen, wie Amazon Linux in das breitere Linux-Ökosystem passt, vergleichen Sie sein `/etc/os-release` Format mit anderen wichtigen Distributionen:

Fedora

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

```
NAME="Fedora Linux"
VERSION="42 (Container Image)"
RELEASE_TYPE=stable
ID=fedora
VERSION_ID=42
VERSION_CODENAME=""
PLATFORM_ID="platform:f42"
PRETTY_NAME="Fedora Linux 42 (Container Image)"
ANSI_COLOR="0;38;2;60;110;180"
LOGO=fedora-logo-icon
CPE_NAME="cpe:/o:fedoraproject:fedora:42"
DEFAULT_HOSTNAME="fedora"
HOME_URL="https://fedoraproject.org/"
```

```
DOCUMENTATION_URL="https://docs.fedoraproject.org/en-US/fedora/f42/system-
administrators-guide/"
SUPPORT_URL="https://ask.fedoraproject.org/"
BUG_REPORT_URL="https://bugzilla.redhat.com/"
REDHAT_BUGZILLA_PRODUCT="Fedora"
REDHAT_BUGZILLA_PRODUCT_VERSION=42
REDHAT_SUPPORT_PRODUCT="Fedora"
REDHAT_SUPPORT_PRODUCT_VERSION=42
SUPPORT_END=2026-05-13
VARIANT="Container Image"
VARIANT_ID=container
```

Debian

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

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

Ubuntu

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

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

```
UBUNTU_CODENAME=noble
LOGO=ubuntu-logo
```

Beachten Sie, dass die maschinenlesbaren Felder für eine einheitliche Identifizierung in allen Distributionen sorgen:

- `ID`— Identifiziert das Betriebssystem eindeutig: `amzn` für Amazon Linux, `fedora` für Fedora, `debian` für Debian, `ubuntu` für Ubuntu
- `ID_LIKE`— Zeigt Distributionsbeziehungen: Amazon Linux verwendet `fedora` (AL2023) oder `centos rhel fedora` (AL2), während Ubuntu zeigt, `debian` um auf seine Debian-Herkunft hinzuweisen
- `VERSION_ID`— Stellt maschinenlesbare Versionsinformationen bereit: `2023` für AL2 023, für Fedora, `42` für Debian, für Ubuntu `12` `24.04`

Im Gegensatz dazu sind die menschenlesbaren Felder so konzipiert, dass sie Benutzern angezeigt werden:

- `NAME`— Benutzerfreundlicher Betriebssystemname:`Amazon Linux`,,, `Fedora Linux` `Debian GNU/Linux` `Ubuntu`
- `PRETTY_NAME`— Vollständiger Anzeigename mit Version:`Amazon Linux 2023.8.20250721`,`Fedora Linux 42 (Container Image)`,`Debian GNU/Linux 12 (bookworm)`, `Ubuntu 24.04.2 LTS`
- `VERSION`— Für Menschen lesbare Version mit zusätzlichem Kontext wie Codenamen oder Releasetypen

Verwenden Sie beim Schreiben plattformübergreifender Skripts immer die maschinenlesbaren Felder (`ID`,`VERSION_ID`,`ID_LIKE`) für Logik und Entscheidungen und die menschenlesbaren Felder (`PRETTY_NAME`,) nur zur Anzeige von Informationen für Benutzer. `NAME`

Amazon Linux-spezifisch

Es gibt einige Amazon Linux-spezifische Dateien, anhand derer Amazon Linux und dessen Version identifiziert werden können. Neuer Code sollte den `/etc/os-release` Standard verwenden, um vertriebsübergreifend kompatibel zu sein. Von der Verwendung von Amazon Linux-spezifischen Dateien wird abgeraten.

Themen

- Die Datei /etc/system-release
- Image-Identifikationsdatei
- Beispiele für Amazon Linux-spezifische Dateien

Die Datei `/etc/system-release`

Amazon Linux enthält eine `/etc/system-release`-Datei, in der die aktuell installierte Version angegeben ist. Diese Datei wird mithilfe von Paketmanagern aktualisiert und ist unter Amazon Linux Teil des `system-release` Pakets. Während einige andere Distributionen wie Fedora diese Datei ebenfalls haben, ist sie in Debian-basierten Distributionen wie Ubuntu nicht vorhanden.

> **Note**
>
> Die `/etc/system-release` Datei enthält eine für Menschen lesbare Zeichenfolge und sollte nicht programmgesteuert zur Identifizierung eines Betriebssystems oder einer Version verwendet werden. Verwenden Sie stattdessen die maschinenlesbaren Felder in `/etc/os-release` (oder `/usr/lib/os-release` falls nicht `/etc/os-release` vorhanden).

Amazon Linux enthält auch eine maschinenlesbare Version davon`/etc/system-release`, die der Common Platform Enumeration (CPE) -Spezifikation in der Datei entspricht. `/etc/system-release-cpe`

Image-Identifikationsdatei

Jedes Amazon Linux-Image enthält eine eindeutige `/etc/image-id` Datei, die zusätzliche Informationen über das vom Amazon Linux-Team generierte Original-Image enthält. Diese Datei ist spezifisch für Amazon Linux und befindet sich nicht in anderen Linux-Distributionen wie Debian, Ubuntu oder Fedora. Diese Datei enthält die folgenden Informationen über das Image:

- `image_name`,`image_version`, `image_arch` — Werte aus dem Build-Rezept, das zur Erstellung des Images verwendet wurde.
- `image_stamp` – Ein eindeutiger, beliebiger Hexadezimalwert, der bei der Erstellung des Images generiert wurde.
- `image_date`— Die UTC-Zeit der Image-Erstellung im YYYYMMDDhhmmssFormat.

- `recipe_name`, `recipe_id` — Der Name und die ID des Build-Rezepts, das zur Erstellung des Images verwendet wurde.

Beispiele für Amazon Linux-spezifische Dateien

Die folgenden Abschnitte enthalten Beispiele für die Amazon Linux-spezifischen Identifikationsdateien für jede Hauptversion von Amazon Linux.

> **Note**
>
> `/usr/lib/os-release`Sollte in jedem realen Code verwendet werden, wenn die `/etc/os-release` Datei nicht existiert.

AL2023

Die folgenden Beispiele zeigen die Identifikationsdateien für AL2 023.

Beispiel `/etc/image-id` für AL2 023:

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

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

Beispiel `/etc/system-release` für für AL2 023:

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

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

AL2

Die folgenden Beispiele zeigen die Identifikationsdateien für AL2.

Beispiel `/etc/image-id` für AL2:

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

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

Beispiel `/etc/system-release` für AL2:

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

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

Amazon Linux-AMI

Die folgenden Beispiele zeigen die Identifikationsdateien für Amazon Linux AMI.

Beispiel `/etc/image-id` für Amazon Linux AMI:

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

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

Beispiel `/etc/system-release` für Amazon Linux AMI:

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

```
Amazon Linux AMI release 2018.03
```

Beispielcode für die Betriebssystemerkennung

Die folgenden Beispiele zeigen, wie das Betriebssystem und die Version mithilfe der Datei `/etc/os-release` (oder `/usr/lib/os-release` falls `/etc/os-release` nicht vorhanden) programmgesteuert erkannt werden. Diese Beispiele zeigen, wie man zwischen Amazon Linux und anderen Distributionen unterscheidet und wie das `ID_LIKE` Feld verwendet wird, um Distributionsfamilien zu bestimmen.

Das folgende Skript ist in verschiedenen Programmiersprachen implementiert, und jede Implementierung erzeugt dieselbe Ausgabe.

Shell

```
#!/bin/bash

# Function to get a specific field from os-release file
get_os_release_field() {
    local field="$1"
    local os_release_file

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

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

```
    )
}

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

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

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

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

# Function to check if this is like Fedora (includes Amazon Linux, CentOS, RHEL,
 etc.)
is_like_fedora() {
    local id="$(get_os_release_field ID)"
    local id_like="$(get_os_release_field ID_LIKE)"
    [ "$id" = "fedora" ] || [[ "$id_like" == *"fedora"* ]]
}

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

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

echo "Operating System Detection Results:"
echo "==================================="
echo "Is Amazon Linux: $(is_amazon_linux && echo YES || echo NO)"
echo "Is Fedora: $(is_fedora && echo YES || echo NO)"
```

```
echo "Is Ubuntu: $(is_ubuntu && echo YES || echo NO)"
echo "Is Debian: $(is_debian && echo YES || echo NO)"
echo "Is like Fedora: $(is_like_fedora && echo YES || echo NO)"
echo "Is like Debian: $(is_like_debian && echo YES || echo NO)"
echo
echo "Detailed OS Information:"
echo "======================="
echo "ID: $ID"
echo "VERSION_ID: $VERSION_ID"
echo "PRETTY_NAME: $PRETTY_NAME"
[ -n "$ID_LIKE" ] && echo "ID_LIKE: $ID_LIKE"

# Amazon Linux specific information
if is_amazon_linux; then
    echo ""
    echo "Amazon Linux Version Details:"
    echo "============================"
    case "$VERSION_ID" in
        2018.03)
            echo "Amazon Linux AMI (version 1)"
            ;;
        2)
            echo "Amazon Linux 2"
            ;;
        2023)
            echo "Amazon Linux 2023"
            ;;
        *)
            echo "Unknown Amazon Linux version: $VERSION_ID"
            ;;
    esac

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

Python 3.7-3.9

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

import os
import sys
```

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

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

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

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

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

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

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

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

```
    return 'fedora' in id_like

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

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

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

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

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

```
            print("Amazon Linux 2023")
        else:
            print(f"Unknown Amazon Linux version: {version_id}")

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

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

Python 3.10+

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

import os
import sys
import platform

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

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

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

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

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

def is_like_debian(os_data):
```

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

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

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

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

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

```
        elif version_id == '2023':
            print("Amazon Linux 2023")
        else:
            print(f"Unknown Amazon Linux version: {version_id}")

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

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

Perl

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

use strict;
use warnings;

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

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

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

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

```
        }
    }

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

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

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

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

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

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

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

```
    return $id_like =~ /debian/;
}

# Main execution
my %os_data = parse_os_release();

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

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

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

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

    # Check for Amazon Linux specific files
    if (-f '/etc/image-id') {
```

```
        print "Amazon Linux image-id file present\n";
    }
}
```

Wenn das Skript auf verschiedenen Systemen ausgeführt wird, erzeugt es die folgende Ausgabe:

AL2023

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

Detailed OS Information:
========================
ID: amzn
VERSION_ID: 2023
PRETTY_NAME: Amazon Linux 2023.8.20250721
ID_LIKE: fedora

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

AL2

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

Detailed OS Information:
========================
```

```
ID: amzn
VERSION_ID: 2
PRETTY_NAME: Amazon Linux 2
ID_LIKE: centos rhel fedora

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

Amazon Linux AMI

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

Detailed OS Information:
========================
ID: amzn
VERSION_ID: 2018.03
PRETTY_NAME: Amazon Linux AMI 2018.03
ID_LIKE: rhel fedora

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

Ubuntu

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

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

Debian

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

Detailed OS Information:
========================
ID: debian
VERSION_ID: 12
PRETTY_NAME: Debian GNU/Linux 12 (bookworm)
```

Fedora

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

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

Dateisystem-Layout

Dieser Abschnitt behandelt das Dateisystem-Layout eines AL2 023-Systems, einschließlich Details, die für Instanzen oder AL2 023-basierte Container spezifisch sein können. Weitere Informationen finden Sie im `file-hierarchy(7)` man Seite für weitere Informationen.

Themen

- /(Das Stammverzeichnis)
- /boot(Kernel, initramfsusw.)
- /etc(Systemkonfiguration)
- /home(Basisverzeichnisse der Benutzer)
- /root (root (Heimatverzeichnis des Benutzers)
- /srv(Nutzlast des Servers)
- /tmp(kleine temporäre Dateien)
- /run(Laufzeitdaten)
- /usr(Systemressourcen)
- /var(Persistente variable Systemdaten)

/(Das Stammverzeichnis)

Standardmäßig sind AL2 023-Bilder mit einem beschreibbaren Wert konfiguriert`/`, sodass privilegierte Benutzer neue Dateien und Verzeichnisse erstellen können.

Es ist möglich, `systemd` Dienste so zu konfigurieren, dass sie einen anderen Pfad oder ein anderes Bild verwenden, um als `/` für diesen Dienst angezeigt zu werden, sowie Zugriffsbeschränkungen für jeden Pfad festzulegen.

> **Note**
>
> Es hat sich bewährt, `systemd` Dienste so zu konfigurieren, dass eingeschränkt wird, worauf der Dienst Zugriff hat. Dies kann die Verwendung der `ReadOnlyPaths=/` Direktive beinhalten, die für diesen Dienst nur `/` Lesezugriff gewährt.
> Weitere Informationen dazu`systemd`, wie Sie den Zugriff eines Dienstes auf das System einschränken können, finden Sie in der `systemd.exec(5)` man Seite.

`/boot`(Kernel, initramfsusw.)

Standardmäßig sind bootfähige AL2 023-Images so konfiguriert, dass sie sich auf dem `/boot` root Dateisystem. Der `/boot` Pfad ist nur für bootfähige Images relevant und wird daher in AL2 023-Container-Images nicht verwendet.

In diesem Verzeichnis befinden sich Dateien, die zum Booten von AL2 023 benötigt werden, z. B. der Linux-Kernel und initramfs. Der Inhalt dieses Verzeichnisses sollte nur mit den im Betriebssystem enthaltenen Tools bearbeitet werden.

`/boot/efi`(EFI-Systempartition)

Standardmäßig werden bootfähige AL2 023-Images mit dem konfiguriert EFI System Partition, auf der gemountet wird. `/boot/efi` Dieses Dateisystem wird vom Betriebssystem verwaltet und enthält Code und Konfiguration, die für das Booten des Systems entscheidend sind.

Dieser Pfad ist für Container-Images nicht relevant.

`/etc`(Systemkonfiguration)

Das `/etc` Verzeichnis auf AL2 023 enthält eine systemspezifische Konfiguration. Standardmäßig befinden sich AL2 023-Bilder auf der `/etc` root Dateisystem und beschreibbar für privilegierte Benutzer.

> **Note**
>
> Es ist üblich, dass Anwendungen (einschließlich`systemd`) die Standardkonfiguration beibehalten`/usr`(Systemressourcen), die durch Einfügen der Konfiguration außer Kraft gesetzt werden kann. `/etc`(Systemkonfiguration)
> Bei diesen Anwendungen führt das Ändern von Dateien in, `/usr`(Systemressourcen) anstatt die Standardkonfiguration zu überschreiben, wahrscheinlich `/etc` dazu, dass die Änderungen bei der Aktualisierung des Pakets außer Kraft gesetzt werden.

`/home`(Basisverzeichnisse der Benutzer)

Normale Benutzer haben ihre Home-Verzeichnisse unter`/home`, aber Software sollte immer nach der benutzerspezifischen `$HOME` Umgebungsvariablen suchen, anstatt sich auf ein Muster wie. `/home/$USER`

Standardmäßig haben AL2 023 Bilder auf /home root Dateisystem, aber Software sollte sich nicht darauf verlassen. Es ist absolut legitim, wenn das Betriebssystem als separates Dateisystem konfiguriert wird, das später beim Booten oder erst, nachdem sich ein Benutzer beim System authentifiziert hat, eingehängt wird. /home

Das Tool root Das Home-Verzeichnis des Benutzers befindet sich nicht im Verzeichnis, /home sondern ist /root (root (Heimatverzeichnis des Benutzers) so, dass es verfügbar ist, falls das /home Dateisystem nicht gemountet werden kann.

> **Note**
>
> Es empfiehlt sich, systemd Dienste, die keinen Schreibzugriff benötigen, mit der ProtectHome=read-only Direktive /home zu konfigurieren. Mit dieser Option /run/user sind /home/root, und für den Dienst schreibgeschützt.
> Es hat sich auch bewährt, Dienste, die keinen Zugriff benötigen, mit der ProtectHome=tmpfs Direktive /home zu konfigurieren. Dadurch wird der Dienst in einer Sandbox ausgeführt, in der es sich bei /home/root, und um leere schreibgeschützte /run/user tmpfs Dateisysteme handelt.
> Weitere Informationen dazu, wie Sie systemd den Zugriff eines Dienstes auf das System einschränken können, finden Sie in der systemd.exec(5) man Seite.

/root (root (Heimatverzeichnis des Benutzers)

Das Home-Verzeichnis der root Benutzer ist das /root Verzeichnis, das absichtlich getrennt ist, / home(Basisverzeichnisse der Benutzer) sodass es vorhanden ist, falls es sich auf einem Dateisystem /home(Basisverzeichnisse der Benutzer) befindet, das nicht verfügbar ist.

Die bewährte Methode für die Konfiguration von systemd Diensten ist dieselbe für /root wie für. / home(Basisverzeichnisse der Benutzer)

/srv(Nutzlast des Servers)

Das /srv Verzeichnis wird vom Administrator des Systems verwaltet und Amazon Linux 2023 schränkt die Organisation dieses Verzeichnisses nicht ein.

Es ist möglich, das /srv Verzeichnis so zu konfigurieren, dass es sich in einem separaten Dateisystem befindet, sodass es möglicherweise erst später beim Systemstart verfügbar ist.

`/tmp`(kleine temporäre Dateien)

> **Note**
>
> Amazon Linux 2023 unterscheidet sich von Amazon Linux 2, da `/tmp` es sich standardmäßig jetzt `tmpfs` nicht um einen Pfad auf dem root Dateisystem.

> **Note**
>
> Bei der Ausführung in einem Container bestimmt in der Regel Ihre Container-Laufzeitkonfiguration`tmpfs`, ob es `/tmp` sich um einen Pfad auf der Festplatte handelt und ob ein Bereinigungsprozess läuft oder nicht.

Das `/tmp` Verzeichnis ist für kleine temporäre Dateien mit begrenzter Größe vorgesehen. Standardmäßig konfiguriert AL2 023 es als `tmpfs` Dateisystem mit einer Größenbeschränkung von 50% des RAM und maximal einer Million inodes.

Anwendungen sollten den Pfad in der `$TMPDIR` Umgebungsvariablen vorziehen. `/tmp` Benutzer können dann die `$TMPDIR` Umgebungsvariable so einstellen, dass sie den Pfad überschreibt, für den eine Anwendung verwenden soll `/tmp`

Für größere temporäre Dateien `/var/tmp` sollte stattdessen verwendet werden.

> **Warning**
>
> Da es gemeinsam genutzt `/tmp` wird, ist es wichtig, sichere Methoden zum Erstellen temporärer Dateien zu verwenden. Einzelheiten finden Sie in der `systemd` Upstream-Dokumentation unter Verwenden `/tmp` und `/var/tmp` Sicher.

> **Note**
>
> Es hat sich bewährt, `systemd` Dienste so zu konfigurieren, dass die `PrivateTmp=` Direktive auf `yes` oder`disconnected`, die den Dienst in einer Sandbox ausführt, gesetzt ist `/tmp` und `/var/tmp` nicht mit dem Host oder anderen Diensten gemeinsam genutzt wird.

Weitere Informationen, einschließlich der Konfiguration von zwei Diensten zur gemeinsamen Nutzung derselben privaten temporären Verzeichnisse, finden Sie im `systemd.exec(5)` man Seite.

Der Inhalt von `/tmp` wird normalerweise beim Systemstart gesäubert, und unbenutzte Dateien werden regelmäßig bereinigt. Standardmäßig wird der Bereinigungsprozess kurz nach dem Systemstart und dann täglich ausgeführt. Informationen zur Konfiguration der Bereinigung temporärer Dateien finden Sie unter und `tmpfiles.d(5)` `systemd-tmpfiles(8)` man Handbuchseiten.

Die `/var/tmp` Pfade `/tmp` und die Pfade sind eng miteinander verknüpft und dienen unterschiedlichen Zwecken.

`/run`(Laufzeitdaten)

Das `/run` Verzeichnis wird von Systempaketen verwendet, um kleine Mengen von Laufzeitdaten (wie Socket-Dateien) zu speichern. Es ist ein `tmpfs` Dateisystem und kann nur von privilegierten Programmen geschrieben werden.

Das `/run/log` Verzeichnis kann von Systemkomponenten verwendet werden, um Logs zu speichern, entweder bevor sie in das Dateisystem geschrieben werden `/var/log` oder bevor das `/var/log` Dateisystem verfügbar ist.

Der `/run/user/` Pfad enthält Laufzeitverzeichnisse pro Benutzer. Standardmäßig handelt es sich dabei um einzelne `tmpfs` Dateisysteme, die `systemd` bei der Anmeldung des Benutzers eingehängt und gelöscht werden, wenn der Benutzer nicht mehr angemeldet ist. Gemäß der XDG-Basisverzeichnisspezifikation sollten diese Pfade nicht direkt, sondern über die `$XDG_RUNTIME_DIR` Umgebungsvariable referenziert werden.

`/usr`(Systemressourcen)

Die `/usr` Hierarchie bezieht sich auf vom Hersteller bereitgestellte Betriebssystemressourcen. Mit Ausnahme der `/usr/local` Hierarchie sollte nichts verändert werden, `/usr` außer dem Betriebssystem-Paketmanager.

Softwareanwendungen müssen davon ausgehen, `/usr` dass sie schreibgeschützt sind. Die `/usr` Hierarchie darf nicht für flüchtige Daten verwendet werden. Mit Ausnahme von: Die `/usr` Hierarchie

darf nicht für Daten verwendet werden/usr/local, die außerhalb der Installation oder Deinstallation von Paketen, wie sie vom Paketmanager des Betriebssystems vorgenommen wird, hinzugefügt oder geändert werden. Der Paketmanager des Betriebssystems kann davon ausgehen, dass die gesamte /usr Hierarchie (außer/usr/local) derselbe Mountpoint ist.

Software, die außerhalb des OS-Paketmanagers installiert wird, sollte keine Daten darin speichern, /usr da dies einen future Aufruf des OS-Paketmanagers behindern kann. Die /usr/local Hierarchie ist die Ausnahme und ist für Software außerhalb des OS-Paketmanagers reserviert.

/usr/bin(Ausführbare Dateien)

Ausführbare Dateien, die in der Standardsuche erscheinen sollten und nützlich sind$PATH, um sie von einer Shell aus aufzurufen. Daemons und ausführbare Dateien, deren Aufruf von einer Shell aus nicht sinnvoll ist, leben stattdessen in oder. /usr/lib /usr/libexec

/usr/include(C/C++-Header)

Das /usr/include Verzeichnis enthält C- und C++-Header-Dateien, die normalerweise in Paketen mit dem -devel Suffix enthalten sind.

/usr/libund **/usr/lib64** (Gemeinsam genutzte Bibliotheken)

Unter Amazon Linux 2023 wird der /usr/lib64 Pfad für gemeinsam genutzte 64-Bit-Bibliotheken und architekturabhängige Paketdaten verwendet. Da AL2 023 keine 32-Bit-Userspace-Unterstützung bietet, sind nur gemeinsam genutzte 64-Bit-Bibliotheken verfügbar.

Der /usr/lib Pfad bezieht sich auf statische Daten aus Betriebssystempaketen und ist mit allen Architekturen kompatibel. Dazu können auch ausführbare Dateien gehören, die normalerweise nicht von einer Shell aus aufgerufen werden. Sie können sich auch in befinden. /usr/libexec Gemeinsam genutzte Bibliotheken befinden sich eher in /usr/lib64 als. /usr/lib

/usr/local(Vom Systemadministrator installierte Software)

Auf Amazon Linux 2023 steht dem Systemadministrator der /usr/local Pfad zur Installation von Software zur Verfügung, in der Software installiert werden kann, die nicht dem Betriebssystem gehört und auch nicht vom Betriebssystem beeinflusst wird. Die /usr/local Standardhierarchie spiegelt die / Hierarchie wider.

`/usr/share`(Gemeinsam genutzte Ressourcen)

Gemeinsam genutzte Ressourcen wie Dokumentation, Schriften und Zeitzonendaten befinden sich in`/usr/share`. Es ist üblich, dass verschiedene Spezifikationen genau vorschreiben, wo und in welchem Format Daten in diesem Verzeichnis gespeichert werden.

`/usr/share/doc`(Dokumentation)

Die Dokumentation, die den Paketen beiliegt, wird in `/usr/share/doc` gespeichert.

`/var`(Persistente variable Systemdaten)

`/var/cache`(Cache)

Im Gegensatz zu `/var/lib` führt das Löschen von Daten nicht zu Datenverlust, da Anwendungen in der Lage sein müssen, ihre `/var/cache` Daten aus anderen Quellen neu aufzubauen. `/var/cache`

`/var/lib`(Persistente Systemdaten)

Das `/var/lib` Verzeichnis wird für persistente Systemdaten verwendet. Verschiedene Systemkomponenten platzieren hier Daten, die für diese Komponente privat sind. Im Gegensatz dazu `/var/cache` führt das Löschen von Daten zu Datenverlust. `/var/lib`

Beispielsweise speichert der PostgreSQL-Datenbankserver standardmäßig Datenbankdaten in. `/var/lib/pgsql` Das Layout und die Dateiformate dieser Daten sind für PostgreSQL privat, und es handelt sich um persistente Daten, als ob sie gelöscht würden, der Benutzer erleidet Datenverlust.

`/var/log`(Persistente Protokolle)

Dieses Verzeichnis wird zum Speichern persistenter Protokolle verwendet. Es wird empfohlen, dass Software die `sd_journal_print(3)` API-Aufrufe `syslog(3)` oder verwendet, anstatt Protokolldateien direkt unter zu speichern`/var/log`.

> **Note**
>
> In AL2 023 das`systemd`Journal ersetzt `rsyslog`, was einen deutlichen Unterschied zur Standardkonfiguration von Amazon Linux 2 darstellt.

Weitere Informationen zum Lesen von Protokollen mithilfe von `journalctl` finden Sie auf der `journalctl`Handbuchseite.

Viele Anwendungen verwenden ihre eigenen Mechanismen zum Schreiben und manchmal zum Rotieren von Protokolldateien in`/var/log`. Informationen zur Konfiguration ihrer Protokolldateien finden Sie in der Dokumentation zu diesen Anwendungen.

`/var/spool`(E-Mail- und Druckerwarteschlangen)

Dieses Verzeichnis wird für persistente Daten wie E-Mail- oder Druckerwarteschlangen verwendet.

`/var/tmp`(größere temporäre Dateien)

Für kleine, größengebundene temporäre Dateien `/tmp` sollte es möglicherweise stattdessen verwendet werden.

`/tmp`While ist standardmäßig als `tmpfs` Volume konfiguriert, `/var/tmp` ist aber standardmäßig als Pfad im Root-Dateisystem konfiguriert und ist somit der Ort für größere und beständigere temporäre Dateien. Standardmäßig wird nach einem regelmäßigen Zeitplan ein Bereinigungsauftrag ausgeführt, bei dem Dateien entfernt werden, auf die in letzter Zeit nicht zugegriffen wurde.

Informationen zur Konfiguration der Säuberung temporärer Dateien finden Sie unter und `tmpfiles.d(5)` `systemd-tmpfiles(8)` man Handbuchseiten.

Ebenso wie bei `/tmp` sollten Anwendungen den in der `$TMPDIR` Umgebungsvariablen angegebenen Pfad vorziehen`/var/tmp`. Benutzer können dann die `$TMPDIR` Umgebungsvariable so einstellen, dass sie den Pfad überschreibt, den eine Anwendung verwenden soll`/var/tmp`.

> **Warning**
>
> Da es gemeinsam genutzt `/var/tmp` wird (so wie es ist)`/tmp`, ist es wichtig, sichere Methoden zum Erstellen temporärer Dateien zu verwenden. Einzelheiten finden Sie in der `systemd` Upstream-Dokumentation unter Verwenden /tmp und /var/tmp Sicher.

> **Note**
>
> Es hat sich bewährt, `systemd` Dienste so zu konfigurieren, dass die `PrivateTmp=` Direktive auf `yes` oder`disconnected`, die den Dienst in einer Sandbox ausführt, gesetzt ist `/tmp` und `/var/tmp` nicht mit dem Host oder anderen Diensten gemeinsam genutzt wird.

Weitere Informationen, einschließlich der Konfiguration von zwei Diensten zur gemeinsamen Nutzung derselben privaten temporären Verzeichnisse, finden Sie im `systemd.exec(5)` man Seite.

Die `/var/tmp` Pfade `/tmp` und die Pfade sind eng miteinander verknüpft und dienen unterschiedlichen Zwecken.

Aktualisierung AL2 023

Es ist wichtig, sich über die AL2 023-Versionen auf dem Laufenden zu halten, damit Sie von Sicherheitsupdates und neuen Funktionen profitieren können. Mit AL2 023 können Sie die Konsistenz zwischen Paketversionen und Updates in Ihrer gesamten Umgebung sicherstellen. Deterministische Upgrades durch versionierte Repositorys auf 023 AL2

> **Warning**
>
> `dnf --releasever=latest update`Das Ausführen ist keine bewährte Methode und führt wahrscheinlich dazu, dass ein Betriebssystem-Update zuerst in der Produktion getestet wird. Verwenden Sie statt der Verwendung `latest` eine bestimmte AL2 023-Release-Version. Dadurch wird sichergestellt, dass Sie dieselben Änderungen in allen Produktionsinstanzen implementieren, die Sie zuvor getestet haben. `dnf --releasever=2023.8.20250721 update`Wird beispielsweise immer auf die Version 2023.8.20250721 aktualisiert. Weitere Informationen finden Sie im Abschnitt Aktualisierung von AL2 023 im 023-Benutzerhandbuch. AL2

Themen

- Bewährte Methoden für die sichere Bereitstellung von Updates
- Erhalten Sie Benachrichtigungen über neue Updates
- Deterministische Upgrades durch versionierte Repositorys auf 023 AL2
- Paket- und Betriebssystemupdates in AL2 023 verwalten
- Kernel Live Patching auf 023 AL2
- Aktualisierung des Linux-Kernels auf AL2 023

Bewährte Methoden für die sichere Bereitstellung von Updates

Amazon Linux 2023 (AL2023) verfügt über mehrere Funktionen, die dazu beitragen, Updates für das Betriebssystem sicher bereitzustellen und zu wissen, was sich zwischen den Updates geändert hat, und bei Bedarf einfach zur älteren Version zurückzukehren. In diesem Abschnitt werden Erkenntnisse AWS aus mehr als einem Jahrzehnt interner und externer Nutzung von Amazon Linux untersucht.

> **Warning**
>
> `dnf --releasever=latest update`Das Ausführen ist keine bewährte Methode und führt wahrscheinlich dazu, dass ein Betriebssystem-Update zuerst in der Produktion getestet wird. Verwenden Sie statt der Verwendung `latest` eine bestimmte AL2 023-Release-Version. Dadurch wird sichergestellt, dass Sie dieselben Änderungen in allen Produktionsinstanzen implementieren, die Sie zuvor getestet haben. `dnf --releasever=2023.8.20250721 update`Wird beispielsweise immer auf die Version 2023.8.20250721 aktualisiert. Weitere Informationen finden Sie im Abschnitt Aktualisierung von AL2 023 im 023-Benutzerhandbuch. AL2

Ohne die Sicherheit bei der Bereitstellung von Betriebssystemupdates zu berücksichtigen, können die Auswirkungen einer unerwarteten negativen Interaktion zwischen Ihnen application/service und einem Betriebssystem-Update erheblich größer sein, bis hin zu einem Totalausfall. Wie bei jedem Softwareproblem gilt: Je früher das Problem erkannt wird, desto weniger Auswirkungen kann es auf die Endbenutzer haben.

Es ist wichtig, nicht in die Falle zu tappen, zwei Dinge zu glauben, die grundsätzlich nicht wahr sind:

1. Der Betriebssystemhersteller wird bei einem Betriebssystem-Update niemals einen Fehler machen.
2. Das spezifische Verhalten oder die Schnittstelle zu dem Betriebssystem, auf das Sie sich verlassen, entspricht dem Verhalten und den Schnittstellen, auf die sich der Betriebssystemhersteller verlassen würde.

   d. h. sowohl der Betriebssystemhersteller als auch Sie würden zustimmen, dass es ein Problem mit dem Update gab.

Verlassen Sie sich nicht auf gute Absichten, sondern richten Sie Systeme ein, die sicherstellen, dass die Sicherheit bei der Bereitstellung jedes Updates des Betriebssystems gewährleistet ist.

Es wird nicht empfohlen, neue Betriebssystemupdates durch Bereitstellung in Produktionsumgebungen zu testen. Es empfiehlt sich, das Betriebssystem als weiteren Teil Ihrer Bereitstellung zu betrachten und darüber nachzudenken, dieselben Sicherheitsmechanismen für die Bereitstellung anzuwenden, die Sie für jede andere Änderung an einer Produktionsumgebung für geeignet halten.

Es empfiehlt sich, alle Betriebssystemupdates vor der Bereitstellung auf Produktionssystemen zu testen. Bei der Bereitstellung werden schrittweise Rollouts in Kombination mit einer guten Überwachung empfohlen. Schrittweise Rollouts können sicherstellen, dass bei Auftreten eines Problems, auch wenn es nicht unmittelbar auftritt, die Auswirkungen auf eine Teilmenge einer Flotte beschränkt sind und die weitere Bereitstellung des Updates gestoppt werden kann, während weitere Untersuchungen und Abhilfemaßnahmen getroffen werden können.

Die Minderung aller negativen Auswirkungen eines Betriebssystem-Updates hat oft oberste Priorität, gefolgt von der Lösung des Problems, wo auch immer es sich befinden mag. Wenn die Einführung eines Betriebssystemupdates mit negativen Auswirkungen einhergeht, ist die Möglichkeit, zur zweifelsfrei funktionierenden Vorgängerversion des Betriebssystems zurückzukehren, ein mächtiges Hilfsmittel.

Amazon Linux 2023 stellt eine leistungsstarke neue Funktion vorDeterministische Upgrades durch versionierte Repositorys, die sicherstellt, dass jede Änderung an der Version des Betriebssystems (oder einzelner Pakete) wiederholbar ist. Wenn also beim Wechsel von einer Betriebssystemversion zur nächsten ein Problem auftritt, stehen einfach zu verwendende Mechanismen zur Verfügung, um bei der bekanntermaßen funktionierenden Betriebssystemversion zu bleiben und gleichzeitig herauszufinden, wie das Problem gelöst werden kann.

Bei AL2 023 gibt es jedes Mal, wenn wir neue Paket-Updates veröffentlichen, eine neue Version, auf die man sich beschränken kann, und neue, die sich auf AMIs diese Version beschränken. Die Versionshinweise zu AL2 023 behandeln die Änderungen in den einzelnen Versionen und behandeln Sicherheitsprobleme, Amazon Linux-Sicherheitsempfehlungen für 023 AL2 die in Paket-Updates behoben wurden.

Wenn Sie beispielsweise von dem Problem in der Version 2023.6.20241028 betroffen wären, könnten Sie sofort wieder die Container-Images AMIs und die Container-Images der vorherigen Version 2023.6.20241010 verwenden. In diesem Fall gab es einen Fehler in einem Paket, der in der nachfolgenden Version 2023.6.20241031 behoben wurde. Jeder Betroffene konnte jedoch sofort einfache Maßnahmen zur Behebung ergreifen: Verwenden Sie einfach die vorherigen Images. Deterministische Upgrades durch versionierte Repositorys

Deterministische Upgrades durch versionierte Repositorysgewährleistet außerdem, dass jegliche Bereitstellung eines Betriebssystem-Updates, entweder direkt vor Ort oder durch das Starten neuer Images AMIs oder Container-Images, nicht durch später veröffentlichte Betriebssystemupdates beeinträchtigt wird.

In unserem ersten Beispiel handelt es sich bei Flotte A um eine große Flotte, die zur Hälfte mit der Bereitstellung des Updates von 2023.5.20241001 auf die Version 2023.6.20241010 fertig ist, wenn die Version 2023.6.20241028 veröffentlicht wird. Deterministische Upgrades durch versionierte Repositorysbedeutet, dass die Bereitstellung für Flotte A fortgesetzt wird, ohne dass sich daran ändert, welche Updates angewendet werden.

Der Zweck von wellen- oder phasenbasierten Implementierungsstrategien, wie z. B. der Einsatz zunächst auf 1% einer Flotte, dann auf 5%, 10%, 20%, 40%, bis 100% erreicht sind, besteht darin, eine Änderung in begrenztem Umfang testen zu können, bevor sie in größerem Umfang eingeführt wird. Diese Art von Implementierungsstrategie wird allgemein als bewährte Methode für die Implementierung beliebiger Produktionsänderungen angesehen.

Bei einer wellenbasierten Bereitstellungsstrategie und dem Flotten-A-Update auf 2023.6.20241010 befindet sich das Update in einem Stadium, in dem es auf vielen Hosts gleichzeitig bereitgestellt wird, hat die Tatsache, dass 2023.6.20241028 veröffentlicht wurde, keine Auswirkungen auf die laufende Bereitstellung, dank der Verwendung von. Deterministische Upgrades durch versionierte Repositorys

Wenn Flotte B eine ältere Version ausführen würde, beispielsweise 2023.5.20240708, und mit der Bereitstellung des Updates für 2023.6.20241028 begonnen hätte und Flotte B von dem Problem in dieser Version betroffen wäre, würde dies zu Beginn der Bereitstellung bemerkt werden. Zu diesem Zeitpunkt kann entschieden werden, ob der Rollout unterbrochen werden soll, bis eine Lösung für dieses Problem verfügbar ist, oder ob in der Zwischenzeit eine Bereitstellung derselben Version gestartet werden soll, die Flotte A ausgeführt hat, 2023.6.20241010, sodass Flotte B alle Updates zwischen 2023.5.20240708 und 2023.6.20241010 erhält.

Es ist wichtig zu beachten, dass es zu Problemen kommen kann, wenn Betriebssystemupdates nicht umgehend durchgeführt werden. Neue Updates enthalten wahrscheinlich Fehler- und Sicherheitskorrekturen, die für Ihre Umgebung relevant sein können. Weitere Informationen erhalten Sie unter Sicherheit und Compliance in Amazon Linux 2023 und Paket- und Betriebssystemupdates in AL2 023 verwalten.

Es ist wichtig, dass Sie Ihre Bereitstellungssysteme so konfigurieren, dass neue Betriebssystemupdates problemlos übernommen, vor der Bereitstellung in der Produktion getestet und Mechanismen wie wellenbasierte Bereitstellungen verwendet werden können, um negative Auswirkungen zu minimieren. Um negative Auswirkungen eines Betriebssystemupdates abzumildern, ist es wichtig zu wissen, wie Sie Ihre Bereitstellungssysteme auf eine frühere, zweifelsfrei funktionierende Version des Betriebssystems verweisen können. Sobald das Problem behoben ist, sollten Sie nicht mehr an die ältere, zweifelsfrei funktionierende Version gebunden sein, sondern zu einer neuen, zweifelsfrei funktionierenden Version wechseln.

Vorbereitung kleinerer Updates

Die Vorbereitung kleinerer Updates für das Betriebssystem, wie z. B. ein neues Point-Release von AL2 023, soll auf Null beschränkt werden. Lesen Sie unbedingt die Versionshinweise zu AL2 023, um sich über bevorstehende Änderungen zu informieren.

Der Supportzeitraum eines Pakets, das sich dem Ende zuneigt, kann die Umstellung auf eine neuere Version der Language Runtime (z. B. withPHP im Jahr AL2 023) beinhalten. Es empfiehlt sich, sich darauf im Voraus vorzubereiten, indem Sie bequem vor Ablauf des Supportzeitraums zu neuen Sprachversionen wechseln.

Bei Paketen wie gibt es auch die Möglichkeitpcre Version 1, im Voraus zu planen und Ihren gesamten Code auf den Ersatzcode, in diesem Fall `pcre` Version 2, zu migrieren. Es empfiehlt sich, dies so schnell wie möglich zu tun, um Zeit für etwaige Rückschläge einzuplanen.

Wenn es keinen direkten Ersatz gibt, z. B. durchDatenbank Berkeley () `libdb`, müssen Sie möglicherweise eine Entscheidung treffen, die auf Ihrem Anwendungsfall basiert.

Wir bereiten uns auf wichtige Updates vor

Die Aktualisierung auf eine neue Hauptversion eines Betriebssystems wird fast allgemein als etwas angesehen, das Planung, Anpassung an geänderte oder veraltete Funktionen sowie Tests vor der Bereitstellung erfordert. Es ist nicht ungewöhnlich, dass man sich schrittweise auf die nächste Hauptversion von Amazon Linux 2023 vorbereiten kann, indem beispielsweise jegliche Verwendung veralteter oder entfernter Funktionen behoben wird, bevor zur nächsten Hauptversion übergegangen wird.

Wenn Sie beispielsweise von Version 023 AL2 auf Version AL2 023 umsteigen, kann das Lesen des Funktionalität wurde in 023 veraltet AL2 und wurde in Version 023 entfernt AL2 Abschnitts zu einer Reihe sicherer und kleiner Schritte führen, die während der Vorbereitung auf Version 023 durchgeführt AL2 werden können. AL2 Beispielsweise kann jede Python 2.7 wurden durch Python 3 ersetzt Verwendung (außerhalb der Betriebssystemnutzung, z. B. im `yum` Paketmanager) zur Vorbereitung der Verwendung Python im Jahr AL2 023 auf Python 3 migriert werden. Wenn Sie PHP verwenden, werden sowohl AL2 (über PHP 8.2 AL2 Extra) als auch AL2 023 PHP 8.2 ausgeliefert, sodass sowohl die PHP-Versionsmigration als auch die Betriebssystemmigration nicht gleichzeitig erfolgen müssen.

Wenn Sie AL2 023 verwenden, können Sie sich heute auch auf die nächste Hauptversion von Amazon Linux 2023 vorbereiten, während Sie AL2 023 verwenden. Dieser In 023 veraltet AL2

Abschnitt behandelt Funktionen und Pakete, die in Version AL2 023 veraltet sind und demnächst entfernt werden.

Wenn Sie beispielsweise alle verbleibenden System V init (`sysvinit`) Verwendungszwecke, wie z. B. `init` Skripts, auf ihre `systemd` Entsprechungen migrieren, sind Sie auf die future vorbereitet und können den gesamten `systemd` Funktionsumfang nutzen, um den Dienst zu überwachen, wie und ob er neu gestartet werden muss, welche anderen Dienste er benötigt und ob Ressourcen- oder Berechtigungsbeschränkungen angewendet werden sollten.

Funktionen wie die 32-Bit-Unterstützung können mehrere Hauptversionen des Betriebssystems als veraltet bezeichnen. Für 32-Bit ist Amazon Linux 1 (AL1) veraltet32-Bit-x86 (i686) AMIs, Amazon Linux 2 als veraltet und Amazon Linux 2023 32-Bit-x86-Pakete (i686) als veraltet. 32-Bit-x86-Laufzeitunterstützung (i686) Die Umstellung weg von umfasst IMDSv1 auch mehrere Hauptversionen des Betriebssystems. Es versteht sich, dass einige Kunden für diese Art von Änderungen eine längere Zeit benötigen, um sich an sie anzupassen. Daher besteht ein großer Spielraum, bis die Funktionalität in Amazon Linux 2023 nicht mehr verfügbar ist.

Die Liste der veralteten Funktionen wird im Laufe der Lebensdauer des Betriebssystems aktualisiert, und es ist ratsam, sich über Änderungen auf dem Laufenden zu halten.

Erhalten Sie Benachrichtigungen über neue Updates

Sie können Benachrichtigungen erhalten, wenn ein neues AL2 023-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 AL2 023-AMI veröffentlicht wird. 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 abonnirte 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.

Deterministische Upgrades durch versionierte Repositorys auf 023 AL2

> **Note**
>
> Standardmäßig erhält Ihre AL2 023-Instance beim Start nicht automatisch zusätzliche kritische und wichtige Sicherheitsupdates. Ihre Instance enthält zunächst die Updates, die in der Version AL2 023 und dem ausgewählten AMI verfügbar waren.

Kontrolle über die Updates, die Sie aus Haupt- und Nebenversionen erhalten

Mit AL2 023 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.

Immer wenn wir neue Paket-Updates veröffentlichen, gibt es eine neue Version, auf die man sich beschränken kann, und neue Versionen, AMIs die sich auf diese Version beschränken.

AL2023 sperrt auf eine bestimmte Version Ihres Repositorys. Dieses Verhalten wird für Haupt- und Nebenversionen unterstützt. Das AL2 023-AMI, das über unsere SSM-Parameter verfügbar gemacht wird, 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 AL2 023 enthalten umfangreiche Updates und können 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 AL2 023 enthalten Funktions- und Sicherheitsupdates, 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.

Wissen, wann Updates verfügbar sind

Um ein Update anwenden zu können, müssen Sie wissen, dass eines verfügbar ist, und dann wissen, wie das Update bereitgestellt wird.

Für Builds, die AMIs bei der Veröffentlichung der neuen AMIs Version AL2 023 abgeleitet wurden, kann EC2 Image Builder automatisch Build, Patches und Tests durchführen AMIs. Sie können Ihre eigenen AMI-Gebäude-Pipelines auslösen oder die Basis AMIs nutzen. Erhalten Sie Benachrichtigungen über neue Updates

Für das direkte Patchen können Sie Tools wie AWS Systems Manager Patch Manager verwenden, um die Installation von Updates innerhalb einer Flotte zu koordinieren.

Für andere Veröffentlichungen, die auf AL2 023 AMIs basieren, AMIs können die Anbieter dieser Produkte ihren eigenen Veröffentlichungszeitplan und ihre eigenen Benachrichtigungsmethoden haben. Wenn Sie abgeleitete Images AMIs oder Container-Images verwenden, lesen Sie in der Dokumentation des Herausgebers nach, wann Updates veröffentlicht werden.

Die Änderungen in den einzelnen Versionen sind in den Versionshinweisen für Version AL2 023 dokumentiert. Sicherheitsupdates werden im Amazon Linux Security Center (ALAS) veröffentlicht.

Kontrollieren Sie die Paket-Updates, die in den AL2 023-Repositorys verfügbar sind

Wenn wir eine neue Version der AL2 023-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 upgrade --releasever=version
```

