



ユーザーガイド

Amazon Linux 2023



Amazon Linux 2023: ユーザーガイド

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

Amazon の商標およびトレードドレスは、お客様に混乱を招く可能性がある態様、または Amazon の信用を傷つけたり、失わせたりする態様において、Amazon のものではない製品またはサービスに関連して使用してはなりません。Amazon が所有しない他の商標はすべてそれぞれの所有者に帰属します。所有者は必ずしも Amazon との提携や関連があるわけではありません。また、Amazon の支援を受けているとはかぎりません。

Table of Contents

Amazon Linux 2023 とは	1
リリース頻度	1
メジャーリリースおよびマイナーリリース	3
新しいリリースの消費	4
長期的なサポートポリシー	4
命名およびバージョンニング	4
パフォーマンスと運用の最適化	6
Fedora との関係	7
カスタマイズされた cloud-init	7
セキュリティ更新および機能	9
更新の管理	9
クラウド内のセキュリティ	10
SELinux モード	10
コンプライアンスプログラム	10
SSH サーバーのデフォルト	10
OpenSSL 3 の主な機能	10
ネットワークサービス	11
コアツールチェーンパッケージ glibc、gcc、binutils	11
パッケージ管理ツール	12
デフォルトの SSH サーバー設定	13
非推奨の機能	15
compat- パッケージ	15
非推奨の機能は AL1 で廃止され、AL2 では削除されました	15
32 ビット x86 (i686) AMI	16
aws-apitools-* に置き換えられました。AWS CLI	16
systemdAL2 upstart では置き換えられません。	17
AL2 では機能が非推奨になり、AL2023 では機能が削除されました	17
32 ビット x86 (i686) パッケージ	18
aws-apitools-* を に置き換えました AWS CLI	18
bzip2 リビジョン管理システム	19
cgroup v1	19
log4j ホットパッチ (log4j-cve-2021-44228-hotpatch)	19
lsb_release および system-lsb-core パッケージ	20
mccrypt	20

OpenJDK 7 (java-1.7.0-openjdk)	20
Python 2.7	21
rsyslog-openssl が を置き換える rsyslog-gnutls	21
Network Information Service (NIS)/ yp	21
AL2023 では非推奨	21
32 ビット x86 (i686) ランタイムのサポート	22
バークレー DB (libdb)	22
cron	23
IMDSv1	23
pcre バージョン 1	23
System V init (sysvinit)	24
AL2 と AL2023 の比較	25
追加、更新、削除されたパッケージ	26
各リリースのサポート	26
命名およびバージョンニングの変更	26
最適化	26
Python 2.7 は Python 3 に置き換えられました	27
セキュリティ更新	27
SELinux	27
OpenSSL 3	28
IMDSv2	28
log4j ホットパッチ (log4j-cve-2021-44228-hotpatch) の削除	29
安定性向上のための確定的な更新	29
複数のアップストリームから供給されています。	30
AMI ルートファイルシステムとデフォルトの Amazon EBS ボリュームタイプ	30
ネットワークシステムサービス	30
統合コントロールグループ階層 (cgroup v2)	30
タスクのスケジューリング	31
glibc、gcc、および binutils のパッケージ	31
パッケージマネージャー	32
ロギングシステム	32
curl および libcurl のパッケージ変更	32
GNU プライバシーガード (GNUPG)	32
デフォルト JVM としての Amazon Corretto	33
AWS CLI v2	33
UEFI Preferred	33

SSH サーバーのデフォルト設定の変更	33
Extra Packages for Enterprise Linux (EPEL)	34
cloud-init の使用	34
グラフィカルデスクトップサポート	34
コンパイラトリプレット	35
32 ビット x86 (i686) パッケージ	35
lsb_release および system-lsb-core パッケージ	35
AL2023 の AL2 からのカーネルの変更	36
セキュリティに重点を置いたカーネル設定の変更	36
その他のカーネル設定の変更	40
カーネルファイルシステムのサポート	41
Amazon Linux 2 と AL2023 AMI の比較	47
Amazon Linux 2 と AL2023 最小 AMI の比較	80
Amazon Linux 2 と AL2023 コンテナの比較	100
AL1 と AL2023 の比較	109
各リリースのサポート	109
init システムとして systemd が upstart を置き換えます。	110
Python 2.6 および 2.7 は Python 3 に置き換えられました	110
最も古い JDK としての OpenJDK 8	110
AL2023 の AL1 からのカーネルの変更	110
カーネルライブパッチ	110
カーネルファイルシステムのサポート	110
セキュリティ重視のカーネル設定の変更	112
その他のカーネル設定の変更	114
AL1 と AL2023 AMI の比較	115
AL1 と AL2023 の最小 AMI の比較	149
AL1 と AL2023 のコンテナの比較	169
システム要件	178
AL2023 を実行するための CPU 要件	178
AL2023 の ARM CPU 要件	178
AL2023 の x86-64 CPU 要件	179
AL2023 を実行するためのメモリ (RAM) 要件	180
での AL2023 の使用 AWS	181
の開始方法 AWS	181
にサインアップする AWS アカウント	181
管理アクセスを持つユーザーを作成する	182

プログラムによるアクセス権を付与する	183
Amazon EC2 での AL2023	185
Amazon EC2 コンソールを使用した AL2023 の起動	185
SSM パラメータと を使用した AL2023 の起動 AWS CLI	186
を使用した最新の AL2023 AMI の起動 AWS CloudFormation	188
特定の AMI ID を使用した AL2023 の起動	189
AL2023 AMI の非推奨とライフサイクル	190
AL2023 インスタンスへの接続	190
AL2023 標準 (デフォルト) と最小 AMI の比較	191
コンテナでの AL2023	218
AL2023 ベースコンテナイメージ	219
AL2023 最小コンテナイメージ	221
ベアボーン AL2023 コンテナイメージの構築	223
AL2023 コンテナイメージパッケージリストの比較	227
AL2023 コンテナイメージと最小 AMI の比較	233
Elastic Beanstalk での AL2023	250
AL-2023 CloudShell	251
Amazon ECS コンテナホスト用 AL2023	251
AL2 以降の Amazon ECS 関連の変更	251
Amazon ECS 最適化 AMI をカスタマイズする	252
AL2023 での Amazon EFS	253
amazon-efs-utils	253
Amazon EFS ファイルシステムのマウンティング	253
AL2023 での Amazon EMR	254
AL2023 ベースの Amazon EMR リリース	254
EKS での Amazon EMR	254
AL2023 オン AWS Lambda	254
provided.al2023Lambda ランタイム	255
AL2023 ベースのランタイム	255
チュートリアル	256
AL2023 に LAMP をインストールする	256
ステップ 1: LAMP サーバーを準備する	257
ステップ 2: LAMP サーバーをテストする	262
ステップ 3: データベースサーバーをセキュリティで保護する	264
ステップ 4: (オプション) をインストールする phpMyAdmin	265
トラブルシューティング	268

関連トピック	269
AL2023 で SSL/TLS を設定する	269
前提条件	271
ステップ 1: サーバーで TLS を有効にする	272
ステップ 2: CA 署名証明書を取得する	275
ステップ 3: セキュリティ設定をテストして強化する	282
トラブルシューティング	286
AL2023 で WordPress ブログをホストする	287
前提条件	288
のインストール WordPress	288
次のステップ	299
ヘルプ! パブリック DNS 名が変更されたため、ブログが壊れました	300
Amazon EC2 以外の AL2023	302
AL2023 VM イメージをダウンロードする	302
サポートされる設定	302
KVM の要件	303
VMware の要件	305
Hyper-V の要件	307
AL2023 VM 設定	309
NoCloud seed.iso ベース設定	310
VMwareゲスト情報ベースの設定	314
標準 AMI および KVM イメージの AL2023 パッケージリスト比較	316
標準 AMI と VMware OVA イメージの AL2023 パッケージリスト比較	341
標準 AMI と Hyper-V イメージの AL2023 パッケージリスト比較	366
AL2023 の更新	392
新しいアップデートの通知を受け取る	392
更新の管理	393
使用可能なパッケージ更新の確認	393
DNF およびリポジトリバージョンを使用してセキュリティ更新を適用します。	395
(セキュリティ) 更新後の自動サービス再起動	398
最新のリポジトリバージョンを有効にしたインスタンスの起動	399
パッケージサポート情報の取得	400
新しいリポジトリバージョンの確認	400
新しいリポジトリの追加、有効化、無効化	403
cloud-init によるリポジトリの追加	406
AL2023のバージョン管理されたリポジトリによる確定的アップグレードの使用	407

メジャーリリースとマイナーリリースから受け取る更新を管理する	407
メジャーバージョンとマイナーバージョンのアップグレードの違い	408
AL2023 リポジトリから入手できるパッケージアップデートを管理します。	408
バージョン管理されたリポジトリの使用による確定的なアップグレード	409
カーネルライブパッチ	415
制限事項	416
サポートされている構成と前提条件	416
カーネルライブパッチを使用する	417
プログラミング言語とランタイム	423
C/C++ と Fortran	423
Go	424
AL2023 Lambda 関数:Go	425
Java	425
Perl	425
Perl モジュール	426
PHP	426
新しい PHP バージョンへの移行	426
PHP 7.x からの移行	426
PHP モジュール	427
Python	427
Python モジュール	428
Rust	428
AL2023 Lambda 関数:Rust	429
セキュリティおよびコンプライアンス	430
セキュリティアドバイザリ	431
ALAS Announcements	431
ALAS のよくある質問	431
AL2023 の SELinux モードの設定	432
AL2023 のデフォルト SELinux ステータスとモード	432
enforcing モードへの変更	433
SELinux を無効にするオプション	434
AL2023 で FIPS モードを有効にする	436
カーネルを強化しています	437
カーネル強化オプション (アーキテクチャに依存していません)	437
x86-64 固有のカーネル強化オプション	450
aarch64 固有のカーネル強化オプション	453

AL2023 での UEFI セキュアブート	454
AL2023 で UEFI Secure Boot を有効にする	455
既存のインスタンスの登録	455
スナップショットからイメージを登録する	456
失効の更新	457
AL2023 での UEFI Secure Boot の仕組み	457
独自のキーを登録する	458
.....	cdlix

Amazon Linux 2023 とは

Amazon Linux 2023 (AL2023) は、Amazon Web Services () の次世代 Amazon Linux です。AWS AL2023 を使用すると、安全で安定した、高性能なランタイム環境で、クラウドアプリケーションやエンタープライズアプリケーションを開発して実行できます。また、Linux の最新のイノベーションを利用できる長期サポートを提供するアプリケーション環境も手に入ります。AL2023 は追加料金なしで提供されます。

AL2023 は Amazon リナックス 2 (AL2) の後継機種です。AL2023 と AL2 の違いについては、「AL2023 [における Package の変更点](#)」を参照してください [AL2 と AL2023 の比較](#)。








トピック








- [リリース頻度](#)
- [命名およびバージョンニング](#)
- [パフォーマンスと運用の最適化](#)
- [Fedora との関係](#)
- [カスタマイズされた cloud-init](#)
- [セキュリティ更新および機能](#)
- [ネットワークサービス](#)
- [コアツールチェーンパッケージ glibc、gcc、binutils](#)
- [パッケージ管理ツール](#)
- [デフォルトの SSH サーバー設定](#)

リリース頻度

Amazon Linux の新しいメジャーバージョンは 2 年ごとにリリースし、5 年間のサポートが含まれます。各リリースには 2 つのフェーズのサポートが含まれます。標準サポートフェーズは最初の 2 年を対象としています。次に、メンテナンスフェーズでさらに 3 年間のサポートが継続されます。

標準サポートフェーズでは、リリースは四半期ごとにマイナーバージョン更新されます。メンテナンスフェーズでは、リリースはセキュリティ更新および重要なバグ修正のみであり、使用可能になり次第公開されます。

年	Amazon Linux 2023	Amazon Linux 2025	Amazon Linux 2027	Amazon Linux 2029
2023	 標準サポート			
2024	 標準サポート			
2025	メンテナンス	 標準サポート		
2026	メンテナンス	 標準サポート		
2027	メンテナンス	メンテナンス	 標準サポート	
2028	 サービス終了	メンテナンス	 標準サポート	

年	Amazon Linux 2023	Amazon Linux 2025	Amazon Linux 2027	Amazon Linux 2029
2029	 サービス終了	メンテナンス	メンテナンス	 標準サポート
2030	 サービス終了	 サービス終了	メンテナンス	 標準サポート
2031	 サービス終了	 サービス終了	メンテナンス	メンテナンス

メジャーリリースおよびマイナーリリース

Amazon Linux のリリース (メジャーバージョン、マイナーバージョン、またはセキュリティリリース) ごとに、新しい Linux Amazon マシンイメージ (AMI) がリリースされます。

- メジャーバージョンリリース — スタック全体のセキュリティとパフォーマンスの新機能と改善が含まれています。改善には、カーネル、ツールチェーン Glib C、OpenSSL、およびその他のシステムライブラリとユーティリティの大幅な変更が含まれる場合があります。Amazon Linux のメジャーリリースの一部は、アップストリームの Fedora Linux 配布の最新バージョンに基づいています。AWS は Fedora 以外の他のアップストリームの特定のパッケージを追加したり、置き換えたりする場合があります。
- マイナーバージョンリリース — セキュリティ更新、バグ修正、および新機能およびパッケージが含まれる四半期ごとの更新です。各マイナーバージョンは、新しい機能やパッケージに加えて、セキュリティやバグの修正を含む更新の累積リストです。これらのリリースには、PHP などの最新の言語ランタイムが含まれる場合があります。Ansible や Docker など、他の一般的なソフトウェアパッケージも含まれる場合があります。

新しいリリースの消費

更新は、新しい Amazon マシンイメージ (AMI) リリースと対応する新しいリポジトリを組み合わせで提供されます。デフォルトでは、新しい AMI とそれが指すリポジトリが結合されます。ただし、実行中の Amazon EC2 インスタンスに更新を適用するために、時間の経過とともに新しいリポジトリバージョンを指定できます。最新の AMI の新しいインスタンスを起動して更新することもできます。

長期的なサポートポリシー

Amazon Linux では、すべてのパッケージの更新が提供され、Amazon Linux 上に構築されたアプリケーションのメジャーバージョン内での互換性が維持されます。glibc ライブラリ、OpenSSL、OpenSSH、および DNF パッケージマネージャなどのコアパッケージは、AL2023 のメジャーリリースの有効期間中サポートを受けられます。コアパッケージに含まれていないパッケージは、特定のアップストリームソースに基づいてサポートされます。以下のコマンドを実行すると、個々のパッケージの具体的なサポート状況および日付を確認できます。

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

以下のコマンドを実行すると、現在インストールされているすべてのパッケージに関する情報を取得できます。

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

コアパッケージのすべてのリストはプレビュー中に完成します。コアパッケージとして含まれるパッケージをさらに確認する場合はお問い合わせください。私たちは、フィードバックを収集して評価しています。AL2023 に関するフィードバックは、指定の AWS 担当者を通じて提供するか、GitHub の [amazon-linux-2023 リポジトリ](#) に問題を提出することで提供できます。

命名およびバージョンニング

AL2023 は、2 年間の標準サポート期間中、3 か月ごとにマイナーリリースを提供します。各リリースは 0 から N までの数字で識別されます。0 はそのイテレーションの元のメジャーリリースを指します。すべてのリリースは Amazon Linux 2023 という名前になります。Amazon Linux 2025 がリリースされると、AL2023 は延長サポートに入り、セキュリティ更新と重大なバグ修正の更新を受け取ります。

例えば、AL2023 のマイナーリリースのフォーマットは以下のとおりです。

- 2023.0.20230301
- 2023.1.20230601
- 2023.2.20230901

対応する AL2023 AMI のフォーマットは以下のとおりです。

- 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

特定のマイナーバージョン内では、AMI リリース日のタイムスタンプが付いた通常の AMI リリースが行われます。

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

AL2 または AL2023 インスタンスを識別するための推奨方法は、から共通プラットフォーム列挙 (CPE) 文字列を読み取ることから始めます。/etc/system-release-cpe次に、文字列をフィールドに分割します。最後に、プラットフォームとバージョンの値を読み取ります。

AL2023 では、プラットフォーム識別用の新しいファイルも導入されています。

- /etc/system-release への /etc/amazon-linux-release シンボリックリンク
- /etc/amazon-linux-release-cpe への /etc/system-release-cpe シンボリックリンク

この 2 つのファイルは、インスタンスが Amazon Linux であることを示しています。特定のプラットフォームとバージョンの値を知りたい場合を除いて、ファイルを読み取ったり、文字列をフィールドに分割したりする必要はありません。

パフォーマンスと運用の最適化

Amazon Linux 6.1 カーネル

- AL2023は、エラスティックネットワークアダプター (ENA) およびElastic Fabric Adapter (EFA) デバイス用の最新のドライバーを使用しています。AL2023は、Amazon EC2インフラストラクチャのハードウェアのパフォーマンスと機能のバックポートに重点を置いています。
- x86_64 および aarch64 インスタンスタイプにはカーネルライブパッチが適用されています。これにより、頻繁に再起動する必要が少なくなります。
- すべてのカーネルビルドとランタイム設定には、AL2 と同じパフォーマンスと運用の最適化が数多く含まれています。

基本ツールチェーンの選択とデフォルトのビルドフラグ

- AL2023 パッケージは、コンパイラ最適化 () がデフォルトで有効化された状態でビルドされます。-O2
- AL2023 パッケージは x86-64 システム (-march=x86-64-v2) の x86-64v2 が必要で、aarch64 (-march=armv8.2-a+crypto -mtune=neoverse-n1) の Graviton 2 以降が必要です。
- AL2023 パッケージは、自動ベクトル化を有効にして構築されています (-ftree-vectorize)。
- AL2023 パッケージはリンクタイム最適化 (LTO) を有効にしてビルドされています。
- AL2023 は Rust、Clang/LLVM、および Go の更新バージョンを使用しています。

パッケージの選択とバージョン

- 主要なシステムコンポーネントへの一部のバックポートには、Amazon EC2 インフラストラクチャ、特に Graviton インスタンスで実行するためのいくつかのパフォーマンスの向上が含まれています。
- AL2023 には複数の機能が統合されています。AWS のサービス これには AWS CLI、SSM エージェント、Amazon Kinesis エージェント、およびが含まれます。CloudFormation
- AL2023 では、Java 開発キット (JDK) として Amazon Corretto が使用されます。
- AL2023 は、上流プロジェクトによってリリースされる新しいバージョンへのデータベースエンジンとプログラミング言語のランタイム更新を提供します。新しいバージョンのプログラミング言語ランタイムは、リリース時に追加されます。

クラウド環境へのデプロイ

- ベースの AL2023 AMI とコンテナイメージは、パッチ適用インスタンスの置換をサポートするように頻繁に更新されます。
- カーネルの更新は AL2023 AMI の更新に含まれています。つまり、カーネルを更新するために `yum update` や `reboot` などのコマンドを使用する必要がないということです。
- 標準の AL2023 AMI に加えて、最小 AMI とコンテナイメージも利用できます。サービスの実行に必要な最小限のパッケージ数で環境を実行するには、最小 AMI を選択します。
- デフォルトでは、AL2023 AMI とコンテナは特定のバージョンのパッケージリポジトリにロックされます。起動時の自動更新はありません。つまり、パッケージ更新をいつ取り込むかを常に制御できるということです。本番環境にロールアウトする前に、いつでもベータ/ガンマ環境でテストできます。問題が発生した場合は、事前に検証されたロールバックパスを使用できます。

Fedora との関係

AL2023は、Fedoraとは独立した独自のリリースおよびサポートライフサイクルを維持しています。AL2023 では、オープンソースソフトウェアの更新バージョン、多種多様なパッケージ、頻繁なリリースが提供されています。これにより、使い慣れた RPM ベースのオペレーティングシステムが維持されます。

AL2023 の一般提供 (GA) バージョンは、特定の Fedora リリースと直接比較できるものではありません。AL2023 GA バージョンには Fedora 34、35、36 のコンポーネントが含まれています。一部のコンポーネントは Fedora のコンポーネントと同じで、一部は変更されています。他のコンポーネントは、CentOS 9 Streams のコンポーネントによく似ているか、独自に開発されたものです。Amazon Linux カーネルは、Fedora とは別に選択された kernel.org にある長期サポートオプションから供給されています。

カスタマイズされた cloud-init

cloud-init パッケージは、クラウドコンピューティング環境で Linux イメージをブートストラップするオープンソースアプリケーションです。詳細については、[「cloud-init ドキュメント」](#)を参照してください。

AL2023 には、cloud-init のカスタマイズバージョンが含まれています。cloud-init を使用すると、起動時のインスタンスの動作を指定できます。

インスタンスを起動するときに、ユーザーデータフィールドを使用してにアクションを渡すことができますcloud-init。つまり、さまざまなユースケースに対して共通の Amazon マシンイメージ (AMI) を使用し、インスタンスの起動時にその AMI を動的に設定できます。AL2023 は ec2-user アカウントの設定にも cloud-init を使用します。

AL2023 は /etc/cloud/cloud.cfg.d と /etc/cloud/cloud.cfg にある cloud-init アクションを使用します。独自の cloud-init アクションファイルを /etc/cloud/cloud.cfg.d ディレクトリに作成できます。Cloud-init はこのディレクトリ内のすべてのファイルを辞書順に読み込みます。後のファイルは以前のファイルの値を上書きします。cloud-init がインスタンスを起動すると、cloud-init パッケージは以下の設定タスクを実行します。

- デフォルトのロケールの設定
- ホスト名の設定
- ユーザーデータの解析および処理
- ホストプライベート SSH キーの生成
- 容易にログインおよび管理できるように、ユーザーのパブリック SSH キーを .ssh/authorized_keys に追加
- パッケージ管理のためのリポジトリの準備
- user-data で定義されたパッケージアクションの処理
- user-data に含まれるユーザースクリプトの実行
- インスタンスストアボリュームのマウント (該当する場合)
 - デフォルトでは、ephemeral0 インスタンスストアボリュームが存在し有効なファイルシステムを含む場合、インスタンスストアボリュームは /media/ephemeral0 にマウントされます。それ以外の場合は、マウントされません。
 - デフォルトでは、m1.small および c1.medium インスタンスタイプの場合、インスタンスに関連付けられたすべてのスワップボリュームがマウントされます。
 - 以下の cloud-init ディレクティブを使用して、デフォルトのインスタンスストアボリュームマウントを上書きできます。

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

マウントをより制御するには、「cloud-init ドキュメント」の「[マウント](#)」を参照してください。

- インスタンスが起動しても、TRIM をサポートするインスタンスストアボリュームはフォーマット化されません。インスタンスストアボリュームをマウントする前に、インスタンスストアボリュームを分割してフォーマット化する必要があります。

詳細については、「Amazon EC2 [ユーザーガイド](#)」の「[インスタンスストアボリュームの TRIM サポート](#)」を参照してください。 Amazon EC2

- インスタンスを起動する際、disk_setup モジュールを使用してインスタンスストアボリュームを分割およびフォーマット化することができます。

詳細については、「cloud-init ドキュメント」の「[ディスクの設定](#)」を参照してください。

SELinux で cloud-init を使用方法については、「[cloud-init を使用して enforcing モードを有効にする](#)」を参照してください。

cloud-init ユーザーデータフォーマットについては、「cloud-init ドキュメント」の「[User-Data フォーマット](#)」を参照してください。

セキュリティ更新および機能

AL2023は、多くのセキュリティアップデートとソリューションを提供します。

トピック

- [更新の管理](#)
- [クラウド内のセキュリティ](#)
- [SELinux モード](#)
- [コンプライアンスプログラム](#)
- [SSH サーバーのデフォルト](#)
- [OpenSSL 3 の主な機能](#)

更新の管理

DNFおよびリポジトリバージョンを使用してセキュリティアップデートを適用します。詳細については、「[AL2023 でパッケージとオペレーティングシステムの更新を管理する](#)」を参照してください。

クラウド内のセキュリティ

AWS セキュリティはユーザーとユーザー間で共有される責任です。[責任分担モデル](#)では、これをクラウドのセキュリティとクラウドのセキュリティと表現しています。詳細については、「[Amazon Linux 2 のセキュリティとコンプライアンス](#)」を参照してください。

SELinux モード

AL2023 では SELinux はデフォルトで有効になっており、許可モードに設定されています。許可モードでは、アクセス拒否は記録されますが、強制ではありません。

SELinux ポリシーは、ユーザー、プロセス、プログラム、ファイル、デバイスのアクセス許可を定義します。SELinux では、2 つのポリシーから 1 つを選択できます。ポリシーはターゲットを絞ったセキュリティ、またはマルチレベルセキュリティ (MLS) です。

SELinux のモードとポリシーの詳細については、「[AL2023 の SELinux モードの設定 および SELinux プロジェクト Wiki](#)」を参照してください。

コンプライアンスプログラム

独立監査人は、AWS AL2023のセキュリティとコンプライアンスを多くのコンプライアンスプログラムとともに評価します。

SSH サーバーのデフォルト

AL2023 には OpenSSH 8.7 が含まれています。デフォルトでは、OpenSSH 8.7 は ssh-rsa キー交換アルゴリズムを無効にします。詳細については、「[デフォルトの SSH サーバー設定](#)」を参照してください。

OpenSSL 3 の主な機能

- 証明書管理プロトコル (CMP、RFC 4210) には CRMF (RFC 4211) と HTTP 転送 (RFC 6712) の両方が含まれています。
- libcrypto の HTTP または HTTPS クライアントは、GET および POST アクション、リダイレクト、プレーンおよび ASN.1 にエンコードされたコンテンツ、プロキシ、タイムアウトをサポートします。
- EVP_KDF はキー派生関数と連携します。
- EVP_MAC API は MACs と連携します。
- Linux カーネル TLS サポート。

詳細については、「[OpenSSL の移行ガイド](#)」を参照してください。

ネットワークサービス

オープンソースプロジェクト `systemd-networkd` は、最新の Linux 配布で広く使用できます。このプロジェクトでは、`systemd` フレームワークの他の部分と同様の宣言型設定言語を使用しています。主な設定ファイルタイプは `.network` および `.link` ファイルです。

`amazon-ec2-net-utils` パッケージはインターフェイス固有の設定を `/run/systemd/network` ディレクトリに生成します。これらの設定では、インターフェイスがインスタンスにアタッチされている場合、IPv4 と IPv6 の両方のネットワークが有効になります。これらの構成では、ローカルソースのトラフィックが対応するインスタンスのネットワークインターフェースを介してネットワークにルーティングされるようにするポリシールーティングルールもインストールされます。これらのルールにより、適切なトラフィックが、関連付けられたアドレスまたはプレフィックスから Elastic Network Interface (ENI) を介してルーティングされます。ENI の使用の詳細については、「Amazon EC2 [ユーザーガイド](#)」の「[ENI の使用](#)」を参照してください。Amazon EC2

このネットワーク動作は、`/etc/systemd/network` ディレクトリにカスタム設定ファイルを配置して、`/run/systemd/network` に含まれるデフォルトの設定を上書きすることでカスタマイズできます。

[systemd.network](#) のドキュメントには、`systemd-networkd` サービスが特定のインターフェースに適用される設定を決定する方法が説明されています。また、ENI-backed インターフェイスにさまざまな AWS リソースのプロパティを反映するために `altnames`、と呼ばれる代替名も生成します。ENI ベースのインターフェースプロパティは ENI アタッチメントの ENI ID および `DeviceIndex` フィールドです。これらのインターフェースは、`ip` コマンドなどのさまざまなツールの使用時に、そのプロパティを使用して参照できます。

AL2023 インスタンスインターフェイス名は、`systemd` スロット命名スキームを使用して生成されます。詳細については、「[systemd.net 命名スキーム](#)」を参照してください。

さらに、AL2023 は `fq_code1` アクティブキュー管理ネットワークの送信スケジューリングアルゴリズムをデフォルトで使用します。詳細については、[CoDel](#) 「[概要](#)」を参照してください。

コアツールチェーンパッケージ `glibc`、`gcc`、`binutils`

Amazon Linux のパッケージのサブセットはコアツールチェーンパッケージとして指定されています。AL2023 の主要な部分として、コアパッケージは5年間のサポートを受けます。パッケージの