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 AL2 023 verwalten.

Deterministische Updates durch Instanzersatz

Die Deterministische Upgrades durch versionierte Repositorys auf 023 AL2 Funktion von Amazon Linux 2023 macht den Instance-Austausch zu einer einfachen Möglichkeit, aktualisierte Versionen von AL2 023 deterministisch und sicher einzuführen. Deterministische Updates bedeuten, dass bei

der schrittweisen Einführung einer neuen Version, wenn ein Problem gefunden wird, es einfach ist, zum vorherigen AMI zurückzukehren und gleichzeitig die Ursache des Problems zu ermitteln.

Durch die Verwendung von Instance-Ersatz anstelle von direkten Patches sind Updates deterministischer und vorhersehbarer, da die Einführung neuer Kapazitäten ein gut getesteter Codepfad mit klaren A- und B-Zuständen sein kann. Jeder Vorher- und Nachher-Status kann in einem CI/CD-System gründlich getestet werden, bevor die Bereitstellung beginnt.

Beim direkten Patchen gibt es viele Zwischenzustände zwischen der Zeit vor und nach der Installation von Updates, was für alle Kombinationen von Zuständen schwieriger zu testen ist.

Eine Strategie für Betriebssystemupdates, bei der Instanzen durch deterministische Updates ersetzt werden, passt gut zu blau/grün-, wellen- und phasenbasierten Bereitstellungsmodellen.

Verwendung deterministischer Upgrades über versionierte Repositorys

Themen

- Verwendung eines aktualisierten deterministischen Systems
- Selektives Update eines deterministisch aktualisierten Systems
- Persistente Überschreibung bei einem deterministischen Upgrade

Verwendung eines aktualisierten deterministischen Systems

> **Note**
>
> Das Standardverhalten des Paketmanagers wurde von AL2 geändert.

Deterministische Upgrades sind eine leistungsstarke Methode, um sicherzustellen, dass alle Änderungen an Produktionsumgebungen vor der umfassenden Bereitstellung vollständig getestet werden können. Jedes neue AL2 023-AMI ist an eine bestimmte Version von AL2 023 gebunden. Dies ermöglicht ein deterministisches Verhalten darüber, welche Versionen von Betriebssystempaketen installiert werden, wenn das spezifische AMI gestartet wird. Direkte Updates können sich auf eine bestimmte Release-Version beziehen, wodurch ein deterministisches Verhalten innerhalb einer Flotte gewährleistet wird. Bei der Umstellung auf neue AMIs oder direkte Aktualisierungsversionen können Sie jede Version in Ihrer CI/CD Pipeline testen und dabei potenzielle Probleme erkennen, bevor Sie sie in Produktionsumgebungen bereitstellen.

Sie können Tools wie AWS Systems Manager Patch Manager verwenden, um die Installation von Updates innerhalb einer Flotte zu koordinieren. Für Builds, die abgeleitet werden, AMIs wenn neue AL2 023 veröffentlicht AMIs werden, kann EC2 Image Builder automatisch erstellen, patchen und testen AMIs, oder Sie können Erhalten Sie Benachrichtigungen über neue Updates wissen, wann neue Basisversionen verfügbar AMIs sind, oder um Ihre eigenen AMI-Building-Pipelines auszulösen.

Informationen zur Beschränkung von Updates auf die Updates aus einem bestimmten Advisory finden Sie unter Direktes Anwenden von Sicherheitsupdates

Für direkte Patches können Sie den `dnf` Paketmanager verwenden. 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 *`latest`* oder eine Version mit Datumsstempel wie. *`2023.8.20250721`*

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 lautet 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.

Wenn Sie immer angeben, auf welche Version `releasever` (d. h. Version AL2 023) aktualisiert werden soll, haben Sie eine deterministische Reihe von Änderungen in einer Flotte. Sie haben die Version gestartet *AB*, auf aktualisiert und dann auf aktualisiert. *C*

Selektives Update eines deterministisch aktualisierten Systems

> **Note**
>
> Wir empfehlen, dass alle Updates in einer neuen Version installiert werden, anstatt bestimmte Updates auszuwählen. Es sollte eine Ausnahme von der üblichen Praxis sein, das gesamte Update nur teilweise auf das Betriebssystem anzuwenden.

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

> **Note**
>
> Mit deterministischen Updates können Sie Betriebssystemänderungen in Ihre CI/CD Pipeline integrieren. Durch die Deaktivierung deterministischer Updates können keine Tests vor der Bereitstellung durchgeführt werden.

Anstatt etwas hinzuzufügen`--releasever=latest`, können Sie das System mit persistenter Überschreibung entsperren, indem Sie den Variablenwert auf setzen. *`latest`* Wenn Sie diese `latest` Option immer verwenden, wird das Verhalten von AL2 023 auf das AL2 Aktualisierungsmodell zurückgesetzt, bei dem bei jedem Aufruf des Paketmanagers immer die neueste Version angezeigt wird und nicht an eine bestimmte Version des Betriebssystems gebunden ist.

> **Warning**
>
> Wenn Sie den Paketmanager mithilfe einer persistenten Überschreibung deterministischer Updates entsperren, gehen Sie das Risiko ein, mögliche Inkompatibilitäten zwischen Ihrer Anwendung und einem Betriebssystemupdate in der Produktion zu entdecken. Inkompatibilitäten sind zwar selten, aber mit einem Betriebssystem-Update integrieren Sie neue Codeänderungen in Ihre Umgebung. Integrationstests können jedoch verhindern, dass Codeänderungen implementiert werden, die sich negativ auf Produktionsumgebungen auswirken.

```
$ 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
```

Die Verwendung einer dauerhaften Überschreibung der Verwendung `latest` anstelle einer bestimmten Version entspricht dem Standardverhalten von. AL2 Es gibt Dienste, die darauf AMIs aufbauen AL2 , dieses Verhalten deaktivieren und sich auf bestimmte Paketversionen beschränken, wie Sie es standardmäßig auf AL2 023 bekommen.

Anstatt deterministische Updates zu deaktivieren, empfehlen wir, Instances durch Instances zu ersetzen, die über ein neues AMI gestartet wurden. Wenn der Austausch von Instances keine Option ist, empfehlen wir die Verwendung von Tools wie AWS Systems Manager Patch Manager, um die Installation von Updates innerhalb einer Flotte zu koordinieren. EC2 Image Builder kann auch automatisch Ihre eigenen, aus AL2 023 Basis-Images AMIs abgeleiteten Images erstellen, patchen und testen. Sie können dies auch verwendenErhalten Sie Benachrichtigungen über neue Updates, um Ihre eigenen AMI-Gebäude-Pipelines auszulösen.

Die Verwendung `latest` in einer Vorproduktionsumgebung und die anschließende Bereitstellung in der Produktionsumgebung bieten `latest` keinen Schutz vor Problemen zwischen einem Betriebssystem-Update und Ihrer Anwendung. Eine neue Version AL2 023 kann zu jedem beliebigen Zeitpunkt verfügbar sein, weshalb jede Verwendung `latest` in der Produktion mit Risiken verbunden ist.

Paket- und Betriebssystemupdates in AL2 023 verwalten

Im Gegensatz zu früheren Versionen von Amazon Linux AMIs sind AL2 023 an eine bestimmte Version des Amazon Linux-Repositorys gebunden. Um sowohl Sicherheits- als auch Bugfixes auf eine AL2 023-Instance anzuwenden, aktualisieren Sie DNF Konfiguration auf die neueste verfügbare Release-Version. Alternativ können Sie eine neuere AL2 023-Instanz starten.

In diesem Abschnitt wird die Verwaltung beschrieben DNF Pakete und Repositorys auf einer laufenden Instanz. Es beschreibt auch, wie man konfiguriert DNF aus einem Benutzerdatenskript, um das neueste verfügbare Amazon Linux-Repository beim Start zu aktivieren. Weitere Informationen finden Sie in der DNF-Befehlsreferenz.

Es wird empfohlen, alle in einer neuen Version AL2 023 verfügbaren Updates zu installieren. Die Auswahl nur von Sicherheitsupdates oder nur bestimmten Updates sollte eher die Ausnahme als die Regel sein. Eine Liste der Sicherheitsempfehlungen für eine bestimmte Instanz relevanten Versionen finden Sie unterListe der zutreffenden Hinweise. Informationen dazu, wie Sie nur Updates installieren können, die für eine bestimmte Empfehlung relevant sind, finden Sie unterDirektes Anwenden von Sicherheitsupdates.

> **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

Themen

- Prüfen auf verfügbare Paket-Updates
- Anwenden von Sicherheitsupdates mit DNF und Repository-Versionen
- Automatischer Neustart des Dienstes nach (Sicherheits-) Updates
- Wann ist ein Neustart erforderlich, um Sicherheitsupdates anzuwenden?
- Starten einer Instance mit aktivierter neuester Repository-Version
- Abrufen von Paketunterstützunginformationen
- Suche nach neueren Repository-Versionen mit dnf check-release-update
- 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 AL2 023 empfehlen wir, dass Sie dem Befehl die `--releasever=`*`version-number`* Option hinzufügen.

Wenn Sie diese Option hinzufügen, DNF sucht auch nach Updates für eine neuere Version des Repositorys. Verwenden Sie beispielsweise nach der Ausführung des `dnf check-update`-Befehls die zuletzt zurückgegebene Version als Wert für *`version-number`*.

Wenn die Instanz 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.

Updates in a specific version

In diesem Beispiel werden wir uns ansehen, welche Updates in der Version 2023.1.20230628 verfügbar sind, wenn wir einen Container der Version 2023.0.20230315 gestartet haben.

> **Note**
>
> In diesem Beispiel werden die Versionen 2023.0.20230315 und 2023.1.20230628verwendet, und dies sind nicht die neueste Version von 023. In den Versionshinweisen zu 023 finden Sie die neuesten Versionen, die die neuesten Sicherheitsupdates enthalten. AL2 AL2

In diesem Beispiel beginnen wir mit einem Container-Image für die Version 2023.0.20230315.

Zuerst holen wir uns dieses Container-Image aus der Container-Registry. Das `.0` am Ende steht für die Version des Images für eine bestimmte Version; diese Image-Version ist normalerweise Null.

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
   2023.0.20230315.0: Pulling from amazonlinux/amazonlinux
b76f3b09316a: Pull complete
Digest: sha256:94e7183b0739140dbd5b639fb7600f0a2299cec5df8780c26d9cb409da5315a9
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.0.20230315.0
public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
```

Wir können jetzt eine Shell innerhalb des Containers erzeugen, von der aus wir nach Updates suchen.

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
   bash-5.2#
```

Der `dnf check-update` Befehl wird jetzt verwendet, um nach Updates zu suchen, die in der Version 2023.1.20230628 verfügbar sind.

> **Note**
>
> Das Anwenden von Paket-Updates ist ein privilegierter Vorgang. Eine Erhöhung der Rechte ist bei der Ausführung in einem Container in der Regel nicht erforderlich. Wenn Sie jedoch in einer nicht containerisierten Umgebung wie einer EC2 Amazon-Instance ausgeführt werden, können Sie nach Updates suchen, ohne die Rechte zu erhöhen.

```
$ dnf check-update --releasever=2023.1.20230628
Amazon Linux 2023 repository                      60 MB/s |  15 MB     00:00
Last metadata expiration check: 0:00:02 ago on Mon Jul 22 17:25:34 2024.

amazon-linux-repo-cdn.noarch        2023.1.20230628-0.amzn2023        amazonlinux
ca-certificates.noarch              2023.2.60-1.0.amzn2023.0.2        amazonlinux
curl-minimal.x86_64                 8.0.1-1.amzn2023                  amazonlinux
glib2.x86_64                        2.74.7-688.amzn2023.0.1           amazonlinux
glibc.x86_64                        2.34-52.amzn2023.0.3              amazonlinux
glibc-common.x86_64                 2.34-52.amzn2023.0.3              amazonlinux
glibc-minimal-langpack.x86_64       2.34-52.amzn2023.0.3              amazonlinux
gnupg2-minimal.x86_64               2.3.7-1.amzn2023.0.4              amazonlinux
keyutils-libs.x86_64                1.6.3-1.amzn2023                  amazonlinux
libcap.x86_64                       2.48-2.amzn2023.0.3               amazonlinux
libcurl-minimal.x86_64              8.0.1-1.amzn2023                  amazonlinux
libgcc.x86_64                       11.3.1-4.amzn2023.0.3             amazonlinux
libgomp.x86_64                      11.3.1-4.amzn2023.0.3             amazonlinux
libstdc++.x86_64                    11.3.1-4.amzn2023.0.3             amazonlinux
libxml2.x86_64                      2.10.4-1.amzn2023.0.1             amazonlinux
ncurses-base.noarch                 6.2-4.20200222.amzn2023.0.4       amazonlinux
ncurses-libs.x86_64                 6.2-4.20200222.amzn2023.0.4       amazonlinux
openssl-libs.x86_64                 1:3.0.8-1.amzn2023.0.3            amazonlinux
python3-rpm.x86_64                  4.16.1.3-12.amzn2023.0.6          amazonlinux
rpm.x86_64                          4.16.1.3-12.amzn2023.0.6          amazonlinux
rpm-build-libs.x86_64               4.16.1.3-12.amzn2023.0.6          amazonlinux
rpm-libs.x86_64                     4.16.1.3-12.amzn2023.0.6          amazonlinux
rpm-sign-libs.x86_64                4.16.1.3-12.amzn2023.0.6          amazonlinux
system-release.noarch               2023.1.20230628-0.amzn2023        amazonlinux
tzdata.noarch                       2023c-1.amzn2023.0.1              amazonlinux
bash-5.2#
```

Die Version des `system-release` Pakets zeigt die Version, auf die ein `dnf upgrade` Befehl aktualisiert werden würde, nämlich die Version 2023.1.20230628, die im Befehl angefordert wurde. `dnf check-update --releasever=2023.1.20230628`

Updates in the latest version

In diesem Beispiel werden wir uns ansehen, welche Updates in der Version 0.23 verfügbar sind, wenn wir einen Container der `latest` Version AL2 2023.4.20240319 gestartet haben. Zum Zeitpunkt der Erstellung dieses Artikels ist die `latest` Version 2023.5.20240708, sodass die aufgelisteten Updates in diesem Beispiel den aktuellen Stand dieser Version haben.

> **Note**
>
> In diesem Beispiel werden die Versionen 2023.4.20240319 und 2023.5.20240708 verwendet, wobei letztere die neueste Version zum Zeitpunkt der Erstellung dieses Artikels ist. Weitere Informationen zu den neuesten Versionen finden Sie in den Versionshinweisen zu 023. AL2

In diesem Beispiel beginnen wir mit einem Container-Image für die Version 2023.4.20240319.

Zuerst holen wir uns dieses Container-Image aus der Container-Registry. Das `.1` am Ende gibt die Version des Images für eine bestimmte Version an. Während die Image-Version normalerweise Null ist, wird in diesem Beispiel eine Version verwendet, bei der es sich um eine Image-Version handelt.

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
   2023.4.20240319.1: Pulling from amazonlinux/amazonlinux
6de065fda9a2: Pull complete
Digest: sha256:b4838c4cc9211d966b6ea158dacc9eda7433a16ba94436508c2d9f01f7658b4e
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.4.20240319.1
public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
```

Wir können jetzt eine Shell innerhalb des Containers erzeugen, von der aus wir nach Updates suchen.

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
   bash-5.2#
```

Der `dnf check-update` Befehl wird nun verwendet, um nach Updates zu suchen, die in der `latest` Version verfügbar waren, die zum Zeitpunkt des Schreibens 2023.5.20240708 lautete.

> **Note**
>
> Das Anwenden von Paket-Updates ist ein privilegierter Vorgang. Eine Erhöhung der Rechte ist bei der Ausführung in einem Container in der Regel nicht erforderlich. Wenn Sie jedoch in einer nicht containerisierten Umgebung wie einer EC2 Amazon-Instance ausgeführt werden, können Sie nach Updates suchen, ohne die Rechte zu erhöhen.

```
$ dnf --releasever=latest check-update
Amazon Linux 2023 repository                    78 MB/s |  25 MB     00:00
Last metadata expiration check: 0:00:04 ago on Mon Jul 22 17:39:13 2024.

amazon-linux-repo-cdn.noarch        2023.5.20240708-1.amzn2023        amazonlinux
curl-minimal.x86_64                 8.5.0-1.amzn2023.0.4              amazonlinux
dnf.noarch                          4.14.0-1.amzn2023.0.5             amazonlinux
dnf-data.noarch                     4.14.0-1.amzn2023.0.5             amazonlinux
expat.x86_64                        2.5.0-1.amzn2023.0.4              amazonlinux
glibc.x86_64                        2.34-52.amzn2023.0.10             amazonlinux
glibc-common.x86_64                 2.34-52.amzn2023.0.10             amazonlinux
glibc-minimal-langpack.x86_64       2.34-52.amzn2023.0.10             amazonlinux
krb5-libs.x86_64                    1.21-3.amzn2023.0.4               amazonlinux
libblkid.x86_64                     2.37.4-1.amzn2023.0.4             amazonlinux
libcurl-minimal.x86_64              8.5.0-1.amzn2023.0.4              amazonlinux
libmount.x86_64                     2.37.4-1.amzn2023.0.4             amazonlinux
libnghttp2.x86_64                   1.59.0-3.amzn2023.0.1             amazonlinux
libsmartcols.x86_64                 2.37.4-1.amzn2023.0.4             amazonlinux
libuuid.x86_64                      2.37.4-1.amzn2023.0.4             amazonlinux
openssl-libs.x86_64                 1:3.0.8-1.amzn2023.0.12           amazonlinux
python3.x86_64                      3.9.16-1.amzn2023.0.8             amazonlinux
python3-dnf.noarch                  4.14.0-1.amzn2023.0.5             amazonlinux
python3-libs.x86_64                 3.9.16-1.amzn2023.0.8             amazonlinux
system-release.noarch               2023.5.20240708-1.amzn2023        amazonlinux
yum.noarch                          4.14.0-1.amzn2023.0.5             amazonlinux
bash-5.2#
```

Die Version des `system-release` Pakets zeigt die Version, auf die ein `dnf upgrade` Befehl aktualisiert werden würde.

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 Sicherheitsupdates mit DNF und Repository-Versionen

Neue Paket- und Sicherheits-Updates werden nur für neue Repository-Versionen bereitgestellt. Für Instances, die Sie von früheren AL2 023 AMI-Versionen aus gestartet haben, müssen Sie die Repository-Version aktualisieren, bevor Sie Sicherheitsupdates 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.

> **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 Updates für installierte Pakete, die in einer späteren Version des Repositorys vorhanden sind, nicht angewendet werden.

Dieser Abschnitt behandelt den empfohlenen Upgrade-Pfad, bei dem alle verfügbaren Updates installiert werden, anstatt einzelne Updates oder nur solche auszuwählen, die als Sicherheitsupdates gekennzeichnet sind. Durch das Anwenden aller Updates werden vorhandene Instances in denselben Paketsatz verschoben wie beim Starten eines aktualisierten AMI. Diese Konsistenz reduziert die Variation der Paketversionen innerhalb einer Flotte. Weitere Informationen zur Anwendung bestimmter Updates finden Sie unterDirektes Anwenden von Sicherheitsupdates.

Applying updates in a specific version

In diesem Beispiel werden wir Updates anwenden, die in der Version 2023.1.20230628 verfügbar sind, wenn wir einen Container der Version 2023.0.20230315 gestartet haben.

> **Note**
>
> In diesem Beispiel werden die Versionen 2023.0.20230315 und 2023.1.20230628verwendet, und dies sind nicht die neueste Version von 023. Die neuesten Versionen, die die neuesten Sicherheitsupdates enthalten, finden Sie in den Versionshinweisen zu 023. AL2 AL2

In diesem Beispiel beginnen wir mit einem Container-Image für die Version 2023.0.20230315.

Zuerst holen wir uns dieses Container-Image aus der Container-Registry. Das `.0` am Ende steht für die Version des Images für eine bestimmte Version; diese Image-Version ist normalerweise Null.

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
   2023.0.20230315.0: Pulling from amazonlinux/amazonlinux
b76f3b09316a: Pull complete
Digest: sha256:94e7183b0739140dbd5b639fb7600f0a2299cec5df8780c26d9cb409da5315a9
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.0.20230315.0
public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
```

Wir können jetzt eine Shell innerhalb des Containers erzeugen, von der aus wir Updates anwenden werden.

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
   bash-5.2#
```

Der `dnf upgrade` Befehl wird jetzt verwendet, um alle in der Version 2023.1.20230628 enthaltenen Updates anzuwenden.

> **Note**
>
> Das Anwenden von Paket-Updates ist ein privilegierter Vorgang. Eine Erhöhung der Rechte ist bei der Ausführung in einem Container in der Regel nicht erforderlich. Wenn Sie jedoch in einer nicht containerisierten Umgebung wie einer EC2 Amazon-Instance ausgeführt werden, müssen Sie den `dnf upgrade` Befehl als Benutzer ausführen. `root` Dies kann mit den Befehlen oder erfolgen. `sudo su`

```
$ dnf upgrade --releasever=2023.1.20230628
Amazon Linux 2023 repository                    38 MB/s |  15 MB     00:00
Last metadata expiration check: 0:00:02 ago on Mon Jul 22 17:49:08 2024.
Dependencies resolved.
================================================================================
 Package                 Arch    Version                     Repository    Size
================================================================================
Upgrading:
```

```
 amazon-linux-repo-cdn   noarch  2023.1.20230628-0.amzn2023   amazonlinux   18 k
 ca-certificates         noarch  2023.2.60-1.0.amzn2023.0.2   amazonlinux  829 k
 curl-minimal            x86_64  8.0.1-1.amzn2023             amazonlinux  150 k
 glib2                   x86_64  2.74.7-688.amzn2023.0.1      amazonlinux  2.7 M
 glibc                   x86_64  2.34-52.amzn2023.0.3         amazonlinux  1.9 M
 glibc-common            x86_64  2.34-52.amzn2023.0.3         amazonlinux  307 k
 glibc-minimal-langpack  x86_64  2.34-52.amzn2023.0.3         amazonlinux   35 k
 gnupg2-minimal          x86_64  2.3.7-1.amzn2023.0.4         amazonlinux  421 k
 keyutils-libs           x86_64  1.6.3-1.amzn2023             amazonlinux   33 k
 libcap                  x86_64  2.48-2.amzn2023.0.3          amazonlinux   67 k
 libcurl-minimal         x86_64  8.0.1-1.amzn2023             amazonlinux  249 k
 libgcc                  x86_64  11.3.1-4.amzn2023.0.3        amazonlinux  105 k
 libgomp                 x86_64  11.3.1-4.amzn2023.0.3        amazonlinux  280 k
 libstdc++               x86_64  11.3.1-4.amzn2023.0.3        amazonlinux  744 k
 libxml2                 x86_64  2.10.4-1.amzn2023.0.1        amazonlinux  706 k
 ncurses-base            noarch  6.2-4.20200222.amzn2023.0.4  amazonlinux   60 k
 ncurses-libs            x86_64  6.2-4.20200222.amzn2023.0.4  amazonlinux  328 k
 openssl-libs            x86_64  1:3.0.8-1.amzn2023.0.3       amazonlinux  2.2 M
 python3-rpm             x86_64  4.16.1.3-12.amzn2023.0.6     amazonlinux   88 k
 rpm                     x86_64  4.16.1.3-12.amzn2023.0.6     amazonlinux  486 k
 rpm-build-libs          x86_64  4.16.1.3-12.amzn2023.0.6     amazonlinux   90 k
 rpm-libs                x86_64  4.16.1.3-12.amzn2023.0.6     amazonlinux  309 k
 rpm-sign-libs           x86_64  4.16.1.3-12.amzn2023.0.6     amazonlinux   21 k
 system-release          noarch  2023.1.20230628-0.amzn2023   amazonlinux   29 k
 tzdata                  noarch  2023c-1.amzn2023.0.1         amazonlinux  433 k

Transaction Summary
================================================================================
Upgrade  25 Packages

Total download size: 12 M
Is this ok [y/N]:
```

Die Version des `system-release` Pakets zeigt die Version, auf die ein `dnf upgrade` Befehl aktualisiert werden würde, nämlich die Version 2023.1.20230628, die im Befehl angefordert wurde. `dnf upgrade --releasever=`*`2023.1.20230628`*

Standardmäßig `dnf` werden Sie aufgefordert, zu bestätigen, dass Sie die Updates anwenden möchten. Sie können diese Aufforderung umgehen, indem Sie das `-y` Flag to verwenden`dnf`. In diesem Beispiel fragt der `dnf upgrade -y --releasever=`*`2023.1.20230628`* Befehl nicht nach einer Bestätigung, bevor die Updates installiert werden. Dies ist in Skripten oder anderen Automatisierungsumgebungen nützlich.

Sobald Sie bestätigt haben, dass Sie die Updates anwenden möchten, werden sie `dnf` angewendet.

```
Is this ok [y/N]:y
   Downloading Packages:
(1/25): libcap-2.48-2.amzn2023.0.3.x86_64.rpm    1.5 MB/s |  67 kB     00:00
(2/25): python3-rpm-4.16.1.3-12.amzn2023.0.6.x86 2.1 MB/s |  88 kB     00:00
(3/25): libcurl-minimal-8.0.1-1.amzn2023.x86_64. 2.6 MB/s | 249 kB     00:00
(4/25): glib2-2.74.7-688.amzn2023.0.1.x86_64.rpm  26 MB/s | 2.7 MB     00:00
(5/25): glibc-minimal-langpack-2.34-52.amzn2023. 1.3 MB/s |  35 kB     00:00
(6/25): rpm-build-libs-4.16.1.3-12.amzn2023.0.6. 2.8 MB/s |  90 kB     00:00
(7/25): rpm-libs-4.16.1.3-12.amzn2023.0.6.x86_64 6.6 MB/s | 309 kB     00:00
(8/25): libgcc-11.3.1-4.amzn2023.0.3.x86_64.rpm  3.9 MB/s | 105 kB     00:00
(9/25): glibc-common-2.34-52.amzn2023.0.3.x86_64  11 MB/s | 307 kB     00:00
(10/25): glibc-2.34-52.amzn2023.0.3.x86_64.rpm    31 MB/s | 1.9 MB     00:00
(11/25): rpm-sign-libs-4.16.1.3-12.amzn2023.0.6. 877 kB/s |  21 kB     00:00
(12/25): gnupg2-minimal-2.3.7-1.amzn2023.0.4.x86  15 MB/s | 421 kB     00:00
(13/25): openssl-libs-3.0.8-1.amzn2023.0.3.x86_6  35 MB/s | 2.2 MB     00:00
(14/25): libxml2-2.10.4-1.amzn2023.0.1.x86_64.rp  14 MB/s | 706 kB     00:00
(15/25): curl-minimal-8.0.1-1.amzn2023.x86_64.rp 4.2 MB/s | 150 kB     00:00
(16/25): rpm-4.16.1.3-12.amzn2023.0.6.x86_64.rpm  11 MB/s | 486 kB     00:00
(17/25): libgomp-11.3.1-4.amzn2023.0.3.x86_64.rp 7.0 MB/s | 280 kB     00:00
(18/25): libstdc++-11.3.1-4.amzn2023.0.3.x86_64.  14 MB/s | 744 kB     00:00
(19/25): keyutils-libs-1.6.3-1.amzn2023.x86_64.r 1.6 MB/s |  33 kB     00:00
(20/25): ncurses-libs-6.2-4.20200222.amzn2023.0.  10 MB/s | 328 kB     00:00
(21/25): tzdata-2023c-1.amzn2023.0.1.noarch.rpm   11 MB/s | 433 kB     00:00
(22/25): amazon-linux-repo-cdn-2023.1.20230628-0 781 kB/s |  18 kB     00:00
(23/25): ca-certificates-2023.2.60-1.0.amzn2023.  16 MB/s | 829 kB     00:00
(24/25): system-release-2023.1.20230628-0.amzn20 1.5 MB/s |  29 kB     00:00
(25/25): ncurses-base-6.2-4.20200222.amzn2023.0. 3.1 MB/s |  60 kB     00:00
--------------------------------------------------------------------------------
Total                                             28 MB/s |  12 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Upgrading        : libgcc-11.3.1-4.amzn2023.0.3.x86_64                   1/50
  Running scriptlet: libgcc-11.3.1-4.amzn2023.0.3.x86_64                   1/50
  Upgrading        : system-release-2023.1.20230628-0.amzn2023.noarch       2/50
  Upgrading        : amazon-linux-repo-cdn-2023.1.20230628-0.amzn2023.no    3/50
  Upgrading        : ncurses-base-6.2-4.20200222.amzn2023.0.4.noarch        4/50
```

```
Upgrading        : tzdata-2023c-1.amzn2023.0.1.noarch                    5/50
Upgrading        : glibc-common-2.34-52.amzn2023.0.3.x86_64              6/50
Running scriptlet: glibc-2.34-52.amzn2023.0.3.x86_64                     7/50
Upgrading        : glibc-2.34-52.amzn2023.0.3.x86_64                     7/50
Running scriptlet: glibc-2.34-52.amzn2023.0.3.x86_64                     7/50
Upgrading        : glibc-minimal-langpack-2.34-52.amzn2023.0.3.x86_64    8/50
Upgrading        : libcap-2.48-2.amzn2023.0.3.x86_64                     9/50
Upgrading        : gnupg2-minimal-2.3.7-1.amzn2023.0.4.x86_64           10/50
Upgrading        : libgomp-11.3.1-4.amzn2023.0.3.x86_64                 11/50
Running scriptlet: ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch    12/50
Upgrading        : ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch    12/50
Running scriptlet: ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch    12/50
Upgrading        : openssl-libs-1:3.0.8-1.amzn2023.0.3.x86_64           13/50
Upgrading        : libcurl-minimal-8.0.1-1.amzn2023.x86_64              14/50
Upgrading        : curl-minimal-8.0.1-1.amzn2023.x86_64                 15/50
Upgrading        : rpm-libs-4.16.1.3-12.amzn2023.0.6.x86_64             16/50
Upgrading        : rpm-4.16.1.3-12.amzn2023.0.6.x86_64                  17/50
Upgrading        : rpm-build-libs-4.16.1.3-12.amzn2023.0.6.x86_64       18/50
Upgrading        : rpm-sign-libs-4.16.1.3-12.amzn2023.0.6.x86_64        19/50
Upgrading        : python3-rpm-4.16.1.3-12.amzn2023.0.6.x86_64          20/50
Upgrading        : glib2-2.74.7-688.amzn2023.0.1.x86_64                 21/50
Upgrading        : libxml2-2.10.4-1.amzn2023.0.1.x86_64                 22/50
Upgrading        : libstdc++-11.3.1-4.amzn2023.0.3.x86_64               23/50
Upgrading        : keyutils-libs-1.6.3-1.amzn2023.x86_64                24/50
Upgrading        : ncurses-libs-6.2-4.20200222.amzn2023.0.4.x86_64      25/50
Cleanup          : glib2-2.73.2-680.amzn2023.0.3.x86_64                 26/50
Cleanup          : libstdc++-11.3.1-4.amzn2023.0.2.x86_64               27/50
Cleanup          : libxml2-2.10.3-2.amzn2023.0.1.x86_64                 28/50
Cleanup          : python3-rpm-4.16.1.3-12.amzn2023.0.5.x86_64          29/50
Cleanup          : rpm-build-libs-4.16.1.3-12.amzn2023.0.5.x86_64       30/50
Cleanup          : rpm-sign-libs-4.16.1.3-12.amzn2023.0.5.x86_64        31/50
Cleanup          : rpm-libs-4.16.1.3-12.amzn2023.0.5.x86_64             32/50
Cleanup          : libcap-2.48-2.amzn2023.0.2.x86_64                    33/50
Cleanup          : gnupg2-minimal-2.3.7-1.amzn2023.0.3.x86_64           34/50
Cleanup          : ncurses-libs-6.2-4.20200222.amzn2023.0.3.x86_64      35/50
Cleanup          : libgomp-11.3.1-4.amzn2023.0.2.x86_64                 36/50
Cleanup          : rpm-4.16.1.3-12.amzn2023.0.5.x86_64                  37/50
Cleanup          : curl-minimal-7.88.1-1.amzn2023.0.1.x86_64            38/50
Cleanup          : libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64         39/50
Cleanup          : openssl-libs-1:3.0.8-1.amzn2023.0.1.x86_64           40/50
Cleanup          : keyutils-libs-1.6.1-2.amzn2023.0.2.x86_64            41/50
Cleanup          : amazon-linux-repo-cdn-2023.0.20230315-1.amzn2023.no  42/50
Cleanup          : system-release-2023.0.20230315-1.amzn2023.noarch      43/50
Cleanup          : ca-certificates-2023.2.60-1.0.amzn2023.0.1.noarch    44/50
```

```
Cleanup          : ncurses-base-6.2-4.20200222.amzn2023.0.3.noarch       45/50
Cleanup          : glibc-minimal-langpack-2.34-52.amzn2023.0.2.x86_64    46/50
Cleanup          : glibc-2.34-52.amzn2023.0.2.x86_64                     47/50
Cleanup          : glibc-common-2.34-52.amzn2023.0.2.x86_64              48/50
Cleanup          : tzdata-2022g-1.amzn2023.0.1.noarch                    49/50
Cleanup          : libgcc-11.3.1-4.amzn2023.0.2.x86_64                   50/50
Running scriptlet: libgcc-11.3.1-4.amzn2023.0.2.x86_64                   50/50
Running scriptlet: ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch     50/50
Running scriptlet: rpm-4.16.1.3-12.amzn2023.0.6.x86_64                   50/50
Running scriptlet: libgcc-11.3.1-4.amzn2023.0.2.x86_64                   50/50
Verifying        : libcurl-minimal-8.0.1-1.amzn2023.x86_64                1/50
Verifying        : libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64           2/50
Verifying        : libcap-2.48-2.amzn2023.0.3.x86_64                      3/50
Verifying        : libcap-2.48-2.amzn2023.0.2.x86_64                      4/50
Verifying        : glib2-2.74.7-688.amzn2023.0.1.x86_64                   5/50
Verifying        : glib2-2.73.2-680.amzn2023.0.3.x86_64                   6/50
Verifying        : python3-rpm-4.16.1.3-12.amzn2023.0.6.x86_64            7/50
Verifying        : python3-rpm-4.16.1.3-12.amzn2023.0.5.x86_64            8/50
Verifying        : glibc-minimal-langpack-2.34-52.amzn2023.0.3.x86_64     9/50
Verifying        : glibc-minimal-langpack-2.34-52.amzn2023.0.2.x86_64    10/50
Verifying        : rpm-libs-4.16.1.3-12.amzn2023.0.6.x86_64              11/50
Verifying        : rpm-libs-4.16.1.3-12.amzn2023.0.5.x86_64              12/50
Verifying        : rpm-build-libs-4.16.1.3-12.amzn2023.0.6.x86_64        13/50
Verifying        : rpm-build-libs-4.16.1.3-12.amzn2023.0.5.x86_64        14/50
Verifying        : glibc-2.34-52.amzn2023.0.3.x86_64                     15/50
Verifying        : glibc-2.34-52.amzn2023.0.2.x86_64                     16/50
Verifying        : libgcc-11.3.1-4.amzn2023.0.3.x86_64                   17/50
Verifying        : libgcc-11.3.1-4.amzn2023.0.2.x86_64                   18/50
Verifying        : glibc-common-2.34-52.amzn2023.0.3.x86_64              19/50
Verifying        : glibc-common-2.34-52.amzn2023.0.2.x86_64              20/50
Verifying        : rpm-sign-libs-4.16.1.3-12.amzn2023.0.6.x86_64         21/50
Verifying        : rpm-sign-libs-4.16.1.3-12.amzn2023.0.5.x86_64         22/50
Verifying        : openssl-libs-1:3.0.8-1.amzn2023.0.3.x86_64            23/50
Verifying        : openssl-libs-1:3.0.8-1.amzn2023.0.1.x86_64            24/50
Verifying        : gnupg2-minimal-2.3.7-1.amzn2023.0.4.x86_64            25/50
Verifying        : gnupg2-minimal-2.3.7-1.amzn2023.0.3.x86_64            26/50
Verifying        : libxml2-2.10.4-1.amzn2023.0.1.x86_64                  27/50
Verifying        : libxml2-2.10.3-2.amzn2023.0.1.x86_64                  28/50
Verifying        : curl-minimal-8.0.1-1.amzn2023.x86_64                  29/50
Verifying        : curl-minimal-7.88.1-1.amzn2023.0.1.x86_64             30/50
Verifying        : rpm-4.16.1.3-12.amzn2023.0.6.x86_64                   31/50
Verifying        : rpm-4.16.1.3-12.amzn2023.0.5.x86_64                   32/50
Verifying        : libstdc++-11.3.1-4.amzn2023.0.3.x86_64                33/50
Verifying        : libstdc++-11.3.1-4.amzn2023.0.2.x86_64                34/50
```

```
  Verifying        : libgomp-11.3.1-4.amzn2023.0.3.x86_64                  35/50
  Verifying        : libgomp-11.3.1-4.amzn2023.0.2.x86_64                  36/50
  Verifying        : keyutils-libs-1.6.3-1.amzn2023.x86_64                 37/50
  Verifying        : keyutils-libs-1.6.1-2.amzn2023.0.2.x86_64             38/50
  Verifying        : ncurses-libs-6.2-4.20200222.amzn2023.0.4.x86_64       39/50
  Verifying        : ncurses-libs-6.2-4.20200222.amzn2023.0.3.x86_64       40/50
  Verifying        : ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch     41/50
  Verifying        : ca-certificates-2023.2.60-1.0.amzn2023.0.1.noarch     42/50
  Verifying        : tzdata-2023c-1.amzn2023.0.1.noarch                    43/50
  Verifying        : tzdata-2022g-1.amzn2023.0.1.noarch                    44/50
  Verifying        : amazon-linux-repo-cdn-2023.1.20230628-0.amzn2023.no   45/50
  Verifying        : amazon-linux-repo-cdn-2023.0.20230315-1.amzn2023.no   46/50
  Verifying        : system-release-2023.1.20230628-0.amzn2023.noarch      47/50
  Verifying        : system-release-2023.0.20230315-1.amzn2023.noarch      48/50
  Verifying        : ncurses-base-6.2-4.20200222.amzn2023.0.4.noarch       49/50
  Verifying        : ncurses-base-6.2-4.20200222.amzn2023.0.3.noarch       50/50

Upgraded:
  amazon-linux-repo-cdn-2023.1.20230628-0.amzn2023.noarch
  ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch
  curl-minimal-8.0.1-1.amzn2023.x86_64
  glib2-2.74.7-688.amzn2023.0.1.x86_64
  glibc-2.34-52.amzn2023.0.3.x86_64
  glibc-common-2.34-52.amzn2023.0.3.x86_64
  glibc-minimal-langpack-2.34-52.amzn2023.0.3.x86_64
  gnupg2-minimal-2.3.7-1.amzn2023.0.4.x86_64
  keyutils-libs-1.6.3-1.amzn2023.x86_64
  libcap-2.48-2.amzn2023.0.3.x86_64
  libcurl-minimal-8.0.1-1.amzn2023.x86_64
  libgcc-11.3.1-4.amzn2023.0.3.x86_64
  libgomp-11.3.1-4.amzn2023.0.3.x86_64
  libstdc++-11.3.1-4.amzn2023.0.3.x86_64
  libxml2-2.10.4-1.amzn2023.0.1.x86_64
  ncurses-base-6.2-4.20200222.amzn2023.0.4.noarch
  ncurses-libs-6.2-4.20200222.amzn2023.0.4.x86_64
  openssl-libs-1:3.0.8-1.amzn2023.0.3.x86_64
  python3-rpm-4.16.1.3-12.amzn2023.0.6.x86_64
  rpm-4.16.1.3-12.amzn2023.0.6.x86_64
  rpm-build-libs-4.16.1.3-12.amzn2023.0.6.x86_64
  rpm-libs-4.16.1.3-12.amzn2023.0.6.x86_64
  rpm-sign-libs-4.16.1.3-12.amzn2023.0.6.x86_64
  system-release-2023.1.20230628-0.amzn2023.noarch
  tzdata-2023c-1.amzn2023.0.1.noarch
```

```
  Complete!
bash-5.2#
```

Updates in the latest version

In diesem Beispiel werden wir Updates anwenden, die in der `latest` Version AL2 023 verfügbar sind, wenn wir einen Container der Version 2023.4.20240319 gestartet haben. Zum Zeitpunkt der Erstellung dieses Artikels ist die `latest` Version 2023.5.20240708, sodass die in diesem Beispiel aufgelisteten Updates ab dieser Version verfügbar sein werden.