バージョンは変更される場合がありますが、Amazon Linux リリースに含まれるパッケージには長期サポートが適用されます。

これら 3 つのコアパッケージは、Amazon Linux 配布のほとんどのソフトウェアの構築に使用されるシステムツールチェーンを提供します。

パッケージ	定義	目的
glibc 2.34	システム C ライブラリ	標準機能を提供するほとんどのバイナリプログラムや、プログラムとカーネル間のインターフェースで使用されません。
gcc 11.2	gcc コンパイラスイート	コンパイル C、C++、Fortran。
binutils 2.35	アセンブラとリンカー、その他のバイナリツール	バイナリプログラムを操作または検査します。

glibc ライブラリを更新した後も再起動が必要です。サービスを制御する更新の場合は、サービスの再起動のみで更新を取得できる場合があります。しかし、システムを再起動すると、パッケージとライブラリの以前の更新はすべて完了します。

パッケージ管理ツール

AL2023 のデフォルトのソフトウェアパッケージ管理ツールは DNF DNFAL2 のパッケージ管理ツールの後継です。YUM

DNF の使い方は YUM と似ています。DNF 多くのコマンドとコマンドオプションはコマンドと同じです YUM。コマンドラインインターフェイス (CLI) のコマンドでは、ほとんどの場合、dnf が yum に置き換えられます。

たとえば、次の AL2 yum コマンドの場合:

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

AL2023 では、これらは以下のコマンドになります。

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

AL2023 では、yum コマンドは引き続き使用できますが、dnf コマンドへのポインタとして利用できます。そのため、yum コマンドをシェルまたはスクリプトで使用すると、すべてのコマンドとオプションは DNF CLI と同じになります。YUM CLI と DNF CLI の相違点の詳細については、「[YUM との比較における DNF CLI の変更点](#)」を参照してください。

コマンドと dnf コマンドのオプションの詳細については、マニュアルページ「man dnf」を参照してください。詳細については、「[DNFコマンドリファレンス](#)」を参照してください。

デフォルトの SSH サーバー設定

数年前の SSH クライアントを使用している場合、インスタンスに接続するとエラーが表示される可能性があります。一致するホストキータイプが見つからないというエラーが表示された場合は、SSH ホストキーを更新してこの問題を解決します。

デフォルトの **ssh-rsa** 署名の無効化

AL2023には、ssh-rsa従来のホストキーアルゴリズムを無効にし、ホストキーの削減セットを生成するデフォルト設定が含まれています。クライアントは ssh-ed25519 または ecdsa-sha2-nistp256 ホストキーアルゴリズムをサポートする必要があります。

デフォルト設定では以下のキー交換アルゴリズムを受け入れます。

- 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

デフォルトでは、AL2023 は ed25519 および ECDSA はホストキーを生成します。クライアントは ssh-ed25519 または ecdsa-sha2-nistp256 ホストキーアルゴリズムのいずれかをサポートします。SSH でインスタンスに接続する場合、ssh-ed25519 または ecdsa-sha2-nistp256 などの互換性のあるアルゴリズムをサポートするクライアントを使用する必要があります。他の種類のキーを使用する必要がある場合は、生成されたキーのリストを user-data 内の cloud-config フラグメントで上書きします。

以下の例では、cloud-config は ecdsa および ed25519 キーを使用して rsa ホストキーを生成します。

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

公開キー認証に RSA キーペアを使用する場合、SSH クライアントは rsa-sha2-256 または rsa-sha2-512 署名をサポートしている必要があります。互換性のないクライアントを使用していてアップグレードできない場合は、インスタンスの ssh-rsa サポートを再度有効にします。ssh-rsa サポートを再度有効にするには、LEGACY 以下のコマンドを使用してシステム暗号化ポリシーを有効にします。

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

ホストキーの管理の詳細については、「[Amazon Linux ホストキー](#)」を参照してください。

AL2023 で廃止された機能

AL2 では廃止され、AL2023 には存在しない機能については、こちらに記載されています。これは AL2 にはあるが AL2023 にはない機能やパッケージなどの機能で、AL2023 には追加されない予定です。[AL2 での機能のサポート期間について詳しくは、「AL2 で廃止予定の機能」を参照してください。](#)

AL2023 には廃止された機能もあり、future のリリースで削除される予定です。この章では、その機能の内容、サポートされなくなった時期、および Amazon Linux から削除される時期について説明します。廃止予定の機能を理解しておく、AL2023 のデプロイに役立つだけでなく、Amazon Linux の次のメジャーバージョンへの準備にも役立ちます。

トピック

- [compat- パッケージ](#)
- [非推奨の機能は AL1 で廃止され、AL2 では削除されました](#)
- [AL2 では機能が非推奨になり、AL2023 では機能が削除されました](#)
- [AL2023 では非推奨](#)

compat- パッケージ

AL2 compat- のプレフィックスがのパッケージはすべて、最新バージョンのパッケージ用にまだ再ビルドされていない古いバイナリとのバイナリ互換性を保つために提供されています。Amazon Linux の新しいメジャーバージョンはそれぞれ、compat-以前のリリースのパッケージを引き継ぐことはありません。

Amazon Linux のリリース (例:AL2) compat- のすべてのパッケージは非推奨であり、それ以降のバージョン (例:AL2023) には存在しません。ライブラリを更新したバージョンに合わせてソフトウェアを再構築することを強くお勧めします。

非推奨の機能は AL1 で廃止され、AL2 では削除されました

このセクションでは、AL1 で使用でき、AL2 では使用できなくなった機能について説明します。

Note

AL1 のメンテナンスサポートフェーズの一環として、end-of-life (EOL) 日が AL1 の EOL よりも早いパッケージがありました。詳細については、「[AL1 Package サポートステートメント](#)」を参照してください。

Note

AL1 の一部の機能は以前のリリースで廃止されました。詳細については、[AL1](#) リリースノート参照してください。

トピック

- [32 ビット x86 \(i686\) AMI](#)
- [aws-apitools-*に置き換えられました。AWS CLI](#)
- [systemdAL2 upstart では置き換えられます。](#)

32 ビット x86 (i686) AMI

[AL1 の 2014.09 リリースの一環として](#)、Amazon Linux は 32 ビット AMI を提供する最後のリリースになると発表しました。そのため、[AL1 の 2015.03 リリース以降](#)、Amazon Linux は 32 ビットモードでのシステムの実行をサポートしなくなりました。AL2 では x86-64 ホスト上の 32 ビットバイナリのランタイムサポートが限定されており、新しい 32 ビットバイナリの構築を可能にする開発パッケージは提供されていません。AL2023 には 32 ビットユーザースペースパッケージは含まれなくなりました。AL2023 に移行する前に 64 ビットコードへの移行を完了することをお勧めします。

AL2023 で 32 ビットバイナリを実行する必要がある場合は、AL2023 上で実行されている AL2 コンテナ内の AL2 の 32 ビットユーザースペースを使用できます。

aws-apitools-*に置き換えられました。AWS CLI

2013 AWS CLI 年 9 月にリリースされる前に、AWS には、ユーザーが Amazon EC2 API 呼び出しを行えるようにするコマンドラインユーティリティのセットが実装されていました。Javaこれらのツールは2015年に廃止され、コマンドラインから Amazon EC2 API AWS CLI を操作する方法として好まれるようになりました。aws-apitools-*コマンドラインユーティリティセットには以下のパッケージが含まれています。

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

`aws-apitools-*`パッケージのアップストリームサポートは 2017 年 3 月に終了しました。アップストリームサポートがないにもかかわらず、Amazon Linux は、ユーザーに下位互換性を提供するなどして `aws-apitools-ec2`、これらのコマンドラインユーティリティの一部を引き続き出荷していました。AWS CLI は活発にメンテナンスされており、すべての AWS API を使用する手段を提供しているため、`aws-apitools-*`パッケージよりも堅牢で完全なツールです。

`aws-apitools-*`これらのパッケージは 2017 年 3 月に廃止され、今後更新される予定はありません。これらのパッケージのいずれかを利用しているすべてのユーザーは、AWS CLI できるだけ早くに移行する必要があります。これらのパッケージは AL2023 には存在しません。

AL1 `aws-apitools-iam` `aws-apitools-rds` ではおよびパッケージも提供していましたが、これらは AL1 では廃止され、AL2 以降の Amazon Linux には存在しません。

systemdAL2 upstart では置き換えられます。

AL2 は、`systemd`初期化システムを使用した最初の Amazon Linux リリースで、AL1 `upstart` に置き換えられました。AL1 から新しいバージョンの Amazon Linux への移行の一環として、`upstart` 特定の設定を変更する必要があります。AL1 `systemd` では使用できないため、`upstart` からへの移行は、AL2 や AL2023 などのより新しいメジャーバージョンの Amazon Linux `systemd` への移行の一環としてのみ実行できます。

AL2 では機能が非推奨になり、AL2023 では機能が削除されました

このセクションでは、AL2 で使用可能で、AL2023 では使用できなくなった機能について説明します。

トピック

- [32 ビット x86 \(i686\) パッケージ](#)
- [aws-apitools-* を置き換えました AWS CLI](#)

- [bzd リビジョン管理システム](#)
- [cgroup v1](#)
- [log4j ホットパッチ \(log4j-cve-2021-44228-hotpatch \)](#)
- [lsb_release および system-lsb-core パッケージ](#)
- [mccrypt](#)
- [OpenJDK 7 \(java-1.7.0-openjdk \)](#)
- [Python 2.7](#)
- [rsyslog-openssl が置き換える rsyslog-gnutls](#)
- [Network Information Service \(NIS\)/ yp](#)

32 ビット x86 (i686) パッケージ

[AL1 の 2014 年 9 月リリース](#)の一環として、32 ビット AMIs。したがって、[AL1 の 2015 年 3 月リリース](#)以降、Amazon Linux は 32 ビットモードでのシステムの実行をサポートしなくなりました。AL2 は x86-64 ホストでの 32 ビットバイナリのランタイムサポートが制限されており、新しい 32 ビットバイナリの構築を可能にする開発パッケージは提供していません。AL2023 には 32 ビットユーザースペースパッケージが含まれなくなりました。64 ビットコードへの移行を完了することをお勧めします。

AL2023 で 32 ビットバイナリを実行する必要がある場合は、AL2023 上で実行されている AL2 コンテナ内で AL2023 AL2 ビットユーザースペースを使用できます。AL2

aws-apitools-* を置き換えました AWS CLI

2013 年 9 AWS CLI 月の のリリース以前は、 で実装された一連のコマンドラインユーティリティが利用可能 AWS になりました。これにより Java、お客様は Amazon EC2 API コールを実行できます。これらのツールは 2015 年に廃止され、コマンドラインから Amazon EC2 APIs を操作するための推奨方法 AWS CLI になりました。これには、次の aws-apitools-* パッケージが含まれます。

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

aws-apitools-* パッケージのアップストリームサポートは 2017 年 3 月に終了しました。アップストリームサポートがないにもかかわらず、Amazon Linux はこれらのコマンドラインユーティリティ (などaws-apitools-ec2) の一部を引き続き出荷し、お客様に下位互換性を提供していました。AWS CLI は、パッケージよりも堅牢で完全なツールです。aws-apitools-*パッケージはアクティブにメンテナンスされており、すべての AWS APIsを使用する手段を提供します。

aws-apitools-* パッケージは 2017 年 3 月に廃止され、それ以降の更新は受け取られません。これらのパッケージのすべてのユーザーは AWS CLI、できるだけ早くに移行する必要があります。これらのパッケージは AL2023 には存在しません。

bzr リビジョン管理システム

[GNU Bazaar](#) (bzr) リビジョン管理システムは AL2 では廃止され、AL2023 では使用できなくなります。

のユーザーはbzr、リポジトリをに移行することをお勧めしますgit。

cgroup v1

AL2023 は統合コントロールグループ階層 (cgroup v2) に移行し、AL2 は cgroup v1 を使用します。AL2 は cgroup v2 をサポートしていないため、この移行は AL2023 への移行の一環として完了する必要があります。

log4j ホットパッチ (**log4j-cve-2021-44228-hotpatch**)

Note

log4j-cve-2021-44228-hotpatch パッケージは AL2 では廃止され、AL2023 では削除されます。

[CVE-2021-44228 用の Apache Log4j 用 Hotpatch](#) の RPM パッケージバージョンをリリースしました。AL1 AL2 [Amazon Linux へのホットパッチの追加に関する発表](#)では、「ホットパッチのインストールは、CVE-2021-44228 または CVE-2021-45046 を軽減する log4j バージョンへの更新に代わるものではありません」と記載されています。

ホットパッチは、log4j へのパッチ適用までの時間を確保するための緩和策でした。AL2023 の最初の一般提供リリースは [CVE-2021-44228](#)5 か月後であるため、AL2023 にはホットパッチ (有効かどうかにかかわらず) が付属していません。

Amazon Linux で独自の log4j バージョンを実行しているお客様は、[CVE-2021-44228](#) または [CVE-2021-45046](#) の影響を受けないバージョンに更新していることを確認することをお勧めします。

lsb_release および system-lsb-core パッケージ

これまで、一部のソフトウェアは (system-lsb-core パッケージによって AL2 で提供されている) lsb_release コマンドを呼び出して、実行されている Linux 配布に関する情報を取得していました。Linux 標準ベース (LSB) ではこのコマンドが導入され、Linux 配布でも採用されました。Linux 配布は、この情報を /etc/os-release およびその他の関連ファイルに保持するという、より単純な標準を使用するように進化しました。

os-release 標準は systemd から生まれました。詳細については、「[systemd os-release ドキュメント](#)」を参照してください。

AL2023 には lsb_release コマンドおよび system-lsb-core パッケージは含まれません。Amazon Linux やその他の主要な Linux 配布との互換性を維持するために、ソフトウェアは os-release 標準への移行を完了する必要があります。

mcrypt

mcrypt ライブラリおよび関連する PHP 拡張機能は AL2 では非推奨となり、AL2023 には存在しなくなりました。

Upstream PHP は [7.1 PHP で mcrypt 拡張機能を非推奨にしました](#)。これは、2016 年 12 月に初めてリリースされ、2019 年 10 月に最終リリースされました。

アップストリーム mcrypt ライブラリは [2007 年に最後にリリースされましたが、SourceForge 2017 年に新しいコミットに必要な cvs リビジョンコントロールからの移行は行われていません](#)。最新のコミット (3 年前のみ) は 2011 年からで、プロジェクトに保守者がいるという言及がなくなりました。

の残りのユーザーは mcrypt、AL2023 に追加 mcrypt されないため OpenSSL、コードを に移植することをお勧めします。

OpenJDK 7 (java-1.7.0-openjdk)

Note

AL2023 には、Java ベースのワークロードをサポートするために、[Amazon Corretto](#) のいくつかのバージョンが用意されています。OpenJDK 7 パッケージは AL2 では非推奨とな

り、AL2023 では使用できなくなりました。AL2023 で利用可能な最も古い JDK は Corretto 8 によって提供されます。

Amazon Linux での Java の詳細については、「」を参照してください [AL2023 での Java](#)。

Python 2.7

Note

AL2023 は Python 2.7 を削除したため、Python を必要とする OS コンポーネントはすべて Python 3 で動作するように記述されています。Amazon Linux によって提供されサポートされているバージョンの Python を引き続き使用するには、Python 2 コードを Python 3 に変換します。

Amazon Linux での Python の詳細については、「」を参照してください [AL2023 での Python](#)。

rsyslog-openssl が置き換える rsyslog-gnutls

rsyslog-gnutls パッケージは AL2 では非推奨となり、AL2023 には存在しなくなります。rsyslog-openssl パッケージは、rsyslog-gnutls パッケージの使用に対するドロップインの置き換えである必要があります。

Network Information Service (NIS)/ yp

Network Information Service (NIS) は、もともと Yellow Pages またはと呼ばれていました YP が、AL2 では非推奨となり、AL2023 では使用できなくなりました。これには、ypbind、ypserv のパッケージが含まれます yp-tools。と統合されている他のパッケージ NIS では、この機能は AL2023 で削除されます。

AL2023 では非推奨

このセクションでは、AL2023 に存在し、将来のバージョンの Amazon Linux で削除される可能性が高い機能について説明します。各セクションでは、機能とは何か、またいつ Amazon Linux から削除されるかについて説明します。

Note

このセクションは、Linux エコシステムが進化し、将来の Amazon Linux のメジャーバージョンがリリースに近づくにつれて、時間の経過とともに更新されます。

トピック

- [32 ビット x86 \(i686\) ランタイムのサポート](#)
- [バークレー DB \(libdb \)](#)
- [cron](#)
- [IMDSv1](#)
- [pcre バージョン 1](#)
- [System V init \(sysvinit\)](#)

32 ビット x86 (i686) ランタイムのサポート

AL2023 は、32 ビット x86 (i686) バイナリを実行する機能を保持します。Amazon Linux の次のメジャーバージョンでは、32 ビットユーザースペースバイナリの実行がサポートされなくなる可能性があります。

バークレー DB (**libdb**)

AL2023 には、バークレー DB (libdb) ライブラリのバージョン 5.3.28 が付属しています。これは、ライセンスが制限の緩い sleepycat ライセンスから GNU Affero GPLv3 (AGPL) ライセンスに変更される前の Berkeley DB の最後のバージョンです。

AL2023 には、バークレー DB (libdb) に依存するパッケージはほとんどなく、ライブラリは Amazon Linux の次のメジャーリリースで削除されます。

Note

AL2023 の dnf パッケージマネージャーは、バークレー DB (BDB) 形式の rpm データベースの読み取り専用サポートを保持します。このサポートは、Amazon Linux の次のメジャーリリースで削除されます。

cron

cronie パッケージは AL2 AMI にデフォルトでインストールされ、定期的なタスクをスケジューリングする従来の crontab 方法をサポートしています。AL2023 では、cronie はデフォルトでは含まれていません。したがって、のサポート crontab はデフォルトでは提供されなくなりました。

AL2023 では、オプションで cronie パッケージをインストールしてクラシック cron ジョブを使用できます。systemd には機能が追加されているため、systemd タイマーに移行することをお勧めします。

Amazon Linux の将来のバージョン、場合によっては次のメジャーバージョンには、クラシック cron ジョブのサポートが含まれなくなり、systemd タイマーへの移行が完了する可能性があります。の使用から移行することをお勧めします cron。

IMDSv1

デフォルトでは、AL2023 AMIs は IMDSv2 専用モードで起動するように設定され、の使用を無効にします IMDSv1。IMDSv1 を有効にした状態で AL2023 を使用するオプションはまだあります。IMDSv1 将来のバージョンの Amazon Linux では、IMDSv2 のみが適用される可能性があります。

AMIs Amazon EC2 ユーザーガイド」の「[AMI の設定](#)」を参照してください。

pcre バージョン 1

レガシー pcre パッケージは廃止され、Amazon Linux の次のメジャーリリースで削除されます。pcre2 パッケージは後継パッケージです。AL2023 の最初のバージョンには、に対して構築されるパッケージの数は限られていますが pcre、これらのパッケージは AL2023 pcre2 内での移行されます。廃止された pcre ライブラリは AL2023 で引き続き利用できます。

Note

の非推奨バージョン pcre は、AL2023 の全有効期間にわたってセキュリティ更新プログラムを受信しません。pcre サポート ライフサイクルとパッケージがセキュリティ更新を受信する時間の詳細については、パッケージ [pcre のパッケージサポートステートメント](#) を参照してください。

System V init (**sysvinit**)

AL2023 は System V サービス (init) スクリプトとの下位互換性を保持しますが、アップストリーム systemd プロジェクトは [v254 リリースの一部として](#)、[System V サービススクリプトのサポートの廃止](#) を発表しました。また、の将来のバージョンでサポートが削除されることが示されました systemd。詳細については、「[systemd](#)」を参照してください。

AL2023 は System V サービス (init) スクリプトとの下位互換性を保持しますが、System V サービス (init) スクリプトのサポートが次のメジャーリリースで Amazon Linux から削除される際に準備できるように、ネイティブ systemd ユニットファイルの使用に移行することをお勧めします。

AL2 と AL2023 の比較

以下のトピックでは、AL2 と AL2023 の主な違いについて説明します。

トピック

- [追加、更新、削除されたパッケージ](#)
- [各リリースのサポート](#)
- [命名およびバージョンングの変更](#)
- [最適化](#)
- [Python 2.7 は Python 3 に置き換えられました](#)
- [セキュリティ更新](#)
- [安定性向上のための確定的な更新](#)
- [複数のアップストリームから供給されています。](#)
- [AMI ルートファイルシステムとデフォルトの Amazon EBS ボリュームタイプ](#)
- [ネットワークシステムサービス](#)
- [統合コントロールグループ階層 \(cgroup v2\)](#)
- [タスクのスケジューリング](#)
- [glibc、gcc、および binutils のパッケージ](#)
- [パッケージマネージャー](#)
- [ロギングシステム](#)
- [curl および libcurl のパッケージ変更](#)
- [GNU プライバシーガード \(GNUPG\)](#)
- [デフォルト JVM としての Amazon Corretto](#)
- [AWS CLI v2](#)
- [UEFI Preferred](#)
- [SSH サーバーのデフォルト設定の変更](#)
- [Extra Packages for Enterprise Linux \(EPEL\)](#)
- [cloud-init の使用](#)
- [グラフィカルデスクトップサポート](#)
- [コンパイラトリプレット](#)

- [32 ビット x86 \(i686\) パッケージ](#)
- [lsb_release および system-lsb-core パッケージ](#)
- [AL2からのAL2023カーネルの変更](#)
- [Amazon Linux 2 AMI と Amazon Linux 2023 AMI にインストールされているパッケージの比較](#)
- [Amazon Linux 2 と Amazon Linux 2023 最小 AMIにインストールされているパッケージの比較](#)
- [Amazon Linux 2 と Amazon Linux 2023 ベースコンテナイメージにインストールされているパッケージの比較](#)

追加、更新、削除されたパッケージ

AL2023 には、何千もの使用可能なソフトウェアパッケージが含まれています。AL2023 で追加、更新、または削除されたすべてのパッケージを以前の Amazon Linux バージョンと比較したリストについては、「[AL2023 でのパッケージ変更](#)」を参照してください。

AL2023 でパッケージの追加または変更をリクエストするには、の [amazon-linux-2023 リポジトリ](#) に問題を提出します GitHub。

各リリースのサポート

AL2023 については、5 年間のサポートを提供しています。

詳細については、「[リリース頻度](#)」を参照してください。

命名およびバージョニングの変更

AL2023 は、AL2 がプラットフォーム識別用にサポートしているのと同じメカニズムをサポートしています。AL2023 では、プラットフォーム識別用の新しいファイルも導入されています。

詳細については、「[命名およびバージョニング](#)」を参照してください。

最適化

AL2023 は起動時間を最適化して、インスタンス起動からお客様のワークロードの実行までの時間を短縮します。これらの最適化は、Amazon EC2 インスタンスのカーネル設定、cloud-init 設定、および kmod または systemd などの OS のパッケージに組み込まれた機能に及びます。

最適化の詳細については、「[パフォーマンスと運用の最適化](#)」を参照してください。

Python 2.7 は Python 3 に置き換えられました

AL2 は、AL2 コアパッケージの長期サポート (LTS) の一環として、2025 年 6 月まで Python 2.7 のサポートパッチとセキュリティパッチを提供しています。このサポートは、2020 年 1 月の Python 2.7 end-of-life のアップストリーム Python コミュニティ宣言を超えて拡張されます。

AL2 は、Python 2.7 に大きく依存する yum パッケージマネージャーを使用します。AL2023 では、dnf パッケージマネージャーは Python 3 に移行され、Python 2.7 を必要としなくなりました。AL2023 は Python 3 に完全に移行しました。

Note

AL2023 は Python 2.7 を削除したため、Python を必要とする OS コンポーネントはすべて Python 3 で動作するように記述されています。Amazon Linux によって提供されサポートされているバージョンの Python を引き続き使用するには、Python 2 コードを Python 3 に変換します。

Amazon Linux での Python の詳細については、「[AL2023 での Python](#)」を参照してください。

セキュリティ更新

SELinux

デフォルトでは、AL2023 の Security Enhanced Linux (SELinux) は、enabled および permissive モードに設定されています。permissive モードでは、アクセスの拒否は記録されますが、強制ではありません。

SELinux は AL2 では disabled であった Amazon Linux カーネルのセキュリティ機能です。SELinux は、カーネルの主要なサブシステムに必須のアクセスコントロール (MAC) アーキテクチャを提供するカーネル機能とユーティリティのコレクションです。

詳細については、「[AL2023 の SELinux モードの設定](#)」を参照してください。

SELinux リポジトリ、ツール、ポリシーについての詳細は、「[SELinux ノートブック](#)」、「[SELinux ポリシーの種類](#)」および「[SELinux プロジェクト](#)」を参照してください。

OpenSSL 3

AL2023 には Open Secure Sockets Layer version 3 (OpenSSL 3) 暗号化ツールキットが搭載されています。AL2023 は TLS 1.3 および TLS 1.2 ネットワークプロトコルをサポートしています。

デフォルトでは AL2 には OpenSSL 1.0.2 が含まれています。OpenSSL 1.1.1 に対してアプリケーションを構築できます。

OpenSSL の詳細については、「[OpenSSL の移行ガイド](#)」を参照してください。

セキュリティの詳細については、「[セキュリティ更新および機能](#)」を参照してください。

IMDSv2

デフォルトでは、AL2023 AMI で起動されるインスタンスには IMDSv2のみが必要で、コンテナ化されたワークロードのサポートを可能にするためにデフォルトのホップ制限が 2 に設定されます。imds-support パラメータを v2.0 に設定してこれを行います。詳細については、「[Amazon EC2 ユーザーガイド](#)」の「[AMI の設定](#)」を参照してください。Amazon EC2

Note

セッショントークンの有効期間は 1 秒から 6 時間の間です。IMDSv2 クエリの API リクエストを送信するアドレスは以下のとおりです。

- IPv4: 169.254.169.254
- IPv6: fd00:ec2::254

これらの設定を手動で上書きし、インスタンスメタデータオプションの起動プロパティ IMDSv1 を使用してを有効にできます。IAM コントロールを使用して、さまざまな IMDS 設定を適用することもできます。インスタンスメタデータサービスのセットアップと使用の詳細については、[Amazon EC2 ユーザーガイドの IMDSv2 「 の使用」](#)、[「新しいインスタンスのインスタンスメタデータオプションの設定」](#)、[「既存のインスタンスのインスタンスメタデータオプションの変更」](#) を参照してください。

log4j ホットパッチ ([log4j-cve-2021-44228-hotpatch](#)) の削除

Note

AL2023 は [log4j-cve-2021-44228-hotpatch](#) パッケージに含まれていません。

[CVE-2021-44228 用の Apache Log4j 用 Hotpatch](#) の RPM パッケージバージョンをリリースしました。AL1 AL2 [Amazon Linux へのホットパッチの追加に関する発表](#)の中で、「ホットパッチのインストールは、CVE-2021-44228 または CVE-2021-45046 を軽減する log4j バージョンへの更新に代わるものではない」ことが示されています。

ホットパッチは、log4j へのパッチ適用までの時間を確保するための緩和策でした。AL2023 の最初の一般提供 (GA) リリースは [CVE-2021-44228](#) の 15 か月後だったため、AL2023 にはホットパッチ (有効化されているかどうかにかかわらず) が含まれていません。

Amazon Linux で独自の log4j バージョンを実行しているユーザーは、[CVE-2021-44228](#) または [CVE-2021-45046](#) の影響を受けないバージョンに更新されていることを確認する必要があります。

AL2023 では、[AL2023 の更新](#) でセキュリティパッチを最新の状態に保つためのガイダンスを提供しています。セキュリティアドバイザリは [Amazon Linux セキュリティセンター](#) で公開されています。

安定性向上のための確定的な更新

バージョン管理されたリポジトリによる確定的なアップグレード機能では、デフォルトですべての AL2023 AMI は特定のリポジトリバージョンにロックされます。確定的な更新を使用すると、パッケージのバージョンと更新の一貫性を高めることができます。メジャーかマイナーかにかかわらず、各リリースには特定のリポジトリバージョンが含まれます。

AL2023 の新機能では、確定的な更新がデフォルトで有効になっています。これは、AL2 や他の以前のバージョンで使用されていた手動の段階的なロック方法を改良したものです。

詳細については、「[AL2023のバージョン管理されたリポジトリによる確定的アップグレードの使用](#)」を参照してください。

複数のアップストリームから供給されています。

AL2023 は RPM ベースで、Fedora の複数のバージョンや CentOS 9 Stream などの他の配布から供給されたコンポーネントが含まれています。Amazon Linux カーネルは、他の配布とは別に選んだ kernel.org から直接提供される長期サポート (LTS) リリースから供給されています。