> **Note**
>
> In diesem Beispiel werden die Versionen 2023.4.20240319 und 2023.5.20240708 verwendet, wobei letztere die neueste Version zum Zeitpunkt der Erstellung dieses Artikels ist. Weitere Informationen zu den neuesten Versionen finden Sie in den Versionshinweisen zu 023. AL2

In diesem Beispiel beginnen wir mit einem Container-Image für die Version 2023.4.20240319.

Zuerst holen wir uns dieses Container-Image aus der Container-Registry. Das `.1` am Ende gibt die Version des Images für eine bestimmte Version an. Während die Image-Version normalerweise Null ist, wird in diesem Beispiel eine Version verwendet, bei der es sich um eine Image-Version handelt.

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
   2023.4.20240319.1: Pulling from amazonlinux/amazonlinux
6de065fda9a2: Pull complete
Digest: sha256:b4838c4cc9211d966b6ea158dacc9eda7433a16ba94436508c2d9f01f7658b4e
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.4.20240319.1
public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
```

Wir können jetzt eine Shell innerhalb des Containers erzeugen, von der aus wir Updates anwenden werden.

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
   bash-5.2#
```

Der `dnf upgrade` Befehl wird jetzt verwendet, um Updates anzuwenden, die in der `latest` Version verfügbar waren, die zum Zeitpunkt der Erstellung dieses Artikels 2023.5.20240708 lautete.

> **Note**
>
> Das Anwenden von Paket-Updates ist ein privilegierter Vorgang. Eine Erhöhung der Rechte ist bei der Ausführung in einem Container in der Regel nicht erforderlich. Wenn Sie jedoch in einer nicht containerisierten Umgebung wie einer EC2 Amazon-Instance ausgeführt werden, müssen Sie den `dnf upgrade` Befehl als Benutzer ausführen. `root` Dies kann mit den Befehlen oder erfolgen. `sudo su`

Standardmäßig `dnf` werden Sie aufgefordert, zu bestätigen, dass Sie die Updates anwenden möchten. In diesem Beispiel umgehen wir diese Aufforderung, indem wir das `-y` Flag to `dnf` verwenden.

```
$ dnf -y --releasever=latest update
Amazon Linux 2023 repository                    75 MB/s |  25 MB     00:00
Last metadata expiration check: 0:00:04 ago on Mon Jul 22 18:00:10 2024.
Dependencies resolved.
================================================================================
 Package                 Arch    Version                       Repository    Size
================================================================================
Upgrading:
 amazon-linux-repo-cdn   noarch  2023.5.20240708-1.amzn2023    amazonlinux   17 k
 curl-minimal            x86_64  8.5.0-1.amzn2023.0.4          amazonlinux  160 k
 dnf                     noarch  4.14.0-1.amzn2023.0.5         amazonlinux  460 k
 dnf-data                noarch  4.14.0-1.amzn2023.0.5         amazonlinux   34 k
 expat                   x86_64  2.5.0-1.amzn2023.0.4          amazonlinux  117 k
 glibc                   x86_64  2.34-52.amzn2023.0.10         amazonlinux  1.9 M
 glibc-common            x86_64  2.34-52.amzn2023.0.10         amazonlinux  295 k
 glibc-minimal-langpack  x86_64  2.34-52.amzn2023.0.10         amazonlinux   23 k
 krb5-libs               x86_64  1.21-3.amzn2023.0.4           amazonlinux  758 k
 libblkid                x86_64  2.37.4-1.amzn2023.0.4         amazonlinux  105 k
 libcurl-minimal         x86_64  8.5.0-1.amzn2023.0.4          amazonlinux  275 k
 libmount                x86_64  2.37.4-1.amzn2023.0.4         amazonlinux  132 k
 libnghttp2              x86_64  1.59.0-3.amzn2023.0.1         amazonlinux   79 k
 libsmartcols            x86_64  2.37.4-1.amzn2023.0.4         amazonlinux   62 k
 libuuid                 x86_64  2.37.4-1.amzn2023.0.4         amazonlinux   26 k
 openssl-libs            x86_64  1:3.0.8-1.amzn2023.0.12       amazonlinux  2.2 M
```

```
 python3                 x86_64  3.9.16-1.amzn2023.0.8        amazonlinux   27 k
 python3-dnf             noarch  4.14.0-1.amzn2023.0.5        amazonlinux  409 k
 python3-libs            x86_64  3.9.16-1.amzn2023.0.8        amazonlinux  7.3 M
 system-release          noarch  2023.5.20240708-1.amzn2023   amazonlinux   28 k
 yum                     noarch  4.14.0-1.amzn2023.0.5        amazonlinux   32 k

 Transaction Summary
================================================================================
Upgrade  21 Packages

Total download size: 14 M
Downloading Packages:
(1/21): amazon-linux-repo-cdn-2023.5.20240708-1. 345 kB/s |  17 kB     00:00
(2/21): dnf-4.14.0-1.amzn2023.0.5.noarch.rpm     6.8 MB/s | 460 kB     00:00
(3/21): dnf-data-4.14.0-1.amzn2023.0.5.noarch.rp 1.6 MB/s |  34 kB     00:00
(4/21): expat-2.5.0-1.amzn2023.0.4.x86_64.rpm    4.6 MB/s | 117 kB     00:00
(5/21): glibc-2.34-52.amzn2023.0.10.x86_64.rpm    38 MB/s | 1.9 MB     00:00
(6/21): glibc-common-2.34-52.amzn2023.0.10.x86_6 8.8 MB/s | 295 kB     00:00
(7/21): glibc-minimal-langpack-2.34-52.amzn2023. 1.7 MB/s |  23 kB     00:00
(8/21): curl-minimal-8.5.0-1.amzn2023.0.4.x86_64 998 kB/s | 160 kB     00:00
(9/21): libblkid-2.37.4-1.amzn2023.0.4.x86_64.rp 4.1 MB/s | 105 kB     00:00
(10/21): krb5-libs-1.21-3.amzn2023.0.4.x86_64.rp  16 MB/s | 758 kB     00:00
(11/21): libmount-2.37.4-1.amzn2023.0.4.x86_64.r 7.9 MB/s | 132 kB     00:00
(12/21): libnghttp2-1.59.0-3.amzn2023.0.1.x86_64 5.6 MB/s |  79 kB     00:00
(13/21): libsmartcols-2.37.4-1.amzn2023.0.4.x86_ 4.4 MB/s |  62 kB     00:00
(14/21): libcurl-minimal-8.5.0-1.amzn2023.0.4.x8 7.1 MB/s | 275 kB     00:00
(15/21): libuuid-2.37.4-1.amzn2023.0.4.x86_64.rp 1.1 MB/s |  26 kB     00:00
(16/21): python3-3.9.16-1.amzn2023.0.8.x86_64.rp 1.5 MB/s |  27 kB     00:00
(17/21): python3-dnf-4.14.0-1.amzn2023.0.5.noarc  19 MB/s | 409 kB     00:00
(18/21): system-release-2023.5.20240708-1.amzn20 1.9 MB/s |  28 kB     00:00
(19/21): yum-4.14.0-1.amzn2023.0.5.noarch.rpm    1.6 MB/s |  32 kB     00:00
(20/21): openssl-libs-3.0.8-1.amzn2023.0.12.x86_  26 MB/s | 2.2 MB     00:00
(21/21): python3-libs-3.9.16-1.amzn2023.0.8.x86_  59 MB/s | 7.3 MB     00:00
--------------------------------------------------------------------------------
Total                                             34 MB/s |  14 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Upgrading        : glibc-common-2.34-52.amzn2023.0.10.x86_64              1/42
  Upgrading        : glibc-minimal-langpack-2.34-52.amzn2023.0.10.x86_64    2/42
  Running scriptlet: glibc-2.34-52.amzn2023.0.10.x86_64                     3/42
```

```
  Upgrading        : glibc-2.34-52.amzn2023.0.10.x86_64                    3/42
  Running scriptlet: glibc-2.34-52.amzn2023.0.10.x86_64                    3/42
  Upgrading        : libuuid-2.37.4-1.amzn2023.0.4.x86_64                  4/42
  Upgrading        : openssl-libs-1:3.0.8-1.amzn2023.0.12.x86_64           5/42
  Upgrading        : krb5-libs-1.21-3.amzn2023.0.4.x86_64                  6/42
  Upgrading        : libblkid-2.37.4-1.amzn2023.0.4.x86_64                 7/42
  Running scriptlet: libblkid-2.37.4-1.amzn2023.0.4.x86_64                 7/42
  Upgrading        : expat-2.5.0-1.amzn2023.0.4.x86_64                     8/42
  Upgrading        : python3-3.9.16-1.amzn2023.0.8.x86_64                  9/42
  Upgrading        : python3-libs-3.9.16-1.amzn2023.0.8.x86_64            10/42
  Upgrading        : libnghttp2-1.59.0-3.amzn2023.0.1.x86_64              11/42
  Upgrading        : libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64          12/42
  Upgrading        : system-release-2023.5.20240708-1.amzn2023.noarch     13/42
  Upgrading        : amazon-linux-repo-cdn-2023.5.20240708-1.amzn2023.no  14/42
  Upgrading        : dnf-data-4.14.0-1.amzn2023.0.5.noarch                15/42
  Upgrading        : python3-dnf-4.14.0-1.amzn2023.0.5.noarch             16/42
  Upgrading        : dnf-4.14.0-1.amzn2023.0.5.noarch                     17/42
  Running scriptlet: dnf-4.14.0-1.amzn2023.0.5.noarch                     17/42
  Upgrading        : yum-4.14.0-1.amzn2023.0.5.noarch                     18/42
  Upgrading        : curl-minimal-8.5.0-1.amzn2023.0.4.x86_64             19/42
  Upgrading        : libmount-2.37.4-1.amzn2023.0.4.x86_64                20/42
  Upgrading        : libsmartcols-2.37.4-1.amzn2023.0.4.x86_64            21/42
  Cleanup          : yum-4.14.0-1.amzn2023.0.4.noarch                     22/42
  Running scriptlet: dnf-4.14.0-1.amzn2023.0.4.noarch                     23/42
  Cleanup          : dnf-4.14.0-1.amzn2023.0.4.noarch                     23/42
  Running scriptlet: dnf-4.14.0-1.amzn2023.0.4.noarch                     23/42
  Cleanup          : python3-dnf-4.14.0-1.amzn2023.0.4.noarch             24/42
  Cleanup          : amazon-linux-repo-cdn-2023.4.20240319-1.amzn2023.no  25/42
  Cleanup          : libmount-2.37.4-1.amzn2023.0.3.x86_64                26/42
  Cleanup          : curl-minimal-8.5.0-1.amzn2023.0.2.x86_64             27/42
  Cleanup          : libcurl-minimal-8.5.0-1.amzn2023.0.2.x86_64          28/42
  Cleanup          : krb5-libs-1.21-3.amzn2023.0.3.x86_64                 29/42
  Cleanup          : libblkid-2.37.4-1.amzn2023.0.3.x86_64                30/42
  Cleanup          : libnghttp2-1.57.0-1.amzn2023.0.1.x86_64              31/42
  Cleanup          : libsmartcols-2.37.4-1.amzn2023.0.3.x86_64            32/42
  Cleanup          : system-release-2023.4.20240319-1.amzn2023.noarch     33/42
  Cleanup          : dnf-data-4.14.0-1.amzn2023.0.4.noarch                34/42
  Cleanup          : python3-3.9.16-1.amzn2023.0.6.x86_64                 35/42
  Cleanup          : python3-libs-3.9.16-1.amzn2023.0.6.x86_64            36/42
  Cleanup          : openssl-libs-1:3.0.8-1.amzn2023.0.11.x86_64          37/42
  Cleanup          : libuuid-2.37.4-1.amzn2023.0.3.x86_64                 38/42
  Cleanup          : expat-2.5.0-1.amzn2023.0.3.x86_64                    39/42
  Cleanup          : glibc-2.34-52.amzn2023.0.8.x86_64                    40/42
  Cleanup          : glibc-minimal-langpack-2.34-52.amzn2023.0.8.x86_64   41/42
```

```
Cleanup          : glibc-common-2.34-52.amzn2023.0.8.x86_64              42/42
Running scriptlet: glibc-common-2.34-52.amzn2023.0.8.x86_64              42/42
Verifying        : amazon-linux-repo-cdn-2023.5.20240708-1.amzn2023.no    1/42
Verifying        : amazon-linux-repo-cdn-2023.4.20240319-1.amzn2023.no    2/42
Verifying        : curl-minimal-8.5.0-1.amzn2023.0.4.x86_64               3/42
Verifying        : curl-minimal-8.5.0-1.amzn2023.0.2.x86_64               4/42
Verifying        : dnf-4.14.0-1.amzn2023.0.5.noarch                       5/42
Verifying        : dnf-4.14.0-1.amzn2023.0.4.noarch                       6/42
Verifying        : dnf-data-4.14.0-1.amzn2023.0.5.noarch                  7/42
Verifying        : dnf-data-4.14.0-1.amzn2023.0.4.noarch                  8/42
Verifying        : expat-2.5.0-1.amzn2023.0.4.x86_64                      9/42
Verifying        : expat-2.5.0-1.amzn2023.0.3.x86_64                     10/42
Verifying        : glibc-2.34-52.amzn2023.0.10.x86_64                    11/42
Verifying        : glibc-2.34-52.amzn2023.0.8.x86_64                     12/42
Verifying        : glibc-common-2.34-52.amzn2023.0.10.x86_64             13/42
Verifying        : glibc-common-2.34-52.amzn2023.0.8.x86_64              14/42
Verifying        : glibc-minimal-langpack-2.34-52.amzn2023.0.10.x86_64   15/42
Verifying        : glibc-minimal-langpack-2.34-52.amzn2023.0.8.x86_64    16/42
Verifying        : krb5-libs-1.21-3.amzn2023.0.4.x86_64                  17/42
Verifying        : krb5-libs-1.21-3.amzn2023.0.3.x86_64                  18/42
Verifying        : libblkid-2.37.4-1.amzn2023.0.4.x86_64                 19/42
Verifying        : libblkid-2.37.4-1.amzn2023.0.3.x86_64                 20/42
Verifying        : libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64           21/42
Verifying        : libcurl-minimal-8.5.0-1.amzn2023.0.2.x86_64           22/42
Verifying        : libmount-2.37.4-1.amzn2023.0.4.x86_64                 23/42
Verifying        : libmount-2.37.4-1.amzn2023.0.3.x86_64                 24/42
Verifying        : libnghttp2-1.59.0-3.amzn2023.0.1.x86_64               25/42
Verifying        : libnghttp2-1.57.0-1.amzn2023.0.1.x86_64               26/42
Verifying        : libsmartcols-2.37.4-1.amzn2023.0.4.x86_64             27/42
Verifying        : libsmartcols-2.37.4-1.amzn2023.0.3.x86_64             28/42
Verifying        : libuuid-2.37.4-1.amzn2023.0.4.x86_64                  29/42
Verifying        : libuuid-2.37.4-1.amzn2023.0.3.x86_64                  30/42
Verifying        : openssl-libs-1:3.0.8-1.amzn2023.0.12.x86_64           31/42
Verifying        : openssl-libs-1:3.0.8-1.amzn2023.0.11.x86_64           32/42
Verifying        : python3-3.9.16-1.amzn2023.0.8.x86_64                  33/42
Verifying        : python3-3.9.16-1.amzn2023.0.6.x86_64                  34/42
Verifying        : python3-dnf-4.14.0-1.amzn2023.0.5.noarch              35/42
Verifying        : python3-dnf-4.14.0-1.amzn2023.0.4.noarch              36/42
Verifying        : python3-libs-3.9.16-1.amzn2023.0.8.x86_64             37/42
Verifying        : python3-libs-3.9.16-1.amzn2023.0.6.x86_64             38/42
Verifying        : system-release-2023.5.20240708-1.amzn2023.noarch      39/42
Verifying        : system-release-2023.4.20240319-1.amzn2023.noarch      40/42
Verifying        : yum-4.14.0-1.amzn2023.0.5.noarch                      41/42
Verifying        : yum-4.14.0-1.amzn2023.0.4.noarch                      42/42
```

```
Upgraded:
  amazon-linux-repo-cdn-2023.5.20240708-1.amzn2023.noarch
  curl-minimal-8.5.0-1.amzn2023.0.4.x86_64
  dnf-4.14.0-1.amzn2023.0.5.noarch
  dnf-data-4.14.0-1.amzn2023.0.5.noarch
  expat-2.5.0-1.amzn2023.0.4.x86_64
  glibc-2.34-52.amzn2023.0.10.x86_64
  glibc-common-2.34-52.amzn2023.0.10.x86_64
  glibc-minimal-langpack-2.34-52.amzn2023.0.10.x86_64
  krb5-libs-1.21-3.amzn2023.0.4.x86_64
  libblkid-2.37.4-1.amzn2023.0.4.x86_64
  libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64
  libmount-2.37.4-1.amzn2023.0.4.x86_64
  libnghttp2-1.59.0-3.amzn2023.0.1.x86_64
  libsmartcols-2.37.4-1.amzn2023.0.4.x86_64
  libuuid-2.37.4-1.amzn2023.0.4.x86_64
  openssl-libs-1:3.0.8-1.amzn2023.0.12.x86_64
  python3-3.9.16-1.amzn2023.0.8.x86_64
  python3-dnf-4.14.0-1.amzn2023.0.5.noarch
  python3-libs-3.9.16-1.amzn2023.0.8.x86_64
  system-release-2023.5.20240708-1.amzn2023.noarch
  yum-4.14.0-1.amzn2023.0.5.noarch

Complete!
bash-5.2#
```

Um AL2 023-Updates zu finden, führen Sie einen oder mehrere der folgenden Schritte aus:

- Führen Sie den Befehl `dnf check-update` aus. Dadurch wird nach nicht angewendeten Updates in der Version von Amazon Linux gesucht, für die Sie gesperrt sind. Möglicherweise werden Updates angezeigt, wenn Sie nur das `system-release` Paket aktualisiert haben und dabei die Version der Repositorys verschoben haben, für die die Instance gesperrt ist, aber keine der darin verfügbaren Updates angewendet haben.
- 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 für Version AL2 023.

- Entdecken Sie neue Versionen von[Suche nach neueren Repository-Versionen mit dnf check-release-update](#).

> **Important**
>
> Neue Versionen von AL2 023, die Sicherheitsupdates enthalten, werden häufig veröffentlicht. Halten Sie sich unbedingt über die entsprechenden Sicherheitspatches auf dem Laufenden.

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 Folgendes aus DNF Befehl (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

`smart-restart`Bietet zusätzlich zur Denylisting 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
```

Wann ist ein Neustart erforderlich, um Sicherheitsupdates anzuwenden?

In einigen Situationen erfordert Amazon Linux einen Neustart, um Updates anzuwenden:

- Aktualisierungen des Linux-Kernelpakets erfordern einen Neustart, um den neuen Kernel mit den neuesten Sicherheitsupdates zu aktivieren. Durch Kernel-Livepatching können Sie Sicherheitsupdates möglicherweise für einen begrenzten Zeitraum verschieben. Einzelheiten finden Sie unter. Kernel Live Patching auf 023 AL2
- Auf EC2 Metal-Instances stellt Amazon Linux Mikrocode-Updates bereit (über das `microcode_ctl` Paket für Intel CPUs und das `amd-ucode-firmware` Paket für AMD) CPUs. Diese Mikrocode-Updates werden nur bei nachfolgenden Instance-Neustarts aktiviert. Bei virtualisierten EC2 Instanzen wickelt das zugrunde liegende AWS Nitro-System Mikrocode-Updates für Sie ab.

- Einige laufende Systemd-Dienste funktionieren erst nach einem vollständigen Systemneustart korrekt. Der `smart-restart` Mechanismus informiert Sie über solche Situationen, indem er Ihnen Hinweise zum Neustart gibt. Siehe Automatischer Neustart des Dienstes nach (Sicherheits-) Updates.

Starten einer Instance mit aktivierter neuester Repository-Version

Sie können hinzufügen DNF Befehle zu einem Benutzerdatenskript, um zu steuern, was RPM Pakete werden auf einem Amazon Linux-AMI installiert, wenn es gestartet wird. 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 upgrade --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 EC2 Amazon-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. Um Ihnen Amazon Linux-spezifische Informationen zu diesen verschiedenen Paketen zur Verfügung zu stellen, DNF `supportinfo`Das Plugin stellt 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.
```

Informationen zur Paketunterstützung finden Sie auch im Abschnitt Support Statements der AL2023-Versionshinweise.

Suche nach neueren Repository-Versionen mit **`dnf check-release-update`**

In einer AL2 023-Instanz können Sie den verwenden DNF Hilfsprogramm zum Verwalten von Repositorys und zum Anwenden aktualisierter Dateien RPM Pakete. Diese Pakete sind in den Amazon-Linux-Repositorys verfügbar. Sie können das DNF Befehl `dnf check-release-update` zur Suche nach neuen Versionen von DNF Repository.

> **Note**
>
> AL2023 Container-Images enthalten den `dnf check-release-update` Befehl standardmäßig nicht.
>
> ```
> $ dnf check-release-update
> No such command: check-release-update. Please use /usr/bin/dnf --help
> It could be a DNF plugin command, try: "dnf install 'dnf-command(check-release-
> update)'"
> ```

Wenn `dnf install 'dnf-command(check-release-update)'` es ausgeführt wird, `dnf` wird das Paket installiert, das den `check-release-update` Befehl bereitstellt, also das `dnf-plugin-release-notification` Paket. Im folgenden Beispiel wird das `-q` Argument angegeben, dass `dnf` es leise ausgegeben werden soll.

```
$ dnf -y -q install 'dnf-command(check-release-update)'
Installed:
  dnf-plugin-release-notification-1.2-1.amzn2023.0.2.noarch
```

In nicht containerisierten Umgebungen wie einer EC2 Amazon-Instance ist der `check-release-update` Befehl standardmäßig enthalten.

```
$ 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 upgrade --releasever=2023.0.20230210

    Release notes:
     https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes.html
```

Dies gibt eine vollständige Liste aller neueren Versionen von DNF Repositorys, die verfügbar sind. Wenn nichts zurückgegeben wird, bedeutet das DNF ist derzeit so konfiguriert, dass die neueste verfügbare Version verwendet wird. Die Version des aktuell installierten `system-release` Pakets legt fest `releasever` DNF variabel. Mit folgendem Befehl können Sie die aktuelle Repository-Version abfragen.

```
$ rpm -q system-release --qf "%{VERSION}\n"
```

Wenn du rennst DNF Bei Pakettransaktionen (wie Befehle zum Installieren, Aktualisieren oder Entfernen) werden Sie mit einer Warnmeldung über neue Repository-Versionen informiert. Wenn Sie

das `httpd` Paket beispielsweise auf einer Instance installieren, die von einer älteren Version von AL2 023 aus gestartet wurde, wird die folgende Ausgabe 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

> **Warning**
>
> Fügen Sie nur Repositorys hinzu, die für die Verwendung mit AL2 023 konzipiert sind. Auch wenn Repositorys, die für andere Distributionen entworfen wurden, heute funktionieren, gibt es keine Garantie dafür, dass sie dies auch weiterhin tun werden, wenn ein Paket-Update in AL2 023 oder ein Projektarchiv nicht für die Verwendung mit 023 konzipiert ist. AL2

Um ein Paket aus einem anderen Repository als den standardmäßigen Amazon Linux-Repositorys zu installieren, müssen Sie das DNF Paketverwaltungssystem so konfigurieren, dass es weiß, wo sich das Repository befindet

Um `dnf` mehr über ein Paket-Repository zu erfahren, fügen Sie die Repository-Informationen zu einer Konfigurationsdatei für dieses Repository im Verzeichnis hinzu. `/etc/yum.repos.d/` Viele Repositorys von Drittanbietern bieten entweder den Inhalt der Konfigurationsdatei oder ein installierbares Paket, das die Konfigurationsdatei enthält.

> **Note**
>
> Repositorys können zwar direkt in der `/etc/dnf/dnf.conf` Datei konfiguriert werden, dies wird jedoch nicht empfohlen. Es wird empfohlen, jedes Repository in einer eigenen Datei in `/etc/yum.repos.d/` zu konfigurieren.

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.

Um ein yum Repository in zu aktivieren`/etc/yum.repos.d`, 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
```

Kernel Live Patching auf 023 AL2

Sie können Kernel Live Patching für AL2 023 verwenden, um bestimmte 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 Ihrer Anwendung zu verbessern und gleichzeitig diese Fixes anzuwenden, bis das System neu gestartet werden kann.

AWS veröffentlicht zwei Arten von Kernel-Live-Patches für 023: AL2

- 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.
- Fehlerkorrekturen — Beinhaltet Korrekturen für kritische Fehler und Stabilitätsprobleme, die nicht damit CVEs zusammenhängen.

AWS bietet Kernel-Live-Patches für eine AL2 023-Kernel-Version für bis zu 3 Monate nach ihrer Veröffentlichung. 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 AL2 023-Repositorys zur Verfügung gestellt. Die Patches können mithilfe vorhandener DNF-Paketmanager-Workflows auf einzelnen Instances installiert werden. Sie können auch mit AWS Systems Manager auf einer Gruppe verwalteter Instanzen installiert werden.

Kernel Live Patching auf AL2 023 wird ohne zusätzliche Kosten bereitgestellt.

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.

> **Note**
>
> Aufgrund technischer Einschränkungen können einige Probleme nicht mit Live-Patching behoben werden. Aus diesem Grund werden diese Fixes nicht im Kernel-Live-Patch-Paket, sondern nur im nativen Kernel-Paket-Update mitgeliefert. Sie können das native Kernel-Paket installieren und das System aktualisieren und neu starten, um die Patches wie gewohnt zu aktivieren.

Unterstützte Konfigurationen und Voraussetzungen

Kernel Live Patching wird auf EC2 Amazon-Instances und lokalen virtuellen Maschinen unterstützt, auf denen 023 ausgeführt wird AL2.

Um Kernel Live Patching auf AL2 023 zu verwenden, müssen Sie Folgendes verwenden:

- Eine 64-Bit `x86_64`- oder `ARM64`-Architektur
- Kernel-Version `6.1`

Richtlinienanforderungen

Um Pakete aus AL2 023 Repositorys herunterzuladen, EC2 benötigt Amazon 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- -de612dc2/* *region*	Amazon S3 S3-Bucket mit AL2 023 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 Verwendung von Kernel Live Patching für eine Gruppe verwalteter Instances finden Sie unter Verwenden von Kernel Live Patching auf AL2 023-Instances im Benutzerhandbuch.AWS Systems Manager

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 auf 023 standardmäßig deaktiviert. AL2 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 AL2 023 mit Kernel-Version verfügbar. `6.1` 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 anfängliche leere Kernel-Live-Patch-RPM (und ein weiteres RPM, das das DNF-Repository mit den Livepatches einrichtet) enthält.

   ```
   $ sudo rpm -qa | grep kernel-livepatch
   kernel-livepatch-repo-s3-2023.7.20250428-0.amzn2023.noarch
   kernel-livepatch-6.1.134-150.224-1.0-0.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 upgrade 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 zu den AL2 023-Sicherheitswarnungen, einschließlich Warnungen für Kernel-Live-Patches, 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 sich auch die verfügbaren Kernel-Live-Patches ansehen, um Hinweise zu erhalten und die Befehlszeile zu CVEs verwenden.