詳細については、「[Fedora との関係](#)」を参照してください。

AMI ルートファイルシステムとデフォルトの Amazon EBS ボリュームタイプ

AL2023 AMI と AL2 はどちらもルートファイルシステム上の XFS ファイルシステムを使用します。AL2023 では、ルートデバイスファイルシステムの mkfs オプションが Amazon EC2 向けにさらに最適化されています。AL2023 は他にも多数のファイルシステムをサポートしており、特定の要件に合わせて他のボリュームでも使用できます。

AL2023 AMI はデフォルトで Amazon EBS gp3 ボリュームを使用しますが、AL2 AMI はデフォルトで Amazon EBS gp2 ボリュームを使用します。インスタンスの起動時にボリュームタイプを変更できます。

Amazon EBS ボリュームタイプの詳細については、「[Amazon EBS 汎用ボリューム](#)」を参照してください。

Amazon EC2 インスタンスの起動の詳細については、「Amazon EC2 [ユーザーガイド](#)」の「[インスタンスの Amazon EC2](#)」を参照してください。

ネットワークシステムサービス

systemd-networkd システムサービスは AL2023 のネットワークインターフェースを管理します。これは、ISC dhclient または dhclient を使用する AL2 からの変更です。

詳細については、「[ネットワークサービス](#)」を参照してください。

統合コントロールグループ階層 (cgroup v2)

コントロールグループ (cgroup) は、プロセスを階層的に整理し、システムリソースをプロセス間で分散するための Linux カーネルの機能です。コントロールグループは、コンテナランタイムの実装や、さまざまな用途に systemd によって使用されています。

AL2 は [cgroupv1](#) をサポートし、AL2023 は [cgroupv2](#) をサポートします。これは [AL2023 ベースの Amazon ECS AMI を使用してコンテナ化されたワークロードをホストする](#) の場合などの、コンテナ化されたワークロードの実行時に特に顕著です。

AL2023 には、[cgroupv1](#) を使用してシステムを実行できるコードが引き続き含まれていますが、これは推奨またはサポートされている設定ではなく、Amazon Linux の今後のメジャーリリースで完全に削除されます。

「[systemd cgroup 委任ドキュメント](#)」など、[低レベルの Linux カーネルインターフェース](#)に関するドキュメントは豊富です。

コンテナ以外の一般的なユースケースは、使用できるシステムリソースに制限がある `systemd` ユニットを作成することです。詳細については、[「systemd.resource-control」](#) を参照してください。

タスクのスケジューリング

`cronie` パッケージは AL2 AMI にデフォルトでインストールされ、定期的なタスクをスケジューリングする従来の `crontab` 方法をサポートしています。AL2023 では、`cronie` はデフォルトでは含まれていません。したがって、`crontab` のサポートはデフォルトでは提供されなくなりました。

オプションで `cronie` パッケージをインストールして元の `cron` ジョブを使用できます。`systemd` には機能が追加されているため、`systemd` タイマーに移行することをお勧めします。

glibc、gcc、および binutils のパッケージ

AL2023 には AL2 と同じコアパッケージが多数含まれています。

AL2023 用に以下の 3 つのコアツールチェーンパッケージが更新されました。

パッケージ名	AL2	AL2023
glibc	2.26	2.34
gcc	7.3	11.3
binutils	2.29	2.39

詳細については、「[コアツールチェーンパッケージ glibc、gcc、binutils](#)」を参照してください。

パッケージマネージャー

AL2023 のデフォルトのソフトウェアパッケージ管理ツールは DNF です。DNF は YUM の後継で、AL2 のパッケージ管理ツールです。

詳細については、「[パッケージ管理ツール](#)」を参照してください。

ロギングシステム

AL2023 では、ロギングシステムパッケージが AL2 から変更されました。AL2023 はデフォルトでは rsyslog をインストールしないため、`/var/log/messages` のような AL2 にあったテキストベースのログファイルはデフォルトでは使用できません。AL2023 のデフォルト設定は `systemd-journal` です。これについては `journalctl` を使用して確認できます。rsyslog は AL2023 のオプションパッケージですが、新しい `systemd` ベースの `journalctl` インターフェースおよび関連パッケージをお勧めします。詳細については、「[journalctl 手動ページ](#)」を参照してください。

curl および libcurl のパッケージ変更

AL2023 では、`curl` および `libcurl` パッケージの一般的なプロトコルと機能が `curl-minimal` と `libcurl-minimal` に分離されています。これにより、ほとんどのユーザーのディスク、メモリ、依存関係のフットプリントが削減され、AL2023 AMI とコンテナのデフォルトパッケージとなっています。

`gopher://` のサポートなどのために `curl` の全機能が必要な場合は、以下のコマンドを実行して `curl-full` および `libcurl-full` パッケージをインストールします。

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

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

GNU プライバシーガード (GNUPG)

AL2023 は、`gnupg2` パッケージの最低限の機能や完全な機能を `gnupg2-minimal` と `gnupg2-full` パッケージに分離しています。デフォルトで `gnupg2-minimal` パッケージのみがインストールされています。これにより、`rpm` パッケージのデジタル署名を検証するのに必要な最小限の機能が提供されます。

キーサーバからキーをダウンロードする機能など、gnupg2 のより多くの機能を利用するには、gnupg2-full パッケージがインストールされていることを確認してください。gnupg2-full の gnupg2-minimal をスワップするには、以下のコマンドを実行します。

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

デフォルト JVM としての Amazon Corretto

AL2023 には、デフォルトで (かつ唯一の) Java 開発キット (JDK) として [Amazon Corretto](#) が含まれています。AL2023 のすべての Java ベースパッケージは、すべてで構築されています Amazon Corretto 17。

AL2 から移行する場合は、AL2 の同等の OpenJDK バージョンから にスムーズに移行できます Amazon Corretto。

AWS CLI v2

AL2023 には AWS CLI バージョン 2 が付属していますが、AL2 には のバージョン 1 が付属していません AWS CLI。

UEFI Preferred

デフォルトでは、UEFI ファームウェアをサポートするインスタンスタイプで AL2023 AMI で起動されたインスタンスはすべて UEFI モードで起動します。これは、ブートモード AMI パラメータを uefi-preferred に設定することで行われます。詳細については、[Amazon EC2 ユーザーガイド](#)の「[ブートモード](#)」を参照してください。

SSH サーバーのデフォルト設定の変更

AL2023 AMI では、リリース時に生成する sshd ホストキーの種類が変更されました。また、起動時に生成されないように、一部のレガシーキータイプが削除されました。クライアントは、rsa-sha2-256 および rsa-sha2-512 プロトコルをサポートするか、ed25519 キーを使用する ssh-ed25519 が必要です。デフォルトでは、ssh-rsa 署名は無効になっています。

さらに、デフォルトの sshd_config ファイルの AL2023 構成設定には UseDNS=no が含まれます。この新しい設定により、DNS 障害によってインスタンスとの ssh セッションを確立できなくなる可能性が低下します。その対価として、authorized_keys ファイル内の from=*hostname.domain*,*hostname.domain* ラインのエントリが解決されなくなります。sshd

が DNS 名を解決しようとしなくなるため、カンマで区切られた `hostname.domain` のそれぞれの値を IP address に対応する値に変換する必要があります。

詳細については、「[デフォルトの SSH サーバー設定](#)」を参照してください。

Extra Packages for Enterprise Linux (EPEL)

Extra Packages for Enterprise Linux (EPEL) は、エンタープライズレベルの Linux オペレーティングシステム用の多数のパッケージを作成することを目的とした Fedora コミュニティ内のプロジェクトです。このプロジェクトは主に、RHEL および CentOS パッケージを作成してきました。AL2 は CentOS 7 との互換性が高いのが特徴です。そのため、多くの EPEL7 パッケージが AL2 で動作します。しかし、AL2023 は EPEL または EPEL のようなリポジトリをサポートしていません。

cloud-init の使用

AL2023 では、cloud-init はパッケージリポジトリを管理します。Amazon Linux の以前のバージョンでは、cloud-init はデフォルトでセキュリティ更新がインストールされていました。これは AL2023 ではデフォルトではありません。AL2023 が起動時に `releasever` を更新できるようにする新しい確定的アップグレード機能には、起動時にパッケージ更新を有効にする方法が説明されています。詳細については、「[AL2023 でパッケージとオペレーティングシステムの更新を管理する](#)」および「[安定性向上のための確定的な更新](#)」を参照してください。

AL2023 では、cloud-init および SELinux を使用できます。詳細については、「[cloud-init を使用して enforcing モードを有効にする](#)」を参照してください。

Cloud-init は cloud-init で HTTP(S) を使用してリモートロケーションから設定コンテンツをロードします。以前のバージョンでは、Amazon Linux はリモートリソースが使用できなくなっても警告しません。AL2023 では、リモートリソースが使用できないと致命的なエラーが発生し、cloud-init の実行に失敗します。AL2 からのこの動作変更により、「フェールクローズ」のデフォルト動作がより安全になりました。

詳細については、「[カスタマイズされた cloud-init](#)」、および「[cloud-init ドキュメント](#)」を参照してください。

グラフィカルデスクトップサポート

AL2023 はクラウド中心で、Amazon EC2 での使用に最適化されており、現時点ではグラフィカル環境やデスクトップ環境は含まれていません。に関するフィードバックについては GitHub、<https://github.com/> を参照してください。

コンパイラトリプレット

AL2023 は、GCC および LLVM に amazon がベンダーであることを示すコンパイラトリプレットを設定します。

したがって、AL2 aarch64-redhat-linux-gcc は AL2023 上の aarch64-amazon-linux-gcc になります。

これはほとんどのユーザーにとって完全に透過的である必要があり、AL2023 でコンパイラを構築しているユーザーにのみ影響する可能性があります。

32 ビット x86 (i686) パッケージ

[AL1 の 2014 年 9 月リリース](#)の一環として、32 ビット AMIs を生成する最後のリリースになると発表されました。そのため、[AL1 の 2015.03 リリース](#)以降、Amazon Linux は 32 ビットモードでのシステム実行をサポートしなくなりました。AL2 は x86-64 ホスト上の 32 ビットバイナリのランタイムサポートを限定的に提供しており、新しい 32 ビットバイナリの構築を可能にする開発パッケージも提供していませんでした。AL2023 には 32 ビットユーザースペースパッケージは含まれなくなりました。64 ビットコードへの移行を完了することをお勧めします。

AL2023 で 32 ビットバイナリを実行する必要がある場合は、AL2023 上で動作する AL2 コンテナ内の AL2 の 32 ビットユーザースペースを使用できます。

lsb_release および system-lsb-core パッケージ

これまで、一部のソフトウェアは (system-lsb-core パッケージによって AL2 で提供されている) lsb_release コマンドを呼び出して、実行されている Linux 配布に関する情報を取得していました。Linux 標準ベース (LSB) ではこのコマンドが導入され、Linux 配布でも採用されました。Linux 配布は、この情報を /etc/os-release およびその他の関連ファイルに保持するという、より単純な標準を使用するように進化しました。

os-release 標準は systemd から生まれました。詳細については、「[systemd os-release ドキュメント](#)」を参照してください。

AL2023 には lsb_release コマンドおよび system-lsb-core パッケージは含まれません。Amazon Linux やその他の主要な Linux 配布との互換性を維持するために、ソフトウェアは os-release 標準への移行を完了する必要があります。

AL2からのAL2023カーネルの変更

AL2023では、6.1カーネルのほか、Amazon Linuxをクラウド向けにさらに最適化するための多くの設定変更が加えられています。ほとんどのユーザーにとって、これらの変更は完全に分かりやすいはずですが。

セキュリティに重点を置いたカーネル設定の変更

CONFIG オプション	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_BUG_ON_DATA_CORRUPTION	n	y	n	y	y	y
CONFIG_DEFAULT_FAULT_MMAP_MIN_ADDR	4096	4096	4096	4096	65536	65536
CONFIG_DEFAULT_VMEM	n	y	n	y	n	n
CONFIG_DEFAULT_VPORT	n	y	n	y	n	n
CONFIG_FORTIFY_SOURCE	n	y	n	y	y	y
CONFIG_HARDENED_USERCOPY_FALLBACK	該当なし	該当なし	y	y	該当なし	該当なし
CONFIG_INIT_ON_ALL	該当なし	該当なし	n	n	n	n

CONFIG オプション	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
OC_DEFAULT_T_ON						
CONFIG_INIT_ON_FREEMEM_DEFAULT_ON	該当なし	該当なし	n	n	n	n
CONFIG_IOMMU_DEFAULT_DMA_STRICT	該当なし	該当なし	該当なし	該当なし	n	n
CONFIG_LD_ISC_AUTOLOAD	y	y	y	y	n	n
CONFIG_SCHED_CORE	該当なし	該当なし	該当なし	該当なし	該当なし	y
CONFIG_SCHED_STACK_END_CHECK	n	y	n	y	y	y
CONFIG_SECURITY_DMESG_RESTRICT	n	n	n	n	y	y
CONFIG_SECURITY_SELINUX_DISABLE	y	y	y	y	n	n

CONFIG オプション	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_SHUFFLE_PAGE_ALLOCATOR	該当なし	該当なし	y	y	y	y
CONFIG_SLAB_FREELIST_HARDENED	n	y	y	y	y	y
CONFIG_SLAB_FREELIST_RANDOM	n	n	y	y	y	y

x86-64 特定セキュリティに焦点を当てたカーネル設定の変更

CONFIG オプション	AL2/4.14/x86_64	AL2/5.10/x86_64	AL2023/6.1/x86_64
CONFIG_AMD_IOMMU	y	y	y
CONFIG_AMD_IOMMU_V2	m	m	y
CONFIG_RANDOMIZE_MEMORY	該当なし	y	y

aarch64 (ARM/Graviton) 特定のセキュリティに焦点を当てたカーネル設定の変更

CONFIG オプション	AL2/4.14/aarch64	AL2/5.10/aarch64	AL2023/6.1/aarch64
CONFIG_AR M64_PTR_AUTH	該当なし	y	y
CONFIG_AR M64_PTR_A UTH_KERNEL	該当なし	該当なし	y
CONFIG_AR M64_SW_TT BR0_PAN	y	y	y

/dev/mem、/dev/kmem、および /dev/port

Amazon Linux 2023 は /dev/mem、AL2 ですでに実施されている制限を /dev/port (CONFIG_DEVMEMそしてCONFIG_DEVPORT) 完全に無効にします。

/dev/kmemこのコードは 5.13 カーネルで Linux から完全に削除され、AL2 では無効になっていましたが、AL2023 には適用されなくなりました。

このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の 1 つです。

FORTIFY_SOURCE

AL2023 はサポートされているすべてのアーキテクチャーで有効になります。CONFIG_FORTIFY_SOURCEこの機能はセキュリティを強化する機能です。コンパイラがバッファサイズを判別して検証できる場合、この機能は一般的な文字列やメモリ関数のバッファオーバーフローを検出できます。

このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の 1 つです。

ライン・ディシプリン・オートロード () CONFIG_LDISC_AUTOLOAD

AL2023 カーネルは、権限を持つプロセスからのリクエストでない限りTIOCSETDioctl、を使用するソフトウェアなどによってラインディシプリンを自動的にロードしません。CAP_SYS_MODULE

このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の 1 つです。

dmesg権限のないユーザーによるアクセス () **CONFIG_SECURITY_DMESG_RESTRICT**

デフォルトでは、AL2023 は権限のないユーザーにはアクセスを許可しません。dmesg

このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

SELinux **selinuxfs** は無効になっています。

AL2023 CONFIG_SECURITY_SELINUX_DISABLE は廃止予定のカーネルオプションを無効にします。これにより、ポリシーが読み込まれる前に SELinux を無効にするランタイムメソッドが有効になりました。

このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

その他のカーネル設定の変更

CONFIG オプション	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_HZ	100	250	100	250	100	100
CONFIG_NR_CPUS	4096	8192	4096	8192	512	512
CONFIG_PANIC_ON_OOPS	y	n	y	n	y	y
CONFIG_PANIC_ON_OOPS_VALUE	1	0	1	0	1	1
CONFIG_PPC	m	m	m	m	n	n
CONFIG_SMP	m	m	m	m	n	n

CONFIG オプション	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_XEN_PV	該当なし	y	該当なし	n	該当なし	n

CONFIG_HZ

AL2023 は、両方のプラットフォームで 100 に設定されています。CONFIG_HZ x86-64 aarch64

CONFIG_NR_CPUS

AL2023 は CONFIG_NR_CPUS Amazon EC2 にある CPU コアの最大数に近い数値に設定されています。

OOPS でのパニック

AL2023 カーネルは、ポーズをとるとパニックに陥ります。この機能は、カーネルコマンドラインで `oops=panic` を起動するのと同じです。

カーネル oops とは、システムのさらなる信頼性に影響を与える可能性のある内部エラーをカーネルが検出したときです。

PPP とスリップのサポート

AL2023 は PPP プロトコルや SLIP プロトコルをサポートしていません。

Xen PV ゲストのサポート

AL2023 は Xen PV ゲストとしての実行をサポートしていません。

カーネルファイルシステムのサポート

AL2 のカーネルがマウントをサポートするファイルシステムにいくつかの変更が加えられたほか、カーネルが解析するパーティショニングスキームも変更されました。

CONFIG オプション	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
<u>CONFIG_AFS_FS</u>	n	m	n	m	n	n
<u>CONFIG_AFS_RRPC</u>	n	m	n	m	n	n
<u>CONFIG_BSD_DISKLABEL</u>	y	y	y	y	n	n
<u>CONFIG_CRAMFS</u>	m	m	m	m	n	n
<u>CONFIG_CRAMFS_BLOCKDEV</u>	該当なし	該当なし	y	n	該当なし	該当なし
<u>CONFIG_DM_CLONE</u>	該当なし	該当なし	n	n	n	n
<u>CONFIG_DM_ERA</u>	m	n	m	n	n	n
<u>CONFIG_DM_INTEGRITY</u>	n	m	n	m	m	m
<u>CONFIG_DM_LOG_WRITES</u>	n	n	m	m	m	m
<u>CONFIG_DM_SWITCH</u>	m	n	m	n	n	n
<u>CONFIG_DM_VERITY</u>	m	n	m	n	n	n

CONFIG オプション	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
<u>CONFIG_ECRYPT_FS</u>	n	m	n	m	n	n
<u>CONFIG_EXFAT_FS</u>	該当なし	該当なし	m	m	m	m
<u>CONFIG_EXT2_FS</u>	n	m	n	m	n	n
<u>CONFIG_EXT3_FS</u>	n	m	n	m	n	n
<u>CONFIG_GFS2_FS</u>	m	m	m	m	n	n
<u>CONFIG_HFSPLUS_FS</u>	n	m	n	m	n	n
<u>CONFIG_HFS_FS</u>	n	m	n	m	n	n
<u>CONFIG_JFS_FS</u>	n	m	n	m	n	n
<u>CONFIG_LDM_PARTITION</u>	n	y	n	y	n	n
<u>CONFIG_MSC_PARTITION</u>	n	y	n	y	n	n
<u>CONFIG_NFS_S_V2</u>	n	m	n	m	n	n

CONFIG オプション	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_NTFS_FS	n	m	n	n	n	n
CONFIG_ROMFS_FS	n	m	n	m	n	n
CONFIG_SOLARIS_X86_PARTITION	n	y	n	y	n	n
CONFIG_SQUASHFS	n	y	n	y	y	y
CONFIG_SUN_PARTITION	n	y	n	y	n	n

Andrew File System (AFS) のサポート

カーネルは afs ファイルシステムをサポートするように構築されなくなりました。AL2 にはユーザースペースのサポートは付属していませんでした。afs

cramfs のサポート

カーネルは cramfs ファイルシステムをサポートするように構築されなくなりました。AL2023 の後継はファイルシステムです。squashfs

BSD ディスクラベルのサポート

カーネルは BSD ディスクラベルをサポートするように構築されなくなりました。BSD ディスクラベル付きのボリュームを読み込む必要がある場合は、さまざまな BSD を起動できます。

デバイスマップャーの変更

AL2023 カーネルで設定されているデバイスマップャーのターゲットにいくつかの変更が加えられました。

eCryptFs サポート

ecryptfs ファイルシステムは Amazon Linux では廃止されました。ecryptfsのユーザースペースコンポーネントは AL1 に存在し、AL2 では削除され、AL2023 はもはやサポート付きのカーネルをビルドしません。ecryptfs

exFAT

exFATファイルシステムのSupportはAL2の5.10カーネルに追加されました。4.14カーネルでAL2を起動した時点ではサポートされていませんでした。AL2023は引き続きファイルシステムをサポートします。exFAT

ext2、ext3、および ext4 のファイルシステム

AL2023 CONFIG_EXT4_USE_FOR_EXT2にはオプションが付属しています。つまり、ext4ファイルシステムコードを使用してレガシーファイルシステムを読み取ることとなります。ext2

CONFIG_GFS2_FS

カーネルは CONFIG_GFS2_FS で構築されなくなりました。

Apple Extended HFS ファイルシステムのサポート (HFS+)

AL2では、x86-64カーネルだけがファイルシステムサポート付きでビルドされました。hfsplusAL2 5.15カーネルには、hfsplusどのアーキテクチャのサポートも含まれていません。AL2023では、Amazon Linux hfsplusでのサポートの廃止が完了しました。

HFS ファイルシステムのサポート

AL2では、x86-64カーネルだけがファイルシステムサポート付きでビルドされました。hfsAL2 5.15カーネルには、hfsどのアーキテクチャのサポートも含まれていません。AL2023では、Amazon Linux hfsでのサポートの廃止が完了しました。

JFS ファイルシステムのサポート

AL2では、x86-64カーネルだけがファイルシステムサポート付きでビルドされました。jfsAL2 5.15カーネルには、jfsどのアーキテクチャのサポートも含まれていません。AL1とAL2のどちら

も JFS ユーザースペースには付属していませんでした。AL2023 では、Amazon Linux jfs でのサポートの廃止が完了しました。

[アップストリームの Linux カーネルは、の削除を検討しています。JFS](#)そのため、JFSファイルシステムにデータがある場合は、それを別のファイルシステムに移行する必要があります。

Windowsロジカルディスクマネージャー (ダイナミックディスク) サポート (CONFIG_LDM_PARTITION)

AL2023 はWindows 2000、Windows XP、Windows VistaMS-DOSスタイルパーティションのあるダイナミックディスクをサポートしなくなりました。このコードは、で導入された新しい GPT ベースのダイナミックディスクをサポートしていませんでした。Windows Vista

Macintosh パーティションマップのサポート

AL2023 は従来の Macintosh パーティションマップをサポートしなくなりました。最新の macOS バージョンでは、デフォルトでこの古いタイプではなく、最新の GPT パーティションテーブルが作成されます。

NFSv2 のサポート

AL2023 は NFSv2 をサポートしなくなりましたが、NFSv3、NFSv4、NFSv4.1、および NFSv4.2 は引き続きサポートします。NFSv3 以降に移行することをお勧めします。

NTFS (CONFIG_NTFS_FS)

ntfs3AL2 の 5.10 カーネルの時点で、Amazon Linux 上の NTFS ntfs ファイルシステムにアクセスするためのコードが置き換えられました。AL2023 にはコードが含まれなくなり、ntfs NTFS ファイルシステムへのアクセスはこのコードのみに依存するようになりました。ntfs3

romfs ファイルシステム

squashfs ファイルシステムは Amazon Linux の romfs ファイルシステムの後継であり、AL2023 カーネルは romfs をサポートするように構築されなくなりました。

Solaris x86 ハードディスクパーティションフォーマット

AL2023 は Solaris x86 ハードディスクパーティションフォーマットをサポートしなくなりました。

squashfszstd 圧縮

AL2023 では、zstdsquashfs サポートされているすべてのアーキテクチャで圧縮ファイルシステムのサポートが追加されています。

Sun パーティションテーブルのサポート

AL2023 には Sun パーティションテーブル形式 () のサポートが含まれなくなりました。CONFIG_SUN_PARTITION

Amazon Linux 2 AMI と Amazon Linux 2023 AMI にインストールされているパッケージの比較

Amazon Linux 2 および AL2023 標準 AMIs に存在する RPMs の比較。

パッケージ	AL2 AMI	AL2023 AMI
acl	2.2.51	2.3.1
acpid	2.0.19	2.0.32
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.4.1
amazon-linux-extras	2.0.3	
amazon-linux-extras-yum-plugin	2.0.3	
amazon-linux-repo-s3		2023.4.20240513
amazon-linux-sb-keys		2023.1
amazon-rpm-config		228
amazon-ssm-agent	3.3.131.0	3.3.380.0

パッケージ	AL2 AMI	AL2023 AMI
at	3.1.13	3.1.23
attr	2.4.46	2.5.1
audit	2.8.1	3.0.6
audit-libs	2.8.1	3.0.6
authconfig	6.2.8	
aws-cfn-bootstrap	2.0	2.0
awscli	1.18.147	
awscli-2		2.15.30
basesystem	10.0	11
bash	4.2.46	5.2.15
bash-completion	2.1	2.11
bc	1.06.95	1.07.1
bind-export-libs	9.11.4	
bind-libs	9.11.4	9.16.48
bind-libs-lite	9.11.4	
bind-license	9.11.4	9.16.48
bind-utils	9.11.4	9.16.48
binutils	2.29.1	2.39
blktrace	1.0.5	
boost-date-time	1.53.0 (x86_64)	

パッケージ	AL2 AMI	AL2023 AMI
boost-filesystem		1.75.0
boost-system	1.53.0 (x86_64)	1.75.0
boost-thread	1.53.0 (x86_64)	1.75.0
bridge-utils	1.5	
bzip2	1.0.6	1.0.8
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023 年 2 月 64 日	2023 年 2 月 64 日
c-ares		1.19.0
checkpolicy		3.4
chkconfig	1.7.4	1.15
chrony	4.2	4.3
cloud-init	19.3	22.2.2
cloud-init-cfg-ec2		22.2.2
cloud-utils-growpart	0.31	0.31
coreutils	8.22	8.32
coreutils-common		8.32
cpio	2.12	2.13
cracklib	2.9.0	2.9.6
cracklib-dicts	2.9.0	2.9.6
cronie	1.4.11	

パッケージ	AL2 AMI	AL2023 AMI
cronie-anacron	1.4.11	
crontabs	1.11	1.11
crypto-policies		20220428
crypto-policies-scripts		20220428
cryptsetup	1.7.4	2.6.1
cryptsetup-libs	1.7.4	2.6.1
curl	8.3.0	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.26	2.1.27
cyrus-sasl-plain	2.1.26	2.1.27
dbus	1.10.24	1.12.28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.10.24	1.12.28
device-mapper	1.02.170	1.02.185
device-mapper-event	1.02.170	
device-mapper-event-libs	1.02.170	
device-mapper-libs	1.02.170	1.02.185

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

パッケージ	AL2 AMI	AL2023 AMI
dwz		0.14
dyninst	9.3.1 (x86_64)	10.2.1
e2fsprogs	1.42.9	1.46.5
e2fsprogs-libs	1.42.9	1.46.5
ec2-hibinit-agent	1.0.8	1.0.8
ec2-instance-connect	1.1	1.1
ec2-instance-connect-selinux	1.1	1.1
ec2-net-utils	1.7.3	
ec2-utils	1.2	2.2.0
ed	1.9	1.14.2
efibootmgr	15 (aarch64)	
efi-filesystem		5
efi-srpm-macros		5
efivar		38
efivar-libs	31 (aarch64)	38
elfutils-debuginfod-client		0.188
elfutils-default-yama-scope	0.176	0.188
elfutils-libelf	0.176	0.188

パッケージ	AL2 AMI	AL2023 AMI
elfutils-libs	0.176	0.188
ethtool	4.8	5.15
expat	2.1.0	2.5.0
file	5.11	5.39
file-libs	5.11	5.39
filesystem	3.2	3.14
findutils	4.5.11	4.8.0
fipscheck	1.4.1	
fipscheck-lib	1.4.1	
fonts-srpm-macros		2.0.5
freetype	2.8	
fstrm		0.6.1
fuse-libs	2.9.2	2.9.9
gawk	4.0.2	5.1.0
gdbm	1.13	
gdbm-libs		1.19
gdisk	0.8.10	1.0.8
generic-logos	18.0.0	
GeoIP	1.5.0	
gettext	0.19.8.1	0.21