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
```

Um alle verfügbaren Kernel-Live-Patches aufzulisten für CVEs

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 Kernel-Live-Patches, die angewendet werden können.

> **Tip**
>
> Wir empfehlen Ihnen, Ihren Kernel regelmäßig mit Kernel Live Patching zu aktualisieren, um sicherzustellen, dass er bestimmte wichtige und kritische Sicherheitsupdates erhält, bis das System neu gestartet werden kann. Bitte überprüfen Sie auch, ob zusätzliche Fixes für das native Kernel-Paket verfügbar gemacht wurden, die nicht als Live-Patches bereitgestellt werden können, und führen Sie in diesen Fällen ein Update durch und starten Sie das Kernel-Update neu.

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 AL2 023-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 AL2 023-Kernelversion an `6.1.12-17.42`

   ```
   $ 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 upgrade --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 AL2 023-Kernel erhält Kernel-Live-Patches für 3 Monate. 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
The current version of the Linux kernel you are running will no longer receive
 live patches after 2025-07-22.
```

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. Deaktiviere das kpatch Dienst:

     ```
     $ sudo systemctl disable --now kpatch.service
     ```

- Entfernen Sie das vollständig livepatch Werkzeuge:
  1. Das Plugin entfernen:

     ```
     $ sudo dnf remove kpatch-dnf
     ```

  2. Remove kpatch-runtime:

     ```
     $ sudo dnf remove kpatch-runtime
     ```

  3. Entfernen Sie alle installierten livepatches:

     ```
     $ sudo dnf remove kernel-livepatch\*
     ```

Aktualisierung des Linux-Kernels auf AL2 023

Themen

- Linux-Kernel-Versionen auf 023 AL2
- Aktualisierung von 0.23 auf Kernel 6.12 AL2
- AL2023 Kernel — Häufig gestellte Fragen

Linux-Kernel-Versionen auf 023 AL2

AL2023 enthält regelmäßig neue Kernel-Versionen, die auf Long-Term Support (LTS) -Versionen des Linux-Kernels basieren.

AL2023 wurde ursprünglich im März 2023 mit Kernel 6.1 veröffentlicht.

Im April 2025 fügte AL2 023 Unterstützung für den Linux-Kernel 6.12 hinzu. Dieser Kernel fügte neue Funktionen hinzu, darunter EEVDF-Planung, I/O FUSE-Passthrough-Unterstützung, eine neue Futex-API und Verbesserungen in eBPF. Kernel 6.12 ermöglicht es einem Userspace-Programm außerdem, sich während der Laufzeit mithilfe von User-Space-Shadow-Stacks und Memory Sealing selbst zu sichern.

Aktualisierung von 0.23 auf Kernel 6.12 AL2

Sie können AL2 023 mit Kernel 6.12 ausführen, indem Sie entweder ein AMI mit vorinstalliertem Kernel 6.12 auswählen oder eine bestehende 023-Instanz aktualisieren. AL2 EC2

Ein AL2 023-Kernel-6.12-AMI ausführen

Sie können wählen, ob Sie ein AL2 023-AMI mit vorinstalliertem Kernel 6.12 über die AWS-Konsole ausführen möchten oder indem Sie SSM nach bestimmten Parametern abfragen. Die SSM-Schlüssel, mit denen abgefragt werden soll, beginnen, gefolgt von einem der `/aws/service/ami-amazon-linux-latest/`

- `al2023-ami-kernel-6.12-arm64` für arm64-Architektur
- `al2023-ami-minimal-kernel-6.12-arm64` für arm64-Architektur (Minimal AMI)
- `al2023-ami-kernel-6.12-x86_64` für x86_64-Architektur
- `al2023-ami-minimal-kernel-6.12-x86_64` für x86_64-Architektur (Minimal AMI)

Einzelheiten AL2023 mit dem SSM-Parameter starten und AWS CLI zur Auswahl von AL2 AMIs 023 finden Sie unter.

Aktualisierung einer AL2 023-Instanz auf Kernel 6.12

Sie können eine laufende AL2 023-Instanz mit den folgenden Schritten direkt auf Kernel 6.12 aktualisieren:

1. Installieren Sie das Paket `kernel6.12`:

```
$ sudo dnf install -y kernel6.12
```

2. Holen Sie sich die neueste Version des Pakets: `kernel6.12`

```
$ version=$(rpm -q --qf '%{version}-%{release}.%{arch}\n' kernel6.12 | sort -V |
 tail -1)
```

3. Machen Sie den neuen Kernel `kernel6.12` zu Ihrem Standard-Kernel:

```
$ sudo grubby --set-default "/boot/vmlinuz-$version"
```

4. Starte dein System neu:

```
$ sudo reboot
```

5. Deinstallieren Sie Kernel 6.1:

```
$ sudo dnf remove -y kernel
```

6. Ersetzen Sie zusätzliche Kernel-Pakete durch ihre Kernel-6.12-Entsprechungen:

```
$ declare -A pkgs
$ pkgs=(
[bpftool]=bpftool6.12
[kernel-debuginfo]=kernel6.12-debuginfo
[kernel-debuginfo-common]=kernel6.12-debuginfo-common
[kernel-headers]=kernel6.12-headers
[kernel-libbpf]=kernel6.12-libbpf
[kernel-libbpf-devel]=kernel6.12-libbpf-devel
[kernel-libbpf-static]=kernel6.12-libbpf-static
[kernel-modules-extra-common]=kernel6.12-modules-extra-common
[kernel-tools]=kernel6.12-tools
[kernel-tools-devel]=kernel6.12-tools-devel
[perf]=perf6.12
[python3-perf]=python3-perf6.12
)
$ for pkg in "${!pkgs[@]}"; do
  rpm -q $pkg && sudo dnf -y swap $pkg "${pkgs["$pkg"]}" ;
done
```

7. (Optional) Deinstallieren Sie kernel-devel für Kernel 6.1:

```
$ rpm -q kernel-devel && sudo dnf remove -y kernel-devel
```

Downgrade von Kernel 6.12 auf Kernel 6.1

Wenn Sie zu irgendeinem Zeitpunkt ein Downgrade zurück auf Kernel 6.1 durchführen müssen, gehen Sie wie folgt vor:

1. Ersetzen Sie zusätzliche Kernel6.12-Pakete durch ihre Kernel-6.1-Entsprechungen:

```
$ declare -A pkgs
$ pkgs=(
[bpftool]=bpftool6.12
[kernel-debuginfo]=kernel6.12-debuginfo
[kernel-debuginfo-common]=kernel6.12-debuginfo-common
[kernel-headers]=kernel6.12-headers
[kernel-libbpf]=kernel6.12-libbpf
[kernel-libbpf-devel]=kernel6.12-libbpf-devel
[kernel-libbpf-static]=kernel6.12-libbpf-static
[kernel-modules-extra-common]=kernel6.12-modules-extra-common
[kernel-tools]=kernel6.12-tools
[kernel-tools-devel]=kernel6.12-tools-devel
[perf]=perf6.12
[python3-perf]=python3-perf6.12
)
$ for pkg in "${!pkgs[@]}"; do
  rpm -q "${pkgs["$pkg"]}" && sudo dnf -y swap "${pkgs["$pkg"]}" $pkg ;
done
```

2. Installieren Sie das Paket `kernel`:

```
$ sudo dnf install -y kernel
```

3. Holen Sie sich die neueste Version des Pakets: `kernel`

```
$ version=$(rpm -q --qf '%{version}-%{release}.%{arch}\n' kernel | sort -V | tail
 -1)
```

4. Machen Sie Kernel 6.1 zu Ihrem Standard-Kernel:

```
$ sudo grubby --set-default "/boot/vmlinuz-$version"
```

5. Starte dein System neu:

```
$ sudo reboot
```

6. Deinstallieren Sie Kernel 6.12:

```
$ sudo dnf remove -y kernel6.12
```

AL2023 Kernel — Häufig gestellte Fragen

1. Muss ich nach einem Kernel-Update einen Neustart durchführen?

Jede Änderung am laufenden Kernel erfordert einen Neustart.

2. Wie verwalte ich Kernel up-to-date über mehrere Instanzen hinweg?

Amazon Linux bietet keine Funktionen zur Verwaltung von Instance-Flotten. Wir empfehlen Ihnen, große Flotten mit Tools wie AWS Systems Manager zu patchen.

3. Wie überprüfe ich, welche Kernel-Version ich gerade verwende?

Führen Sie diesen Befehl auf Ihrer AL2 023-Instanz aus:

```
$ uname -r
```

4. Wie installiere ich Kernel-Header, Entwicklungspakete und zusätzliche Module für Kernel 6.12?

Bitte starte:

```
$ sudo dnf install -y kernel6.12-modules-extra-$(uname -r) kernel6.12-headers-$(uname -r) kernel6.12-devel-$(uname -r)
```

Erste Schritte mit der Programmierung von Laufzeiten auf 023 AL2

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 AL2 023 existieren.

Themen

- CC++, und Fortran im Jahr AL2 203
- Go im Jahr AL2 023
- Javaim Jahr AL2 023
- NodeJS im Jahr AL2 023
- Perl im Jahr AL2 023
- PHP im Jahr AL2 023
- Python im Jahr AL2 023
- Rust im Jahr AL2 023

CC++, und Fortran im Jahr AL2 203

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 023 konstant bleiben. AL2 Kleinere Versionen enthalten Fehlerkorrekturen und sind möglicherweise in AL2 023-Versionen enthalten. Andere Fehler-, Leistungs- und Sicherheitskorrekturen werden möglicherweise auf die Hauptversion zurückportiert, die in GCC AL2 023 ausgeliefert wird.

AL2023 enthält Version 11 von GCC als Standard-Compiler mit den Frontends C (`gcc`), C++ () und Fortran (`g++`). `gfortran` Darüber hinaus bietet AL2 023 GCC Version 14 als optionalen alternativen Compiler, der zusammen mit der Standardversion installiert werden kann.

AL2023 aktiviert die Frontends Ada (`gnat`), Go (`gcc-go`), Objective-C oder Objective-C++ nicht.

Zu den Standard-Compiler-Flags, mit denen AL2 023 erstellt wurde, RPMs gehören Optimierungs- und Hardening-Flags. Um Ihren eigenen Code mit GCC zu erstellen, empfehlen wir Ihnen, Optimierungs- und Hardening-Flags zu verwenden.

> **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-Pakete glibc, gcc, binutils

Weitere Informationen zu AL2 023 und seiner Beziehung zu anderen Linux-Distributionen finden Sie unter. Beziehung zu Fedora

Weitere Informationen zur Änderung der Compiler-Tripletts in AL2 023 im Vergleich zu finden Sie unter. AL2 Compiler-Triplet

Themen

- GCC14
- Vergleich der Sprachstandardversionen

GCC14

AL2023 bietet GCC 14 als optionalen Compiler, der zusammen mit dem GCC Standard-Compiler 11 installiert werden kann. GCC14 enthält die neuesten Sprachfunktionen und Optimierungen und eignet sich daher für Projekte, die Unterstützung neuerer C-, C++- oder Fortran-Standards erfordern.

Verwenden Sie den folgenden GCC Befehl, um 14 zu installieren:

```
sudo dnf install gcc14 gcc14-c++ gcc14-gfortran
```

Die GCC 14 Compiler werden mit versionsspezifischen Befehlsnamen installiert, um Konflikte mit dem Standard 11 zu vermeiden: GCC

- `gcc14-gcc`- C-Compiler
- `gcc14-g++`- C++-Compiler
- `gcc14-gfortran`- Fortran-Compiler

Beispielverwendung:

```
gcc14-gcc -o myprogram myprogram.c
gcc14-g++ -o mycppprogram mycppprogram.cpp
gcc14-gfortran -o myfortranprogram myfortranprogram.f90
```

Sie können die installierte Version überprüfen, indem Sie Folgendes ausführen:

```
gcc14-gcc --version
```

Dadurch werden Versionsinformationen angezeigt, die den folgenden ähneln: `gcc14-gcc (GCC) 14.2.1 20250110 (Red Hat 14.2.1-7)`

> **Note**
>
> Sowohl GCC 11 als auch GCC 14 können gleichzeitig auf demselben System installiert werden. Die `gfortran` Standardbefehle `gccg++`, und verwenden weiterhin GCC 11, während auf GCC 14 über die versionsspezifischen Befehle zugegriffen wird.

Vergleich der Sprachstandardversionen

In der folgenden Tabelle werden die Standardversionen der Standardsprachen für verschiedene Amazon Linux-Versionen und GCC Compiler-Versionen verglichen:

Amazon Linux-Version	C-Standard (Standard)	C++-Standard (Standard)	Fortran-Standard
AL2 mit GCC 7 (Standard)	C11 (201112L)	C+14 (201402L)	Fortran 2008

Amazon Linux-Version	C-Standard (Standard)	C++-Standard (Standard)	Fortran-Standard
AL2 mit GCC 10 (optional)	C17/C18 (201710L)	C+14 (201402L)	Fortran 2008
AL2023 mit GCC 11 (Standard)	C17/C18 (201710L)	C+17 (201703L)	Fortran 2008
AL2023 mit GCC 14 (optional)	C17/C18 (201710L)	C+17 (201703L)	Fortran 2008

Wichtigste Verbesserungen je nach VersionGCC:

- GCC10 gegenüber GCC 7: Der Standard-C-Standard wurde von C11 auf C17/C18 aktualisiert, Unterstützung für C++20-Funktionen hinzugefügt und die Optimierungsmöglichkeiten verbessert.
- GCC11 vs GCC 10: Der Standard-C++-Standard wurde von C++14 auf C++17 aktualisiert, die C++20-Unterstützung verbessert und experimentelle C++23-Funktionen hinzugefügt.
- GCC14 gegenüber GCC 11: Vollständige C23-Standardunterstützung, erweiterte C++23-Funktionen, verbesserte Optimierung und bessere Standardkonformität hinzugefügt.

Unterstützte Sprachstandards:

- C-Standards: Alle Versionen unterstützen C90, C99, C11 und C17/C18. GCC10+ unterstützt C2x (Entwurf C23), während 14 volle C23-Unterstützung bietet. GCC
- C++-Standards: Alle Versionen unterstützen C++98, C++03, C++11, C++14, C++17 und C++20. GCC11+ bietet experimentelle C++23-Unterstützung, wobei 14 erweiterte C++23-Funktionen bietet. GCC
- Fortran-Standards: Alle Versionen unterstützen hauptsächlich Fortran 2008, wobei die Funktionen von Fortran 2018 je nach Version unterschiedlich sind. GCC

> **Note**
>
> Während die Standardstandards zwischen GCC 11 und 14 konsistent bleiben, bietet GCC 14 eine deutlich verbesserte Unterstützung von Sprachfunktionen, eine bessere Optimierung,

erweiterte Diagnosen und eine vollständigere Implementierung neuerer Standards, wenn sie ausdrücklich mithilfe von Flags angefordert werden. `-std=`

Go im Jahr AL2 023

Vielleicht möchten Sie Ihren eigenen Code erstellen, der in geschrieben ist Goauf Amazon Linux und möchten möglicherweise eine mit AL2 023 bereitgestellte Toolchain verwenden. Ähnlich AL2 wie AL2 023 aktualisiert 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.

Go ist eine Sprache, die sich relativ schnell bewegt. Es kann eine Situation geben, in der bestehende Anwendungen geschrieben wurden Go müssen sich an neue Versionen anpassen Go Toolchain. Weitere Informationen zur Go, siehe Go 1 und die Zukunft von Go Programme.

Obwohl AL2 023 neue Versionen von enthalten wird Go Die Toolchain wird während ihrer Laufzeit nicht im Gleichschritt mit der Upstream-Version laufen Go Veröffentlichungen. Daher verwenden Sie die Go Die in AL2 023 bereitgestellte Toolchain ist möglicherweise nicht geeignet, wenn Sie Folgendes erstellen möchten Go Code unter Verwendung modernster Funktionen von Go Sprache und Standardbibliothek.

Während der Lebensdauer von AL2 023 werden frühere Paketversionen nicht aus den Repositorien entfernt. Wenn ein früherer Go Wenn eine Toolchain erforderlich ist, können Sie sich dafür entscheiden, auf die Fehler- und Sicherheitskorrekturen neuerer Versionen zu verzichten Go Toolchains und installieren Sie eine frühere Version aus den Repositorys mit den gleichen Mechanismen, die für jedes RPM verfügbar sind.

Wenn Sie Ihre eigene bauen möchten Go Code auf AL2 023, du kannst den benutzen Go In AL2 023 enthaltene Toolchain mit dem Wissen, dass sich diese Toolchain während der Lebensdauer von 023 weiterentwickeln könnte. AL2

AL2023 Lambda-Funktionen, geschrieben in Go

Als Go kompiliert zu nativem Code, Lambda behandelt Go als benutzerdefinierte Laufzeit. Sie können die `provided.al2023` Runtime für die Bereitstellung verwenden Go funktioniert auf AL2 023 bis Lambda.

Weitere Informationen finden Sie unter Lambda-Funktionen erstellen mit Go im Entwicklerhandbuch für AWS Lambda .

Javaim Jahr AL2 023

AL2023 bietet mehrere Versionen von Amazon Corretto zur Unterstützung Java von basierten Workloads. Alle in AL2 023 enthaltenen Java basierten Pakete werden mit erstellt. Amazon Corretto 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 AL2 023 wird für den gleichen Zeitraum wie die Corretto-Version oder bis zum Lebensende von AL2 023 unterstützt, je nachdem, welcher Zeitpunkt früher eintritt. Weitere Informationen finden Sie in den Support-Erklärungen für Amazon Linux-Pakete und im Amazon Corretto FAQs.

NodeJS im Jahr AL2 023

NodeJSin AL2 023 wird durch die Versionen 18, 20 und 22 repräsentiert. Sie haben einen Namespace und können gleichzeitig auf demselben System installiert werden. NodeJS wird als mehrere Pakete verteilt, die den Knoten, das NPM-Tool einer damit kompatiblen Version, Dokumentation, Bibliotheken usw. enthalten. Zum Beispiel für NodeJS 18, node und npm werden von den `nodejs-npm` Paketen `nodejs` und bereitgestellt. Alle folgenden Versionen von NodeJS haben jedoch Paketnamen mit Namespace, die mit beginnen. `nodejs{MAJOR_VERSION}` Zum Beispiel NodeJS 20, wird mit Node und NPM geliefert, die jeweils als und verpackt sind. `nodejs20` `nodejs20-npm`

Um die gleichzeitige Installation verschiedener Hauptversionen von zu ermöglichen NodeJS, Pakete werden zusammen mit ausführbaren Dateien, Modulen und anderen Dateien mit Namespaces ausgeliefert, um Überschneidungen und Dateisystemkonflikte zu vermeiden. Beispielsweise wird die ausführbare Datei für den Knoten benannt *`/usr/bin/node-{MAJOR_VERSION}`* und die ausführbare NPM-Datei wird benannt. `/usr/bin/npm-{MAJOR_VERSION}` Auf dem laufenden System kann es jedoch nur eins `/usr/bin/node` und eins `/usr/bin/npm` geben. Diese ausführbaren Dateien sind virtuelle Namen (Symlinks) und verweisen auf die tatsächlichen

ausführbaren Dateien der aktuell aktiven Version von NodeJS. Dies wird durch die Verwendung eines alternativen Systems erreicht.

Durch die Verwendung von Alternativen können Sie mit einem einzigen Befehl auswählen, welche NodeJS Die Konfigurationsdateien der Version, die Binärdateien (wie `node` und`npm`) und die global installierten Module werden verwendet. Standardmäßig ist Alternatives so konfiguriert, dass es sich im auto Modus befindet, der Prioritäten verwendet, um die aktuell aktive Version von auszuwählen NodeJS. Sie können jedoch jederzeit zwischen den installierten Versionen wechseln, indem Sie Folgendes ausführen`alternatives --config node`. Derzeit haben alle unterstützten NodeJS-Versionen die gleiche Priorität.

Einige nützliche alternative Befehle:

1. Prüfen Sie, was in Alternativen konfiguriert ist

   ```
   alternatives --list
   ```

2. Überprüfen Sie die aktuelle Konfiguration des Knotens

   ```
   alternatives --display node
   ```

3. Ändern Sie interaktiv die NodeJS version

   ```
   alternatives --config node
   ```

4. Wechseln Sie in den manuellen Modus und wählen Sie eine bestimmte Version

   ```
   alternatives --set node /usr/bin/node-{MAJOR_VERSION}
   ```

5. Wechseln Sie zurück zum auto Versionsauswahlmodus

   ```
   alternatives --auto node
   ```

Perl im Jahr AL2 023

AL2023 bietet Version 5.32 von Perl Programmiersprache.

Obwohl Perl hat im Rahmen von für ein hohes Maß an Sprachkompatibilität gesorgt Perl 5 Releases in den letzten Jahrzehnten, von denen Amazon Linux voraussichtlich nicht abrücken

wird Perl 5.32 während der Veröffentlichung von AL2 2023. Amazon Linux wird weiterhin den Sicherheitspatch verwenden Perl für die gesamte Lebensdauer von AL2 023 gemäß unseren Angaben zur Paketunterstützung.

Perl Module im Jahr 023 AL2

Verschiedene Perl Module sind wie RPMs in AL2 023 verpackt. Obwohl es viele gibt Perl Module verfügbar als RPMs, Amazon Linux ist nicht bestrebt, alle möglichen Module zu verpacken Perl Modul. Module, die so verpackt sind, wie sie von anderen RPM-Paketen für Betriebssysteme verwendet werden RPMs könnten, sodass Amazon Linux diesen Sicherheitspatches Vorrang vor reinen Funktionsupdates einräumt.

AL20.23 beinhaltet `CPAN` auch, dass Perl Entwickler können den idiomatischen Paketmanager verwenden für Perl Module.

PHP im Jahr AL2 023

AL2023 bietet derzeit die PHPProgrammiersprache, Versionen 8.1, 8.2, 8.3 und 8.4. Jede Version wird für den gleichen Zeitraum wie die Upstream-Version unterstützt PHP. Weitere Informationen finden Sie unter Erklärungen zur Paketunterstützung.

Von einer älteren Version migrieren PHP versions

Der Upstream PHP Die Community hat eine umfassende Migrationsdokumentation für den Umzug zusammengestellt:

- von PHP 8.3.x bis PHP 8.4.x
- von PHP 8.2.x bis PHP 8.3.x
- von PHP 8.1.x bis PHP 8.2.x
- von PHP 8.0.x bis PHP 8.1.x

AL2 beinhaltet PHP 8.0, 8.1 und 8.2 `amazon-linux-extras` ermöglichen einen einfachen Upgrade-Pfad auf Version AL2 023.

Migration von PHP 7.x-Versionen

> **Note**
>
> Die PHPDas Projekt führt eine Liste und einen Zeitplan der unterstützten Versionen sowie eine Liste der nicht unterstützten Zweige.
> Als AL2 023 veröffentlicht wurde, waren alle 7.x- und 5.x-Versionen von PHPwurden von der nicht unterstützt PHP Community und waren in AL2 023 nicht als Optionen enthalten.

Der Upstream PHP Die Community hat eine umfassende Migrationsdokumentation für den Umzug zusammengestellt PHP 8,0 von PHP 7.4. Kombiniert mit der Dokumentation, auf die im vorherigen Abschnitt zur Migration verwiesen wurde PHP 8.1 und PHP 8.2, Sie können Ihre migrieren PHP basierte Anwendung auf moderne PHP.

> **Note**
>
> AL2 beinhaltet PHP 7,1, 7,2, 7,3 und 7,4 `amazon-linux-extras` Zoll Es ist wichtig zu beachten, dass für all diese Extras garantiert weitere Sicherheitsupdates verfügbar sind end-of-life und nicht garantiert werden.

PHP Module im Jahr AL2 023

AL2023 beinhaltet viele PHP Module, die enthalten sind in PHP Kern. AL2023 zielt nicht darauf ab, alle Pakete in das aufzunehmen PHP Extension Community Library (PECL).

Python im Jahr AL2 023

AL2023 entfernt Python 2.7 und alle Komponenten, die Python sind jetzt so geschrieben, dass sie damit arbeiten Python 3.

AL2203 Marken Python 3 verfügbar`/usr/bin/python3`, um die Kompatibilität mit dem Kundencode zu wahren, sowie mit AL2 0.23 ausgeliefertem Python-Code. Dieser bleibt wie folgt Python 3.9 für die Lebenszeit von AL2 023.

Die Version von Python, auf die `/usr/bin/python3` verwiesen wird, wird als das System Python betrachtet, und für AL2 0.23 ist dies Python 3.9.

Neuere Versionen von Python, wie beispielsweise Python 3.11 werden in Version AL2 023 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.

Mehrere Versionen von Python kann gleichzeitig auf AL2 023 installiert werden. `/usr/bin/python3`Wird es aber immer sein Python 3.9, jede Version von Python hat 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 023 beeinträchtigen könnte. AL2

Python Module in 023 AL2

Verschiedene Python Module sind wie RPMs in AL2 023 verpackt. Typischerweise RPMs für Python Module werden nur für die Systemversion von erstellt Python.

Rust im Jahr AL2 023

Vielleicht möchten Sie ihren Code erstellen, in den Sie geschrieben haben Rustauf Amazon Linux und möchten möglicherweise eine mit AL2 023 bereitgestellte Toolchain verwenden.

Ähnlich AL2 wie AL2 023 aktualisiert 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. Obwohl AL2 023 neue Versionen von enthalten wird Rust Die Toolchain wird während ihrer Laufzeit nicht im Gleichschritt mit der Upstream-Version laufen Rust Veröffentlichungen. Daher verwenden Sie die Rust Die in AL2 023 bereitgestellte Toolchain ist möglicherweise nicht geeignet, wenn Sie Folgendes erstellen möchten Rust Code unter Verwendung modernster Funktionen von Rust Sprache.

Während der Lebensdauer von AL2 023 werden alte Paketversionen nicht aus den Repositorien entfernt. Wenn ein älterer Rust Wenn eine Toolchain erforderlich ist, können Sie auf Fehler- und Sicherheitskorrekturen neuerer Versionen verzichten Rust Toolchains und installiere eine ältere Version aus den Repositorys mit den gleichen Mechanismen, die für jedes RPM verfügbar sind.

Wenn Sie Ihre eigene bauen möchten Rust Code auf AL2 023, du kannst den benutzen Rust In AL2 023 enthaltene Toolchain mit dem Wissen, dass sich diese Toolchain während der Lebensdauer von 023 weiterentwickeln könnte. AL2

AL2023 Lambda-Funktionen, geschrieben in Rust

Weil Rust kompiliert zu nativem Code, Lambda behandelt Rust als benutzerdefinierte Laufzeit. Sie können die `provided.al2023` Runtime für die Bereitstellung verwenden Rust funktioniert auf AL2 023 bis Lambda.

Weitere Informationen finden Sie unter Lambda-Funktionen erstellen mit Rust im Entwicklerhandbuch für AWS Lambda .

AL2023 Reservierte Benutzer und Gruppen

AL2023 weist bestimmte Benutzer und Gruppen sowohl bei der Bereitstellung des Images als auch bei der Installation bestimmter Pakete vorab zu. Die Benutzer, Gruppen und die ihnen zugewiesenen Benutzer GIDs sind hier aufgeführt, um UIDs Konflikte zu vermeiden.

Themen

- Liste der AL2 023 reservierten Benutzer
- Liste der AL2 023 reservierten Gruppen

Liste der AL2 023 reservierten Benutzer

Benutzername	Benutzerkennung (UID)
Root	0
bin	1
Dämon	2
adm	3
lp	4
Sync	5
shutdown	6
Anhalten	7
Post	8
operator	11
Spiele	12
ftp	14

Benutzername	Benutzerkennung (UID)
Tintenfisch	23
genannt	25
postgres	26
mysql-	27
nscd	28
nscd	28
Rpcuser	29
rpc	32
mailnull	47
Apache	48
smmsp	51
Tomcat	53
ldap	55
tss	59
nslcd	65
Avahi	70
tcpdump	72
sshd	74
radvd	75
dbus	81

Benutzername	Benutzerkennung (UID)
postfix	89
Taubenbett	97
Stapus	156
Stapsys	157
stapdev	158
Avahi-AutoIPD	170
Impuls	171
RT-Kit	172
Sanlock	179
systemd-Netzwerk	192
systemd-resolve	193
uuidd	961
stap-Server	962
systemd-journal-remote	963
redis6	970
unterzeichnen	971
smtpq	972
smtpd	973
nginx	974
munge	975

Benutzername	Benutzerkennung (UID)
memcached	976
sphinx	977
haproxy	978
Flatpak	979
Debug-Informationen	980
Taubenull	981
dnsmasq	982
ungebunden	983
Muschelscan	984
beansprucht	985
klammer aktualisieren	986
Farben	987
Quoten	988
aws-kinesis-agent-user	989
Saslauth	990
C-Agent	991
höflich	992
ec2-instance-connect	993
chrony	994
Systemd-Zeitsynchronisierung	995

Benutzername	Benutzerkennung (UID)
systemd-coredump	996
libstoragemgmt	997
System-Raum	999
ec2-Benutzer	1000
niemand	65534

Nach Namen gelistet

Benutzername	Benutzerkennung (UID)
adm	3
Apache	48
Avahi	70
Avahi-AutoIPD	170
aws-kinesis-agent-user	989
bin	1
chrony	994
beansprucht	985
Muschelscan	984
Klammer aktualisieren	986
Farben	987
Kaagenz	991

Benutzername	Benutzerkennung (UID)
Dämon	2
dbus	81
Informationen zum Debuggen	980
dnsmasq	982
Taubenbett	97
Taubenull	981
ec2-instance-connect	993
ec2-Benutzer	1000
Flatpak	979
ftp	14
Spiele	12
Anhalten	7
haproxy	978
ldap	55
libstoragemgmt	997
lp	4
Post	8
mailnull	47
memcached	976
munge	975

Benutzername	Benutzerkennung (UID)
mysql-	27
genannt	25
nginx	974
niemand	65534
nscd	28
nscd	28
nslcd	65
Quoten	988
operator	11
unterzeichnen	971
gepolt	992
postfix	89
postgres	26
Impuls	171
radvd	75
redis6	970
Root	0
rpc	32
RPC-Benutzer	29
rtkit	172

Benutzername	Benutzerkennung (UID)
Sanlock	179
Saslauth	990
shutdown	6
smmsp	51
smtpd	973
smtpq	972
sphinx	977
Tintenfisch	23
sshd	74
stap-Server	962
stapdev	158
Stapsys	157
Stapus	156
Sync	5
systemd-coredump	996
systemd-journal-remote	963
systemd-Netzwerk	192
systemd-oom	999
systemd-resolve	193
systemd-timesync	995

Benutzername	Benutzerkennung (UID)
tcpdump	72
Tomcat	53
tss	59
ungebunden	983
uuidd	961

Liste der AL2 023 reservierten Gruppen

Group name (Gruppenname)	GID
Root	0
bin	1
Dämon	2
Sys	3
adm	4
TTY	5
Datenträger	6
Datenträger	6
lp	7
mem	8
Kem	9
Rad	10

Group name (Gruppenname)	GID
CD-ROM	11
Post	12
Post	12
man	15
wählen	18
Diskette	19
Spiele	20
slokieren	21
utmp	22
Tintenfisch	23
genannt	25
postgres	26
mysql-	27
nscd	28
nscd	28
Rpcuser	29
rpc	32
Band	33
Unverlockung	35
kvm	36

Group name (Gruppenname)	GID
video	39
mailnull	47
Apache	48
ftp	50
smmsp	51
Tomcat	53
Verriegelung	54
ldap	55
tss	59
audio	63
Avahi	70
tcpdump	72
sshd	74
radvd	75
Saslauth	76
dbus	81
screen	84
wbpriv	88
postfix	89
Postdrop	90

Group name (Gruppenname)	GID
Taubenbett	97
Benutzer	100
input	104
leisten	105
sgx	106
verspotten	135
Stapus	156
Stapus	156
Stapsys	157
Stapsys	157
stapdev	158
stapdev	158
Avahi-AutoIPD	170
Impuls	171
RT-Kit	172
Sanlock	179
systemd-journal	190
Systemd-Netzwerk	192
systemd-resolve	193
beschwören	959

Group name (Gruppenname)	GID
Drahthaie	960
uuidd	961
stap-Server	962
systemd-journal-remote	963
Benutzeraktien	964
redis6	965
unterzeichnen	966
smtpq	967
smtpd	968
nginx	969
munge	970
memcached	971
sphinx	972
Nachverfolgung	973
haproxy	974
Flatpak	975
Debug-Informationen	976
Taubenull	977
dnsmasq	978
ungebunden	979

Group name (Gruppenname)	GID
Muschelscan	980
beansprucht	981
Virengruppe	982
Virusgruppe	982
Virusgruppe	982
Klammer aktualisieren	983
admin drucken	984
Farben	985
Quoten	986
Docker	987
aws-kinesis-agent-user	988
Kagenz	989
Puls-RT	990
Puls-Zugriff	991
ec2-instance-connect	993
chrony	994
Systemd-Zeitsynchronisierung	995
systemd-coredump	996
libstoragemgmt	997
SSH-Schlüssel	998

Group name (Gruppenname)	GID
System-Raum	999
ec2-Benutzer	1000
Ein	1001
Polkitt	920
niemand	65534

Nach Namen gelistet

Group name (Gruppenname)	GID
adm	4
Apache	48
audio	63
Avahi	70
Avahi-AutoIPD	170
aws-kinesis-agent-user	988
bin	1
CD-ROM	11
chrony	994
beansprucht	981
Muschelscan	980
klammer aktualisieren	983

Group name (Gruppenname)	GID
Farben	985
Kaagenz	989
Dämon	2
dbus	81
Informationen zum Debuggen	976
Wählen Sie aus	18
Datenträger	6
Datenträger	6
dnsmasq	978
Docker	987
Taubenbett	97
Taubenull	977
ec2-instance-connect	993
ec2-Benutzer	1000
Flatpak	975
Diskette	19
ftp	50
Spiele	20
haproxy	974
input	104

Group name (Gruppenname)	GID
kmem	9
kvm	36
ldap	55
libstoragemgmt	997
Verriegelung	54
lp	7
Post	12
Post	12
mailnull	47
man	15
mem	8
memcached	971
verspotten	135
munge	970
mysql-	27
genannt	25
Eins	1001
nginx	969
niemand	65534
nscd	28

Group name (Gruppenname)	GID
nscd	28
Götter	986
unterzeichnen	966
höflich	920
Postdrop	90
postfix	89
postgres	26
admin drucken	984
Impuls	171
Puls-Zugriff	991
Puls-RT	990
radvd	75
redis6	965
leisten	105
Root	0
rpc	32
RPC-Benutzer	29
rtkit	172
Sanlock	179
Saslauth	76

Group name (Gruppenname)	GID
screen	84
sgx	106
slokieren	21
smmsp	51
smtpd	968
smtpq	967
sphinx	972
Tintenfisch	23
ssh_keys	998
sshd	74
stap-Server	962
stapdev	158
stapdev	158
Stapsys	157
Stapsys	157
Stapus	156
Stapus	156
sagt	3
systemd-coredump	996
systemd-journal	190

Group name (Gruppenname)	GID
systemd-journal-remote	963
systemd-Netzwerk	192
systemd-oom	999
systemd-resolve	193
systemd-timesync	995
Band	33
tcpdump	72
Tomcat	53
Nachverfolgung	973
tss	59
TTY	5
ungebunden	979
heraufbeschwören	959
Benutzer	100
Benutzeraktien	964
Unverlockung	35
utmp	22
uuidd	961
video	39
Virengruppe	982

Group name (Gruppenname)	GID
Virusgruppe	982
Virusgruppe	982
wbpriv	88
Rad	10
Wireshark	960

Liste der in 023 verfügbaren Codecs AL2

AL2023 bietet über seine Standard-Repositorys eine Auswahl an Multimedia-Codecs. Diese Seite bietet einen Überblick über die Codecs und ihre typischen Anwendungsfälle.

> **Important**
>
> Die Verwendung und Verbreitung der in Amazon Linux enthaltenen Codecs kann erfordern, dass Sie Lizenzrechte von Dritten erwerben, einschließlich Eigentümern oder Lizenzgebern bestimmter Audio- und Videoformate von Drittanbietern. Sie sind allein dafür verantwortlich, diese Lizenzen zu erhalten und alle erforderlichen Lizenzgebühren oder Gebühren zu zahlen.

Codec	Beschreibung
flac	Ein kostenloser und verlustfreier Open-Source-Audiocodec, der Audio komprimiert, ohne dabei Daten oder Qualität zu verlieren. Er wird häufig für die Speicherung hochwertiger Audiodateien verwendet
fdk-aac-free	Eine Open-Source-Implementierung des AAC-Standards (Advanced Audio Codec), die eine hochwertige Audiokomprimierung für MP3 Alternativen wie Streaming oder Dateispei cherung bietet
webrtc-audio-processing	Eine Bibliothek für die Audioverarbeitung, die in WebRTC (Web Real-Time Communica tion) verwendet wird und Funktionen wie Geräuschunterdrückung, Echokompensation und Verstärkungssteuerung bietet
opus	Ein äußerst vielseitiger und effizienter Audio-Codec, der für Echtzeit-Streaming entwickelt wurde. Er bietet geringe Latenz und unterstützt

Codec	Beschreibung
	eine Vielzahl von Audioanwendungen, einschließlich VoIP und Musik-Streaming
libsndfile	Eine Bibliothek zum Lesen und Schreiben von Audiodateien in verschiedenen Formaten (wie WAV, AIFF und FLAC), die häufig in Tools zur Audioverarbeitung und -manipulation verwendet wird

Sicherheit und Compliance in Amazon Linux 2023

> **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 023 gelten, finden Sie unter AWS Services im Bereich nach Compliance-Programm 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 023 AL2
- Liste der zutreffenden Hinweise
- Direktes Anwenden von Sicherheitsupdates
- SELinux Einstellmodi für AL2 023
- Aktivieren Sie den FIPS-Modus auf 023 AL2
- Aktivieren Sie den FIPS-Modus in einem AL2 023-Container
- Tauschen Sie OpenSSL FIPS-Anbieter auf 023 aus AL2
- AL2023 Kernelhärtung

- UEFI Secure Boot auf 023 AL2

Amazon Linux-Sicherheitsempfehlungen für 023 AL2

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 Fix 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 AL2 023 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 CVEs von Amazon Linux finden Sie unter Häufig gestellte Fragen zum Amazon Linux Security Center (ALAS) (FAQs).

ALAS-Empfehlungen

Ein Amazon Linux-Advisory enthält wichtige Informationen, die für Amazon Linux-Benutzer relevant sind, in der Regel Informationen zu Sicherheitsupdates. Im Amazon Linux Security Center sind die Advisories im Internet sichtbar. Hinweisinformationen sind auch Teil der Metadaten des RPM-Paket-Repositorys.

Hinweise und RPM-Repositorys

Ein Amazon Linux 2023-Paket-Repository kann Metadaten enthalten, die keine oder mehr Updates beschreiben. Der `dnf updateinfo` Befehl ist nach dem Dateinamen der Repository-Metadaten benannt, der diese Informationen enthält`updateinfo.xml`. Der Befehl ist zwar benannt `updateinfo` und die Metadatendatei bezieht sich auf ein`update`, aber alle beziehen sich auf Paket-Updates, die Teil eines Advisory sind.

Amazon Linux-Advisories werden auf der Amazon Linux Security Center-Website veröffentlicht, zusammen mit Informationen, die in den Metadaten des RPM-Repositorys enthalten sind, auf die sich der `dnf` Paketmanager bezieht. Die Website- und Repository-Metadaten sind letztendlich konsistent, und es kann zu Inkonsistenzen bei den Informationen auf der Website und in den Repository-Metadaten kommen. Dies tritt in der Regel auf, wenn eine neue Version von AL2 023 gerade veröffentlicht wird und ein Advisory nach der letzten AL2 Version von 023 aktualisiert wurde.

Es ist zwar üblich, dass zusammen mit dem Paket-Update, das das Problem behebt, auch ein neues Advisory veröffentlicht wird, aber das ist nicht immer der Fall. Ein Hinweis kann für ein neues Problem erstellt werden, das in bereits veröffentlichten Paketen behoben ist. Eine bestehende Empfehlung kann auch mit neuen aktualisiert werden CVEs , die durch das bestehende Update behoben werden.

Die Deterministische Upgrades durch versionierte Repositorys auf 023 AL2 Funktion von Amazon Linux 2023 bedeutet, dass das RPM-Repository für eine bestimmte Version AL2 023 einen Snapshot der RPM-Repository-Metadaten ab dieser Version enthält. Dazu gehören die Metadaten, die Sicherheitsupdates beschreiben. Das RPM-Repository für eine bestimmte Version AL2 023 wird nach der Veröffentlichung nicht aktualisiert. Neue oder aktualisierte Sicherheitshinweise sind nicht sichtbar, wenn Sie sich eine ältere Version der AL2 023 RPM-Repositorys ansehen. Liste der zutreffenden

HinweiseIn diesem Abschnitt erfahren Sie, wie Sie mit dem `dnf` Paketmanager entweder die `latest` Repository-Version oder eine bestimmte AL2 Version 023 einsehen können.

Hinweis IDs

Auf jedes Advisory wird mit einem verwiesen`id`. Derzeit ist es eine Eigenart von Amazon Linux, dass auf der Amazon Linux Security Center-Website eine Empfehlung als ALAS-2024-581 aufgeführt wird, während der `dnf` Paketmanager diese Empfehlung mit der ID 023-2024-581 auflistet. ALAS2 Wenn Direktes Anwenden von Sicherheitsupdates die Paketmanager-ID verwendet werden muss, wenn auf ein bestimmtes Advisory verwiesen wird.

Für Amazon Linux hat jede Hauptversion des Betriebssystems ihren eigenen IDs Advisory-Namespace. Es sollten keine Annahmen über das Format von Amazon Linux Advisory getroffen IDs werden. In der Vergangenheit folgte Amazon Linux Advisory IDs dem Muster von`NAMESPACE-YEAR-NUMBER`. Der gesamte Bereich möglicher Werte für `NAMESPACE` ist nicht definiert, umfasst aber`ALAS`,`ALASCORRETTO8`, `ALAS2023`ALAS2,`ALASPYTHON3.8`, und`ALASUNBOUND-1.17`. Dabei handelt es `YEAR` sich um das Jahr, in dem das Advisory erstellt wurde. Dabei handelt `NUMBER` es sich um eine eindeutige Ganzzahl innerhalb des Namespace.

Obwohl die Empfehlung IDs normalerweise sequentiell und in der Reihenfolge erfolgt, in der die Updates veröffentlicht werden, gibt es viele Gründe, warum dies nicht der Fall sein könnte. Daher sollte nicht davon ausgegangen werden.

Behandeln Sie die Advisory-ID als undurchsichtige Zeichenfolge, die für jede Hauptversion von Amazon Linux eindeutig ist.

In Amazon Linux 2 befand sich jedes Extra in einem separaten RPM-Repository, und die Advisory-Metadaten sind nur in dem Repository enthalten, für das sie relevant sind. Eine Empfehlung für ein Repository gilt nicht für ein anderes Repository. Auf der Amazon Linux Security Center-Website gibt es derzeit eine Liste mit Hinweisen für jede Hauptversion von Amazon Linux, die nicht in Listen pro Repository unterteilt ist.

Da AL2 023 den Extras-Mechanismus nicht verwendet, um alternative Versionen von Paketen zu verpacken, gibt es derzeit nur zwei RPM-Repositorys, von denen jedes über Advisories verfügt: das Repository und das `core` Repository. `livepatch` Das `livepatch` Repository ist für. Kernel Live Patching auf 023 AL2

Veröffentlichungsdatum der Empfehlung und Aktualisierungsdatum der Empfehlung

Das Veröffentlichungsdatum der Empfehlung für Amazon Linux Advisories gibt an, wann das Sicherheitsupdate zum ersten Mal im RPM-Repository öffentlich verfügbar gemacht wurde. Advisories werden auf der Amazon Linux Security Center-Website veröffentlicht, unmittelbar nachdem Fixes über das RPM-Repository zur Installation zur Verfügung gestellt wurden.

Das Aktualisierungsdatum der Empfehlung gibt an, wann neue Informationen zu einer Empfehlung hinzugefügt wurden, nachdem sie zuvor veröffentlicht wurde.

Es sollten keine Annahmen zwischen der Versionsnummer AL2 023 (z. B. 2023.6.20241031) und dem Veröffentlichungsdatum der zusammen mit dieser Pressemitteilung veröffentlichten Advisories getroffen werden.

Arten von Hinweisen

Die Metadaten des RPM-Repositorys unterstützen Advisories verschiedener Typen. Amazon Linux hat zwar fast überall nur Advisories herausgegeben, bei denen es sich um Sicherheitsupdates handelt, aber davon sollte nicht ausgegangen werden. Es ist möglich, dass Ankündigungen für Ereignisse wie Bugfixes, Verbesserungen und neue Pakete herausgegeben werden und dass die Empfehlung so gekennzeichnet ist, dass sie diese Art von Update enthält.

Schweregrade der Empfehlung

Jede Empfehlung hat ihren eigenen Schweregrad, da jedes Problem separat bewertet wird. In einem einzigen Advisory CVEs können mehrere behandelt werden, und jeder CVE kann eine andere Bewertung haben, aber das Advisory selbst hat einen Schweregrad. Es kann mehrere Advisories geben, die sich auf ein einzelnes Paket-Update beziehen, sodass es für ein bestimmtes Paket-Update mehrere Schweregrade geben kann (einer pro Advisory).

In der Reihenfolge des abnehmenden Schweregrads hat Amazon Linux Kritisch, Wichtig, Moderat und Niedrig verwendet, um den Schweregrad einer Empfehlung anzugeben. Amazon Linux Advisories haben möglicherweise auch keinen Schweregrad, obwohl dieser äußerst selten vorkommt.

Amazon Linux ist eine der RPM-basierten Linux-Distributionen, die den Begriff Moderat verwendet, während einige andere RPM-basierte Linux-Distributionen den entsprechenden Begriff Medium verwenden. Der Amazon Linux-Paketmanager behandelt beide Begriffe als gleichwertig, und Paket-Repositorys von Drittanbietern können den Begriff Medium verwenden.

Amazon Linux-Advisories können den Schweregrad im Laufe der Zeit ändern, wenn mehr über die relevanten Probleme, die in der Empfehlung behandelt wurden, bekannt ist.

Der Schweregrad einer Empfehlung entspricht in der Regel dem höchsten von Amazon Linux bewerteten CVSS-Score für das, CVEs auf das die Empfehlung verweist. Es kann Fälle geben, in denen dies nicht der Fall ist. Ein Beispiel wäre, wenn ein Problem behoben wird, für das kein CVE zugewiesen wurde.

Weitere Informationen darüber, wie Amazon Linux die Schweregrade von Advisory verwendet, finden Sie in den häufig gestellten Fragen zu ALAS.

Ratgeber und Pakete

Für ein einzelnes Paket kann es viele Advisories geben, und nicht für alle Pakete wird jemals ein Advisory veröffentlicht. Auf eine bestimmte Paketversion kann in mehreren Advisories verwiesen werden, von denen jedes seinen eigenen Schweregrad und hat. CVEs

Es ist möglich, dass mehrere Advisories für dasselbe Paket-Update gleichzeitig in einer neuen Version AL2 023 oder schnell hintereinander veröffentlicht werden.

Wie bei anderen Linux-Distributionen kann es ein bis viele verschiedene Binärpakete geben, die aus demselben Quellpaket erstellt wurden. Beispielsweise ist ALAS-2024-698 eine Empfehlung, die im Abschnitt AL2 023 der Amazon Linux Security Center-Website als für das Paket relevant aufgeführt ist. `mariadb105` Dies ist der Name des Quellpakets, und das Advisory selbst bezieht sich neben dem Quellpaket auch auf die Binärpakete. In diesem Fall werden über ein Dutzend Binärpakete aus einem `mariadb105` Quellpaket erstellt. Es ist zwar sehr üblich, dass es ein Binärpaket mit demselben Namen wie das Quellpaket gibt, aber das ist nicht universal.

Obwohl Amazon Linux Advisories in der Regel alle Binärpakete aufgelistet haben, die aus dem aktualisierten Quellpaket erstellt wurden, sollte davon nicht ausgegangen werden. Der Paketmanager und das RPM-Repository-Metadatenformat ermöglichen Advisories, die eine Teilmenge der aktualisierten Binärpakete auflisten.

Ein bestimmter Hinweis kann auch nur für eine bestimmte CPU-Architektur gelten. Es kann Pakete geben, die nicht für alle Architekturen erstellt wurden, oder Probleme, die nicht alle Architekturen betreffen. Für den Fall, dass ein Paket auf allen Architekturen verfügbar ist, ein Problem aber nur für eine auftritt, hat Amazon Linux in der Regel kein Advisory herausgegeben, das nur auf die betroffene Architektur verweist, obwohl davon nicht ausgegangen werden sollte.

Aufgrund der Art der Paketabhängigkeiten ist es üblich, dass ein Advisory auf ein Paket verweist, aber die Installation dieses Updates erfordert andere Paket-Updates, einschließlich Pakete, die nicht in der Empfehlung aufgeführt sind. Der `dnf` Paketmanager kümmert sich um die Installation der erforderlichen Abhängigkeiten.

Ratschläge und CVEs

Ein Advisory kann keine oder mehrere Advisories adressieren CVEs, und es kann mehrere Advisories geben, die sich auf dasselbe CVE beziehen.

Ein Beispiel dafür, wann ein Advisory auf Null verweisen kann, CVEs ist, wenn dem Problem noch kein CVE (oder noch nie) zugewiesen wurde.

Ein Beispiel dafür, dass mehrere Advisories auf denselben CVE verweisen können, wenn der CVE (zum Beispiel) für mehrere Pakete gilt. Beispielsweise gilt CVE-2024-21208 für Corretto 8, 11, 17 und 21. Jede dieser Corretto-Versionen ist ein separates Paket in AL2 023, und für jedes dieser Pakete gibt es ein Advisory: ALAS-2024-754 für Corretto 8, ALAS-2024-753 für Corretto 11, ALAS-2024-752für Corretto 17 und ALAS-2024-752 für Corretto 21. Diese Corretto-Versionen haben zwar alle dieselbe Liste von CVEs, dies sollte jedoch nicht davon ausgegangen werden.

Ein bestimmter CVE kann für verschiedene Pakete unterschiedlich bewertet werden. Wenn beispielsweise in einem Advisory mit dem Schweregrad Wichtig auf ein bestimmtes CVE verwiesen wird, ist es möglich, dass ein anderes Advisory herausgegeben wird, das sich auf denselben CVE mit einem anderen Schweregrad bezieht.

Die Metadaten des RPM-Repositorys ermöglichen eine Referenzliste für jedes Advisory. Während Amazon Linux in der Regel nur referenziert hat CVEs, ermöglicht das Metadatenformat auch andere Referenztypen.

Auf die Metadaten des RPM-Paket-Repositorys wird nur verwiesen, CVEs wenn ein Fix verfügbar ist. Der Abschnitt Erkunden der Amazon Linux Security Center-Website enthält Informationen zu den Informationen CVEs , die Amazon Linux bewertet hat. Diese Bewertung kann zu einer CVSS-Basisbewertung, einem Schweregrad und einem Status für verschiedene Amazon Linux-Versionen und -Pakete führen. Der Status eines CVE für eine bestimmte Amazon Linux-Version oder ein bestimmtes Paket kann „Nicht betroffen“, „Ausstehende Korrektur“ oder „Kein Fix geplant“ lauten. Der Status und die Bewertung von CVEs können sich viele Male und in beliebiger Weise ändern, bevor eine Empfehlung veröffentlicht wird. Dies beinhaltet eine Neubewertung der Anwendbarkeit eines CVE auf Amazon Linux.

Die Liste der Personen, auf die in einer Empfehlung CVEs verwiesen wird, kann sich nach der ersten Veröffentlichung dieser Empfehlung ändern.

Text des Hinweises

Ein Hinweis wird auch einen Text enthalten, der das Problem oder die Probleme beschreibt, die der Grund für die Erstellung des Ratgebers waren. Es ist üblich, dass es sich bei diesem Text um den unveränderten CVE-Text handelt. Dieser Text kann sich auf Upstream-Versionsnummern beziehen, für die ein Fix verfügbar ist, die sich von der Paketversion unterscheiden, auf die Amazon Linux einen Fix angewendet hat. Es ist üblich, dass Amazon Linux Fixes aus neueren Upstream-Versionen zurückportiert. Falls im Text der Ankündigung eine Upstream-Version erwähnt wird, die sich von der Version unterscheidet, die in einer Amazon Linux-Version ausgeliefert wurde, gelten die Amazon Linux-Paketversionen in der Empfehlung für Amazon Linux.

Es ist möglich, dass der Hinweistext in den Metadaten des RPM-Repositorys Platzhaltertext ist, der lediglich auf die Amazon Linux Security Center-Website verweist, um weitere Informationen zu erhalten.

Kernel Live Patch Advisories

Ankündigungen für Live-Patches sind insofern einzigartig, als sie sich auf ein anderes Paket (den Linux-Kernel) beziehen als das Paket, gegen das sich die Ankündigung richtet (z. B.`kernel-livepatch-6.1.15-28.43`).

Eine Empfehlung für einen Kernel-Live-Patch bezieht sich auf die Probleme (z. B. CVEs), die das jeweilige Live-Patch-Paket für die spezifische Kernel-Version, für die das Live-Patch-Paket gilt, beheben kann.

Jeder Live-Patch bezieht sich auf eine bestimmte Kernel-Version. Um einen Live-Patch für eine CVE anzuwenden, muss das richtige Live-Patch-Paket für Ihre Kernel-Version installiert und der Live-Patch angewendet werden.

Zum Beispiel kann CVE-2023-6111 für 023-Kernelversionen,, und live gepatcht werden. AL2 `6.1.56-82.125` `6.1.59-84.139` `6.1.61-85.141` Eine neue Kernelversion mit einem Fix für dieses CVE wurde ebenfalls veröffentlicht, für die es eine separate Empfehlung gibt. Damit CVE-2023-6111 auf AL2 023 adressiert werden kann, muss entweder eine Kernelversion ausgeführt werden, die der Angabe von ALAS2023-2023-461 entspricht oder später ist, oder es muss eine der Kernelversionen mit einem Live-Patch für dieses CVE ausgeführt werden, wobei der entsprechende Livepatch angewendet wurde.

Wenn neue Live-Patches für eine bestimmte Kernel-Version verfügbar sind, für die bereits ein Live-Patch verfügbar ist, wird eine neue Version des Pakets veröffentlicht. `kernel-livepatch-KERNEL_VERSION` Zum Beispiel wurde das ALASLIVEPATCH-2023-003Advisory zusammen mit dem `kernel-livepatch-6.1.15-28.43-1.0-1.amzn2023` Paket veröffentlicht, das Live-Patches für den `6.1.15-28.43` Kernel enthielt, die drei Patches abdeckten CVEs. Später wurde das ALASLIVEPATCH-2023-009Advisory zusammen mit dem `kernel-livepatch-6.1.15-28.43-1.0-2.amzn2023` Paket veröffentlicht, ein Update des vorherigen Live-Patch-Pakets für den `6.1.15-28.43` Kernel, das Live-Patches für drei weitere enthält CVEs. Es gab auch andere Probleme mit Live-Patch-Advisories für andere Kernel-Versionen, wobei Pakete Live-Patches für diese spezifischen Kernel-Versionen enthielten.

Weitere Informationen zum Live-Patchen des Kernels finden Sie unter. Kernel Live Patching auf 023 AL2

Allen, die Tools rund um Sicherheitsempfehlungen entwickeln, wird außerdem empfohlen, diesen XML-Schema für Sicherheitshinweise und `updateinfo.xml` Abschnitt für weitere Informationen zu lesen.

XML-Schema für Sicherheitshinweise und **`updateinfo.xml`**

Die `updateinfo.xml` Datei ist Teil des Paket-Repository-Formats. Es sind die Metadaten, die der `dnf` Paketmanager analysiert, um Funktionen wie Liste der zutreffenden Hinweise und Direktes Anwenden von Sicherheitsupdates zu implementieren.

Wir haben empfohlen, die API des `dnf` Paketmanagers zu verwenden, anstatt benutzerdefinierten Code zu schreiben, um die Metadatenformate des Repositorys zu analysieren. Die Version von `dnf` in AL2 023 kann sowohl das AL2 023- als auch das AL2 Repository-Format analysieren, sodass die API verwendet werden kann, um Beratungsinformationen für beide Betriebssystemversionen zu überprüfen.

Das RPM-Softwaremanagement-Projekt dokumentiert die RPM-Metadatenformate im RPM-Metadaten-Repository unter. GitHub

Denjenigen, die Tools zur direkten Analyse der `updateinfo.xml` Metadaten entwickeln, wird dringend empfohlen, die Dokumentation zu den RPM-Metadaten sorgfältig zu lesen. Die Dokumentation behandelt, was in freier Wildbahn beobachtet wurde, und enthält viele Ausnahmen von dem, was Sie vernünftigerweise als Regel für das Metadatenformat interpretieren könnten.

Es gibt auch eine wachsende Anzahl von Beispielen aus der realen Welt für `updateinfo.xml` Dateien im raw-historical-rpm-repository-examples-Repository unter. GitHub

Falls in der Dokumentation etwas unklar ist, können Sie ein Problem im GitHub Projekt eröffnen, damit wir die Frage beantworten und die Dokumentation entsprechend aktualisieren können. Da es sich um Open-Source-Projekte handelt, sind auch Pull-Requests zur Aktualisierung der Dokumentation willkommen.

Liste der zutreffenden Hinweise

Der `dnf` Paketmanager hat Zugriff auf Metadaten, die beschreiben, welche Advisories in welchen Paketversionen behoben wurden. Somit kann er auflisten, welche Advisories für eine Instanz oder ein Container-Image gelten.

> **Note**
>
> Tools wie AWS Systems Managerkönnen diese Funktionalität nutzen, um zu zeigen, welche Updates für eine Flotte relevant sind und nicht nur für eine einzelne Instanz.

Wenn Sie Updates auflisten, können Sie `dnf` anweisen, sich die Metadaten einer bestimmten Version AL2 023 oder die Metadaten der neuesten Version anzusehen.

> **Note**
>
> Sobald eine Version AL2 023 veröffentlicht wurde, ist sie unveränderlich. Daher werden neue oder aktualisierte Hinweise im Amazon Linux Security Center nur zu den Metadaten neuer Versionen von AL2 0.23 hinzugefügt.

Wir werden uns nun anhand von Beispielen ansehen, welche Hinweise für einige AL2 023-Container-Images gelten. Diese Befehle funktionieren alle in nicht containerisierten Umgebungen wie Instances. EC2

Listing advisories in a specific version

In diesem Beispiel werden wir uns ansehen, welche Hinweise in der Version 2023.1.20230628 in einem Container-Image der Version 2023.0.20230315 relevant sind.

> **Note**
>
> In diesem Beispiel werden die Versionen 2023.0.20230315 und 2023.1.20230628verwendet, und dies sind nicht die neueste Version von 023. In den Versionshinweisen zu 023 finden Sie die neuesten Versionen, die die neuesten Sicherheitsupdates enthalten. AL2 AL2

In diesem Beispiel beginnen wir mit einem Container-Image für die Version 2023.0.20230315.

Zuerst holen wir uns dieses Container-Image aus der Container-Registry. Das `.0` am Ende steht für die Version des Images für eine bestimmte Version; diese Image-Version ist normalerweise Null.

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
   2023.0.20230315.0: Pulling from amazonlinux/amazonlinux
b76f3b09316a: Pull complete
Digest: sha256:94e7183b0739140dbd5b639fb7600f0a2299cec5df8780c26d9cb409da5315a9
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.0.20230315.0
public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
```

Wir können jetzt eine Shell innerhalb des Containers erzeugen, von der aus wir fragen`dnf`, welche Hinweise für die im Container installierten Pakete relevant sind.

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
   bash-5.2#
```

Der `dnf updateinfo` Befehl wird nun verwendet, um eine Zusammenfassung der Empfehlungen in der Version 2023.1.20230628 anzuzeigen, die für unsere installierten Pakete relevant sind.

```
$ dnf updateinfo --releasever=2023.1.20230628
Amazon Linux 2023 repository                    42 MB/s |  15 MB     00:00
Last metadata expiration check: 0:00:02 ago on Mon Jul 22 20:24:24 2024.
Updates Information Summary: available
    8 Security notice(s)
        1 Important Security notice(s)
        5 Medium Security notice(s)
        2 Low Security notice(s)
```

Um eine Liste der Empfehlungen zu erhalten, kann die Option angegeben werden. `--list` `dnf updateinfo`

```
$ dnf updateinfo --releasever=2023.1.20230628 --list
Last metadata expiration check: 0:01:22 ago on Mon Jul 22 20:24:24 2024.
ALAS2023-2023-193 Medium/Sec.    curl-minimal-8.0.1-1.amzn2023.x86_64
ALAS2023-2023-225 Medium/Sec.    glib2-2.74.7-688.amzn2023.0.1.x86_64
ALAS2023-2023-195 Low/Sec.       libcap-2.48-2.amzn2023.0.3.x86_64
ALAS2023-2023-193 Medium/Sec.    libcurl-minimal-8.0.1-1.amzn2023.x86_64
ALAS2023-2023-145 Low/Sec.       libgcc-11.3.1-4.amzn2023.0.3.x86_64
ALAS2023-2023-145 Low/Sec.       libgomp-11.3.1-4.amzn2023.0.3.x86_64
ALAS2023-2023-145 Low/Sec.       libstdc++-11.3.1-4.amzn2023.0.3.x86_64
ALAS2023-2023-163 Medium/Sec.    libxml2-2.10.4-1.amzn2023.0.1.x86_64
ALAS2023-2023-220 Important/Sec. ncurses-base-6.2-4.20200222.amzn2023.0.4.noarch
ALAS2023-2023-220 Important/Sec. ncurses-libs-6.2-4.20200222.amzn2023.0.4.x86_64
ALAS2023-2023-181 Medium/Sec.    openssl-libs-1:3.0.8-1.amzn2023.0.2.x86_64
ALAS2023-2023-222 Medium/Sec.    openssl-libs-1:3.0.8-1.amzn2023.0.3.x86_64
```

Listing advisories in the latest version

In diesem Beispiel werden wir uns ansehen, welche Updates in der Version AL2 023 verfügbar sind, wenn wir einen Container der `latest` Version 2023.4.20240319 gestartet haben. Zum Zeitpunkt der Erstellung dieses Artikels ist die `latest` Version 2023.5.20240708, sodass die in diesem Beispiel aufgelisteten Updates ab dieser Version gültig sind.

> **Note**
>
> In diesem Beispiel werden die Versionen 2023.4.20240319 und 2023.5.20240708 verwendet, wobei letztere die neueste Version zum Zeitpunkt der Erstellung dieses Artikels ist. Weitere Informationen zu den neuesten Versionen finden Sie in den Versionshinweisen zu 023. AL2

In diesem Beispiel beginnen wir mit einem Container-Image für die Version 2023.4.20240319.

Zuerst holen wir uns dieses Container-Image aus der Container-Registry. Das `.1` am Ende gibt die Version des Images für eine bestimmte Version an. Während die Image-Version normalerweise Null ist, wird in diesem Beispiel eine Version verwendet, bei der es sich um eine Image-Version handelt.

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
```

```
   2023.4.20240319.1: Pulling from amazonlinux/amazonlinux
6de065fda9a2: Pull complete
Digest: sha256:b4838c4cc9211d966b6ea158dacc9eda7433a16ba94436508c2d9f01f7658b4e
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.4.20240319.1
public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
```

Wir können jetzt eine Shell im Container erzeugen, von der aus wir nach Updates suchen.

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
   bash-5.2#
```

Der `dnf updateinfo` Befehl wird nun verwendet, um eine Zusammenfassung der Hinweise in der neuesten Version anzuzeigen, die für unsere installierten Pakete relevant sind. Zum Zeitpunkt der Erstellung dieses Artikels war 2023.1.20230628 die neueste Version.

```
$ dnf --releasever=latest updateinfo
Amazon Linux 2023 repository                    76 MB/s |  25 MB     00:00
Last metadata expiration check: 0:00:04 ago on Mon Jul 22 20:59:54 2024.
Updates Information Summary: available
    9 Security notice(s)
        4 Important Security notice(s)
        4 Medium Security notice(s)
        1 Low Security notice(s)
```

Um eine Liste der Empfehlungen zu erhalten, können Sie die Option wählen. `--list` `dnf updateinfo`

```
$ dnf updateinfo --releasever=latest --list
Last metadata expiration check: 0:00:58 ago on Mon Jul 22 20:59:54 2024.
ALAS2023-2024-581 Low/Sec.       curl-minimal-8.5.0-1.amzn2023.0.3.x86_64
ALAS2023-2024-596 Medium/Sec.    curl-minimal-8.5.0-1.amzn2023.0.4.x86_64
ALAS2023-2024-576 Important/Sec. expat-2.5.0-1.amzn2023.0.4.x86_64
ALAS2023-2024-589 Important/Sec. glibc-2.34-52.amzn2023.0.10.x86_64
ALAS2023-2024-589 Important/Sec. glibc-common-2.34-52.amzn2023.0.10.x86_64
ALAS2023-2024-589 Important/Sec. glibc-minimal-langpack-2.34-52.amzn2023.0.10.x86_64
ALAS2023-2024-586 Medium/Sec.    krb5-libs-1.21-3.amzn2023.0.4.x86_64
ALAS2023-2024-581 Low/Sec.       libcurl-minimal-8.5.0-1.amzn2023.0.3.x86_64
ALAS2023-2024-596 Medium/Sec.    libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64
ALAS2023-2024-592 Important/Sec. libnghttp2-1.59.0-3.amzn2023.0.1.x86_64
ALAS2023-2024-640 Medium/Sec.    openssl-libs-1:3.0.8-1.amzn2023.0.12.x86_64
```

```
ALAS2023-2024-605 Medium/Sec.    python3-3.9.16-1.amzn2023.0.7.x86_64
ALAS2023-2024-616 Important/Sec. python3-3.9.16-1.amzn2023.0.8.x86_64
ALAS2023-2024-605 Medium/Sec.    python3-libs-3.9.16-1.amzn2023.0.7.x86_64
ALAS2023-2024-616 Important/Sec. python3-libs-3.9.16-1.amzn2023.0.8.x86_64
```

Direktes Anwenden von Sicherheitsupdates

Eine Übersicht über das Anwenden von Updates finden Sie unterAnwenden von Sicherheitsupdates mit DNF und Repository-Versionen. `--security`Mit der Option „zu“ `dnf upgrade` werden die Paket-Updates auf diejenigen beschränkt, für die ein Advisory verfügbar ist. Im Rest dieses Abschnitts wird beschrieben, wie nur bestimmte Sicherheitsupdates installiert werden.

> **Note**
>
> Es wird empfohlen, alle in einer neuen Version AL2 023 verfügbaren Updates zu installieren. Die Auswahl nur von Sicherheitsupdates oder nur bestimmten Updates sollte eher die Ausnahme als die Regel sein.

Anwenden von Updates, die in einem Advisory erwähnt werden

Die Hinweiskennungen in der ersten Spalte der Ausgabe von `dnf upgradeinfo` können verwendet werden, um Updates für die in der Empfehlung genannten Pakete anzuwenden. Der `dnf` Paketmanager kann angewiesen werden, die Pakete in der Empfehlung entweder auf die neueste verfügbare Version oder nur auf die in der Empfehlung genannten Versionen zu aktualisieren. Wenn die Updates bereits installiert sind, ist der Aktualisierungsbefehl ein No-Op.

Um die Updates für die betroffenen Pakete nur bis zu der in der Empfehlung genannten Version anzuwenden, verwenden Sie den `dnf upgrade-minimal` Befehl und gleichzeitig die `--advisory` Option zur Angabe der Empfehlung. Das folgende Beispiel wird `dnf upgrade-minimal` in einem Container der Version AL2 023, Version 2023.0.20230315, ausgeführt.

```
$ dnf upgrade-minimal -y --releasever=2023.1.20230628 --advisory ALAS2023-2023-193
Amazon Linux 2023 repository                    46 MB/s |  15 MB     00:00
Last metadata expiration check: 0:00:03 ago on Mon Jul 22 20:36:13 2024.
Dependencies resolved.
================================================================================
 Package              Arch        Version                Repository        Size
```

```
================================================================================
Upgrading:
 curl-minimal         x86_64      8.0.1-1.amzn2023       amazonlinux      150 k
 libcurl-minimal      x86_64      8.0.1-1.amzn2023       amazonlinux      249 k

Transaction Summary
================================================================================
Upgrade  2 Packages

Total download size: 399 k
Downloading Packages:
(1/2): curl-minimal-8.0.1-1.amzn2023.x86_64.rpm 2.7 MB/s | 150 kB     00:00
(2/2): libcurl-minimal-8.0.1-1.amzn2023.x86_64. 3.8 MB/s | 249 kB     00:00
--------------------------------------------------------------------------------
Total                                           2.5 MB/s | 399 kB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Upgrading        : libcurl-minimal-8.0.1-1.amzn2023.x86_64                1/4
  Upgrading        : curl-minimal-8.0.1-1.amzn2023.x86_64                   2/4
  Cleanup          : curl-minimal-7.88.1-1.amzn2023.0.1.x86_64              3/4
  Cleanup          : libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64           4/4
  Running scriptlet: libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64           4/4
  Verifying        : libcurl-minimal-8.0.1-1.amzn2023.x86_64                1/4
  Verifying        : libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64           2/4
  Verifying        : curl-minimal-8.0.1-1.amzn2023.x86_64                   3/4
  Verifying        : curl-minimal-7.88.1-1.amzn2023.0.1.x86_64              4/4

Upgraded:
  curl-minimal-8.0.1-1.amzn2023.x86_64  libcurl-minimal-8.0.1-1.amzn2023.x86_64

Complete!
```

Dieselben Paketversionen werden aktualisiert, auch wenn sie `--releasever=latest` als Anforderung für das minimale Update verwendet werden, das *dnf* zur Behebung der Empfehlung erforderlich ist.

Wenn Sie den regulären `dnf upgrade` Befehl mit der `--advisory` Option verwenden, werden die entsprechenden Pakete, die in der Empfehlung erwähnt werden, auf die neueste verfügbare Version aktualisiert, die möglicherweise neuer ist als die in der Empfehlung angegebene Version.

> **Note**
>
> Sofern das `system-release` Paket nicht aktualisiert wird, ändert sich die Version der AL2 023-Repositorys, für die gesperrt `dnf` ist, nicht.

> **Warning**
>
> Wenn Updates aus einer anderen Version von AL2 023 installiert werden, ohne die Version des Projektarchivs zu ändern, für das gesperrt `dnf` ist, muss bei allen nachfolgenden Mutationsvorgängen Vorsicht walten. `dnf` Wenn Sie beispielsweise Pakete installieren oder aktualisieren, kann die ältere Version, die Sie weiterhin verwenden, diese neuen Abhängigkeiten möglicherweise nicht erfüllen, da sich die Paketabhängigkeiten in der neueren Version möglicherweise geändert haben.

Das folgende Beispiel wird in einem Container der Version AL2 023, Version 2023.0.20230315 ausgeführt, der sich auf die neueste Version von AL2 023 bezieht, deren Zeitpunkt der Erstellung 2023.5.20240708 war. Beachten Sie, dass sowohl die Version, auf die aktualisiert `curl` wird, neuer ist als die Version, auf die aktualisiert wurde, dass diese neuere Version jedoch neue Abhängigkeiten mit sich bringt`update-minimal`.

```
$ dnf upgrade -y --releasever=latest --advisory ALAS2023-2023-193
Amazon Linux 2023 repository                    80 MB/s |  25 MB     00:00
Last metadata expiration check: 0:00:04 ago on Mon Jul 22 20:48:38 2024.
Dependencies resolved.
================================================================================
 Package                  Arch     Version                 Repository     Size
================================================================================
Upgrading:
 curl-minimal             x86_64   8.5.0-1.amzn2023.0.4    amazonlinux   160 k
 libcurl-minimal          x86_64   8.5.0-1.amzn2023.0.4    amazonlinux   275 k
 libnghttp2               x86_64   1.59.0-3.amzn2023.0.1   amazonlinux    79 k
Installing dependencies:
 libpsl                   x86_64   0.21.1-3.amzn2023.0.2   amazonlinux    61 k
 publicsuffix-list-dafsa  noarch   20240212-61.amzn2023    amazonlinux    59 k

Transaction Summary
================================================================================
Install  2 Packages
```

```
Upgrade  3 Packages

Total download size: 634 k
Downloading Packages:
(1/5): publicsuffix-list-dafsa-20240212-61.amzn 1.1 MB/s |  59 kB     00:00
(2/5): curl-minimal-8.5.0-1.amzn2023.0.4.x86_64 2.6 MB/s | 160 kB     00:00
(3/5): libpsl-0.21.1-3.amzn2023.0.2.x86_64.rpm  949 kB/s |  61 kB     00:00
(4/5): libnghttp2-1.59.0-3.amzn2023.0.1.x86_64. 3.7 MB/s |  79 kB     00:00
(5/5): libcurl-minimal-8.5.0-1.amzn2023.0.4.x86 6.7 MB/s | 275 kB     00:00
--------------------------------------------------------------------------------
Total                                           3.5 MB/s | 634 kB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Upgrading        : libnghttp2-1.59.0-3.amzn2023.0.1.x86_64                1/8
  Installing       : publicsuffix-list-dafsa-20240212-61.amzn2023.noarch    2/8
  Installing       : libpsl-0.21.1-3.amzn2023.0.2.x86_64                    3/8
  Upgrading        : libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64            4/8
  Upgrading        : curl-minimal-8.5.0-1.amzn2023.0.4.x86_64               5/8
  Cleanup          : curl-minimal-7.88.1-1.amzn2023.0.1.x86_64              6/8
  Cleanup          : libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64           7/8
  Cleanup          : libnghttp2-1.51.0-1.amzn2023.x86_64                    8/8
  Running scriptlet: libnghttp2-1.51.0-1.amzn2023.x86_64                    8/8
  Verifying        : libpsl-0.21.1-3.amzn2023.0.2.x86_64                    1/8
  Verifying        : publicsuffix-list-dafsa-20240212-61.amzn2023.noarch    2/8
  Verifying        : curl-minimal-8.5.0-1.amzn2023.0.4.x86_64               3/8
  Verifying        : curl-minimal-7.88.1-1.amzn2023.0.1.x86_64              4/8
  Verifying        : libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64            5/8
  Verifying        : libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64           6/8
  Verifying        : libnghttp2-1.59.0-3.amzn2023.0.1.x86_64                7/8
  Verifying        : libnghttp2-1.51.0-1.amzn2023.x86_64                    8/8

Upgraded:
  curl-minimal-8.5.0-1.amzn2023.0.4.x86_64
  libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64
  libnghttp2-1.59.0-3.amzn2023.0.1.x86_64
Installed:
  libpsl-0.21.1-3.amzn2023.0.2.x86_64
  publicsuffix-list-dafsa-20240212-61.amzn2023.noarch
```

```
Complete!
```

SELinux Einstellmodi für AL2 023

Standardmäßig ist Security Enhanced Linux (SELinux) auf `enabled` den `permissive` Modus 023 eingestellt. AL2 Im permissiven Modus werden Zugriffsverweigerungen zwar protokolliert, aber nicht erzwungen. SELinux ist eine Sammlung von Kernelfunktionen und Hilfsprogrammen zur Bereitstellung einer starken, flexiblen MAC-Architektur (Mandatory Access Control) für die wichtigsten Subsysteme des Kernels.

SELinux bietet einen verbesserten 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 SELinux Features und Funktionen finden Sie unter SELinux Notebook und Policy Languages“.

Themen

- SELinux Standardstatus und -modi für AL2 023
- Wechseln in den enforcing-Modus
- Option zum Deaktivieren SELinux für AL2 023

SELinux Standardstatus und -modi für AL2 023

Für AL2 023 ist dies SELinux standardmäßig der Modus `enabled` und ist auf Modus eingestellt. `permissive` Im `permissive`-Modus werden Zugriffsverweigerungen protokolliert, aber nicht durchgesetzt.

Die **`sestatus`** Befehle **`getenforce`** oder informieren Sie über den aktuellen SELinux Status, die Richtlinie und den aktuellen Modus.

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 den `permissive` Modus ausführen, beschriften Benutzer Dateien möglicherweise falsch. Wenn Sie SELinux im `disabled` Status ausführen, werden Dateien nicht beschriftet. Falsch benannte oder unbenannte Dateien können Probleme verursachen, wenn Sie in den `enforcing`-Modus wechseln.

SELinux benennt Dateien automatisch neu, um dieses Problem zu vermeiden. SELinux verhindert Probleme mit der automatischen Neubeschriftung, wenn Sie den Status auf ändern. `enabled`

Wechseln in den **`enforcing`**-Modus

Wenn Sie laufen SELinux im `enforcing` Modus, der SELinux Utility ist `enforcing` die konfigurierte Richtlinie. SELinux steuert die Funktionen ausgewählter Anwendungen, indem der Zugriff auf der Grundlage der Richtlinienregeln zugelassen oder verweigert wird.

Um den aktuellen zu finden SELinux Modus, führe den `getenforce` Befehl aus.

```
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 SELinux benennt alle Dateien und Verzeichnisse im System neu. SELinux fügt auch das hinzu SELinux Kontext für Dateien und Verzeichnisse, die erstellt wurden wann SELinux war`disabled`.

Nach dem Wechsel in `enforcing` den Modus SELinux könnte einige Aktionen ablehnen, weil sie falsch sind oder fehlen SELinux politische Regeln. Sie können die Aktionen anzeigen, die SELinux lehnt mit dem folgenden Befehl ab.

```
$ sudo ausearch -m AVC,USER_AVC,SELINUX_ERR,USER_SELINUX_ERR -ts recent
```

Verwenden Sie cloud-init um den Modus zu aktivieren **`enforcing`**

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 SELinux für AL2 023

Wenn du deaktivierst SELinux, SELinux Die Richtlinie ist nicht geladen oder wird nicht durchgesetzt, und Access Vector Cache (AVC) -Meldungen werden nicht protokolliert. Sie verlieren alle Vorteile des Laufens SELinux.

Anstatt zu deaktivieren SELinux, wir empfehlen, den `permissive` Modus zu verwenden. Der Betrieb im `permissive` Modus kostet nur wenig mehr als das Deaktivieren SELinux vollständig. 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.

Änderung SELinux in den **`permissive`** Modus

Wenn du rennst SELinux im `permissive` Modus SELinux Die Richtlinie wird nicht durchgesetzt. Im `permissive` Modus SELinux protokolliert AVC-Nachrichten, verweigert jedoch keine Operationen. Sie können diese AVC-Meldungen zur Problembehandlung, zum Debuggen und SELinux politische Verbesserungen.

Um sich zu ändern SELinux Gehen Sie wie folgt vor, 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 SELinux

Wenn du deaktivierst SELinux, SELinux Die Richtlinie ist nicht geladen oder wird nicht durchgesetzt, und AVC-Meldungen werden nicht protokolliert. Sie verlieren alle Vorteile des Laufens SELinux.

Um zu deaktivieren SELinux, gehen Sie 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 dies zu bestätigen SELinux ist`Disabled`.

```
$ getenforce
Disabled
```

Weitere Informationen zur SELinux, siehe SELinux Notizbuch und SELinux Konfiguration.

Aktivieren Sie den FIPS-Modus auf 023 AL2

In diesem Abschnitt wird erklärt, wie die Federal Information Processing Standards (FIPS) auf 023 aktiviert werden. AL2 Weitere Informationen über FIPS finden Sie unter:

- Federal Information Processing Standard (FIPS)
- Einhaltung FAQs: Bundesstandards für die Informationsverarbeitung

> **Note**
>
> In diesem Abschnitt wird die Aktivierung beschrieben FIPS Modus in AL2 023, er deckt nicht den Zertifizierungsstatus von AL2 023-Kryptografiemodulen ab.

Voraussetzungen

- Eine bestehende EC2 Amazon-Instance AL2 023 (AL2023.2 oder höher) mit Internetzugang zum Herunterladen der erforderlichen Pakete. Weitere Informationen zum Starten einer AL2 EC2 023-Amazon-Instance finden Sie unter AL2023 über die EC2 Amazon-Konsole starten.

- Sie müssen über SSH oder AWS Systems Manager eine Verbindung zu Ihrer EC2 Amazon-Instance herstellen. Weitere Informationen finden Sie unter Verbindung zu 023 Instances herstellen AL2.

> **Important**
>
> ED25519 SSH-Benutzerschlüssel werden im FIPS-Modus nicht unterstützt. Wenn Sie Ihre EC2 Amazon-Instance mit einem ED25519 SSH-Schlüsselpaar gestartet haben, müssen Sie mithilfe eines anderen Algorithmus (z. B. RSA) neue Schlüssel generieren. Andernfalls verlieren Sie möglicherweise den Zugriff auf Ihre Instance, nachdem Sie den FIPS-Modus aktiviert haben. Weitere Informationen finden Sie unter Schlüsselpaare erstellen im EC2 Amazon-Benutzerhandbuch.

Aktivieren des FIPS-Modus

1. Stellen Sie über SSH oder eine Connect zu Ihrer AL2 023-Instance her. AWS Systems Manager
2. Stellen Sie sicher, dass das System auf dem neuesten Stand ist. Weitere Informationen finden Sie unter Paket- und Betriebssystemupdates in AL2 023 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. Dadurch wird der FIPS-Modus systemweit für die in der 023 FAQ aufgeführten Module aktiviert AL2

   ```
   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.
```

Aktivieren Sie den FIPS-Modus in einem AL2 023-Container

In diesem Abschnitt wird erklärt, wie die Federal Information Processing Standards (FIPS) in einem AL2 023-Container aktiviert werden. Weitere Informationen über FIPS finden Sie unter:

- Federal Information Processing Standard (FIPS)
- Einhaltung FAQs: Bundesstandards für die Informationsverarbeitung

> **Note**
>
> In diesem Abschnitt wird beschrieben, wie Sie die Aktivierung durchführen FIPS Modus in einem AL2 023-Container. Er deckt nicht den Zertifizierungsstatus von AL2 023-Kryptografiemodulen ab.

Voraussetzungen

- Eine bestehende EC2 Amazon-Instance AL2 023 (AL2023.2 oder höher) mit Internetzugang zum Herunterladen der erforderlichen Pakete. Weitere Informationen zum Starten einer AL2 EC2 023-Amazon-Instance finden Sie unter AL2023 über die EC2 Amazon-Konsole starten.
- Sie müssen über SSH oder AWS Systems Manager eine Verbindung zu Ihrer EC2 Amazon-Instance herstellen. Weitere Informationen finden Sie unter Verbindung zu 023 Instances herstellen AL2.

> **Important**
>
> Der `fips-mode-setup` Befehl funktioniert innerhalb des Containers nicht richtig. Bitte lesen Sie die folgenden Schritte, um den FIPS-Modus in einem AL2 023-Container richtig zu konfigurieren.

Aktivieren Sie den FIPS-Modus in einem 023-Container AL2

1. Der FIPS-Modus muss zuerst auf dem AL2 023-Container-Host aktiviert werden. Folgen Sie den Anweisungen unterAktivieren Sie den FIPS-Modus auf 023 AL2, um den FIPS-Modus auf dem Host zu aktivieren.
2. Stellen Sie mithilfe von SSH oder eine Connect zu Ihrer AL2 023-Container-Host-Instance her. AWS Systems Manager
3. Der FIPS-Modus wird in einem AL2 023-Container automatisch aktiviert, wenn sich der AL2 023-Host im FIPS-Modus befindet und vom Container aus zugänglich `/proc/sys/crypto/fips_enabled` ist. Wenn der Inhalt von `/proc/sys/crypto/fips_enabled` ist, ist `0` FIPS nicht aktiviert, und ein Wert von 1 gibt an, dass der FIPS-Modus aktiviert ist.