パッケージ	AL2 AMI	AL2023 AMI
gettext-libs	0.19.8.1	0.21
ghc-srpm-macros		1.5.0
glib2	2.56.1	2.74.7
glibc	2.26	2.34
glibc-all-langpacks	2.26	2.34
glibc-common	2.26	2.34
glibc-gconv-extra		2.34
glibc-locale-source	2.26	2.34
glibc-minimal-lang pack	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

パッケージ	AL2 AMI	AL2023 AMI
grub2-efi-aa64	2.06 (aarch64)	
grub2-efi-aa64-ec2	2.06 (aarch64)	2.06 (aarch64)
grub2-efi-aa64-modules	2.06 (ノーク)	
grub2-efi-x64-ec2	2.06 (x86_64)	2.06 (x86_64)
grub2-pc	2.06 (x86_64)	
grub2-pc-modules	2.06 (ノーク)	2.06
grub2-tools	2.06	2.06
grub2-tools-minimal	2.06	2.06
grubby	8.28	8.40
gssproxy	0.7.0	0.8.4
gzip	1.5	1.12
hardlink	1.3	
hibagent	1.1.0	
hostname	3.13	3.23
hunspell	1.3.2	1.7.0
hunspell-en	0.20121024	0.20140811.1
hunspell-en-GB	0.20121024	0.20140811.1
hunspell-en-US	0.20121024	0.20140811.1
hunspell-filesystem		1.7.0
hwdata	0.252	0.353

パッケージ	AL2 AMI	AL2023 AMI
info	5.1	6.7
inih		49
initscripts	9.49.47	10.09
iproute	5.10.0	5.10.0
iptables	1.8.4	
iptables-libs	1.8.4	
iputils	20180629	20210202
irqbalance	1.7.0	1.9.0
jansson	2.10	2.14
jbigkit-libs	2.0	
jitterentropy		3.4.1
jq		1.7.1
json-c	0.11	0.14
kbd	1.15.5	2.4.0
kbd-legacy	1.15.5	
kbd-misc	1.15.5	2.4.0
kernel	5.10.215	6.1.90
kernel-livepatch-r epo-s3		2023.4.20240513
kernel-srpm-macros		1.0
kernel-tools	5.10.215	6.1.90

パッケージ	AL2 AMI	AL2023 AMI
keyutils	1.5.8	1.6.3
keyutils-libs	1.5.8	1.6.3
kmod	25	29
kmod-libs	25	29
kpartx	0.4.9	
kpatch-runtime	0.9.4	0.9.7
krb5-libs	1.15.1	1.21
langtable	0.0.31	
langtable-data	0.0.31	
langtable-python	0.0.31	
less	458	608
libacl	2.2.51	2.3.1
libaio	0.3.109	0.3.111
libarchive		3.5.3
libargon2		20171227
libassuan	2.1.0	2.5.5
libattr	2.4.46	2.5.1
libbasicobjects	0.1.1	0.1.1
libblkid	2.30.2	2.37.4
libcap	2.54	2.48

パッケージ	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		0.1.20
libconfig	1.4.9	1.7.2
libcroco	0.6.12	
libcrypt	2.26	
libcurl	8.3.0	
libcurl-minimal		8.5.0
libdaemon	0.14	
libdb	5.3.21	5.3.28
libdb-utils	5.3.21	
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	

パッケージ	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	2.0.90	
libkcapi		1.4.0
libkcapi-hmaccalc		1.4.0
libldb		2.6.2
libmaxminddb		1.5.2

パッケージ	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_contrack	1.0.6	
libnfnetworking	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	0.1.5	0.1.5
librepo		1.14.5

パッケージ	AL2 AMI	AL2023 AMI
libreport-filessystem		2.15.2
libseccomp	2.5.2	2.5.3
libselinux	2.5	3.4
libselinux-utils	2.5	3.4
libsemanage	2.5	3.4
libsepol	2.5	3.4
libsigsegv		2.13
libsmartcols	2.30.2	2.37.4
libsolv		0.7.22
libss	1.42.9	1.46.5
libssh2	1.4.3	
libsss_certmap		2.9.4
libsss_idmap	1.16.5	2.9.4
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	

パッケージ	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

パッケージ	AL2 AMI	AL2023 AMI
libxml2-python	2.9.1	
libyaml	0.1.4	0.2.5
libzstd		1.5.5
lm_sensors-libs	3.4.0	3.6.0
lmdb-libs		0.9.29
logrotate	3.8.6	3.20.1
lsof	4.87	4.94.0
lua	5.1.4	
lua-libs		5.4.4
lua-srpm-macros		1
lvm2	2.02.187	
lvm2-libs	2.02.187	
lz4	1.7.5	
lz4-libs		1.9.4
make	3.82	
man-db	2.6.3	2.9.3
man-pages	3.53	5.10
man-pages-overrides	7.5.2	
mariadb-libs	5.5.68	
mdadm	4.0	

パッケージ	AL2 AMI	AL2023 AMI
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)
mlocate	0.26	
mpfr		4.1.0
mtr	0.92	
nano	2.9.8	5.8
ncurses	6.0	6.2
ncurses-base	6.0	6.2
ncurses-libs	6.0	6.2
nettle	2.7.1	3.8
net-tools	2.0	2.0
newt	0.52.15	0.52.21
newt-python	0.52.15	
nfs-utils	1.3.0	2.5.4
npth		1.6
nspr	4.35.0	4.35.0
nss	3.90.0	3.90.0
nss-pem	1.0.3	
nss-softokn	3.90.0	3.90.0
nss-softokn-freebl	3.90.0	3.90.0
nss-sysinit	3.90.0	3.90.0

パッケージ	AL2 AMI	AL2023 AMI
nss-tools	3.90.0	
nss-util	3.90.0	3.90.0
ntsysv	1.7.4	1.15
numactl-libs	2.0.9	2.0.14
ocaml-srpm-macros		6
oniguruma		6.9.7.1
openblas-srpm-macros		2
openldap	2.4.44	2.4.57
openssh	7.4p1	8.7p1
openssh-clients	7.4p1	8.7p1
openssh-server	7.4p1	8.7p1
openssl	1.0.2k	3.0.8
openssl-libs	1.0.2k	3.0.8
openssl-pkcs11		0.4.12
os-prober	(1.58)	1.77
p11-kit	0.23.22	0.24.1
p11-kit-trust	0.23.22	0.24.1
package-notes-srpm-macros		0.4
pam	1.1.8	1.5.1
parted	3.1	3.4

パッケージ	AL2 AMI	AL2023 AMI
passwd	0.79	0.80
pciutils	3.5.1	3.7.0
pciutils-libs	3.5.1	3.7.0
pcre	8.32	
pcre2	10.23	10.40
pcre2-syntax		10.40
perl	5.16.3	
perl-Carp	1.26	1.50
perl-Class-Struct		0.66
perl-constant	1.27	1.33
perl-DynaLoader		1.47
perl-Encode	2.51	3.15
perl-Errno		1.30
perl-Exporter	5.68	5.74
perl-Fcntl		1.13
perl-File-Basename		2.85
perl-File-Path	2.09	2.18
perl-File-stat		1.09
perl-File-Temp	0.23.01	0.231.100
perl-Filter	1.49	

パッケージ	AL2 AMI	AL2023 AMI
perl-Getopt-Long	2.40	2.52
perl-Getopt-Std		1.12
perl-HTTP-Tiny	0.033	0.078
perl-if		0.60.800
perl-interpreter		5.32.1
perl-IO		1.43
perl-IPC-Open3		1.21
perl-libs	5.16.3	5.32.1
perl-macros	5.16.3	
perl-MIME-Base64		3.16
perl-mro		1.23
perl-overload		1.31
perl-overloading		0.02
perl-parent	0.225	0.238
perl-PathTools	3.40	3.78
perl-Pod-Escapes	1.04	1.07
perl-podlators	2.5.1	4.14
perl-Pod-Perldoc	3.20	3.28.01
perl-Pod-Simple	3.28	3.42
perl-Pod-Usage	1.63	2.01

パッケージ	AL2 AMI	AL2023 AMI
perl-POSIX		1.94
perl-Scalar-List-Utils	1.27	1.56
perl-SelectSaver		1.02
perl-Socket	2.010	2.032
perl-srpm-macros		1
perl-Storable	2.45	3.21
perl-subst		1.03
perl-Symbol		1.08
perl-Term-ANSIColor		5.01
perl-Term-Cap		1.17
perl-Text-ParseWords	3.29	3.30
perl-Text-Tabs+Wrap		2021.0726
perl-threads	1.87	
perl-threads-shared	1.43	
perl-Time-HiRes	1.9725	
perl-Time-Local	1.2300	1.300
perl-vars		1.05
pinentry	0.8.1	
pkgconf		1.8.0
pkgconfig	0.27.1	

パッケージ	AL2 AMI	AL2023 AMI
pkgconf-m4		1.8.0
pkgconf-pkg-config		1.8.0
plymouth	0.8.9	
plymouth-core-libs	0.8.9	
plymouth-scripts	0.8.9	
pm-utils	1.4.1	
policycoreutils	2.5	3.4
policycoreutils-python-utils		3.4
popt	1.13	1.18
postfix	2.10.1	
procps-ng	3.3.10	3.3.17
protobuf-c		1.4.1
psacct	6.6.1	6.6.4
psmisc	22.20	23.4
pth	2.0.7	
publicsuffix-list-dafsa	20240208	20240212
pygpme	0.3	
pyliblzma	0.5.3	
pystache	0.5.3	

パッケージ	AL2 AMI	AL2023 AMI
python	2.7.18	
python2-botocore	1.18.6	
python2-colorama	0.3.9	
python2-cryptography	1.7.2	
python2-dateutil	2.6.1	
python2-futures	3.0.5	
python2-jmespath	0.9.3	
python2-jsonschema	2.5.1	
python2-oauthlib	2.0.1	
python2-pyasn1	0.1.9	
python2-rpm	4.11.3	
python2-rsa	3.4.1	
python2-s3transfer	0.3.3	
python2-setuptools	41.2.0	
python2-six	1.11.0	
python3	3.7.16	3.9.16
python3-attrs		20.3.0
python3-audit		3.0.6
python3-awscli		0.19.19
python3-babel		2.9.1

パッケージ	AL2 AMI	AL2023 AMI
python3-cffi		1.14.5
python3-chardet		4.0.0
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36.0.1
python3-daemon	2.2.3	2.3.0
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils	0.14	0.16
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-idna		2.10
python3-jinja2		2.11.3
python3-jmespath		0.10.0
python3-jsonpatch		1.21
python3-jsonpointer		2.0
python3-jjsonschema		3.2.0

パッケージ	AL2 AMI	AL2023 AMI
python3-libcomps		0.1.20
python3-libdnf		0.69.0
python3-libs	3.7.16	3.9.16
python3-libselinux		3.4
python3-libsemanage		3.4
python3-libstorage mgmt		1.9.4
python3-lockfile	0.11.0	0.12.2
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip	20.2.2	
python3-pip-wheel		21.3.1
python3-ply		3.11
python3-policycore utils		3.4
python3-prettytable		0.7.2
python3-prompt-too lkit		3.0.24
python3-pycparser		2.20
python3-pyrsistent		0.17.3

パッケージ	AL2 AMI	AL2023 AMI
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pystache	0.5.4	
python3-pytz		2022.7.1
python3-pyyaml		5.4.1
python3-requests		2.25.1
python3-rpm		4.16.1.3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools	49.1.3	59.6.0
python3-setuptools-wheel		59.6.0
python3-simplejson	3.2.0	
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1.25.10
python3-wcwidth		0.2.5
python-babel	0.9.6	
python-backports	1.0	

パッケージ	AL2 AMI	AL2023 AMI
python-backports-s sl_match_hostname	3.5.0.1	
python-cffi	1.6.0	
python-chardet	2.2.1	
python-chevron		0.13.1
python-configobj	4.7.2	
python-daemon	1.6	
python-devel	2.7.18	
python-docutils	0.12	
python-enum34	1.0.4	
python-idna	2.4	
python-iniparse	0.4	
python-ipaddress	1.0.16	
python-jinja2	2.7.2	
python-jsonpatch	1.2	
python-jsonpointer	1.9	
python-jwcrypto	0.4.2	
python-kitchen	1.1.1	
python-libs	2.7.18	
python-lockfile	0.9.1	
python-markupsafe	0.11	

パッケージ	AL2 AMI	AL2023 AMI
python-pillow	2.0.0	
python-ply	3.4	
python-pycparser	2.14	
python-pycurl	7.19.0	
python-repoze-lru	0.4	
python-requests	2.6.0	
python-simplejson	3.2.0	
python-srpm-macros		3.9
python-urlgrabber	3.10	
python-urllib3	1.25.9	
pyxattr	0.5.1	
PyYAML	3.10	
qrencode-libs	3.4.1	
quota	4.01	4.06
quota-nls	4.01	4.06
rdate	1.4	
readline	6.2	8.1
rng-tools	6.8	6.14
rootfiles	8.1	8.1
rpcbind	0.2.0	1.2.6

パッケージ	AL2 AMI	AL2023 AMI
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
rsync	3.1.2	3.2.6
rsyslog	8.24.0	
rust-srpm-macros		21
sbsigntools		0.9.4
scl-utils	20130529	
screen	4.1.0	4.8.0
sed	4.2.2	4.8
selinux-policy	3.13.1	37.22
selinux-policy-targeted	3.13.1	37.22
setserial	2.17	
setup	2.8.71	2.13.7
setuptools	1.19.11	
sgpio	1.2.0.10	

パッケージ	AL2 AMI	AL2023 AMI
shadow-utils	4.1.5.1	4.9
shared-mime-info	1.8	
slang	2.2.4	2.3.2
sqlite	3.7.17	
sqlite-libs		3.40.0
sssd-client	1.16.5	2.9.4
sssd-common		2.9.4
sssd-kcm		2.9.4
sssd-nfs-idmap		2.9.4
strace	4.26	6.8
sudo	1.8.23	1.9.15
sysctl-defaults	1.0	1.0
sysstat	10.1.5	12.5.6
systemd	219	252.16
systemd-libs	219	252.16
systemd-networkd		252.16
systemd-pam		252.16
systemd-resolved		252.16
systemd-sysv	219	
systemd-udev		252.16

パッケージ	AL2 AMI	AL2023 AMI
system-release	2	2023.4.20240513
systemtap-runtime	4.5	4.8
sysvinit-tools	2.88	
tar	1.26	1.34
tbb		2020年3月
tcp_wrappers	7.6	
tcp_wrappers-libs	7.6	
tcpdump	4.9.2	4.99.1
tcsch	6.18.01	6.24.07
teamd	1.27	
time	1.7	1.9
traceroute	2.0.22	2.1.3
tzdata	2024年a	2024年a
unzip	6.0	6.0
update-motd	1.1.2	2.2
usermode	1.111	
userspace-rcu		0.12.1
ustr	1.0.4	
util-linux	2.30.2	2.37.4
util-linux-core		2.37.4

パッケージ	AL2 AMI	AL2023 AMI
vim-common	9.0.2153	9.0.2153
vim-data	9.0.2153	9.0.2153
vim-enhanced	9.0.2153	9.0.2153
vim-filesystem	9.0.2153	9.0.2153
vim-minimal	9.0.2153	9.0.2153
virt-what	1.18	
wget	1.14	1.21.3
which	2.20	2.21
words	3.0	3.0
xfsdump	3.1.8	3.1.11
xfspgrog	5.0.0	5.18.0
xxd	9.0.2153	9.0.2153
xxhash-libs		0.8.0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yajl	2.0.4	
yum	3.4.3	4.14.0
yum-langpacks	0.4.2	
yum-metadata-parser	1.1.4	
yum-plugin-priorities	1.1.31	

パッケージ	AL2 AMI	AL2023 AMI
yum-utils	1.1.31	
zip	3.0	3.0
zlib	1.2.7	1.2.11
zram-generator		1.1.2
zram-generator-defaults		1.1.2
zstd		1.5.5

Amazon Linux 2 と Amazon Linux 2023 最小 AMI にインストールされているパッケージの比較

Amazon Linux 2 および AL2023 最小 AMIs に存在する RPMs の比較。

パッケージ	AL2 最小	AL2023 最小
acl	2.2.51	
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.4.1
amazon-linux-extras	2.0.3	
amazon-linux-repo-s3		2023.4.20240513
amazon-linux-sb-keys		2023.1
audit	2.8.1	3.0.6
audit-libs	2.8.1	3.0.6

パッケージ	AL2 最小	AL2023 最小
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 年 2 月 64 日	2023 年 2 月 64 日
checkpolicy		3.4
chkconfig	1.7.4	
chrony	4.2	4.3
cloud-init	19.3	22.2.2
cloud-init-cfg-ec2		22.2.2
cloud-utils-growpart	0.31	0.31
coreutils	8.22	8.32
coreutils-common		8.32
cpio	2.12	2.13
cracklib	2.9.0	2.9.6
cracklib-dicts	2.9.0	2.9.6
cronie	1.4.11	
cronie-anacron	1.4.11	

パッケージ	AL2 最小	AL2023 最小
crontabs	1.11	
crypto-policies		20220428
cryptsetup-libs	1.7.4	2.6.1
curl	8.3.0	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.26	2.1.27
dbus	1.10.24	1.12.28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.10.24	1.12.28
device-mapper	1.02.170	1.02.185
device-mapper-libs	1.02.170	1.02.185
dhclient	4.2.5	
dhcp-common	4.2.5	
dhcp-libs	4.2.5	
diffutils	3.3	3.8
dnf		4.14.0
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0

パッケージ	AL2 最小	AL2023 最小
dnf-plugin-support-info		1.2
dracut	033	055
dracut-config-ec2	2.0	3.0
dracut-config-generic	033	055
e2fsprogs	1.42.9	1.46.5
e2fsprogs-libs	1.42.9	1.46.5
ec2-utils	1.2	2.2.0
efibootmgr	15 (aarch64)	
efi-filesystem		5
efivar		38
efivar-libs	31 (aarch64)	38
elfutils-default-yama-scope	0.176	0.188
elfutils-libelf	0.176	0.188
elfutils-libs	0.176	0.188
expat	2.1.0	2.5.0
file	5.11	5.39
file-libs	5.11	5.39
filesystem	3.2	3.14

パッケージ	AL2 最小	AL2023 最小
findutils	4.5.11	4.8.0
fipscheck	1.4.1	
fipscheck-lib	1.4.1	
freetype	2.8	
fuse-libs	2.9.2	2.9.9
gawk	4.0.2	5.1.0
gdbm	1.13	
gdbm-libs		1.19
gdisk	0.8.10	1.0.8
gettext	0.19.8.1	0.21
gettext-libs	0.19.8.1	0.21
glib2	2.56.1	2.74.7
glibc	2.26	2.34
glibc-all-langpacks	2.26	2.34
glibc-common	2.26	2.34
glibc-locale-source	2.26	2.34
glibc-minimal-lang pack	2.26	
gmp	6.0.0	6.2.1
gnupg2	2.0.22	
gnupg2-minimal		2.3.7

パッケージ	AL2 最小	AL2023 最小
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 (ノーク)	
grub2-efi-x64-ec2	2.06 (x86_64)	2.06 (x86_64)
grub2-pc	2.06 (x86_64)	
grub2-pc-modules	2.06 (ノーク)	2.06
grub2-tools	2.06	2.06
grub2-tools-minimal	2.06	2.06
grubby	8.28	8.40
gzip	1.5	1.12
hardlink	1.3	
hostname	3.13	3.23
hwdata		0.353
info	5.1	

パッケージ	AL2 最小	AL2023 最小
inih		49
initscripts	9.49.47	10.09
iproute	5.10.0	5.10.0
iptables	1.8.4	
iptables-libs	1.8.4	
iputils	20180629	20210202
irqbalance	1.7.0	1.9.0
jansson		2.14
jitterentropy		3.4.1
jq		1.7.1
json-c		0.14
kbd		2.4.0
kbd-misc		2.4.0
kernel	4.14.343	6.1.90
kernel-livepatch-r epo-s3		2023.4.20240513
keyutils-libs	1.5.8	1.6.3
kmod	25	29
kmod-libs	25	29
kpartx	0.4.9	
krb5-libs	1.15.1	1.21

パッケージ	AL2 最小	AL2023 最小
less	458	608
libacl	2.2.51	2.3.1
libarchive		3.5.3
libargon2		20171227
libassuan	2.1.0	2.5.5
libattr	2.4.46	2.5.1
libblkid	2.30.2	2.37.4
libcap	2.54	2.48
libcap-ng	0.7.5	0.8.2
libcbor		0.7.0
libcom_err	1.42.9	1.46.5
libcomps		0.1.20
libcroco	0.6.12	
libcrypt	2.26	
libcurl	8.3.0	
libcurl-minimal		8.5.0
libdb	5.3.21	5.3.28
libdb-utils	5.3.21	
libdnf		0.69.0
libeconf		0.4.0

パッケージ	AL2 最小	AL2023 最小
libedit	3.0	3.1
libestr	0.1.9	
libfastjson	0.99.4	
libfdisk	2.30.2	2.37.4
libffi	3.0.13	3.4.4
libfido2		1.10.0
libgcc	7.3.1	11.4.1
libgcrypt	1.5.3	1.10.2
libgomp	7.3.1	11.4.1
libgpg-error	1.12	1.42
libicu	50.2	
libidn	1.28	
libidn2	2.3.0	2.3.2
libkcapi		1.4.0
libkcapi-hmaccalc		1.4.0
libmetalink	0.1.3	
libmnl	1.0.3	1.0.4
libmodulemd		2.13.0
libmount	2.30.2	2.37.4
libnetfilter_connt rack	1.0.6	

パッケージ	AL2 最小	AL2023 最小
libnfnetwork	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-filestore		2.15.2
libseccomp	2.5.2	2.5.3
libselinux	2.5	3.4
libselinux-utils	2.5	3.4
libsemanage	2.5	3.4
libsepol	2.5	3.4
libsigsegv		2.13
libsmartcols	2.30.2	2.37.4
libsolv		0.7.22
libss	1.42.9	1.46.5
libssh2	1.4.3	
libstdc++	7.3.1	11.4.1

パッケージ	AL2 最小	AL2023 最小
libsysfs	2.1.0	
libtasn1	4.10	4.19.0
libtextstyle		0.21
libunistring	0.9.3	0.9.10
libuser	0.60	0.63
libutempter	1.1.6	1.2.1
libuuid	2.30.2	2.37.4
libverto	0.2.5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libyaml	0.1.4	0.2.5
libzstd		1.5.5
logrotate	3.8.6	3.20.1
lua	5.1.4	
lua-libs		5.4.4
lz4	1.7.5	
lz4-libs		1.9.4
make	3.82	
man-db	2.6.3	2.9.3
mariadb-libs	5.5.68	

パッケージ	AL2 最小	AL2023 最小
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)
mpfr		4.1.0
ncurses	6.0	6.2
ncurses-base	6.0	6.2
ncurses-libs	6.0	6.2
nettle	2.7.1	3.8
net-tools	2.0	2.0
newt	0.52.15	
newt-python	0.52.15	
npth		1.6
nspr	4.35.0	
nss	3.90.0	
nss-pem	1.0.3	
nss-softokn	3.90.0	
nss-softokn-freebl	3.90.0	
nss-sysinit	3.90.0	
nss-tools	3.90.0	
nss-util	3.90.0	
numactl-libs	2.0.9	2.0.14
oniguruma		6.9.7.1

パッケージ	AL2 最小	AL2023 最小
openldap	2.4.44	2.4.57
openssh	7.4p1	8.7p1
openssh-clients	7.4p1	8.7p1
openssh-server	7.4p1	8.7p1
openssl	1.0.2k	3.0.8
openssl-lib	1.0.2k	3.0.8
openssl-pkcs11		0.4.12
os-prober	(1.58)	1.77
p11-kit	0.23.22	0.24.1
p11-kit-trust	0.23.22	0.24.1
pam	1.1.8	1.5.1
passwd	0.79	0.80
pciutils		3.7.0
pciutils-lib		3.7.0
pcre	8.32	
pcre2	10.23	10.40
pcre2-syntax		10.40
pinentry	0.8.1	
pkgconfig	0.27.1	
policycoreutils	2.5	3.4

パッケージ	AL2 最小	AL2023 最小
popt	1.13	1.18
postfix	2.10.1	
procps-ng	3.3.10	3.3.17
psmisc	22.20	23.4
pth	2.0.7	
publicsuffix-list-dafsa	20240208	20240212
pygpgme	0.3	
pyliblzma	0.5.3	
python	2.7.18	
python2-cryptography	1.7.2	
python2-jsonschema	2.5.1	
python2-oauthlib	2.0.1	
python2-pyasn1	0.1.9	
python2-rpm	4.11.3	
python2-setuptools	41.2.0	
python2-six	1.11.0	
python3		3.9.16
python3-attrs		20.3.0
python3-audit		3.0.6
python3-awscli		0.19.19

パッケージ	AL2 最小	AL2023 最小
python3-babel		2.9.1
python3-cffi		1.14.5
python3-chardet		4.0.0
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36.0.1
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils		0.16
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-idna		2.10
python3-jinja2		2.11.3
python3-jmespath		0.10.0
python3-jsonpatch		1.21
python3-jsonpointer		2.0
python3-jjsonschema		3.2.0

パッケージ	AL2 最小	AL2023 最小
python3-libcomps		0.1.20
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-policycore utils		3.4
python3-prettytable		0.7.2
python3-prompt-too lkit		3.0.24
python3-pycparser		2.20
python3-pyrsistent		0.17.3
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pytz		2022.7.1
python3-pyyaml		5.4.1

パッケージ	AL2 最小	AL2023 最小
python3-requests		2.25.1
python3-rpm		4.16.1.3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools		59.6.0
python3-setuptools-wheel		59.6.0
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1.25.10
python3-wcwidth		0.2.5
python-babel	0.9.6	
python-backports	1.0	
python-backports-s sl_match_hostname	3.5.0.1	
python-cffi	1.6.0	
python-chardet	2.2.1	
python-configobj	4.7.2	
python-devel	2.7.18	

パッケージ	AL2 最小	AL2023 最小
python-enum34	1.0.4	
python-idna	2.4	
python-iniparse	0.4	
python-ipaddress	1.0.16	
python-jinja2	2.7.2	
python-jsonpatch	1.2	
python-jsonpointer	1.9	
python-jwcrypto	0.4.2	
python-libs	2.7.18	
python-markupsafe	0.11	
python-ply	3.4	
python-pycparser	2.14	
python-pycurl	7.19.0	
python-repoze-lru	0.4	
python-requests	2.6.0	
python-urlgrabber	3.10	
python-urllib3	1.25.9	
pyattr	0.5.1	
PyYAML	3.10	
qrencode-libs	3.4.1	

パッケージ	AL2 最小	AL2023 最小
readline	6.2	8.1
rng-tools	6.8	6.14
rootfiles	8.1	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
rsyslog	8.24.0	
sbsigntools		0.9.4
sed	4.2.2	4.8
selinux-policy	3.13.1	37.22
selinux-policy-targeted	3.13.1	37.22
setup	2.8.71	2.13.7
shadow-utils	4.1.5.1	4.9
shared-mime-info	1.8	
slang	2.2.4	
sqlite	3.7.17	

パッケージ	AL2 最小	AL2023 最小
sqlite-libs		3.40.0
sudo	1.8.23	1.9.15
sysctl-defaults	1.0	1.0
systemd	219	252.16
systemd-libs	219	252.16
systemd-networkd		252.16
systemd-pam		252.16
systemd-resolved		252.16
systemd-sysv	219	
systemd-udev		252.16
system-release	2	2023.4.20240513
sysvinit-tools	2.88	
tar	1.26	1.34
tcp_wrappers-libs	7.6	
tzdata	2024年a	2024年a
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

パッケージ	AL2 最小	AL2023 最小
vim-data	9.0.2153	9.0.2153
vim-minimal	9.0.2153	9.0.2153
which	2.20	2.21
xfspgrog	5.0.0	5.18.0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	1.1.4	
yum-plugin-priorities	1.1.31	
zlib	1.2.7	1.2.11
zram-generator		1.1.2
zram-generator-defaults		1.1.2
zstd		1.5.5

Amazon Linux 2 と Amazon Linux 2023 ベースコンテナイメージにインストールされているパッケージの比較

Amazon Linux 2 および AL2023 ベースコンテナイメージに存在する RPMs の比較。

パッケージ	AL2 コンテナ	AL2023 コンテナ
alternatives		1.15

パッケージ	AL2 コンテナ	AL2023 コンテナ
amazon-linux-extras	2.0.3	
amazon-linux-repo-cdn		2023.4.20240513
audit-libs		3.0.6
basesystem	10.0	11
bash	4.2.46	5.2.15
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023 年 2 月 64 日	2023 年 2 月 64 日
chkconfig	1.7.4	
coreutils	8.22	
coreutils-single		8.32
cpio	2.12	
crypto-policies		20220428
curl	8.3.0	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.26	
diffutils	3.3	
dnf		4.14.0
dnf-data		4.14.0
elfutils-default-yama-scope		0.188

パッケージ	AL2 コンテナ	AL2023 コンテナ
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-lang pack	2.26	2.34
gmp	6.0.0	6.2.1
gnupg2	2.0.22	
gnupg2-minimal		2.3.7
gpgme	1.3.2	1.15.1
grep	2.20	3.8
info	5.1	

パッケージ	AL2 コンテナ	AL2023 コンテナ
json-c		0.14
keyutils-libs	1.5.8	1.6.3
krb5-libs	1.15.1	1.21
libacl	2.2.51	2.3.1
libarchive		3.5.3
libassuan	2.1.0	2.5.5
libattr	2.4.46	2.5.1
libblkid	2.30.2	2.37.4
libcap	2.54	2.48
libcap-ng		0.8.2
libcom_err	1.42.9	1.46.5
libcomps		0.1.20
libcrypt	2.26	
libcurl	8.3.0	
libcurl-minimal		8.5.0
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

パッケージ	AL2 コンテナ	AL2023 コンテナ
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.12	1.42
libidn2	2.3.0	2.3.2
libmetalink	0.1.3	
libmodulemd		2.13.0
libmount	2.30.2	2.37.4
libnghttp2	1.41.0	1.59.0
libpsl	0.21.5	0.21.1
librepo		1.14.5
libreport-filesystem		2.15.2
libselinux	2.5	3.4
libsepol	2.5	3.4
libsigsegv		2.13
libsmartcols		2.37.4
libsolv		0.7.22
libssh2	1.4.3	
libstdc++	7.3.1	11.4.1
libtasn1	4.10	4.19.0
libunistring	0.9.3	0.9.10

パッケージ	AL2 コンテナ	AL2023 コンテナ
libuuid	2.30.2	2.37.4
libverto	0.2.5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libyaml		0.2.5
libzstd		1.5.5
lua	5.1.4	
lua-libs		5.4.4
lz4-libs		1.9.4
mpfr		4.1.0
ncurses	6.0	
ncurses-base	6.0	6.2
ncurses-libs	6.0	6.2
npth		1.6
nspr	4.35.0	
nss	3.90.0	
nss-pem	1.0.3	
nss-softokn	3.90.0	
nss-softokn-freebl	3.90.0	
nss-sysinit	3.90.0	

パッケージ	AL2 コンテナ	AL2023 コンテナ
nss-tools	3.90.0	
nss-util	3.90.0	
openldap	2.4.44	
openssl-libs	1.0.2k	3.0.8
p11-kit	0.23.22	0.24.1
p11-kit-trust	0.23.22	0.24.1
pcre	8.32	
pcre2		10.40
pcre2-syntax		10.40
pinentry	0.8.1	
popt	1.13	1.18
pth	2.0.7	
publicsuffix-list-dafsa	20240208	20240212
pygpgme	0.3	
pyliblzma	0.5.3	
python	2.7.18	
python2-rpm	4.11.3	
python3		3.9.16
python3-dnf		4.14.0
python3-gpg		1.15.1

パッケージ	AL2 コンテナ	AL2023 コンテナ
python3-hawkey		0.69.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0
python3-libs		3.9.16
python3-pip-wheel		21.3.1
python3-rpm		4.16.1.3
python3-setuptools-wheel		59.6.0
python-iniparse	0.4	
python-libs	2.7.18	
python-pycurl	7.19.0	
python-urlgrabber	3.10	
pyxattr	0.5.1	
readline	6.2	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
sed	4.2.2	4.8
setup	2.8.71	2.13.7
shared-mime-info	1.8	

パッケージ	AL2 コンテナ	AL2023 コンテナ
sqlite	3.7.17	
sqlite-libs		3.40.0
system-release	2	2023.4.20240513
tzdata	2024年a	2024年a
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

AL1 と AL2023 の比較

以下のトピックでは、AL2 との比較でまだカバーされていない AL1 と AL2023 [AL2](#) の主な違いについて説明します。

Note

AL1 は 2023 年 12 月 31 日に end-of-life (EOL) に到達し、2024 年 1 月 1 日以降、セキュリティ更新プログラムやバグ修正プログラムを受け取りません。AL1 EOL とメンテナンスサポートの詳細については、ブログ記事「[Amazon Linux AMI での更新 end-of-life](#)」を参照してください。アプリケーションを AL2023 にアップグレードすることをお勧めします。これには 2028 年までの長期サポートが含まれます。

トピック

- [各リリースのサポート](#)
- [init システムとして systemd が upstart を置き換えます。](#)
- [Python 2.6 および 2.7 は Python 3 に置き換えられました](#)
- [最も古い JDK としての OpenJDK 8](#)
- [Amazon リナックス 1 \(AL1\) からの AL2023 カーネルの変更](#)
- [Amazon Linux 1 \(AL1\) と Amazon Linux 2023 AMI にインストールされているパッケージの比較](#)
- [Amazon Linux 1 \(AL1\) と Amazon Linux 2023 最小 AMI にインストールされているパッケージの比較](#)
- [Amazon Linux 1 \(AL1\) と Amazon Linux 2023 ベースコンテナイメージにインストールされているパッケージの比較](#)

各リリースのサポート

AL2023 では、リリース日から 5 年間のサポートを提供しています。AL1 は 2020 年 12 月 31 日に標準サポートを終了し、2023 年 12 月 31 日にメンテナンスサポートを終了しました。

詳細については、「[リリース頻度](#)」を参照してください。

init システムとして systemd が upstart を置き換えます。

AL2 では、init システム systemd として に置き換え upstart られました。AL2023 は を init システム systemd として使用し、 の新機能をさらに採用しています systemd。

Python 2.6 および 2.7 は Python 3 に置き換えられました

AL1 は 2018.03 リリースで Python 2.6 を EOL としてマークしましたが、パッケージはリポジトリでインストール可能のままでした。AL2 には、サポートされている最も古い Python バージョンとして Python 2.7 が付属しており、AL2023 は Python 3 への移行を完了します。AL2023 リポジトリには Python 2.x バージョンは含まれていません。

Amazon Linux での Python の詳細については、[AL2023 での Python](#) を参照してください。

最も古い JDK としての OpenJDK 8

AL2023 には、デフォルトで (かつ唯一の) Java 開発キット (JDK) として [Amazon Corretto](#) が含まれています。AL2023 のすべての Java ベースパッケージは で構築されています Amazon Corretto 17。

AL1 では、OpenJDK 1.6.0 (java-1.6.0-openjdk) は最初の 2018 年 3 月リリースで EOL になり、OpenJDK 1.7.0 (java-1.7.0-openjdk) は 2020 年半ばに EOL になりましたが、両方のバージョンが AL1 リポジトリで利用可能でした。AL2023 で利用可能な最も古い OpenJDK バージョンは、 が提供する OpenJDK 8 です Amazon Corretto 8。

Amazon リナックス 1 (AL1) からの AL2023 カーネルの変更

カーネルライブパッチ

AL2023 と AL2 はどちらもカーネルのライブパッチ機能のサポートを追加しています。これにより、Linux カーネルの重要かつ重要なセキュリティ脆弱性に、再起動やダウンタイムなしでパッチを適用できます。詳細については、「[AL2023 でのカーネルライブパッチ](#)」を参照してください。

カーネルファイルシステムのサポート

AL1 のカーネルがマウントをサポートするファイルシステムにはいくつかの変更が加えられ、カーネルが解析するパーティションスキームも変更されました。

CONFIG オプション	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
<u>CONFIG_AFS_FS</u>	m	n	n
<u>CONFIG_AF_RXRPC</u>	m	n	n
<u>CONFIG_BSD_DISKLABEL</u>	y	n	n
<u>CONFIG_CRAMFS</u>	m	n	n
<u>CONFIG_CRAMFS_BLOCKDEV</u>	該当なし	該当なし	該当なし
<u>CONFIG_DM_CLONE</u>	該当なし	n	n
<u>CONFIG_DM_ERA</u>	n	n	n
<u>CONFIG_DM_INTEGRITY</u>	m	m	m
<u>CONFIG_DM_LOG_WRITES</u>	n	m	m
<u>CONFIG_DM_SWITCH</u>	n	n	n
<u>CONFIG_DM_VERITY</u>	n	n	n
<u>CONFIG_ECRYPT_FS</u>	m	n	n
<u>CONFIG_EXFAT_FS</u>	該当なし	m	m
<u>CONFIG_EXT2_FS</u>	m	n	n
<u>CONFIG_EXT3_FS</u>	m	n	n
<u>CONFIG_GFS2_FS</u>	n	n	n

CONFIG オプション	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
<u>CONFIG_HF SPLUS_FS</u>	m	n	n
<u>CONFIG_HFS_FS</u>	m	n	n
<u>CONFIG_JFS_FS</u>	m	n	n
<u>CONFIG_LD M_PARTITION</u>	y	n	n
<u>CONFIG_MA C_PARTITION</u>	y	n	n
<u>CONFIG_NFS_V2</u>	m	n	n
<u>CONFIG_NTFS_FS</u>	m	n	n
<u>CONFIG_ROMFS_FS</u>	m	n	n
<u>CONFIG_S0 LARIS_X86 _PARTITION</u>	y	n	n
<u>CONFIG_SQ UASHFS_ZSTD</u>	y	y	y
<u>CONFIG_SU N_PARTITION</u>	y	n	n

セキュリティ重視のカーネル設定の変更

CONFIG オプション	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
<u>CONFIG_BU G_ON_DATA _CORRUPTION</u>	y	y	y

CONFIG オプション	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
<u>CONFIG_DE FAULT_MMA P_MIN_ADDR</u>	4096	65536	65536
<u>CONFIG_DEVMEM</u>	y	n	n
<u>CONFIG_DEVPOR T</u>	y	n	n
<u>CONFIG_FO RTIFY_SOURCE</u>	y	y	y
<u>CONFIG_HA RDENED_US ERCOPY_FA LLBACK</u>	該当なし	該当なし	該当なし
<u>CONFIG_IN IT_ON_ALL OC_DEFAULT_ON</u>	該当なし	n	n
<u>CONFIG_IN IT_ON_FRE E_DEFAULT_ON</u>	該当なし	n	n
<u>CONFIG_IO MMU_DEFAU LT_DMA_STRICT</u>	該当なし	n	n
<u>CONFIG_LD ISC_AUTOLOAD</u>	y	n	n
<u>CONFIG_SC HED_CORE</u>	該当なし	該当なし	y
<u>CONFIG_SC HED_STACK _END_CHECK</u>	y	y	y

CONFIG オプション	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_SE CURITY_DM ESG_RESTRICT	n	y	y
CONFIG_SE CURITY_SE LINUX_DISABLE	y	n	n
CONFIG_SH UFFLE_PAG E_ALLOCATOR	該当なし	y	y
CONFIG_SL AB_FREELI ST_HARDENED	y	y	y
CONFIG_SL AB_FREELI ST_RANDOM	n	y	y

その他のカーネル設定の変更

CONFIG オプション	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_HZ	250	100	100
CONFIG_NR_CPUS	8192	512	512
CONFIG_PA NIC_ON_OOPS	n	y	y
CONFIG_PA NIC_ON_00 PS_VALUE	0	1	1

CONFIG オプション	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_PPP	m	n	n
CONFIG_SLIP	m	n	n
CONFIG_XEN_PV	y	該当なし	n

Amazon Linux 1 (AL1) と Amazon Linux 2023 AMI にインストールされているパッケージの比較

AL1 および AL2023 標準 AMIs に存在する RPMs の比較。

パッケージ	AL1 AMI	AL2023 AMI
acl	2.2.49	2.3.1
acpid	2.0.19	2.0.32
alsa-lib	1.0.22	
alternatives		1.15
amazon-chroney-config		4.3
amazon-ec2-net-utils		2.4.1
amazon-linux-repo-s3		2023.4.20240513
amazon-linux-sb-keys		2023.1
amazon-rpm-config		228
amazon-ssm-agent	3.2.2222.0	3.3.380.0
at	3.1.10	3.1.23
attr	2.4.46	2.5.1

パッケージ	AL1 AMI	AL2023 AMI
audit	2.6.5	3.0.6
audit-libs	2.6.5	3.0.6
authconfig	6.2.8	
aws-amitools-ec2	1.5.13	
aws-cfn-bootstrap	1.4	2.0
aws-cli	1.18.107	
awscli-2		2.15.30
basesystem	10.0	11
bash	4.2.46	5.2.15
bash-completion		2.11
bc	1.06.95	1.07.1
bind-libs	9.8.2	9.16.48
bind-license		9.16.48
bind-utils	9.8.2	9.16.48
binutils	2.27	2.39
boost-filesystem		1.75.0
boost-system		1.75.0
boost-thread		1.75.0
bzip2	1.0.6	1.0.8
bzip2-libs	1.0.6	1.0.8

パッケージ	AL1 AMI	AL2023 AMI
ca-certificates	2023 年 2 月 62 日	2023 年 2 月 64 日
c-ares		1.19.0
checkpolicy	2.1.10	3.4
chkconfig	1.3.49.3	1.15
chrony		4.3
cloud-disk-utils	0.27	
cloud-init	0.7.6	22.2.2
cloud-init-cfg-ec2		22.2.2
cloud-utils-growpart		0.31
copy-jdk-configs	3.3	
coreutils	8.22	8.32
coreutils-common		8.32
cpio	2.10	2.13
cracklib	2.8.16	2.9.6
cracklib-dicts	2.8.16	2.9.6
cronie	1.4.4	
cronie-anacron	1.4.4	
crontabs	1.10	1.11
crypto-policies		20220428
crypto-policies-scripts		20220428

パッケージ	AL1 AMI	AL2023 AMI
cryptsetup	1.6.7	2.6.1
cryptsetup-libs	1.6.7	2.6.1
curl	7.61.1	
curl-minimal		8.5.0
cyrus-sasl	2.1.23	
cyrus-sasl-lib	2.1.23	2.1.27
cyrus-sasl-plain	2.1.23	2.1.27
dash	0.5.5.1	
db4	4.7.25	
db4-utils	4.7.25	
dbus	1.6.12	1.12.28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.6.12	1.12.28
dejavu-fonts-common	2.33	
dejavu-sans-fonts	2.33	
dejavu-serif-fonts	2.33	
device-mapper	1.02.135	1.02.185
device-mapper-event	1.02.135	
device-mapper-event-libs	1.02.135	

パッケージ	AL1 AMI	AL2023 AMI
device-mapper-libs	1.02.135	1.02.185
device-mapper-persistent-data	0.6.3	
dhclient	4.1.1	
dhcp-common	4.1.1	
diffutils	3.3	3.8
dmraid	1.0.0.rc16	
dmraid-events	1.0.0.rc16	
dnf		4.14.0
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0
dnf-plugin-support-info		1.2
dnf-utils		4.3.0
dosfstools		4.2
dracut	004	055
dracut-config-ec2		3.0
dracut-config-generic		055

パッケージ	AL1 AMI	AL2023 AMI
dracut-modules-gro wroot	0.20	
dump	0.4	
dwz		0.14
dyninst		10.2.1
e2fsprogs	1.43.5	1.46.5
e2fsprogs-libs	1.43.5	1.46.5
ec2-hibinit-agent	1.0.0	1.0.8
ec2-instance-connect		1.1
ec2-instance-conne ct-selinux		1.1
ec2-net-utils	0.7	
ec2-utils	0.7	2.2.0
ed	1.1	1.14.2
efi-filesystem		5
efi-srpm-macros		5
efivar		38
efivar-libs		38
elfutils-debuginfod- client		0.188
elfutils-default-y ama-scope		0.188

パッケージ	AL1 AMI	AL2023 AMI
elfutils-libelf	0.168	0.188
elfutils-libs		0.188
epel-release	6	
ethtool	3.15	5.15
expat	2.1.0	2.5.0
file	5.37	5.39
file-libs	5.37	5.39
filesystem	2.4.30	3.14
findutils	4.4.2	4.8.0
fipscheck	1.3.1	
fipscheck-lib	1.3.1	
fontconfig	2.8.0	
fontpackages-files system	1.41	
fonts-srpm-macros		2.0.5
freetype	2.3.11	
fstrm		0.6.1
fuse-libs	2.9.4	2.9.9
gawk	3.1.7	5.1.0
gdbm	1.8.0	
gdbm-libs		1.19

パッケージ	AL1 AMI	AL2023 AMI
gdisk	0.8.10	1.0.8
generic-logos	17.0.0	
get_reference_source	1.2	
gettext		0.21
gettext-libs		0.21
ghc-srpm-macros		1.5.0
giflib	4.1.6	
glib2	2.36.3	2.74.7
glibc	2.17	2.34
glibc-all-langpacks		2.34
glibc-common	2.17	2.34
glibc-gconv-extra		2.34
glibc-locale-source		2.34
gmp	6.0.0	6.2.1
gnupg2	2.0.28	
gnupg2-minimal		2.3.7
gnutls		3.8.0
go-srpm-macros		3.2.0
gpgme	1.4.3	1.15.1
gpm-libs	1.20.6	1.20.7

パッケージ	AL1 AMI	AL2023 AMI
grep	2.20	3.8
groff	1.22.2	
groff-base	1.22.2	1.22.4
grub	0.97	
grub2-common		2.06
grub2-efi-x64-ec2		2.06
grub2-pc-modules		2.06
grub2-tools		2.06
grub2-tools-minimal		2.06
grubby	7.0.15	8.40
gssproxy		0.8.4
gzip	1.5	1.12
hesiod	3.1.0	
hibagent	1.0.0	
hmaccalc	0.9.12	
hostname		3.23
hunspell		1.7.0
hunspell-en		0.20140811.1
hunspell-en-GB		0.20140811.1
hunspell-en-US		0.20140811.1

パッケージ	AL1 AMI	AL2023 AMI
hunspell-filesystem		1.7.0
hwdata	0.233	0.353
info	5.1	6.7
inih		49
initscripts	9.03.58	10.09
iproute	4.4.0	5.10.0
iptables	1.4.21	
iputils	20121221	20210202
irqbalance	1.5.0	1.9.0
jansson		2.14
java-1.7.0-openjdk	1.7.0.321	
javapackages-tools	0.9.1	
jitterentropy		3.4.1
jpackage-utils	1.7.5	
jq		1.7.1
json-c		0.14
kbd	1.15	2.4.0
kbd-misc	1.15	2.4.0
kernel	4.14.336	6.1.90
kernel-livepatch-r epo-s3		2023.4.20240513

パッケージ	AL1 AMI	AL2023 AMI
kernel-srpm-macros		1.0
kernel-tools	4.14.336	6.1.90
keyutils	1.5.8	1.6.3
keyutils-libs	1.5.8	1.6.3
kmod	14	29
kmod-libs	14	29
kpartx	0.4.9	
kpatch-runtime		0.9.7
krb5-libs	1.15.1	1.21
lcms2	2.6	
less	436	608
libacl	2.2.49	2.3.1
libaio	0.3.109	0.3.111
libarchive		3.5.3
libargon2		20171227
libassuan	2.0.3	2.5.5
libattr	2.4.46	2.5.1
libbasicobjects		0.1.1
libblkid	2.23.2	2.37.4
libcap	2.16	2.48

パッケージ	AL1 AMI	AL2023 AMI
libcap54	2.54	
libcap-ng	0.7.5	0.8.2
libcbor		0.7.0
libcgroup	0.40.rc1	
libcollection		0.7.0
libcom_err	1.43.5	1.46.5
libcomps		0.1.20
libconfig		1.7.2
libcurl	7.61.1	
libcurl-minimal		8.5.0
libdb		5.3.28
libdhash		0.5.0
libdnf		0.69.0
libeconf		0.4.0
libedit	2.11	3.1
libev		4.33
libevent	2.0.21	2.1.12
libfdisk		2.37.4
libffi	3.0.13	3.4.4
libfido2		1.10.0

パッケージ	AL1 AMI	AL2023 AMI
libfontenc	1.0.5	
libgcc		11.4.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.11	1.42
libgssglue	0.1	
libibverbs		48.0
libICE	1.0.6	
libicu	50.2	
libidn	1.18	
libidn2	2.3.0	2.3.2
libini_config		1.3.1
libjpeg-turbo	1.2.90	
libkcapi		1.4.0
libkcapi-hmaccalc		1.4.0
libldb		2.6.2
libmaxminddb		1.5.2
libmetalink		0.1.3
libmnl	1.0.3	1.0.4

パッケージ	AL1 AMI	AL2023 AMI
libmodulemd		2.13.0
libmount	2.23.2	2.37.4
libnetfilter_contrack	1.0.4	
libnfnetworking	1.0.1	
libnfsidmap	0.25	2.5.4
libnghttp2	1.33.0	1.59.0
libnih	1.0.1	
libnl	1.1.4	
libnl3		3.5.0
libpath_utils		0.2.1
libpcap		1.10.1
libpipeline	1.2.3	1.5.3
libpkgconf		1.8.0
libpng	1.2.49	
libpsl	0.6.2	0.21.1
libpwquality	1.2.3	1.4.4
libref_array		0.1.5
librepo		1.14.5
libreport-filesystem		2.15.2
libseccomp		2.5.3

パッケージ	AL1 AMI	AL2023 AMI
libselinux	2.1.10	3.4
libselinux-utils	2.1.10	3.4
libsemanage	2.1.6	3.4
libsepol	2.1.7	3.4
libsigsegv		2.13
libSM	1.2.1	
libsmartcols	2.23.2	2.37.4
libsolv		0.7.22
libss	1.43.5	1.46.5
libssh2	1.4.2	
libsss_certmap		2.9.4
libsss_idmap		2.9.4
libsss_nss_idmap		2.9.4
libsss_sudo		2.9.4
libstdc++		11.4.1
libstdc++72	7.2.1	
libstoragemgmt		1.9.4
libsysfs	2.1.0	
libtalloc		2.3.4
libtasn1	2.3	4.19.0

パッケージ	AL1 AMI	AL2023 AMI
libtdb		1.4.7
libtevent		0.13.0
libtextstyle		0.21
libtirpc	0.2.4	1.3.3
libudev	173	
libunistring	0.9.3	0.9.10
libuser	0.60	0.63
libutempter	1.1.5	1.2.1
libuuid	2.23.2	2.37.4
libuv		1.47.0
libverto	0.2.5	0.3.2
libverto-libev		0.3.2
libX11	1.6.0	
libX11-common	1.6.0	
libXau	1.0.6	
libxcb	1.11	
libXcomposite	0.4.3	
libxcrypt		4.4.33
libXext	1.3.2	
libXfont	1.4.5	

パッケージ	AL1 AMI	AL2023 AMI
libXi	1.7.2	
libxml2	2.9.1	2.10.4
libxml2-python27	2.9.1	
libXrender	0.9.8	
libxslt	1.1.28	
libXtst	1.2.2	
libyaml	0.1.6	0.2.5
libzstd		1.5.5
lm_sensors-libs		3.6.0
lmdb-libs		0.9.29
log4j-cve-2021-44228-hotpatch	1.3	
logrotate	3.7.8	3.20.1
lsof	4.82	4.94.0
lua	5.1.4	
lua-libs		5.4.4
lua-srpm-macros		1
lvm2	2.02.166	
lvm2-libs	2.02.166	
lz4-libs		1.9.4
mailcap	2.1.31	

パッケージ	AL1 AMI	AL2023 AMI
make	3.82	
man-db	2.6.3	2.9.3
man-pages	4.10	5.10
mdadm	3.2.6	
microcode_ctl	2.1	2.1
mingetty	1.08	
mpfr		4.1.0
nano	2.5.3	5.8
nc	1.84	
ncurses	5.7	6.2
ncurses-base	5.7	6.2
ncurses-libs	5.7	6.2
nettle		3.8
net-tools	1.60	2.0
newt	0.52.11	0.52.21
newt-python27	0.52.11	
nfs-utils	1.3.0	2.5.4
npth		1.6
nspr	4.25.0	4.35.0
nss	3.53.1	3.90.0

パッケージ	AL1 AMI	AL2023 AMI
nss-pem	1.0.3	
nss-softokn	3.53.1	3.90.0
nss-softokn-freebl	3.53.1	3.90.0
nss-sysinit	3.53.1	3.90.0
nss-tools	3.53.1	
nss-util	3.53.1	3.90.0
ntp	4.2.8p15	
ntpdate	4.2.8p15	
ntsysv	1.3.49.3	1.15
numactl	2.0.7	
numactl-libs		2.0.14
ocaml-srpm-macros		6
oniguruma		6.9.7.1
openblas-srpm-macros		2
openldap	2.4.40	2.4.57
openssh	7.4p1	8.7p1
openssh-clients	7.4p1	8.7p1
openssh-server	7.4p1	8.7p1
openssl	1.0.2k	3.0.8
openssl-libs		3.0.8

パッケージ	AL1 AMI	AL2023 AMI
openssl-pkcs11		0.4.12
os-prober		1.77
p11-kit	0.18.5	0.24.1
p11-kit-trust	0.18.5	0.24.1
package-notes-srpm-macros		0.4
pam	1.1.8	1.5.1
pam_ccreds	10	
pam_krb5	2.3.11	
pam_passwdqc	1.0.5	
parted	2.1	3.4
passwd	0.79	0.80
pciutils	3.1.10	3.7.0
pciutils-libs	3.1.10	3.7.0
pcre	8.21	
pcre2		10.40
pcre2-syntax		10.40
perl	5.16.3	
perl-Carp	1.26	1.50
perl-Class-Struct		0.66
perl-constant	1.27	1.33

パッケージ	AL1 AMI	AL2023 AMI
perl-Digest	1.17	
perl-Digest-HMAC	1.03	
perl-Digest-MD5	2.52	
perl-Digest-SHA	5.85	
perl-DynaLoader		1.47
perl-Encode	2.51	3.15
perl-Errno		1.30
perl-Exporter	5.68	5.74
perl-Fcntl		1.13
perl-File-Basename		2.85
perl-File-Path	2.09	2.18
perl-File-stat		1.09
perl-File-Temp	0.23.01	0.231.100
perl-Filter	1.49	
perl-Getopt-Long	2.40	2.52
perl-Getopt-Std		1.12
perl-HTTP-Tiny	0.033	0.078
perl-if		0.60.800
perl-interpreter		5.32.1
perl-IO		1.43

パッケージ	AL1 AMI	AL2023 AMI
perl-IPC-Open3		1.21
perl-libs	5.16.3	5.32.1
perl-macros	5.16.3	
perl-MIME-Base64		3.16
perl-mro		1.23
perl-overload		1.31
perl-overloading		0.02
perl-parent	0.225	0.238
perl-PathTools	3.40	3.78
perl-Pod-Escapes	1.04	1.07
perl-podlators	2.5.1	4.14
perl-Pod-Perldoc	3.20	3.28.01
perl-Pod-Simple	3.28	3.42
perl-Pod-Usage	1.63	2.01
perl-POSIX		1.94
perl-Scalar-List-Utils	1.27	1.56
perl-SelectSaver		1.02
perl-Socket	2.010	2.032
perl-srpm-macros		1
perl-Storable	2.45	3.21

パッケージ	AL1 AMI	AL2023 AMI
perl-subvars		1.03
perl-Symbol		1.08
perl-Term-ANSIColor		5.01
perl-Term-Cap		1.17
perl-Text-ParseWords	3.29	3.30
perl-Text-Tabs+Wrap		2021.0726
perl-threads	1.87	
perl-threads-shared	1.43	
perl-Time-HiRes	1.9725	
perl-Time-Local	1.2300	1.300
perl-vars		1.05
pinentry	0.7.6	
pkgconf		1.8.0
pkgconfig	0.27.1	
pkgconf-m4		1.8.0
pkgconf-pkg-config		1.8.0
pm-utils	1.4.1	
policycoreutils	2.1.12	3.4
policycoreutils-python-utils		3.4
popt	1.13	1.18