   Sie können überprüfen, ob FIPS aktiviert ist, indem Sie den folgenden Befehl sowohl auf dem AL2 023-Host als auch auf dem Container ausführen:

   ```
   cat /proc/sys/crypto/fips_enabled
   ```

4. Als Nächstes aktivieren Sie die FIPS-Kryptorichtlinien im Container. Es gibt mehrere Möglichkeiten, dies zu erreichen, die in den folgenden Optionen beschrieben werden. Verwenden Sie die Option, die für Ihre Umgebung am besten geeignet ist.

   a. Aktivieren Sie die FIPS-Kryptorichtlinien manuell im Container mit dem folgenden Befehl: `update-crypto-policies`

      ```
      # Run these commands inside the container
      dnf install -y crypto-policies-scripts
      update-crypto-policies --set FIPS
      ```

   b. Erstellen Sie `bind` Mounts innerhalb des AL2 023-Containers (das `podman` funktioniert ähnlich wie in anderen Distributionen):

      ```
      # Run these commands inside the container
      mount --bind /usr/share/crypto-policies/back-ends/FIPS /etc/crypto-policies/back-
      ends
      echo "FIPS" > /usr/share/crypto-policies/default-fips-config
      mount --bind /usr/share/crypto-policies/default-fips-config /etc/crypto-policies/
      config
      ```

   c. Es ist auch möglich, einen Bind-Mount zu erstellen, sodass der AL2 023-Container den crypto-policies des AL2 023-Hosts entspricht. Das Folgende dient nur als Beispiel. Diese Konfiguration

kann zu Problemen führen, wenn es inkompatible Unterschiede in den crypto-policies und Paketversionen zwischen dem Container und dem Host gibt:

```
sudo docker pull amazonlinux:2023
sudo docker run --mount type=bind,readonly,src=/etc/crypto-policies,dst=/etc/
crypto-policies -it amazonlinux:2023
```

5. Nachdem Sie die obigen Schritte ausgeführt haben, können Sie mit den folgenden Befehlen erneut überprüfen, ob FIPS im Container aktiviert ist:

```
$ cat /etc/crypto-policies/config
FIPS

$ cat /proc/sys/crypto/fips_enabled
1
```

Tauschen Sie OpenSSL FIPS-Anbieter auf 023 aus AL2

In diesem Abschnitt wird erklärt, wie Sie auf 023 zwischen dem `latest` und dem `certified` OpenSSL FIPS-Anbieter wechseln können. AL2

Weitere Informationen über FIPS finden Sie unter:

- Federal Information Processing Standard (FIPS)
- Einhaltung FAQs: Bundesstandards für die Informationsverarbeitung
- FedRAMP-Richtlinie für die Auswahl und Verwendung von kryptografischen Modulen

> **Important**
>
> Auf AL2 Version 023.7 und höher ist der Standard-OpenSSL-FIPS-Anbieter das `openssl-fips-provider-latest` Paket, das regelmäßig Bugfix- und Sicherheitsupdates erhält. Die folgenden Anweisungen gelten nur für Kunden, die sich an das Paket anheften möchten. `openssl-fips-provider-certified` Diese Version des FIPS-Anbieters entspricht der Prüfsumme auf dem NIST-Zertifikat und verfügt möglicherweise nicht über die neuesten Updates.

Weitere Informationen zu FIPS-zertifizierten Modulen und Paketversionen finden Sie in den häufig gestellten Fragen zu AL2 023.

Voraussetzungen

- Eine bestehende EC2 Amazon-Instance AL2 023 (AL2023.7 oder höher) mit Internetzugang zum Herunterladen der erforderlichen Pakete. Weitere Informationen zum Starten einer AL2 EC2 023-Amazon-Instance finden Sie unter AL2023 über die EC2 Amazon-Konsole starten.
- Sie müssen über SSH oder AWS Systems Manager eine Verbindung zu Ihrer EC2 Amazon-Instance herstellen. Weitere Informationen finden Sie unter Verbindung zu 023 Instances herstellen AL2.
- Folgen Sie den Anweisungen unter, um den FIPS-Modus auf AL2 023 zu aktivieren. Aktivieren Sie den FIPS-Modus auf 023 AL2

Wechseln Sie zwischen und **`openssl-fips-provider-latestopenssl-fips-provider-certified`**

1. Verwenden Sie`dnf`, um den OpenSSL FIPS-Anbieter zu wechseln:

   ```
   sudo dnf -y swap openssl-fips-provider-latest openssl-fips-provider-certified
   ```

2. Vergewissern Sie sich, dass Sie den zertifizierten OpenSSL FIPS-Anbieter verwenden. Wenn sich AL2 023 im FIPS-Modus befindet, führen Sie den folgenden Befehl aus:

   ```
   openssl list -providers
   ```

   Die Ausgabe sollte folgendermaßen aussehen:

   ```
   Providers:
     base
       name: OpenSSL Base Provider
       version: 3.2.2
       status: active
     default
       name: OpenSSL Default Provider
       version: 3.2.2
       status: active
   ```

```
fips
  name: Amazon Linux 2023 - OpenSSL FIPS Provider
  version: 3.0.8-d694bfa693b76001
  status: active
```

AL2023 Kernelhärtung

Der 6.1-Linux-Kernel in AL2 023 ist mit verschiedenen Hardening-Optionen und -Funktionen konfiguriert und gebaut.

Kernel-Hardening-Optionen (architekturunabhängig)

CONFIG-Option	AL2023/6.1/ aarch64	AL2023/6,1/ x86_64	AL2023/6.12/ aarch64	AL2023/6,12/ x86_64
CONFIG_ACPI_CUSTOM_METHOD	n	n	N/A	–
CONFIG_BINFMT_MISC	m	m	m	m
CONFIG_BUG	y	y	y	y
CONFIG_BUG_ON_DATA_CORRUPTION	y	y	y	y
CONFIG_CFI_CLANG	–	–	–	–
CONFIG_CFI_PERMISSIVE	–	–	–	–
CONFIG_COMPAT	y	y	y	y

CONFIG-Option	AL2023/6.1/ aarch64	AL2023/6,1/ x86_64	AL2023/6.12/ aarch64	AL2023/6,12/ x86_64
CONFIG_COMPAT_BRK	n	n	n	n
CONFIG_COMPAT_VDSO	–	n	–	n
CONFIG_DEBUG_CREDENTIALS	n	n	–	–
CONFIG_DEBUG_LIST	y	y	y	y
CONFIG_DEBUG_NOTIFIERS	n	n	n	n
CONFIG_DEBUG_SG	n	n	n	n
CONFIG_DEBUG_VIRTUAL	n	n	n	n
CONFIG_DEBUG_WX	n	n	n	n
CONFIG_DEFAULT_MMAP_MIN_ADDR	65536	65536	65536	65536
CONFIG_DEVKMEM	–	–	–	–
CONFIG_DEVMEM	n	n	n	n

CONFIG-Option	AL2023/6.1/ aarch64	AL2023/6,1/ x86_64	AL2023/6.12/ aarch64	AL2023/6,12/ x86_64
CONFIG_EFI_DISABLE_PCI_DMA	n	n	n	n
CONFIG_FORTIFY_SOURCE	y	y	y	y
CONFIG_HARDENED_USERCOPY	y	y	y	y
CONFIG_HARDENED_USERCOPY_FALLBACK	–	–	–	–
CONFIG_HARDENED_USERCOPY_PAGESPAN	–	–	–	–
CONFIG_HIBERNATION	y	y	y	y
CONFIG_HW_RANDOM_TPM	–	–	–	–
CONFIG_INET_DIAG	m	m	m	m

CONFIG-Option	AL2023/6.1/ aarch64	AL2023/6,1/ x86_64	AL2023/6.12/ aarch64	AL2023/6,12/ x86_64
CONFIG_INIT_ON_ALLOC_DEFAULT_ON	n	n	n	n
CONFIG_INIT_ON_FREE_DEFAULT_ON	n	n	n	n
CONFIG_INIT_STACK_ALL_ZERO	–	–	–	–
CONFIG_IOMMU_DEFAULT_DMA_STRICT	n	n	n	n
CONFIG_IOMMU_SUPPORT	y	y	y	y
CONFIG_IO_STRICT_DEVMEM	–	–	–	–
CONFIG_KEXEC	y	y	y	y
CONFIG_KFENCE	n	n	n	n

CONFIG-Option	AL2023/6.1/ aarch64	AL2023/6,1/ x86_64	AL2023/6.12/ aarch64	AL2023/6,12/ x86_64
CONFIG_LDISC_AUTOLOAD	n	n	n	n
CONFIG_LEGACY_PTYS	n	n	n	n
CONFIG_LOCK_DOWN_KERNEL_FORCE_CONFIDENTIALITY	n	n	n	n
CONFIG_MODULES	y	y	y	y
CONFIG_MODULE_SIG	y	y	y	y
CONFIG_MODULE_SIG_ALL	y	y	y	y
CONFIG_MODULE_SIG_FORCE	n	n	n	n
CONFIG_MODULE_SIG_HASH	sha512	sha512	sha512	sha512
CONFIG_MODULE_SIG_KEY	certs/signing_key.pem	certs/signing_key.pem	certs/signing_key.pem	certs/signing_key.pem

CONFIG-Option	AL2023/6.1/ aarch64	AL2023/6,1/ x86_64	AL2023/6.12/ aarch64	AL2023/6,12/ x86_64
CONFIG_MODULE_SIG_SHA512	y	y	y	y
CONFIG_PAGE_POISONING	n	n	n	n
CONFIG_PAGE_POISONING_NO_SANITY	–	–	–	–
CONFIG_PAGE_POISONING_ZERO	–	–	–	–
CONFIG_PANIC_ON_OOPS	y	y	y	y
CONFIG_PANIC_TIMEOUT	0	0	0	0
CONFIG_PROC_KCORE	y	y	y	y
CONFIG_RANDOMIZE_KSTACK_OFFSET_DEFAULT	n	n	n	n

CONFIG-Option	AL2023/6.1/ aarch64	AL2023/6,1/ x86_64	AL2023/6.12/ aarch64	AL2023/6,12/ x86_64
CONFIG_RANDOM_TRUST_BOOTLOADER	y	y	–	–
CONFIG_RANDOM_TRUST_CPU	y	y	–	–
CONFIG_REFCOUNT_FULL	–	–	–	–
CONFIG_SCHED_CORE	–	y	–	y
CONFIG_SCHED_STACK_END_CHECK	y	y	y	y
CONFIG_SECCOMP	y	y	y	y
CONFIG_SECCOMP_FILTER	y	y	y	y
CONFIG_SECURITY	y	y	y	y
CONFIG_SECURITY_DMESG_RESTRICT	y	y	y	y

CONFIG-Option	AL2023/6.1/ aarch64	AL2023/6,1/ x86_64	AL2023/6.12/ aarch64	AL2023/6,12/ x86_64
CONFIG_SECURITY_LANDLOCK	n	n	n	n
CONFIG_SECURITY_LOCKDOWN_LSM	y	y	y	y
CONFIG_SECURITY_LOCKDOWN_LSM_EARLY	y	y	y	y
CONFIG_SECURITY_SELINUX_BOOTPARAM	y	y	y	y
CONFIG_SECURITY_SELINUX_DEVELOP	y	y	y	y
CONFIG_SECURITY_SELINUX_DISABLE	n	n	–	–
CONFIG_SECURITY_WRITABLE_HOOKS	–	–	–	N/A

CONFIG-Option	AL2023/6.1/ aarch64	AL2023/6,1/ x86_64	AL2023/6.12/ aarch64	AL2023/6,12/ x86_64
CONFIG_SECURITY_YAMA	y	y	y	y
CONFIG_SHUFFLE_PAGE_ALLOCATOR	y	y	y	y
CONFIG_SLAB_FREELIST_HARDENED	y	y	y	y
CONFIG_SLAB_FREELIST_RANDOM	y	y	y	y
CONFIG_SLUB_DEBUG	y	y	y	y
CONFIG_STACKPROTECTOR	y	y	y	y
CONFIG_STACKPROTECTOR_STRONG	y	y	y	y
CONFIG_STATIC_USERMODEHELPER	n	n	n	n

CONFIG-Option	AL2023/6.1/ aarch64	AL2023/6,1/ x86_64	AL2023/6.12/ aarch64	AL2023/6,12/ x86_64
CONFIG_STRICT_DEVMEM	n	n	n	n
CONFIG_STRICT_KERNEL_RWX	y	y	y	y
CONFIG_STRICT_MODULE_RWX	y	y	y	y
CONFIG_SYN_COOKIES	y	y	y	y
CONFIG_VMAP_STACK	y	y	y	y
CONFIG_WERROR	n	n	n	n
CONFIG_ZERO_CALL_USED_REGS	n	n	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, 023 legt diese Konfigurationsoption jedoch nicht auf die von KSPP empfohlenen Einstellungen fest. AL2 In AL2 023 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, von `CONFIG_DEBUG` denen gesteuert wird, sind normalerweise für die Verwendung in Kerneln konzipiert, die für Debugging-Probleme gebaut

wurden, und Dinge wie Leistung haben keine Priorität. AL2023 aktiviert die Härtungsoption. `CONFIG_DEBUG_LIST`

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, AL2 023 legt diese Konfigurationsoption jedoch nicht auf die von KSPP empfohlenen Einstellungen fest.

-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 mit AL2 023 ein 6.1-Kernel ausgeliefert wird, ist diese Option für 023 nicht mehr relevant. AL2

Die `CONFIG_HARDENED_USERCOPY_PAGESPAN` Option existierte in Kerneln hauptsächlich als Debugging-Option für Entwickler und gilt in 023 nicht mehr für den 6.1-Kernel. AL2

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

Hibernation-Unterstützung

Obwohl diese Option zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen gehört, setzt AL2 023 diese Konfigurationsoption nicht auf das, was KSPP empfiehlt. Diese Option muss aktiviert sein, um Hibernation der On-Demand Instance und Hibernation für unterbrochene Spot Instances zu unterstützen.

Zufallszahlengenerierung

Der AL2 023-Kernel ist so konfiguriert, dass sichergestellt ist, dass eine angemessene Entropie für die Verwendung innerhalb des Kernels verfügbar ist. EC2

`CONFIG_INET_DIAG`

Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, AL2 023 legt diese Konfigurationsoption jedoch nicht auf die von KSPP empfohlenen Einstellungen fest. In AL2 023 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

Obwohl diese Option zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen gehört, setzt AL2 023 diese Konfigurationsoption nicht auf die von KSPP empfohlenen Einstellungen. Diese Optionen sind in AL2 023 deaktiviert, da sich die standardmäßige Aktivierung dieser Funktion möglicherweise auf die Leistung auswirkt. 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, AL2 023 legt diese Konfigurationsoption jedoch nicht auf die von KSPP empfohlenen Einstellungen fest. Diese Option erfordert GCC 12 oder höher, während AL2 023 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, AL2 023 setzt diese Konfigurationsoption jedoch nicht auf die von KSPP empfohlenen Einstellungen. 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, AL2 023 legt diese Konfigurationsoption jedoch nicht auf die von KSPP empfohlenen Einstellungen fest.

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, AL2 023 legt diese Konfigurationsoption jedoch nicht auf die von KSPP empfohlenen Einstellungen fest. Wenn Secure Boot nicht verwendet wird, kann das Lockdown-LSM aktiviert und nach Bedarf konfiguriert werden.

Page Poisoning

Obwohl diese Option zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen gehört, setzt AL2 023 diese Konfigurationsoption nicht auf das, was KSPP empfiehlt. Ähnlich Der gesamte Kernel-Page- und Slab-Allocator-Speicher wird bei Zuweisung und Freigabe auf Null gesetzt wie diese Option ist im AL2 023-Kernel aufgrund möglicher Auswirkungen auf die Leistung deaktiviert.

Stack Protector

Der AL2 023-Kernel ist mit der Stack-Protector-Funktion von gebaut GCC aktiviert mit der Option. `-fstack-protector-strong`

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

seccomp BPF API

Das Tool seccomp Die Härtungsfunktion wird von Software wie `systemd` Container-Runtimes verwendet, um Userspace-Anwendungen zu schützen.

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

`panic()` timeout

Der AL2 023-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, AL2 023 setzt diese Konfigurationsoption jedoch nicht auf die von KSPP empfohlenen Einstellungen. Dies kann über `sysctl /proc/sys/kernel/panic` oder über die Kernel-Befehlszeile konfiguriert werden.

Sicherheitsmodelle

AL2023 ist standardmäßig im Modus „Zulässig“ aktiviert SELinux . Weitere Informationen finden Sie unter SELinux Einstellmodi für AL2 023.

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, AL2 023 legt diese Konfigurationsoption jedoch nicht auf die von KSPP empfohlenen Einstellungen fest.

Randomisierung des Kernel-Stack-Offsets bei Eingabe von „syscall“

Obwohl diese Option zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen gehört, setzt AL2 023 diese Konfigurationsoption nicht auf das, was KSPP empfiehlt. Dies kann durch eine Einstellung `randomize_kstack_offset=on` in der Kernel-Befehlszeile aktiviert werden.

Prüfungen zur Referenzzählung (`CONFIG_REFCOUNT_FULL`)

Obwohl diese Option zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen gehört, setzt AL2 023 diese Konfigurationsoption nicht auf das, was KSPP empfiehlt. Diese Option ist derzeit aufgrund möglicher Auswirkungen auf die Leistung nicht aktiviert.

Scheduler ist sich bewusst für SMT Kerne () `CONFIG_SCHED_CORE`

Mit dem AL2 023-Kernel wurde gebaut`CONFIG_SCHED_CORE`, der 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 AL2 023-Kernel wurde mit aktivierter Option gebaut. CONFIG_SCHED_STACK_END_CHECK Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Memory Allocator Hardening

Der AL2 023-Kernel ermöglicht das Härten des Kernel-Speicherzuweisers mit den OptionenCONFIG_SHUFFLE_PAGE_ALLOCATOR, undCONFIG_SLAB_FREELIST_HARDENED. CONFIG_SLAB_FREELIST_RANDOM Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

SLUB Debugging-Unterstützung

Der AL2 023-Kernel wird aktiviert, CONFIG_SLUB_DEBUG da diese Option optionale Debugging-Funktionen für den Allocator aktiviert, die in der Kernel-Befehlszeile aktiviert werden können. Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

CONFIG_STATIC_USERMODEHELPER

Obwohl diese Option zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen gehört, setzt AL2 023 diese Konfigurationsoption nicht auf das, was KSPP empfiehlt. 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 (und) **CONFIG_STRICT_KERNEL_RWXCONFIG_STRICT_MODULE_RWX**

Der AL2 023-Kernel ist so konfiguriert, dass er Kernel- und Kernelmodultext markiert und rodata Speicher als schreibgeschützt und Nicht-Text-Speicher als nicht ausführbar markiert. Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

TCP Syncookie-Unterstützung () **CONFIG_SYN_COOKIES**

Der AL2 023-Kernel wurde mit Unterstützung für TCP-Syncookies gebaut. Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Virtuell zugeordneter Stack mit Guard-Seiten (**CONFIG_VMAP_STACK**)

Der AL2 023-Kernel ist mit CONFIG_VMAP_STACK gebaut und ermöglicht virtuell zugeordnete Kernel-Stacks mit Schutzseiten. Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Build mit Compiler-Warnungen als Fehler angezeigt (**CONFIG_WERROR**)

Obwohl diese Option zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen gehört, setzt AL2 023 diese Konfigurationsoption nicht auf die von KSPP empfohlenen Einstellungen.

Nullstellung des Registers bei Funktionsbeendigung (**CONFIG_ZERO_CALL_USED_REGS**)

Obwohl diese Option zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen gehört, setzt AL2 023 diese Konfigurationsoption nicht auf das, was KSPP empfiehlt.

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 023-Kernel besteht aus AL2 GCC eher als clang, sodass die CONFIG_CFI_CLANG Härtungsoption nicht aktiviert werden kann, was auch bedeutet, dass sie CONFIG_CFI_PERMISSIVE nicht anwendbar ist. Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, AL2 023 legt diese Konfigurationsoption jedoch nicht auf die von KSPP empfohlenen Einstellungen fest.

Spezifische Kernel-Hardening-Optionen für x86-64

CONFIG-Option	AL2023/6.1/ aarch64	AL2023/6,1/ x86_64	AL2023/6.12/ aarch64	AL2023/6,12/ x86_64
CONFIG_AMD_IOMMU	N/A	y	–	y
CONFIG_AMD_IOMMU_V2	–	y	–	–

CONFIG-Option	AL2023/6.1/ aarch64	AL2023/6,1/ x86_64	AL2023/6.12/ aarch64	AL2023/6,12/ x86_64
CONFIG_IA32_EMULATION	–	y	–	y
CONFIG_INTEL_IOMMU	–	y	–	y
CONFIG_INTEL_IOMMU_DEFAULT_ON	–	n	–	n
CONFIG_INTEL_IOMMU_SVM	–	n	–	n
CONFIG_LEGACY_VSYSCALL_NONE	–	n	–	n
CONFIG_MODIFY_LDT_SYSCALL	–	n	–	n
CONFIG_PAGE_TABLE_ISOLATION	–	y	–	–
CONFIG_RANDOMIZE_MEMORY	–	y	–	y
CONFIG_X86_64	–	y	–	y

CONFIG-Option	AL2023/6.1/ aarch64	AL2023/6,1/ x86_64	AL2023/6.12/ aarch64	AL2023/6,12/ x86_64
CONFIG_X86_MSR	–	y	–	y
CONFIG_X86_VSYSCALL_EMULATION	–	y	–	y
CONFIG_X86_X32	–	–	–	–
CONFIG_X86_X32_ABI	–	n	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 AL2 023-Kernel wird mit Unterstützung für AMD und Intel gebaut IOMMUs. Diese Option ist eine der empfohlenen Einstellungen für das 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, AL2 023 legt diese Konfigurationsoption jedoch nicht auf die von KSPP empfohlenen Einstellungen fest.

Die `CONFIG_INTEL_IOMMU_SVM` Option ist derzeit in 023 nicht aktiviert. AL2 Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, AL2 023 legt diese Konfigurationsoption jedoch nicht auf die von KSPP empfohlenen Einstellungen fest.

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 AL2 023 die `CONFIG_X86_VSYSCALL_EMULATION` Option nicht und aktiviert 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, AL2 023 legt diese Konfigurationsoption jedoch nicht auf die von KSPP empfohlenen Einstellungen fest.

Das Tool x32 Das native 32-Bit-ABI für 64-Bit-Prozessoren ist nicht aktiviert (und). `CONFIG_X86_X32` `CONFIG_X86_X32_ABI` Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

Modellspezifisches x86-Register (MSR) Unterstützung

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, AL2 023 legt diese Konfigurationsoption jedoch nicht auf die von KSPP empfohlenen Einstellungen fest.

`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 beschädigt wird, z. B. das Ausführen

einiger Programme unter `dosemu` WINEund einige sehr alte Threading-Bibliotheken. 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 er die virtuellen Basisadressen der Kernel-Speicherbereiche nach dem Zufallsprinzip sortiert. 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	AL2023/6.12/ aarch64	AL2023/6,12/ x86_64
CONFIG_ARM64_BTI	y	N/A	y	–
CONFIG_ARM64_BTI_KERNEL	–	–	–	–
CONFIG_ARM64_PTR_AUTH	y	–	y	–
CONFIG_ARM64_PTR_AUTH_KERNEL	y	–	y	–
CONFIG_ARM64_SW_TTBR0_PAN	y	–	y	–

CONFIG-Option	AL2023/6.1/ aarch64	AL2023/6,1/ x86_64	AL2023/6.12/ aarch64	AL2023/6,12/ x86_64
CONFIG_UNMAP_KERNEL_AT_EL0	y	–	y	N/A

Identifizierung des -Abzweigungsziels

Der AL2 023-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 AL2 023 nicht aktiviert, da sie mit erstellt wurde GCC, und die Unterstützung für das Erstellen des Kernels mit dieser Option ist derzeit im Upstream-Kernel aufgrund eines GCC−Fehlers deaktiviert. Diese Option gehört zwar zu den vom Kernel Self Protection Project (KSPP) empfohlenen Einstellungen, AL2 023 setzt diese Konfigurationsoption jedoch nicht auf die von KSPP empfohlenen Einstellungen.

Pointer-Authentifizierung (**CONFIG_ARM64_PTR_AUTH**)

Der AL2 023-Kernel wurde mit Unterstützung für die Pointer Authentication-Erweiterung (Teil der ARMv8 3.3-Erweiterungen) entwickelt, die zur Abschwächung von Techniken der Return Oriented Programming (ROP) 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 AL2 023-Kernel den Absenderadressschutz 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 AL2 023-Kernel ist so konfiguriert, dass er die Zuordnung des Kernels aufhebt, wenn er in userspace () ausgeführt wird. `CONFIG_UNMAP_KERNEL_AT_EL0` Diese Option ist eine der empfohlenen Einstellungen des Kernel Self Protection Project.

UEFI Secure Boot auf 023 AL2

AL2023 unterstützt UEFI Secure Boot ab Version 2023.1. Sie müssen AL2 023 mit EC2 Amazon-Instances verwenden, die sowohl UEFI als auch UEFI Secure Boot unterstützen. Weitere Informationen finden Sie unter Anforderungen für den Start einer EC2 Amazon-Instance im UEFI-Startmodus im EC2 Amazon-Benutzerhandbuch.

AL2023 Instances mit aktiviertem UEFI Secure Boot akzeptieren nur Code auf Kernelebene, einschließlich des Linux-Kernels und der Module, die signiert sind von Amazon so können Sie sicherstellen, dass Ihre Instance nur Codes auf Kernelebene ausführt, die von signiert wurden. AWS

Weitere Informationen zu EC2 Amazon-Instances und UEFI Secure Boot finden Sie unter UEFI Secure Boot for Amazon EC2 Amazon-Instances im EC2 Amazon-Benutzerhandbuch.

Voraussetzungen

- Sie müssen ein AMI mit Version AL2 023 Version 2023.1 oder höher verwenden.
- Der Instance-Typ muss UEFI Secure Boot unterstützen. Weitere Informationen finden Sie unter Anforderungen für den Start einer EC2 Amazon-Instance im UEFI-Startmodus im EC2 Amazon-Benutzerhandbuch.

Aktivieren Sie UEFI Secure Boot auf 023 AL2

Standard AL2 023 AMIs beinhaltet einen Bootloader und einen Kernel, der mit unseren Schlüsseln signiert ist. Sie können UEFI Secure Boot aktivieren, indem Sie entweder vorhandene Instanzen registrieren oder wenn UEFI Secure Boot bereits aktiviert ist, indem Sie ein Image aus AMIs einem Snapshot registrieren. UEFI Secure Boot ist auf dem Standard 023 standardmäßig nicht aktiviert. AL2 AMIs

Der Startmodus AL2 023 AMIs ist auf 023 eingestellt. Dadurch AMIs wird sichergestellt`uefi-preferred`, dass die damit gestarteten Instanzen die UEFI-Firmware verwenden, sofern der

Instanztyp 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 EC2 Amazon-Instances finden Sie unter Instance-Startverhalten mit Amazon EC2 Amazon-Startmodi im EC2 Amazon-Benutzerhandbuch.

Themen

- Registrierung einer vorhandenen Instance
- Image aus einem Snapshot registrieren
- Widerruf-Updates
- So funktioniert UEFI Secure Boot auf 023 AL2
- 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 unterstützt AMIs derzeit kein Nitro Trusted Platform Module (NitroTPM). Wenn Sie NitroTPM zusätzlich zu UEFI Secure Boot benötigen, helfen Ihnen die Informationen im folgenden Abschnitt weiter.

Image aus einem Snapshot registrieren

Wenn Sie ein AMI aus einem Snapshot eines Amazon EBS-Root-Volumes mithilfe der EC2 `register-image` Amazon-API registrieren, können Sie das AMI mit einem binären Blob bereitstellen, der den Status des UEFI-Variablenspeichers enthält. Wenn Sie AL2 023 angeben`UefiData`, aktivieren Sie UEFI Secure Boot und müssen die Schritte im vorherigen Abschnitt nicht ausführen.

Weitere Informationen zum Erstellen und Verwenden eines binären Blobs finden Sie unter Erstellen eines binären Blobs mit einem vorausgefüllten Variablenspeicher im EC2 Amazon-Benutzerhandbuch.

AL2023 bietet einen vorgefertigten Binär-Blob, der direkt auf Amazon-Instances verwendet werden kann. EC2 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 auf AL2 023 AMIs installiert ist.

> **Note**
>
> Um sicherzustellen, dass Sie die neueste Version von Keys and Revocations verwenden, verwenden Sie den Blob aus derselben Version von AL2 023, mit dem 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 für Amazon EC2 Amazon-Instances im Amazon-Benutzerhandbuch. EC2

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.

So funktioniert UEFI Secure Boot auf 023 AL2

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 mit EC2 Amazon-Amazon-Instances im EC2 Amazon-Benutzerhandbuch.

Eigene Schlüssel registrieren

Wie im vorherigen Abschnitt dokumentiert, benötigt Amazon Linux `shim` für UEFI Secure Boot auf Amazon EC2 keinen sicheren Start. Wenn Sie die Dokumentation für andere Linux-Distributionen lesen, finden Sie möglicherweise eine Dokumentation zur Verwaltung der Machine Owner Key

(MOK) -Datenbank`mokutil`, die auf 023 nicht verfügbar ist. AL2 Die Umgebungen `shim` und MOK umgehen einige Einschränkungen der Schlüsselregistrierung in der UEFI-Firmware, die nicht auf die Implementierung von UEFI Secure Boot EC2 durch Amazon zutreffen. Bei Amazon EC2 gibt es Mechanismen, um die Schlüssel im UEFI-Variablenspeicher einfach direkt zu manipulieren.

Wenn Sie Ihre eigenen Schlüssel registrieren möchten, können Sie dies entweder tun, indem Sie den Variablenspeicher innerhalb einer vorhandenen Instance manipulieren (siehe Schlüssel aus der Instance zum Variablenspeicher hinzufügen) oder indem Sie einen binären Blob erstellen, der vorgefüllt ist (siehe Erstellen eines binären Blobs, der einen vorgefüllten Variablenspeicher enthält).

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.