パッケージ	AL1 AMI	AL2023 AMI
procmail	3.22	
procps	3.2.8	
procps-ng		3.3.17
protobuf-c		1.4.1
psacct	6.3.2	6.6.4
psmisc	22.20	23.4
pth	2.0.7	
publicsuffix-list-dafsa		20240212
python27	2.7.18	
python27-babel	0.9.4	
python27-backports	1.0	
python27-backports-ssl_match_hostname	3.4.0.2	
python27-boto	2.48.0	
python27-botocore	1.17.31	
python27-chardet	2.0.1	
python27-colorama	0.4.1	
python27-configobj	4.7.2	
python27-crypto	2.6.1	
python27-daemon	1.5.2	

パッケージ	AL1 AMI	AL2023 AMI
python27-dateutil	2.1	
python27-devel	2.7.18	
python27-docutils	0.11	
python27-ecdsa	0.11	
python27-futures	3.0.3	
python27-imaging	1.1.6	
python27-iniparse	0.3.1	
python27-jinja2	2.7.2	
python27-jmespath	0.9.2	
python27-jsonpatch	1.2	
python27-jsonpointer	1.0	
python27-kitchen	1.1.1	
python27-libs	2.7.18	
python27-lockfile	0.8	
python27-markupsafe	0.11	
python27-paramiko	1.15.1	
python27-pip	9.0.3	
python27-ply	3.4	
python27-pyasn1	0.1.7	
python27-pycurl	7.19.0	

パッケージ	AL1 AMI	AL2023 AMI
python27-pygpme	0.3	
python27-pyliblzma	0.5.3	
python27-pystache	0.5.3	
python27-pyxattr	0.5.0	
python27-PyYAML	3.10	
python27-requests	1.2.3	
python27-rsa	3.4.1	
python27-setuptools	36.2.7	
python27-simplejson	3.6.5	
python27-six	1.8.0	
python27-urlgrabber	3.10	
python27-urllib3	1.24.3	
python27-virtualenv	15.1.0	
python3		3.9.16
python3-attrs		20.3.0
python3-audit		3.0.6
python3-awscli		0.19.19
python3-babel		2.9.1
python3-cffi		1.14.5
python3-chardet		4.0.0

パッケージ	AL1 AMI	AL2023 AMI
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36.0.1
python3-daemon		2.3.0
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils		0.16
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-idna		2.10
python3-jinja2		2.11.3
python3-jmespath		0.10.0
python3-jsonpatch		1.21
python3-jsonpointer		2.0
python3-jsonschemata		3.2.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0

パッケージ	AL1 AMI	AL2023 AMI
python3-libs		3.9.16
python3-libselinux		3.4
python3-libsemanage		3.4
python3-libstorage mgmt		1.9.4
python3-lockfile		0.12.2
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip-wheel		21.3.1
python3-ply		3.11
python3-policycore utils		3.4
python3-prettytable		0.7.2
python3-prompt-too lkit		3.0.24
python3-pycparser		2.20
python3-pyrsistent		0.17.3
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pytz		2022.7.1

パッケージ	AL1 AMI	AL2023 AMI
python3-pyyaml		5.4.1
python3-requests		2.25.1
python3-rpm		4.16.1.3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools		59.6.0
python3-setuptools-wheel		59.6.0
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1.25.10
python3-wcwidth		0.2.5
python-chevront		0.13.1
python-srpm-macros		3.9
quota	4.00	4.06
quota-nls	4.00	4.06
readline	6.2	8.1
rmt	0.4	
rng-tools	5	6.14

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

パッケージ	AL1 AMI	AL2023 AMI
rust-srpm-macros		21
sbsigntools		0.9.4
screen	4.0.3	4.8.0
sed	4.2.1	4.8
selinux-policy		37.22
selinux-policy-targeted		37.22
sendmail	8.14.4	
setserial	2.17	
setup	2.8.14	2.13.7
sgpio	1.2.0.10	
shadow-utils	4.1.4.2	4.9
shared-mime-info	1.1	
slang	2.2.1	2.3.2
sqlite	3.7.17	
sqlite-libs		3.40.0
sssd-client		2.9.4
sssd-common		2.9.4
sssd-kcm		2.9.4
sssd-nfs-idmap		2.9.4
strace		6.8

パッケージ	AL1 AMI	AL2023 AMI
sudo	1.8.23	1.9.15
sysctl-defaults	1.0	1.0
sysfsutils	2.1.0	
sysstat		12.5.6
systemd		252.16
systemd-libs		252.16
systemd-networkd		252.16
systemd-pam		252.16
systemd-resolved		252.16
systemd-udev		252.16
system-release	2018 年 3 月	2023.4.20240513
systemtap-runtime		4.8
sysvinit	2.87	
tar	1.26	1.34
tbb		2020 年 3 月
tcp_wrappers	7.6	
tcp_wrappers-libs	7.6	
tcpdump		4.99.1
tcsh		6.24.07
time	1.7	1.9

パッケージ	AL1 AMI	AL2023 AMI
tmpwatch	2.9.16	
traceroute	2.0.14	2.1.3
ttmkfdir	3.0.9	
tzdata	2023 年	2024年a
tzdata-java	2023 年	
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

パッケージ	AL1 AMI	AL2023 AMI
words	3.0	3.0
xfsdump		3.1.11
xfsprogs		5.18.0
xorg-x11-fonts-Type1	7.2	
xorg-x11-font-utils	7.2	
xxd	9.0.2120	9.0.2153
xxhash-libs		0.8.0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	1.1.4	
yum-plugin-priorities	1.1.31	
yum-plugin-upgrade-helper	1.1.31	
yum-utils	1.1.31	
zip	3.0	3.0
zlib	1.2.8	1.2.11
zram-generator		1.1.2
zram-generator-defaults		1.1.2

パッケージ	AL1 AMI	AL2023 AMI
zstd		1.5.5

Amazon Linux 1 (AL1) と Amazon Linux 2023 最小 AMI にインストールされているパッケージの比較

AL1 および AL2023 最小 AMIs に存在する RPMs の比較。

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

パッケージ	AL1 最小	AL2023 最小
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023 年 2 月 62 日	2023 年 2 月 64 日
checkpolicy	2.1.10	3.4
chkconfig	1.3.49.3	
chrony		4.3
cloud-disk-utils	0.27	
cloud-init	0.7.6	22.2.2
cloud-init-cfg-ec2		22.2.2
cloud-utils-growpart		0.31
coreutils	8.22	8.32
coreutils-common		8.32
cpio	2.10	2.13
cracklib	2.8.16	2.9.6
cracklib-dicts	2.8.16	2.9.6
cronie	1.4.4	
cronie-anacron	1.4.4	
crontabs	1.10	
crypto-policies		20220428
cryptsetup-libs		2.6.1
curl	7.61.1	

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

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

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

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

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

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

パッケージ	AL1 最小	AL2023 最小
libfido2		1.10.0
libgcc		11.4.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.11	1.42
libicu	50.2	
libidn	1.18	
libidn2	2.3.0	2.3.2
libkcapi		1.4.0
libkcapi-hmaccalc		1.4.0
libmnl	1.0.3	1.0.4
libmodulemd		2.13.0
libmount	2.23.2	2.37.4
libnetfilter_contrack	1.0.4	
libnfnetlink	1.0.1	
libnhttp2	1.33.0	1.59.0
libnih	1.0.1	
libpipeline		1.5.3
libpsl	0.6.2	0.21.1

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

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

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

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

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

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

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

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

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

パッケージ	AL1 最小	AL2023 最小
setserial	2.17	
setup	2.8.14	2.13.7
shadow-utils	4.1.4.2	4.9
shared-mime-info	1.1	
slang	2.2.1	
sqlite	3.7.17	
sqlite-libs		3.40.0
sudo	1.8.23	1.9.15
sysctl-defaults	1.0	1.0
sysfsutils	2.1.0	
systemd		252.16
systemd-libs		252.16
systemd-networkd		252.16
systemd-pam		252.16
systemd-resolved		252.16
systemd-udev		252.16
system-release	2018 年 3 月	2023.4.20240513
sysvinit	2.87	
tar	1.26	1.34
tcp_wrappers-libs	7.6	

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

パッケージ	AL1 最小	AL2023 最小
zram-generator		1.1.2
zram-generator-def aults		1.1.2
zstd		1.5.5

Amazon Linux 1 (AL1) と Amazon Linux 2023 ベースコンテナイメージにインストールされているパッケージの比較

AL1 および AL2023 ベースコンテナイメージに存在する RPMs の比較。

パッケージ	AL1 コンテナ	AL2023 コンテナ
alternatives		1.15
amazon-linux-repo- cdn		2023.4.20240513
audit-libs		3.0.6
basesystem	10.0	11
bash	4.2.46	5.2.15
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023 年 2 月 62 日	2023 年 2 月 64 日
chkconfig	1.3.49.3	
coreutils	8.22	
coreutils-single		8.32
crypto-policies		20220428

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

パッケージ	AL1 コンテナ	AL2023 コンテナ
glibc-minimal-langpack		2.34
gmp	6.0.0	6.2.1
gnupg2	2.0.28	
gnupg2-minimal		2.3.7
gpgme	1.4.3	1.15.1
grep	2.20	3.8
gzip	1.5	
info	5.1	
json-c		0.14
keyutils-libs	1.5.8	1.6.3
krb5-libs	1.15.1	1.21
libacl	2.2.49	2.3.1
libarchive		3.5.3
libassuan	2.0.3	2.5.5
libattr	2.4.46	2.5.1
libblkid		2.37.4
libcap	2.16	2.48
libcap-ng		0.8.2
libcom_err	1.43.5	1.46.5
libcomps		0.1.20

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

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

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

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

パッケージ	AL1 コンテナ	AL2023 コンテナ
python3-libs		3.9.16
python3-pip-wheel		21.3.1
python3-rpm		4.16.1.3
python3-setuptools-wheel		59.6.0
readline	6.2	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-python27	4.11.3	
rpm-sign-libs		4.16.1.3
sed	4.2.1	4.8
setup	2.8.14	2.13.7
shared-mime-info	1.1	
sqlite	3.7.17	
sqlite-libs		3.40.0
sysctl-defaults	1.0	
system-release	2018 年 3 月	2023.4.20240513
tar	1.26	
tzdata	2023 年	2024年a
xz-libs	5.2.2	5.2.5

パッケージ	AL1 コンテナ	AL2023 コンテナ
yum	3.4.3	4.14.0
yum-metadata-parser	1.1.4	
yum-plugin-ovl	1.1.31	
yum-plugin-priorities	1.1.31	
yum-utils	1.1.31	
zlib	1.2.8	1.2.11

AL2023 システム要件

このセクションでは、AL2023 を使用するためのシステム要件について説明します。

トピック

- [AL2023 を実行するための CPU 要件](#)
- [AL2023 を実行するためのメモリ \(RAM\) 要件](#)

AL2023 を実行するための CPU 要件

AL2023 コードを実行するには、使用するプロセッサが特定の最小要件を満たしている必要があります。これらの要件を満たしていない CPU で AL2023 を実行しようとすると、コード実行の非常に早い段階で不正な命令エラーが発生する可能性があります。

最小要件は[Amazon EC2 での AL2023](#)、[コンテナでの AL2023](#)およびに適用されます。[Amazon EC2 以外の AL2023](#)

AL2023 の ARM CPU 要件

AL2023 aarch64 (ARM) バイナリはすべて 64 ビット用に構築されています。32 ARM ビットのバイナリは用意されていないため、64 ビット CPU が必要です。ARM

Note

ARM ベースのインスタンスの場合、AL2023 は Graviton2 以降のプロセッサを使用するインスタンスタイプのみをサポートします。AL2023 は A1 インスタンスをサポートしていません。

AL2023 には、暗号化拡張 (ARMv8.2+crypto) を備えた ARMv8.2 準拠のプロセッサが必要です。の AL2023 パッケージはすべて、aarch64コンパイラフラグを使用してビルドされます。-march=armv8.2-a+cryptoAL2023 ARM コードを古いプロセッサで実行しようとすると、わかりやすいエラーメッセージを表示するように努めていますが、最初のエラーメッセージが不正な命令エラーである可能性もあります。

Note

AL2023 aarch64 のベース CPU 要件のため、Raspberry Pi以前のすべてのシステムは CPU Raspberry Pi 5 の最小要件を満たしていません。

AL2023 の x86-64 CPU 要件

AL2023 x86-64 のすべてのバイナリは、x86-64v2x86-64コンパイラーに渡すことでアーキテクチャのリビジョンに合わせてビルドされます。-march=x86-64-v2

x86-64v2アーキテクチャの改訂では、ベースラインアーキテクチャに加えて次の CPU 機能が追加されています。x86-64

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

これは 2009 x86-64 年以降にリリースされたプロセッサとほぼ同等です。例としては、Intel Nehalem、AMD Jaguar、Atom Silvermont、、、VIA NanoEden C やマイクロアーキテクチャなどがあります。

Amazon EC2 では、すべての x86-64 インスタンスタイプは M1、C1、M2 インスタンスファミリーを含む x86-64v2 をサポートします。

32 ビット x86 (i686) AL2023 バイナリはビルドされていません。AL2023 は 32 ビットユーザースペースバイナリの実行を引き続きサポートしていますが、この機能は廃止され、Amazon Linux の future メジャーバージョンで削除される可能性があります。詳細については、「[32 ビット x86 \(i686\) パッケージ](#)」を参照してください。

AL2023 を実行するためのメモリ (RAM) 要件

Amazon EC2 .nano ファミリーのインスタンスタイプ (t2.nano、t3.nano、およびt4g.nano) にはt3a.nano、AL2023 の最小要件である 512 MB の RAM が搭載されています。

Note

512 MB が最小要件ですが、これらのインスタンスタイプはメモリに制約があり、機能やパフォーマンスにも制限がある場合があります。

AL2023 イメージは、RAM が 512 MB 未満のシステムではテストされていません。AL2023 ベースのコンテナイメージを 512 MB 未満の RAM で実行できるかどうかは、コンテナ化されたワークロードによって異なります。

一部の AL2023 dnf update リリースの間など、一部のワークロードでは 512 MB を超える RAM が必要になる場合があります。このため、[AL2023.3](#) リリースでは、RAM が 800 MB zram 未満のインスタンスをデフォルトで有効化するようになりました。つまり、コンテナ化されたワークロードでは、このメモリ量の AL2023 インスタンスでは正常に動作するが、この量のメモリ使用量に制限されたコンテナで実行すると失敗するワークロードがあるということです。

RAM が 800 MB 未満のインスタンスタイプでは、AL2023 ([AL2023.3](#) 以降) はデフォルトで zram ベーススワップを有効にします。メモリが 800 MB 未満の Amazon EC2 インスタンスタイプの例にはt4g.nano、t3a.nano、t3.nano、t2.nano、などがありますt1.micro。AL2023 はオンデマンドでメモリページの圧縮と解凍を行うため、これらのインスタンスタイプではメモリ不足のシナリオが減少します。これにより、圧縮に必要な CPU 使用量を対価として、より多くのメモリを備えたインスタンスタイプを必要とするワークロードが可能になります。

での AL2023 の使用 AWS

AL2023 は、他の で使用するように設定できます AWS のサービス。例えば、[Amazon Elastic Compute Cloud](#) (Amazon EC2) インスタンスを起動するときに AL2023 AMI を選択できます。

これらのセットアップ手順では、AWS Identity and Access Management (IAM) サービスを使用します。IAM の詳細については、以下の資料を参照してください。

- [AWS Identity and Access Management \(IAM\)](#)
- [IAM ユーザーガイド](#)

トピック

- [の開始方法 AWS](#)
- [Amazon EC2 での AL2023](#)
- [コンテナでの AL2023 の使用](#)
- [AL2023 オン AWS Elastic Beanstalk](#)
- [で AL2023 を使用する AWS CloudShell](#)
- [AL2023 ベースの Amazon ECS AMI を使用してコンテナ化されたワークロードをホストする](#)
- [AL2023 での Amazon Elastic File System 使用](#)
- [AL2023 上に構築された Amazon EMR を使用する](#)
- [で AL2023 を使用する AWS Lambda](#)

の開始方法 AWS

にサインアップする AWS アカウント

がない場合は AWS アカウント、次のステップを実行して作成します。

にサインアップするには AWS アカウント

1. <https://portal.aws.amazon.com/billing/signup> を開きます。
2. オンラインの手順に従います。

サインアップ手順の一環として、通話呼び出しを受け取り、電話キーパッドで検証コードを入力するように求められます。

にサインアップすると AWS アカウント、AWS アカウントのルートユーザーが作成されます。ルートユーザーには、アカウントのすべての AWS のサービス とリソースへのアクセス権があります。セキュリティのベストプラクティスとして、ユーザーに管理アクセスを割り当て、ルートユーザーのみを使用して [ルートユーザーアクセスが必要なタスク](#) を実行してください。

AWS サインアッププロセスが完了すると、 から確認メールが送信されます。 <https://aws.amazon.com/> の [アカウント] をクリックして、いつでもアカウントの現在のアクティビティを表示し、アカウントを管理することができます。

管理アクセスを持つユーザーを作成する

にサインアップしたら AWS アカウント、 を保護し AWS アカウントのルートユーザー、 を有効にして AWS IAM Identity Center、日常的なタスクにルートユーザーを使用しないように管理ユーザーを作成します。

のセキュリティ保護 AWS アカウントのルートユーザー

1. ルートユーザーを選択し、AWS アカウント E メールアドレスを入力して、アカウント所有者 [AWS Management Console](#) として にサインインします。次のページでパスワードを入力します。

ルートユーザーを使用してサインインする方法については、AWS サインイン ユーザーガイドの「[ルートユーザーとしてサインインする](#)」を参照してください。

2. ルートユーザーの多要素認証 (MFA) を有効にします。

手順については、「IAM [ユーザーガイド](#)」の AWS アカウント「[ルートユーザーの仮想 MFA デバイスを有効にする \(コンソール\)](#)」を参照してください。

管理アクセスを持つユーザーを作成する

1. IAM アイデンティティセンターを有効にします。

手順については、「AWS IAM Identity Center ユーザーガイド」の「[AWS IAM Identity Center の有効化](#)」を参照してください。

2. IAM アイデンティティセンターで、ユーザーに管理アクセスを付与します。

を ID ソース IAM アイデンティティセンターディレクトリとして使用する方法的チュートリアルについては、「[ユーザーガイド](#)」の「[デフォルトでユーザーアクセスを設定する IAM アイデンティティセンターディレクトリ](#)」AWS IAM Identity Center」を参照してください。

管理アクセス権を持つユーザーとしてサインインする

- IAM アイデンティティセンターのユーザーとしてサインインするには、IAM アイデンティティセンターのユーザーの作成時に E メールアドレスに送信されたサインイン URL を使用します。

IAM Identity Center ユーザーを使用してサインインする方法については、「[AWS サインインユーザーガイド](#)」の AWS 「[アクセスポータルにサインインする](#)」を参照してください。

追加のユーザーにアクセス権を割り当てる

1. IAM アイデンティティセンターで、最小特権のアクセス許可を適用するというベストプラクティスに従ったアクセス許可セットを作成します。

手順については、「[AWS IAM Identity Center ユーザーガイド](#)」の「[権限設定を作成する](#)」を参照してください。

2. グループにユーザーを割り当て、そのグループにシングルサインオンアクセス権を割り当てます。

手順については、「[AWS IAM Identity Center ユーザーガイド](#)」の「[グループの参加](#)」を参照してください。

プログラムによるアクセス権を付与する

ユーザーが の AWS 外部で を操作する場合は、プログラムによるアクセスが必要です AWS Management Console。プログラムによるアクセスを許可する方法は、 にアクセスするユーザーのタイプによって異なります AWS。

ユーザーにプログラマチックアクセス権を付与するには、以下のいずれかのオプションを選択します。

プログラマチックアクセス権を必要とするユーザー	目的	方法
<p>ワークフォースアイデンティティ</p> <p>(IAM Identity Center で管理されているユーザー)</p>	<p>一時的な認証情報を使用して、AWS SDKs AWS CLI、または AWS APIs。</p>	<p>使用するインターフェイス用の手引きに従ってください。</p> <ul style="list-style-type: none"> • については AWS CLI、「ユーザーガイド」の AWS CLI「を使用するための設定 AWS IAM Identity Center AWS Command Line Interface」を参照してください。 • AWS SDKs、ツール、AWS APIs「SDK とツールのリファレンスガイド」の 「IAM Identity Center 認証」を参照してください。 AWS SDKs
IAM	<p>一時的な認証情報を使用して、AWS SDKs AWS CLI、または AWS APIs。</p>	<p>「IAM ユーザーガイド」の 「AWS リソースでの一時的な認証情報の使用」の手順に従います。</p>
IAM	<p>(非推奨)</p> <p>長期認証情報を使用して、AWS SDKs AWS CLI、または AWS APIs。</p>	<p>使用するインターフェイス用の手引きに従ってください。</p> <ul style="list-style-type: none"> • については AWS CLI、「AWS Command Line Interface ユーザーガイド」の 「IAM ユーザー認証情報を使用した認証」を参照してください。 • AWS SDKs「SDK とツールのリファレンスガイド」の 「長期的な認証情報を使

プログラマチックアクセス権を必要とするユーザー	目的	方法
		<p>用した認証」を参照してください。AWS SDKs</p> <ul style="list-style-type: none">• AWS APIs ユーザーガイド」の「IAM ユーザーのアクセスキーの管理」を参照してください。

Amazon EC2 での AL2023

AL2023 AMI で Amazon EC2 インスタンスを起動するには、次のいずれかの手順を使用します。標準 AMI または最小 AMI を選択できます。標準 AMI と最小 AMI の違いの詳細については、「[AL2023 標準 \(デフォルト\) と最小 AMI の比較](#)」を参照してください。

トピック

- [Amazon EC2 コンソールを使用した AL2023 の起動](#)
- [SSM パラメータとを使用した AL2023 の起動 AWS CLI](#)
- [を使用した最新の AL2023 AMI の起動 AWS CloudFormation](#)
- [特定の AMI ID を使用した AL2023 の起動](#)
- [AL2023 AMI の非推奨とライフサイクル](#)
- [AL2023 インスタンスへの接続](#)
- [AL2023 標準 AMI と最小 AMIs の比較](#)

Amazon EC2 コンソールを使用した AL2023 の起動

Amazon EC2 コンソールを使用して、AL2023 AMI を起動します。

Note

ARM ベースのインスタンスの場合、AL2023 は Graviton2 以降のプロセッサを使用するインスタンスタイプのみをサポートします。AL2023 は A1 インスタンスをサポートしていません。

Amazon EC2 コンソールから AL2023 AMI を使用した Amazon EC2 インスタンスを起動するには、以下の手順に従います。

AL2023 AMI を使用して EC2 インスタンスを起動するには AL2023

1. Amazon EC2 コンソール (<https://console.aws.amazon.com/ec2/>) を開きます。
2. ナビゲーションペインで [AMI] を選択します。
3. ドロップダウンメニューから [パブリックイメージ] を選択します。
4. [検索] フィールドに **al2023-ami** を入力します。

Note

[オーナーエイリアス] 列に Amazon が表示されていることを確認します。

5. リストから AMI を選択します。[ソース] で、AMI を標準または最小にするかを決定できます。AL2023 AMI 名は、以下の形式を使用して解釈できます。

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

6. 以下のイメージは、AL2023 AMI のリストの一部を示しています。

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

Amazon EC2 インスタンスの起動の詳細については、[Amazon EC2 ユーザーガイド](#)の「[Amazon EC2 Linux インスタンスの開始方法](#)」を参照してください。Amazon EC2

SSM パラメータと を使用した AL2023 の起動 AWS CLI

では AWS CLI、AMI の SSM パラメータ値を使用して AL2023 の新しいインスタンスを起動できます。具体的には、以下のリストにある動的 SSM パラメータ値のいずれかを使用し、SSM パラメータ value/ の前に /aws/service/ami-amazon-linux-latest/ を追加します。これを使用して、AWS CLI内でインスタンスを起動できます。

- arm64 アーキテクチャの al2023-ami-kernel-default-arm64

- arm64 アーキテクチャ用 (最小 AMI) の `al2023-ami-minimal-kernel-default-arm64`
- x86_64 アーキテクチャの `al2023-ami-kernel-default-x86_64`
- x86_64 アーキテクチャ (最小 AMI) の `al2023-ami-minimal-kernel-default-x86_64`

Note

###各項目はサンプルパラメータです。それらを、ユーザー自身の情報に置き換えます。

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

`--image-id` フラグは SSM パラメータ値を指定します。

`--instance-type` フラグはインスタンスのタイプとサイズを指定します。このフラグは、選択した AMI タイプと互換性がある必要があります。

`--region` フラグは、インスタンスを作成する AWS リージョン を指定します。

`--key-name` フラグは、インスタンスへの接続に使用される の AWS リージョンキーを指定します。インスタンスを作成したリージョンに存在するキーを指定しない場合、SSH を使用してインスタンスに接続することはできません。

`--security-group-ids` フラグは、インバウンドとアウトバウンドのネットワークトラフィックのアクセス権限を決定するセキュリティグループを指定します。

Important

AWS CLI では、ポート 経由でリモートマシンからインスタンスへのアクセスを許可する既存のセキュリティグループを指定する必要がありますTCP:22。セキュリティグループを指定しない場合、新しいインスタンスはデフォルトのセキュリティグループに配置されます。デフォルトのセキュリティグループでは、インスタンスは VPC 内の他のインスタンスとのみ接続できます。

詳細については、「AWS Command Line Interface ユーザーガイド」の「[Amazon EC2 インスタンスの起動、リスト化、終了](#)」を参照してください。

を使用した最新の AL2023 AMI の起動 AWS CloudFormation

を使用して AL2023 AMI を起動するには AWS CloudFormation、次のいずれかのテンプレートを使用します。

Note

x86_64 と Arm64 AMI にはそれぞれ異なるインスタンスタイプが必要です。詳細については、「[Amazon EC2 のインスタンスタイプ](#)」を参照してください。

JSON テンプレート:

```
{
  "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 テンプレート:

```
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
```

必要に応じて、「デフォルト」セクションの末尾にある AMI パラメータを置き換えてください。以下のパラメータ値を使用できます。

- arm64 アーキテクチャの al2023-ami-kernel-6.1-arm64
- arm64 アーキテクチャ用 (最小 AMI) の al2023-ami-minimal-kernel-6.1-arm64
- x86_64 アーキテクチャの al2023-ami-kernel-6.1-x86_64
- x86_64 アーキテクチャ (最小 AMI) の al2023-ami-minimal-kernel-6.1-x86_64

以下は動的カーネルの仕様です。デフォルトのカーネルバージョンは、カーネルバージョンがメジャー更新されるたびに自動的に変更されます。

- arm64 アーキテクチャの al2023-ami-kernel-default-arm64
- arm64 アーキテクチャ用 (最小 AMI) の al2023-ami-minimal-kernel-default-arm64
- x86_64 アーキテクチャの al2023-ami-kernel-default-x86_64
- x86_64 アーキテクチャ (最小 AMI) の al2023-ami-minimal-kernel-default-x86_64

特定の AMI ID を使用した AL2023 の起動

AMI ID を使用して特定の AL2023 AMI を起動できます。Amazon EC2 コンソールの AMI リストを確認することで、どの AL2023 AMI ID が必要かを判断できます。または、[AMI を使用できます AWS Systems Manager](#)。Systems Manager を使用している場合は、前のセクションに記載されているエイリアスから AMI エイリアスを選択します。詳細については、[AWS Systems Manager 「Parameter Store を使用して最新の Amazon Linux AMI IDs」](#)を参照してください。

AL2023 AMI の非推奨とライフサイクル

AL2023 の新しいリリースにはそれぞれ、新しい AMI が含まれています。AMI が登録されると、廃止日が表示されます。各 AL2023 AMI の非推奨日は、[AL2023 でのカーネルライブパッチ](#) が個々のカーネルリリースに設定されている期間に合わせて、リリース日から 90 日です。

Note

90 日間の廃止日は個々の AMI を指し、AL2023 [リリース頻度](#) や製品サポート期間を指すものではありません。

AMI の非推奨の詳細については、Amazon EC2 ユーザーガイド」の「[AMI の非推奨](#)」を参照してください。

更新された AMI を定期的を使用してインスタンスを起動することで、更新されたカーネルを含む最新のセキュリティ更新でインスタンスが起動します。AMI の以前のバージョンを起動して更新を適用した場合、インスタンスに最新のセキュリティ更新が適用されない期間があります。最新の AMI を使用するには、SSM パラメータを使用することをお勧めします。

SSM パラメータを使用したインスタンスの起動方法の詳細については、以下を参照してください。

- [SSM パラメータとを使用した AL2023 の起動 AWS CLI](#)
- [を使用した最新の AL2023 AMI の起動 AWS CloudFormation](#)

AL2023 インスタンスへの接続

SSH または AWS Systems Manager を使用して AL2023 インスタンスに接続します。

SSH を使用したインスタンスへの接続

SSH を使用してインスタンスに接続する方法については、Amazon EC2 [ユーザーガイドの「SSH を使用して Linux インスタンスに接続する」](#)を参照してください。Amazon EC2

を使用してインスタンスに接続する AWS Systems Manager

AWS Systems Manager を使用して AL2023 インスタンスに接続する方法については、Amazon EC2 [ユーザーガイドの「セッションマネージャーを使用して Linux インスタンスに接続する」](#)を参照してください。Amazon EC2

Amazon EC2 Instance Connect の使用

最小 AMI を除く AL2023 AMI には、デフォルトで EC2 Instance Connect エージェントがインストールされています。最小 AMI から起動された AL2023 インスタンスで EC2 Instance Connect を使用するには、`ec2-instance-connect` パッケージをインストールする必要があります。EC2 Instance Connect の使用手順については、「[Amazon EC2 ユーザーガイド](#)」の [EC2 Instance Connect を使用して Linux](#) インスタンスに接続する」を参照してください。Amazon EC2

AL2023 標準 AMI と最小 AMIs の比較

Amazon EC2 インスタンスは、標準 (デフォルト) または最小限の AL2023 AMI を使用して起動できます。標準または最小の AMI タイプで Amazon EC2 インスタンスを起動する方法については、「」を参照してください [Amazon EC2 での AL2023](#)。

標準の AL2023 AMI には、最も一般的に使用されるすべてのアプリケーションとツールがインストールされています。すぐに使い始めたいが AMI のカスタマイズは不要な場合は、標準 AMI をお勧めします。

最小の AL2023 AMI は、オペレーティングシステム (OS) の実行に必要な最も基本的なツールとユーティリティのみを含む、基本で合理化されたバージョンです。OS のフットプリントをできるだけ小さくしたい場合は、最小 AMI を使用することをお勧めします。最小 AMI では、ディスク容量の使用率がわずかに低下し、長期的なコスト効率が向上します。OS を小さくしたいが、ツールやアプリケーションを手動でインストールしてもかまわない場合は、最小 AMI が適しています。

コンテナイメージは、パッケージセット内の AL2023 最小 AMI に近いものです。

Amazon Linux 2023 イメージにインストールされているパッケージの比較

AL2023 AMI、Minimal AMI、コンテナイメージに存在する RPMs の比較。

パッケージ	AMI	Minimal AMI	コンテナ
<code>acl</code>	2.3.1		
<code>acpid</code>	2.0.32		
<code>alternatives</code>	1.15	1.15	1.15
<code>amazon-chroney-config</code>	4.3	4.3	

パッケージ	AMI	Minimal AMI	コンテナ
amazon-ec2-net-utils	2.4.1	2.4.1	
amazon-linux-repo-cdn			2023.4.20240513
amazon-linux-repo-s3	2023.4.20240513	2023.4.20240513	
amazon-linux-sb-keys	2023.1	2023.1	
amazon-rpm-config	228		
amazon-ssm-agent	3.3.380.0		
at	3.1.23		
attr	2.5.1		
audit	3.0.6	3.0.6	
audit-libs	3.0.6	3.0.6	3.0.6
aws-cfn-bootstrap	2.0		
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		

パッケージ	AMI	Minimal AMI	コンテナ
bind-libs	9.16.48		
bind-license	9.16.48		
bind-utils	9.16.48		
binutils	2.39		
boost-fil esystem	1.75.0		
boost-system	1.75.0		
boost-thread	1.75.0		
bzip2	1.0.8		
bzip2-libs	1.0.8	1.0.8	1.0.8
ca-certificates	2023 年 2 月 64 日	2023 年 2 月 64 日	2023 年 2 月 64 日
c-ares	1.19.0		
checkpolicy	3.4	3.4	
chkconfig	1.15		
chrony	4.3	4.3	
cloud-init	22.2.2	22.2.2	
cloud-init-cfg- ec2	22.2.2	22.2.2	
cloud-utils- growpart	0.31	0.31	
coreutils	8.32	8.32	

パッケージ	AMI	Minimal AMI	コンテナ
coreutils-common	8.32	8.32	
coreutils-single			8.32
cpio	2.13	2.13	
cracklib	2.9.6	2.9.6	
cracklib-dicts	2.9.6	2.9.6	
crontabs	1.11		
crypto-policies	20220428	20220428	20220428
crypto-policies-scripts	20220428		
cryptsetup	2.6.1		
cryptsetup-libs	2.6.1	2.6.1	
curl-minimal	8.5.0	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27	
cyrus-sasl-plain	2.1.27		
dbus	1.12.28	1.12.28	
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	

パッケージ	AMI	Minimal AMI	コンテナ
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-n-release-notification	1.2	1.2	
dnf-plugins-core	4.3.0	4.3.0	
dnf-plugin-support-info	1.2	1.2	
dnf-utils	4.3.0		
dosfstools	4.2		
dracut	055	055	
dracut-config-ec2	3.0	3.0	
dracut-config-generic	055	055	
dwz	0.14		
dyninst	10.2.1		
e2fsprogs	1.46.5	1.46.5	
e2fsprogs-libs	1.46.5	1.46.5	

パッケージ	AMI	Minimal AMI	コンテナ
ec2-hibinit-agent	1.0.8		
ec2-instance-connect	1.1		
ec2-instance-connect-selinux	1.1		
ec2-utils	2.2.0	2.2.0	
ed	1.14.2		
efi-filesystem	5	5	
efi-srpm-macros	5		
efivar	38	38	
efivar-libs	38	38	
elfutils-debuginfod-client	0.188		
elfutils-default-yama-scope	0.188	0.188	0.188
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

パッケージ	AMI	Minimal AMI	コンテナ
file	5.39	5.39	
file-libs	5.39	5.39	5.39
filesystem	3.14	3.14	3.14
findutils	4.8.0	4.8.0	
fonts-srpm-macros	2.0.5		
fstrm	0.6.1		
fuse-libs	2.9.9	2.9.9	
gawk	5.1.0	5.1.0	5.1.0
gdbm-libs	1.19	1.19	1.19
gdisk	1.0.8	1.0.8	
gettext	0.21	0.21	
gettext-libs	0.21	0.21	
ghc-srpm-macros	1.5.0		
glib2	2.74.7	2.74.7	2.74.7
glibc	2.34	2.34	2.34
glibc-all-langpacks	2.34	2.34	
glibc-common	2.34	2.34	2.34
glibc-gconv-extra	2.34		

パッケージ	AMI	Minimal AMI	コンテナ
glibc-locale-source	2.34	2.34	
glibc-minimal-langpack			2.34
gmp	6.2.1	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7	2.3.7
gnutls	3.8.0	3.8.0	
go-srpm-macros	3.2.0		
gpgme	1.15.1	1.15.1	1.15.1
gpm-libs	1.20.7		
grep	3.8	3.8	3.8
groff-base	1.22.4	1.22.4	
grub2-common	2.06	2.06	
grub2-efi-aa64-ec2	2.06 (aarch64)	2.06 (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	

パッケージ	AMI	Minimal AMI	コンテナ
gssproxy	0.8.4		
gzip	1.12	1.12	
hostname	3.23	3.23	
hunspell	1.7.0		
hunspell-en	0.20140811.1		
hunspell-en-GB	0.20140811.1		
hunspell-en-US	0.20140811.1		
hunspell-filesystem	1.7.0		
hwdata	0.353	0.353	
info	6.7		
inih	49	49	
initscripts	10.09	10.09	
iproute	5.10.0	5.10.0	
iputils	20210202	20210202	
irqbalance	1.9.0	1.9.0	
jansson	2.14	2.14	
jitterentropy	3.4.1	3.4.1	
jq	1.7.1	1.7.1	
json-c	0.14	0.14	0.14
kbd	2.4.0	2.4.0	

パッケージ	AMI	Minimal AMI	コンテナ
kbd-misc	2.4.0	2.4.0	
kernel	6.1.90	6.1.90	
kernel-li vepatch-repo- s3	2023.4.20240513	2023.4.20240513	
kernel-srpm- macros	1.0		
kernel-tools	6.1.90		
keyutils	1.6.3		
keyutils-libs	1.6.3	1.6.3	1.6.3
kmod	29	29	
kmod-libs	29	29	
kpatch-runtime	0.9.7		
krb5-libs	1.21	1.21	1.21
less	608	608	
libacl	2.3.1	2.3.1	2.3.1
libaio	0.3.111		
libarchive	3.5.3	3.5.3	3.5.3
libargon2	20171227	20171227	
libassuan	2.5.5	2.5.5	2.5.5
libattr	2.5.1	2.5.1	2.5.1

パッケージ	AMI	Minimal AMI	コンテナ
libbasicobjects	0.1.1		
libblkid	2.37.4	2.37.4	2.37.4
libcap	2.48	2.48	2.48
libcap-ng	0.8.2	0.8.2	0.8.2
libcbor	0.7.0	0.7.0	
libcollection	0.7.0		
libcom_err	1.46.5	1.46.5	1.46.5
libcomps	0.1.20	0.1.20	0.1.20
libconfig	1.7.2		
libcurl-minimal	8.5.0	8.5.0	8.5.0
libdb	5.3.28	5.3.28	
libdhash	0.5.0		
libdnf	0.69.0	0.69.0	0.69.0
libeconf	0.4.0	0.4.0	
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	

パッケージ	AMI	Minimal AMI	コンテナ
libgcc	11.4.1	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2	1.10.2
libgomp	11.4.1	11.4.1	11.4.1
libgpg-error	1.42	1.42	1.42
libibverbs	48.0		
libidn2	2.3.2	2.3.2	2.3.2
libini_config	1.3.1		
libkcapi	1.4.0	1.4.0	
libkcapi-hmaccalc	1.4.0	1.4.0	
libldb	2.6.2		
libmaxminddb	1.5.2		
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		

パッケージ	AMI	Minimal AMI	コンテナ
libpipeline	1.5.3	1.5.3	
libpkgconf	1.8.0		
libpsl	0.21.1	0.21.1	0.21.1
libpwquality	1.4.4	1.4.4	
libref_array	0.1.5		
librepo	1.14.5	1.14.5	1.14.5
libreport-filesystem	2.15.2	2.15.2	2.15.2
libseccomp	2.5.3	2.5.3	
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		

パッケージ	AMI	Minimal AMI	コンテナ
libsss_ns s_idmap	2.9.4		
libsss_sudo	2.9.4		
libstdc++	11.4.1	11.4.1	11.4.1
libstoragemgmt	1.9.4		
libtalloc	2.3.4		
libtasn1	4.19.0	4.19.0	4.19.0
libtdb	1.4.7		
libtevent	0.13.0		
libtextstyle	0.21	0.21	
libtirpc	1.3.3		
libunistring	0.9.10	0.9.10	0.9.10
libuser	0.63	0.63	
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

パッケージ	AMI	Minimal AMI	コンテナ
libzstd	1.5.5	1.5.5	1.5.5
lm_sensors-libs	3.6.0		
lmdb-libs	0.9.29		
logrotate	3.20.1	3.20.1	
lsuf	4.94.0		
lua-libs	5.4.4	5.4.4	5.4.4
lua-srpm-macros	1		
lz4-libs	1.9.4	1.9.4	1.9.4
man-db	2.9.3	2.9.3	
man-pages	5.10		
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)	
mpfr	4.1.0	4.1.0	4.1.0
nano	5.8		
ncurses	6.2	6.2	
ncurses-base	6.2	6.2	6.2
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		

パッケージ	AMI	Minimal AMI	コンテナ
npth	1.6	1.6	1.6
nspr	4.35.0		
nss	3.90.0		
nss-softokn	3.90.0		
nss-softokn-freebl	3.90.0		
nss-sysinit	3.90.0		
nss-util	3.90.0		
ntsysv	1.15		
numactl-libs	2.0.14	2.0.14	
ocaml-srpm-macros	6		
oniguruma	6.9.7.1	6.9.7.1	
openblas-srpm-macros	2		
openldap	2.4.57	2.4.57	
openssh	8.7p1	8.7p1	
openssh-clients	8.7p1	8.7p1	
openssh-server	8.7p1	8.7p1	
openssl	3.0.8	3.0.8	
openssl-libs	3.0.8	3.0.8	3.0.8

パッケージ	AMI	Minimal AMI	コンテナ
openssl-pkcs11	0.4.12	0.4.12	
os-prober	1.77	1.77	
p11-kit	0.24.1	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1	0.24.1
package-notes-srpm-macros	0.4		
pam	1.5.1	1.5.1	
parted	3.4		
passwd	0.80	0.80	
pciutils	3.7.0	3.7.0	
pciutils-libs	3.7.0	3.7.0	
pcre2	10.40	10.40	10.40
pcre2-syntax	10.40	10.40	10.40
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		

パッケージ	AMI	Minimal AMI	コンテナ
perl-Fcntl	1.13		
perl-File-Basename	2.85		
perl-File-Path	2.18		
perl-File-stat	1.09		
perl-File-Temp	0.231.100		
perl-Getopt-Long	2.52		
perl-Getopt-Std	1.12		
perl-HTTP-Tiny	0.078		
perl-if	0.60.800		
perl-integerpreter	5.32.1		
perl-IO	1.43		
perl-IPC-Open3	1.21		
perl-libs	5.32.1		
perl-MIME-Base64	3.16		
perl-mro	1.23		
perl-overload	1.31		
perl-overloading	0.02		

パッケージ	AMI	Minimal AMI	コンテナ
perl-parent	0.238		
perl-PathTools	3.78		
perl-Pod-Escapes	1.07		
perl-podlators	4.14		
perl-Pod-Perldoc	3.28.01		
perl-Pod-Simple	3.42		
perl-Pod-Usage	2.01		
perl-POSIX	1.94		
perl-Scalar-List-Utills	1.56		
perl-SelectSaver	1.02		
perl-Socket	2.032		
perl-srpm-macros	1		
perl-Storable	3.21		
perl-subst	1.03		
perl-Symbol	1.08		
perl-Term-ANSIColor	5.01		
perl-Term-Cap	1.17		

パッケージ	AMI	Minimal AMI	コンテナ
perl-Text- ParseWords	3.30		
perl-Text-Tabs +Wrap	2021.0726		
perl-Time-Local	1.300		
perl-vars	1.05		
pkgconf	1.8.0		
pkgconf-m4	1.8.0		
pkgconf-pkg- config	1.8.0		
policycoreutils	3.4	3.4	
policycoreutils-python- utils	3.4		
popt	1.18	1.18	1.18
procps-ng	3.3.17	3.3.17	
protobuf-c	1.4.1		
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	

パッケージ	AMI	Minimal AMI	コンテナ
python3-audit	3.0.6	3.0.6	
python3-awscli	0.19.19	0.19.19	
python3-babel	2.9.1	2.9.1	
python3-cffi	1.14.5	1.14.5	
python3-chardet	4.0.0	4.0.0	
python3-colorama	0.4.4	0.4.4	
python3-configobj	5.0.6	5.0.6	
python3-cryptography	36.0.1	36.0.1	
python3-daemon	2.3.0		
python3-dateutil	2.8.1	2.8.1	
python3-dbus	1.2.18	1.2.18	
python3-distro	1.5.0	1.5.0	
python3-dnf	4.14.0	4.14.0	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

パッケージ	AMI	Minimal AMI	コンテナ
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-libibcomps	0.1.20	0.1.20	0.1.20
python3-libdnf	0.69.0	0.69.0	0.69.0
python3-libs	3.9.16	3.9.16	3.9.16
python3-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	

パッケージ	AMI	Minimal AMI	コンテナ
python3-netifaces	0.10.6	0.10.6	
python3-openssl	3.0.2	3.0.2	
python3-pip-wheel	21.3.1	21.3.1	21.3.1
python3-ply	3.11	3.11	
python3-policycoreutils	3.4	3.4	
python3-pyrettytable	0.7.2	0.7.2	
python3-prompt-toolkit	3.0.24	3.0.24	
python3-pyyaml	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	

パッケージ	AMI	Minimal AMI	コンテナ
python3-rpm	4.16.1.3	4.16.1.3	4.16.1.3
python3-ruamel-yaml	0.16.6	0.16.6	
python3-ruamel-yaml-clib	0.1.2	0.1.2	
python3-setools	4.4.1	4.4.1	
python3-setuptools	59.6.0	59.6.0	
python3-setuptools-wheel	59.6.0	59.6.0	59.6.0
python3-six	1.15.0	1.15.0	
python3-systemd	235	235	
python3-urllib3	1.25.10	1.25.10	
python3-wcwidth	0.2.5	0.2.5	
python-chevron	0.13.1		
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	

パッケージ	AMI	Minimal AMI	コンテナ
rpcbind	1.2.6		
rpm	4.16.1.3	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3	4.16.1.3
rpm-libs	4.16.1.3	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3	4.16.1.3	
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3	
rpm-sign-libs	4.16.1.3	4.16.1.3	4.16.1.3
rsync	3.2.6		
rust-srpm-macros	21		
sbsigntools	0.9.4	0.9.4	
screen	4.8.0		
sed	4.8	4.8	4.8
selinux-policy	37.22	37.22	
selinux-policy-targeted	37.22	37.22	
setup	2.13.7	2.13.7	2.13.7
shadow-utils	4.9	4.9	
slang	2.3.2		
sqlite-libs	3.40.0	3.40.0	3.40.0

パッケージ	AMI	Minimal AMI	コンテナ
sssd-client	2.9.4		
sssd-common	2.9.4		
sssd-kcm	2.9.4		
sssd-nfs-idmap	2.9.4		
strace	6.8		
sudo	1.9.15	1.9.15	
sysctl-defaults	1.0	1.0	
sysstat	12.5.6		
systemd	252.16	252.16	
systemd-libs	252.16	252.16	
systemd-networkd	252.16	252.16	
systemd-pam	252.16	252.16	
systemd-resolved	252.16	252.16	
systemd-udev	252.16	252.16	
system-release	2023.4.20240513	2023.4.20240513	2023.4.20240513
systemtap-runtime	4.8		
tar	1.34	1.34	
tbb	2020年3月		

パッケージ	AMI	Minimal AMI	コンテナ
tcpdump	4.99.1		
tcsh	6.24.07		
time	1.9		
traceroute	2.1.3		
tzdata	2024年a	2024年a	2024年a
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		
xfspgrog	5.18.0	5.18.0	

パッケージ	AMI	Minimal AMI	コンテナ
xxd	9.0.2153		
xxhash-libs	0.8.0		
xz	5.2.5	5.2.5	
xz-libs	5.2.5	5.2.5	5.2.5
yum	4.14.0	4.14.0	4.14.0
zip	3.0		
zlib	1.2.11	1.2.11	1.2.11
zram-generator	1.1.2	1.1.2	
zram-generator-defaults	1.1.2	1.1.2	
zstd	1.5.5	1.5.5	

コンテナでの AL2023 の使用

Note

AL2023 を使用して Amazon ECS でコンテナ化されたワークロードをホストする方法の詳細については、「」を参照してください [Amazon ECS コンテナホスト用 AL2023](#)。

ユースケースによっては、コンテナ内で AL2023 を使用する方法がいくつかあります。 [AL2023 ベースコンテナイメージ](#) は、Amazon Linux 2 コンテナイメージと AL2023 最小 AMI に最も似ています。

上級ユーザー向けに、AL2023.2 リリースで導入された最小限のコンテナイメージと、 [ベアボーンコンテナ](#) の構築方法を説明するドキュメントを提供しています。

AL2023 は、AL2023 ベースのコンテナイメージ、または他の Linux 配布に基づくコンテナのいずれかの、コンテナ化されたワークロードをホストするためにも使用できます。[Amazon ECS コンテナホスト用 AL2023](#) を使用することも、付属のコンテナランタイムパッケージを直接使用することもできます。docker、containerd、nerdctl パッケージは AL2023 にインストールして使用できません。

トピック

- [AL2023 ベースコンテナイメージの使用](#)
- [AL2023 最小コンテナイメージ](#)
- [ベアボーン AL2023 コンテナイメージの構築](#)
- [Amazon Linux 2023 コンテナイメージにインストールされているパッケージの比較](#)
- [Amazon Linux 2023 最小 AMI とコンテナイメージにインストールされているパッケージの比較](#)

AL2023 ベースコンテナイメージの使用

AL2023 コンテナイメージは、AL2023 AMI に含まれているのと同じソフトウェアコンポーネントから構築されています。これは、Docker ワークロードのベースイメージとして任意の環境で使用できます。[Amazon Elastic Compute Cloud](#) (Amazon EC2) 内のアプリケーション用にすでに Amazon Linux AMI を使用している場合、Amazon Linux コンテナイメージでアプリケーションをコンテナ化できます。

ローカル開発環境で Amazon Linux コンテナイメージを使用し、Amazon [Elastic Container Service \(Amazon ECS\)](#) AWS を使用してアプリケーションを にプッシュします。詳細については、「Amazon Elastic Container Registry ユーザーガイド」の「[Amazon ECS で Amazon ECR イメージを使用する](#)」を参照してください。

Amazon Linux コンテナイメージは、Amazon ECR Public で入手できます。AL2023 に関するフィードバックは、指定された AWS 担当者を通じて、またはの [amazon-linux-2023 リポジトリ](#) に問題を提起することで提供できます GitHub。

Amazon ECR Public から Amazon Linux コンテナイメージをプルする方法

1. Amazon Linux Public レジストリに Docker クライアントを認証します。認証トークンは 12 時間有効です。詳細については、「Amazon Elastic Container Registry ユーザーガイド」の「[プライベートレジストリの認証](#)」を参照してください。

Note

get-login-password コマンドは、最新バージョンの AWS CLI バージョン 2 を使用してサポートされています。詳細については、「AWS Command Line Interface ユーザーガイド」の「[AWS Command Line Interfaceのインストール](#)」を参照してください。

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

出力は次のとおりです。

```
Login succeeded
```

2. docker pull コマンドを起動して Amazon Linux コンテナイメージを取得します。Amazon ECR Public Gallery で Amazon Linux コンテナイメージを表示するには、「[Amazon ECR Public Gallery - amazonlinux](#)」を参照してください。

Note

AL2023 Docker コンテナイメージを取得する場合、以下のいずれかの形式でタグを使用できます。

- AL2023 コンテナイメージの最新バージョンを取得するには、:2023 タグを使用します。
- AL2023 の特定のバージョンを入手するには、以下の形式を使用できます。
 - :2023.[0-7 release quarter].[release date].[build number]

以下の例では、タグ :2023 を使用して AL2023 の入手可能な最新のコンテナイメージを取得しています。

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

3. (オプション) コンテナをローカルに実行します。

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

Docker Hub から AL2023 コンテナイメージを取得する方法

1. `docker pull` コマンドを使用して AL2023 コンテナイメージを取得します。

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

2. (オプション) コンテナをローカルに実行します。

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

Note

AL2023 のコンテナイメージは、`dnf` パッケージマネージャーのみを使用してソフトウェアパッケージをインストールします。つまり、追加のソフトウェアに使用できる、`amazon-linux-extras` または同等のコマンドはありません。

AL2023 最小コンテナイメージ

Note

標準の AL2023 コンテナイメージはほとんどのユースケースに適しています。最小のコンテナイメージへの適応は、AL2023 ベースコンテナイメージへの適応よりも機能する可能性が高いです。

AL2023.2 で導入された AL2023 最小コンテナイメージは、他のパッケージのインストールに必要な最低限のパッケージのみが含まれているため、ベースコンテナイメージとは異なります。最小コンテナイメージは、パッケージの便利なセットではなく、パッケージの最小限のセットとして設計されています。

AL2023 の最小コンテナイメージは、AL2023 ですでに利用可能なソフトウェアコンポーネントから構築されています。最小コンテナイメージの主な違いは、`microdnf`を使用して、フル機能の Python ベースではなく `dnf` パッケージマネージャーを提供することです `dnf`。これにより、AL2023

AMIs とベースコンテナイメージに含まれるdnfパッケージマネージャーの機能セットがすべて揃っていないというトレードオフにより、最小限のコンテナイメージを小さくすることができます。

AL2023 最小コンテナイメージは、Lambda provided.al2023 AWS ランタイム環境のベースを形成します。

最小コンテナイメージに含まれるパッケージの詳細なリストについては、「」を参照してください [Amazon Linux 2023 コンテナイメージにインストールされているパッケージの比較](#)。

最小コンテナイメージのサイズ

AL2023 最小コンテナイメージに含まれるパッケージは AL2023 ベースコンテナイメージよりも少ないため、大幅に小さくなります。次の表は、Amazon Linux の現在および過去のリリースのコンテナイメージオプションを比較したものです。

Note

イメージサイズは [Amazon ECR Public Gallery の Amazon Linux](#) に表示されているとおりです。

イメージ	Version	イメージのサイズ	注記
Amazon Linux 1 (AL1)	2018.03.0.20230918 .0	62.3 MB	x86-64 のみ
Amazon Linux 2	2.0.20230926.0	64.2 MB	aaarch64 は x86-64 よりも 1.6 MB 大きくなります
Amazon Linux 2023 コンテナイメージ	2023 年 2 月 2023 日 1002.0	52.4 MB	
Amazon Linux 2023 最小コンテナイメージ	2023.2.20231002.0- minimal	35.2 MB	

AL2023 最小コンテナイメージの使用

AL2023 最小コンテナイメージは で利用でき、2023-minimalタグは常に最新の AL2023 ベースの最小コンテナイメージを指ECRし、minimalタグは AL2023 よりも新しいバージョンの Amazon Linux に更新される場合があります。

これらのタグは、次の例dockerで を使用してプルできます。

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

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

次の例はDockerfile、最小限のコンテナイメージを取得し、その上に GCC をインストールするを示しています。

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

ベアボーン AL2023 コンテナイメージの構築

AL2023 コンテナイメージは、AL2023 AMI に含まれているのと同じソフトウェアコンポーネントから構築されています。これには、ベースコンテナレイヤーがパッケージマネージャーなどの Amazon EC2 インスタンスでの実行と同様に動作できるようにするソフトウェアが含まれていますdnf。このセクションでは、アプリケーションに必要な最低限の依存関係のみを含むコンテナをゼロから構築する方法について説明します。

Note

標準の AL2023 コンテナイメージは、ほとんどのユースケースに適しています。標準のコンテナイメージを使用すると、イメージ上に簡単に構築できます。ベアボーンコンテナイメージを使用すると、イメージ上に構築するのが難しくなります。

アプリケーションの依存関係を最低限に抑えたコンテナを作成する方法

1. ランタイム依存関係を決定します。これはアプリケーションによって異なります。

2. FROM scratch をビルドする Dockerfile または Containerfile を作成します。次の Dockerfile の例では、bash シェルとその依存関係のみを含むコンテナを構築できます。

```
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"]
```

- この Dockerfile は以下のように機能します。
 1. build という名前の AL2023 コンテナを起動します。このコンテナはベアボーンコンテナのブートストラップに使用されます。このコンテナ自体はデプロイされませんが、デプロイするコンテナが生成されます。
 2. /sysroot ディレクトリを作成します。このディレクトリは、build コンテナがベアボーンコンテナに必要な依存関係をインストールする場所になります。以降のステップでは、/sysroot パスはベアボーンイメージのルートディレクトリとなるようにパッケージ化されます。

この `--installroot` オプションをこのように `dnf` に使用して、他の AL2023 イメージを作成します。これは、インストーラーとイメージ作成ツールを動作させる `dnf` の機能です。

3. `dnf` を呼び出して、/sysroot にパッケージをインストールします。

`rpm -q system-release --qf '%{VERSION}'` コマンドは `system-release` パッケージをクエリ (`-q`) し、クエリ対象のパッケージのバージョン (`%{VERSION}`) 変数は RPM のバージョンに対応する `rpm` 変数) を出力するようにクエリ形式 (`--qf`) を設定します。

dnf の `--releasever` 引数を build コンテナ内の `system-release` バージョンに設定すると、Amazon Linux の更新されたコンテナベースイメージがリリースされるたびに、この Dockerfile を使用してベアボーンコンテナを再構築できます。

を 2023.4.20240513 などの `--releasever` 任意の Amazon Linux 2023 バージョンに設定することができます。これにより、build コンテナは最新の AL2023 バージョンとして実行されますが、現在の AL2023 AL2023 リリースに関係なく、2023.4.20240513 からベアボーンコンテナを構築します。

`--setopt=install_weak_deps=False` 設定オプションでは、推奨や提案ではなく、必要な依存関係のみをインストールするように dnf に指示します。

4. インストールしたシステムを空の (FROM scratch) コンテナのルートにコピーします。
 5. この `/bin/bash` の場合は、ENTRYPOINT を目的のバイナリに設定します。
3. 空のディレクトリを作成し、ステップ 2 の例の内容を Dockerfile という名前のファイルに追加します。

```
$ 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. 次のコマンドを実行してコンテナを構築します。

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

5. 以下のコマンドを使用してコンテナを実行し、bash オンリーコンテナがどれほど最小限であるかを確認します。

```
$ 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
```

より実用的な例として、以下の手順では Hello World! を表示する C アプリケーション用のコンテナを構築します。

1. 空のディレクトリを作成し、C ソースコードおよび Dockerfile を追加します。

```
$ 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. 以下のコマンドを使用して、コンテナを構築します。

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

3. 以下のコマンドを実行して、コンテナを起動します。

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

Amazon Linux 2023 コンテナイメージにインストールされているパッケージの比較

AL2023 ベースコンテナイメージに存在する RPMs と AL2023 最小コンテナイメージに存在する RPMs の比較。

パッケージ	コンテナ	最小コンテナ
alternatives	1.15	1.15
amazon-linux-repo-cdn	2023.4.20240513	2023.4.20240513
audit-libs	3.0.6	3.0.6
basesystem	11	11
bash	5.2.15	5.2.15

パッケージ	コンテナ	最小コンテナ
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023年2月64日	2023年2月64日
coreutils-single	8.32	8.32
crypto-policies	20220428	20220428
curl-minimal	8.5.0	8.5.0
dnf	4.14.0	
dnf-data	4.14.0	4.14.0
elfutils-default-yama-scope	0.188	
elfutils-libelf	0.188	
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

パッケージ	コンテナ	最小コンテナ
gmp	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7
gobject-introspection		1.73.0
gpgme	1.15.1	1.15.1
grep	3.8	3.8
json-c	0.14	0.14
keyutils-libs	1.6.3	1.6.3
krb5-libs	1.21	1.21
libacl	2.3.1	2.3.1
libarchive	3.5.3	3.5.3
libassuan	2.5.5	2.5.5
libattr	2.5.1	2.5.1
libblkid	2.37.4	2.37.4
libcap	2.48	2.48
libcap-ng	0.8.2	0.8.2
libcom_err	1.46.5	1.46.5
libcomps	0.1.20	
libcurl-minimal	8.5.0	8.5.0
libdnf	0.69.0	0.69.0
libffi	3.4.4	3.4.4

パッケージ	コンテナ	最小コンテナ
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	
libgpg-error	1.42	1.42
libidn2	2.3.2	2.3.2
libmodulemd	2.13.0	2.13.0
libmount	2.37.4	2.37.4
libnghttp2	1.59.0	1.59.0
libpeas		1.32.0
libpsl	0.21.1	0.21.1
librepo	1.14.5	1.14.5
libreport-filessystem	2.15.2	2.15.2
libselenium	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

パッケージ	コンテナ	最小コンテナ
libuuid	2.37.4	2.37.4
libverto	0.3.2	0.3.2
libxcrypt	4.4.33	
libxml2	2.10.4	2.10.4
libyaml	0.2.5	0.2.5
libzstd	1.5.5	1.5.5
lua-libs	5.4.4	5.4.4
lz4-libs	1.9.4	1.9.4
microdnf		3.8.1
microdnf-dnf		3.8.1
mpfr	4.1.0	4.1.0
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
npth	1.6	1.6
openssl-libs	3.0.8	3.0.8
p11-kit	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1
pcre2	10.40	10.40
pcre2-syntax	10.40	10.40
popt	1.18	1.18

パッケージ	コンテナ	最小コンテナ
publicsuffix-list-dafsa	20240212	20240212
python3	3.9.16	
python3-dnf	4.14.0	
python3-gpg	1.15.1	
python3-hawkey	0.69.0	
python3-libcomps	0.1.20	
python3-libdnf	0.69.0	
python3-libs	3.9.16	
python3-pip-wheel	21.3.1	
python3-rpm	4.16.1.3	
python3-setuptools-wheel	59.6.0	
readline	8.1	8.1
rpm	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	
rpm-libs	4.16.1.3	4.16.1.3
rpm-sign-libs	4.16.1.3	
sed	4.8	4.8
setup	2.13.7	2.13.7
sqlite-libs	3.40.0	3.40.0

パッケージ	コンテナ	最小コンテナ
system-release	2023.4.20240513	2023.4.20240513
tzdata	2024年a	
xz-libs	5.2.5	5.2.5
yum	4.14.0	
zlib	1.2.11	1.2.11

Amazon Linux 2023 最小 AMI とコンテナイメージにインストールされているパッケージの比較

AL2023 最小 AMI に存在する RPMs と AL2023 ベースイメージおよび最小コンテナイメージに存在する RPMs の比較。

パッケージ	Minimal AMI	コンテナ	最小コンテナ
alternatives	1.15	1.15	1.15
amazon-chrorny-config	4.3		
amazon-ec2-net-utils	2.4.1		
amazon-linux-repo-cdn		2023.4.20240513	2023.4.20240513
amazon-linux-repo-s3	2023.4.20240513		
amazon-linux-sb-keys	2023.1		
audit	3.0.6		

パッケージ	Minimal AMI	コンテナ	最小コンテナ
audit-libs	3.0.6	3.0.6	3.0.6
awscli-2	2.15.30		
basesystem	11	11	11
bash	5.2.15	5.2.15	5.2.15
bzip2-libs	1.0.8	1.0.8	1.0.8
ca-certificates	2023 年 2 月 64 日	2023 年 2 月 64 日	2023 年 2 月 64 日
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

パッケージ	Minimal AMI	コンテナ	最小コンテナ
cryptsetup-libs	2.6.1		
curl-minimal	8.5.0	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27		
dbus	1.12.28		
dbus-broker	32		
dbus-common	1.12.28		
dbus-libs	1.12.28		
device-mapper	1.02.185		
device-mapper-libs	1.02.185		
diffutils	3.8		
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		

パッケージ	Minimal AMI	コンテナ	最小コンテナ
dracut-config-generic	055		
e2fsprogs	1.46.5		
e2fsprogs-libs	1.46.5		
ec2-utils	2.2.0		
efi-filesystem	5		
efivar	38		
efivar-libs	38		
elfutils-default-yama-scope	0.188	0.188	
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	

パッケージ	Minimal AMI	コンテナ	最小コンテナ
gdisk	1.0.8		
gettext	0.21		
gettext-libs	0.21		
glib2	2.74.7	2.74.7	2.74.7
glibc	2.34	2.34	2.34
glibc-all-langpacks	2.34		
glibc-common	2.34	2.34	2.34
glibc-locale-source	2.34		
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		

パッケージ	Minimal AMI	コンテナ	最小コンテナ
grub2-efi-aa64-ec2	2.06 (aarch64)		
grub2-efi-x64-ec2	2.06 (x86_64)		
grub2-pc-modules	2.06		
grub2-tools	2.06		
grub2-tools-minimal	2.06		
grubby	8.40		
gzip	1.12		
hostname	3.23		
hwdata	0.353		
inih	49		
initscripts	10.09		
iproute	5.10.0		
iputils	20210202		
irqbalance	1.9.0		
jansson	2.14		
jitterentropy	3.4.1		
jq	1.7.1		
json-c	0.14	0.14	0.14

パッケージ	Minimal AMI	コンテナ	最小コンテナ
kbd	2.4.0		
kbd-misc	2.4.0		
kernel	6.1.90		
kernel-li vepatch-repo- s3	2023.4.20240513		
keyutils-libs	1.6.3	1.6.3	1.6.3
kmod	29		
kmod-libs	29		
krb5-libs	1.21	1.21	1.21
less	608		
libacl	2.3.1	2.3.1	2.3.1
libarchive	3.5.3	3.5.3	3.5.3
libargon2	20171227		
libassuan	2.5.5	2.5.5	2.5.5
libattr	2.5.1	2.5.1	2.5.1
libblkid	2.37.4	2.37.4	2.37.4
libcap	2.48	2.48	2.48
libcap-ng	0.8.2	0.8.2	0.8.2
libcbor	0.7.0		
libcom_err	1.46.5	1.46.5	1.46.5

パッケージ	Minimal AMI	コンテナ	最小コンテナ
libcomps	0.1.20	0.1.20	
libcurl-minimal	8.5.0	8.5.0	8.5.0
libdb	5.3.28		
libdnf	0.69.0	0.69.0	0.69.0
libeconf	0.4.0		
libedit	3.1		
libfdisk	2.37.4		
libffi	3.4.4	3.4.4	3.4.4
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

パッケージ	Minimal AMI	コンテナ	最小コンテナ
libpeas			1.32.0
libpipeline	1.5.3		
libpsl	0.21.1	0.21.1	0.21.1
libpwquality	1.4.4		
librepo	1.14.5	1.14.5	1.14.5
libreport-filesystem	2.15.2	2.15.2	2.15.2
libseccomp	2.5.3		
libselinux	3.4	3.4	3.4
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

パッケージ	Minimal AMI	コンテナ	最小コンテナ
libuser	0.63		
libutempter	1.2.1		
libuuid	2.37.4	2.37.4	2.37.4
libverto	0.3.2	0.3.2	0.3.2
libxcrypt	4.4.33	4.4.33	
libxml2	2.10.4	2.10.4	2.10.4
libyaml	0.2.5	0.2.5	0.2.5
libzstd	1.5.5	1.5.5	1.5.5
logrotate	3.20.1		
lua-libs	5.4.4	5.4.4	5.4.4
lz4-libs	1.9.4	1.9.4	1.9.4
man-db	2.9.3		
microcode_ctl	2.1 (x86_64)		
microdnf			3.8.1
microdnf-dnf			3.8.1
mpfr	4.1.0	4.1.0	4.1.0
ncurses	6.2		
ncurses-base	6.2	6.2	6.2
ncurses-libs	6.2	6.2	6.2
nettle	3.8		

パッケージ	Minimal AMI	コンテナ	最小コンテナ
net-tools	2.0		
npth	1.6	1.6	1.6
numactl-libs	2.0.14		
oniguruma	6.9.7.1		
openldap	2.4.57		
openssh	8.7p1		
openssh-clients	8.7p1		
openssh-server	8.7p1		
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

パッケージ	Minimal AMI	コンテナ	最小コンテナ
policycoreutils	3.4		
popt	1.18	1.18	1.18
procps-ng	3.3.17		
psmisc	23.4		
publicsuffix-list-dafsa	20240212	20240212	20240212
python3	3.9.16	3.9.16	
python3-attrs	20.3.0		
python3-audit	3.0.6		
python3-awscrt	0.19.19		
python3-babel	2.9.1		
python3-cffi	1.14.5		
python3-chardet	4.0.0		
python3-colorama	0.4.4		
python3-configobj	5.0.6		
python3-cryptography	36.0.1		
python3-dateutil	2.8.1		
python3-dbus	1.2.18		

パッケージ	Minimal AMI	コンテナ	最小コンテナ
python3-distro	1.5.0		
python3-dnf	4.14.0	4.14.0	
python3-dnf-plugins-core	4.3.0		
python3-docutils	0.16		
python3-gpg	1.15.1	1.15.1	
python3-hawkey	0.69.0	0.69.0	
python3-idna	2.10		
python3-jinja2	2.11.3		
python3-jmespath	0.10.0		
python3-jsonpatch	1.21		
python3-jsonpointer	2.0		
python3-jsonschema	3.2.0		
python3-libibcomps	0.1.20	0.1.20	
python3-libdnf	0.69.0	0.69.0	
python3-libs	3.9.16	3.9.16	

パッケージ	Minimal AMI	コンテナ	最小コンテナ
python3-l ibselinux	3.4		
python3-l ibsemanage	3.4		
python3-m arkupsafe	1.1.1		
python3-n etifaces	0.10.6		
python3-o authlib	3.0.2		
python3-pip- wheel	21.3.1	21.3.1	
python3-ply	3.11		
python3-p olicycoreutils	3.4		
python3-p rettytable	0.7.2		
python3-prompt- toolkit	3.0.24		
python3-p ycparser	2.20		
python3-p yrsistent	0.17.3		
python3-p yserial	3.4		

パッケージ	Minimal AMI	コンテナ	最小コンテナ
python3-pysocks	1.7.1		
python3-pytz	2022.7.1		
python3-pyyaml	5.4.1		
python3-requests	2.25.1		
python3-rpm	4.16.1.3	4.16.1.3	
python3-ruamel-yaml	0.16.6		
python3-ruamel-yaml-clib	0.1.2		
python3-setools	4.4.1		
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		

パッケージ	Minimal AMI	コンテナ	最小コンテナ
rpm	4.16.1.3	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3	
rpm-libs	4.16.1.3	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3		
rpm-plugin-systemd-inhibit	4.16.1.3		
rpm-sign-libs	4.16.1.3	4.16.1.3	
sbsigntools	0.9.4		
sed	4.8	4.8	4.8
selinux-policy	37.22		
selinux-policy-targeted	37.22		
setup	2.13.7	2.13.7	2.13.7
shadow-utils	4.9		
sqlite-libs	3.40.0	3.40.0	3.40.0
sudo	1.9.15		
sysctl-defaults	1.0		
systemd	252.16		
systemd-libs	252.16		
systemd-networkd	252.16		

パッケージ	Minimal AMI	コンテナ	最小コンテナ
systemd-pam	252.16		
systemd-resolved	252.16		
systemd-udev	252.16		
system-release	2023.4.20240513	2023.4.20240513	2023.4.20240513
tar	1.34		
tzdata	2024年a	2024年a	
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		
xfspgrog	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		

パッケージ	Minimal AMI	コンテナ	最小コンテナ
zram-generator-defaults	1.1.2		
zstd	1.5.5		

AL2023 オン AWS Elastic Beanstalk

AWS Elastic Beanstalk は、ウェブアプリケーションとサービスをデプロイし、スケーリングするためのサービスです。コードをアップロードするだけで、Elastic Beanstalk が、キャパシティーのプロビジョニング、ロードバランシング、自動スケーリングからアプリケーションの状態モニタリングまで、デプロイを自動的に処理します。詳細については、「[AWS Elastic Beanstalk](#)」を参照してください。

Elastic Beanstalk を使用するには、アプリケーションを作成し、アプリケーションソースバンドル (Java .war ファイルなど) の形式でアプリケーションバージョンを Elastic Beanstalk にアップロードした後、アプリケーションに関する情報を提供します。Elastic Beanstalk は自動的に環境を起動し、AWS コードの実行に必要なリソースを作成および設定します。詳細については、「[AWS Elastic Beanstalk デベロッパーガイド](#)」を参照してください。

Elastic Beanstalk Linux プラットフォームは Amazon EC2 インスタンスを使用し、これらのインスタンスは Amazon Linux を実行します。2023 年 8 月 4 日現在、Elastic Beanstalk は、Amazon Linux 2023 ベースの Docker、Tomcat、Java SE、Node.js、PHP、Python プラットフォームブランチを提供しています。Elastic Beanstalk は、AL2023 のサポートをより多くの Elastic Beanstalk プラットフォームにリリースする作業に取り組んでいます。

Elastic Beanstalk プラットフォームのサポートと AL2023 をベースに構築された現在のプラットフォームのすべてのリストは、「[Elastic Beanstalk デベロッパーガイド](#)」の「[Elastic Beanstalk Linux プラットフォーム](#)」セクションに記載されています。

新しい Elastic Beanstalk プラットフォームと既存のプラットフォームのバージョンのリリースノートは、「[Elastic Beanstalk リリースノート](#)」に記載されています。

で AL2023 を使用する AWS CloudShell

AWS CloudShell はブラウザベースの認証済みシェルで、から直接起動できます。AWS Management Consoleには、CloudShell いくつかの異なる方法で移動できます AWS Management Console 。詳細については、「[はじめるには AWS CloudShell?](#)」を参照してください。

AWS CloudShell現在Amazon Linux 2をベースにしているものが、AL2023に移行される予定です。AL2023 への移行は、2023 年 12 月 4 AWS リージョン 日から開始されます。AL2023 CloudShell への移行の詳細については、「[Amazon Linux 2 から Amazon Linux 2023 AWS CloudShell への移行](#)」を参照してください。

AL2023 ベースの Amazon ECS AMI を使用してコンテナ化されたワークロードをホストする

Note

AL2023 をコンテナ内で使用する方法の詳細については、[を参照してください。コンテナでの AL2023](#)

Amazon Elastic Container Service (Amazon ECS) は、コンテナ化されたアプリケーションを簡単にデプロイ、管理、スケーリングできる、完全マネージド型のコンテナオーケストレーションサービスです。完全マネージド型サービスとして、Amazon ECS AWS には設定と運用のベストプラクティスが組み込まれています。Amazon Elastic Container Registry (Amazon ECR) や Docker AWS などのサードパーティツールの両方と統合されています。この統合により、チームは環境ではなくアプリケーションの構築に集中しやすくなります。コントロールプレーンの複雑な管理は必要なく、クラウドの AWS リージョン間で、コンテナワークロードを実行およびスケールできます。

AL2023 ベースの Amazon ECS に最適化された AMI を使用して、コンテナ化されたワークロードを AL2023 でホストできます。詳細については、[Amazon ECS に最適化された AMI](#) を参照してください。

AL2 と比較した Amazon ECS の AL2023 の変更点

AL2 と同様に、AL2023 には Amazon ECS Linux インスタンスとして実行するために必要な基本パッケージが用意されています。AL2 ではcontainerd、docker、ecs-initパッケージはから入手で

きましたがamazon-linux-extras、AL2023 ではコアリポジトリにこれらのパッケージが含まれています。

バージョンングされたリポジトリ機能による確定的なアップグレードにより、すべての AL2023 AMI はデフォルトで特定のリポジトリバージョンにロックされます。これは AL2023 Amazon ECS に最適化された AMI にも当てはまります。環境へのすべての更新は、デプロイ前に慎重に管理およびテストできます。また、問題が発生した場合に以前の AMI の内容に簡単に戻すことができます。AL2023 でのこの機能の詳細については、「[AL2023のバージョン管理されたリポジトリによる確定的アップグレードの使用](#)」を参照してください。

AL2023 は AL2 でサポートされている cgroup v1 インターフェースを介して cgroup v2 に切り替わります。詳細については、「[統合コントロールグループ階層 \(cgroup v2\)](#)」を参照してください。

Note

[2023.2.20230920 \(最初の AL2023.2 リリース\) より前のバージョンの AL2023 には、cgroup 内でのメモリ不足 \(OOM\) 処理に関するバグが含まれていました。systemdOOM Killer が一度に 1 つのプロセスを選択する代わりに、cgroup 内のすべてのプロセスが常に強制終了されました。これは意図した動作です。これは AL2 の動作と比較するとリグレッションであり、AL2023 の 2023.2.20230920 リリースで修正されました。](#)

[Amazon ECS に最適化された AMI を構築するコードはプロジェクトで利用可能です。amazon-ecs-ami GitHub リリースノート](#)には、どの AL2023 バージョンがどの Amazon ECS AMI バージョンに対応しているかが記載されています。

AL2023 ベースの Amazon ECS 最適化 AMI のカスタマイズ

Important

Amazon ECS に最適化された AL2023 AMI を使用することをお勧めします。詳細については、「[Amazon エラスティックコンテナサービス開発者ガイド](#)」の「[Amazon ECS に最適化された AMI](#)」を参照してください。

Amazon ECS がカスタム AMI の作成に使用するのと同じビルドスクリプトを使用できます。詳細については、「[Amazon ECS に最適化された Linux AMI ビルドスクリプト](#)」を参照してください。

AL2023 での Amazon Elastic File System 使用

Amazon Elastic File System (Amazon EFS) は、サーバーレスで伸縮自在なファイルストレージを提供するため、ストレージ容量およびパフォーマンスのプロビジョニングや管理を行うことなくファイルデータを共有できます。Amazon EFS は、アプリケーションを中断することなく、ファイルの追加や削除に伴って自動的に伸縮し、ペタバイト規模までオンデマンドで拡張できるように構築されています。Amazon EFS はシンプルなウェブサービスインターフェイスを提供しているため、ファイルシステムをすばやく簡単に作成、設定できます。このサービスでは、ユーザーに代わってすべてのファイルストレージインフラストラクチャを管理するため、複雑なデプロイ、パッチ適用、および複雑なファイルシステム設定の保守を行う必要がありません。

Amazon EFS はネットワークファイルシステムバージョン 4 (NFSv4.1 および NFSv4.0) プロトコルをサポートするので、現在お使いのアプリケーションやツールも Amazon EFS とシームレスに動作します。Amazon EC2、Amazon ECS、を含む複数のコンピュートインスタンスが Amazon EFS ファイルシステムに同時にアクセスできます。AWS Lambda したがって、EFS ファイルシステムは、複数のコンピュートインスタンスやサーバーで実行されるワークロードやアプリケーションに共通のデータソースを提供できます。

AL2023 への **amazon-efs-utils** のインストール

amazon-efs-utils このパッケージは AL2023 リポジトリに用意されており、インストールして Amazon EFS ファイルシステムへのアクセスに使用できます。

amazon-efs-utils パッケージを AL2023 にインストールする

- **amazon-efs-utils** 以下のコマンドを使用してインストールします。

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

Amazon EFS ファイルシステムの AL2023 マウンティング

をインストールしたら **amazon-efs-utils**、Amazon EFS ファイルシステムを AL2023 インスタンスにマウントできます。

Amazon EFS ファイルシステムを AL2023 にマウントする

- ファイルシステム ID を使用してマウントするには、次のコマンドを使用します。

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

また、TLS を使用して転送中のデータを暗号化するようにファイルシステムをマウントしたり、ファイルシステム ID の代わりに DNS 名またはマウントターゲット IP を使用したりできます。詳細については、「[EFS マウントヘルパーを使用した Amazon EC2 Linux インスタンスへのマウンティング](#)」を参照してください。

AL2023 上に構築された Amazon EMR を使用する

Amazon EMR は、Apache Hadoop と AWS が提供するサービスを使用して、膨大な量のデータを効率良く簡単に処理できるウェブサービスです。

AL2023 ベースの Amazon EMR リリース

Amazon EMR リリース 7.0.0 は AL2023 に基づいて構築された最初のリリースでした。このリリースでは、AL2023 が Amazon EMR のベースオペレーティングシステムとなり、AL2023 のすべての利点が Amazon EMR にもたらされます。詳細については、[Amazon EMR 7.0.0](#) リリースノートを参照してください。

EKS での Amazon EMR

EKS 6.13 での Amazon EMR は、AL2023 をオプションとして導入した最初のリリースです。このリリースでは、Java 17 ランタイムとともに AL2023 をオペレーティングシステムとして Spark を起動できます。詳細については、[EKS 6.13 での Amazon EMR リリースノート](#)と、EKS [での Amazon EMR](#) のすべてのリリースノートを参照してください。

で AL2023 を使用する AWS Lambda

を使用すると AWS Lambda、サーバーのプロビジョニングや管理を行わずにコードを実行できます。お支払いいただくのは、使用したコンピューティング時間に対してのみ発生し、コードが実行されていない場合は課金されません。事実上あらゆる種類のアプリケーションやバックエンドサービスのコードを、すべて管理なしで実行できます。コードをアップロードするだけで、コードの実行とスケールに必要な処理はすべて Lambda により自動的に実行され、高い可用性が維持されます。

AL2023 **provided.al2023** はランタイムとコンテナイメージを管理します。

[provided.al2023ベースランタイムは AL2023 の最小コンテナイメージに基づいており、AL2023 ベースの Lambda マネージドランタイムとコンテナベースイメージを提供します。](#) provided.al2023ランタイムは AL2023 の最小コンテナイメージに基づいているため、約 109 MB のランタイムよりも 40 MB 未満と大幅に小さくなっています。provided.al2

詳細については、「[Lambda ランタイム](#)」と「[Lambda コンテナイメージの使用](#)」を参照してください。

AL2023 ベースのLambda ランタイム

[マネージド言語ランタイムの今後のリリース \(Node.js20、Python 3.12、Java 21、.NET 8 など\) は AL2023 provided.al2023 をベースとしており、AL2023 ベースのランタイムの発表で説明されているようにベースイメージとして使用される予定です。](#)

AL2023 ベースのLambda 関数

- [AL2023 で記述されたLambda 関数 Go](#)
- [AL2023 で記述されたLambda 関数 Rust](#)

詳細については、AWS Lambda デベロッパーガイドの「[Lambda ランタイム](#)」を参照してください。

チュートリアル

以下のチュートリアルでは、Amazon Linux 2023 (AL2023) を実行する Amazon EC2 インスタンスを使用して一般的なタスクを実行する方法を示します。ビデオチュートリアルについては、[AWS「の教育ビデオとラボ」](#)を参照してください。

AL2 の手順については、[Amazon EC2 ユーザーガイド](#)の「[Linux を実行する Amazon EC2 インスタンスのチュートリアル](#)」を参照してください。Amazon EC2

チュートリアル

- [チュートリアル: AL2023 に LAMP サーバーをインストールする](#)
- [チュートリアル: AL2023 で SSL/TLS を設定する](#)
- [チュートリアル: AL2023 で WordPress ブログをホストする](#)

チュートリアル: AL2023 に LAMP サーバーをインストールする

次の手順は、PHP と [MariaDB](#) (コミュニティで開発された MySQL のフォーク) をサポートする Apache ウェブサーバーを AL2023 インスタンス (LAMP ウェブサーバーまたは LAMP スタックと呼ばれることもあります) にインストールするのに役立ちます。このサーバーを使用して静的ウェブサイトホストしたり、データベースとの情報の読み取りと書き込みを行う動的な PHP アプリケーションをデプロイしたりできます。

Important

これらの手順は AL2023 での使用を目的としています。Ubuntu や Red Hat Enterprise Linux などの別のディストリビューションに LAMP ウェブサーバーを設定しようとする、このチュートリアルの通りにはなりません。Ubuntu については、Ubuntu コミュニティドキュメント [ApacheMySQLPHP](#) を参照してください。その他のディストリビューションについては、それぞれのドキュメントを参照してください。

タスク

- [ステップ 1: LAMP サーバーを準備する](#)
- [ステップ 2: LAMP サーバーをテストする](#)
- [ステップ 3: データベースサーバーをセキュリティで保護する](#)

- [ステップ 4: \(オプション\) をインストールする phpMyAdmin](#)
- [トラブルシューティング](#)
- [関連トピック](#)

ステップ 1: LAMP サーバーを準備する

前提条件

- このチュートリアルでは、AL2023 を使用してインターネットからアクセスできるパブリック DNS 名で新しいインスタンスを既に起動していることを前提としています。詳細については、「[Amazon EC2 での AL2023](#)」を参照してください。また、セキュリティグループを設定して、SSH (ポート 22)、HTTP (ポート 80)、HTTPS (ポート 443) 接続を有効にしている必要もあります。これらの前提条件の詳細については、Amazon EC2 ユーザーガイドの「[Linux インスタンスのインバウンドトラフィックの承認](#)」を参照してください。
- 次の手順では、AL2023 で利用可能な最新の PHP バージョン、現在 8.1 をインストールします。このチュートリアルで説明している以外の PHP アプリケーションを使用する場合は、8.1 との互換性を確認する必要があります。

LAMP サーバーを準備するには

1. インスタンスに接続します。詳細については、「[AL2023 インスタンスへの接続](#)」を参照してください。
2. すべてのソフトウェアパッケージが最新の状態であることを確認するため、インスタンスでソフトウェアの更新を実行します。この処理には数分かかりますが、最新のセキュリティアップデートとバグ修正を必ず適用することが重要です。

-y オプションを指定すると、確認メッセージを表示せずに更新をインストールします。インストール前に更新を検査する場合は、このオプションを省略できます。

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

3. AL2023 用の最新バージョンの Apache ウェブサーバーと PHP パッケージをインストールします。

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

4. MariaDB ソフトウェアパッケージをインストールします。dnf install コマンドを使用すると、複数のソフトウェアパッケージと関連するすべての依存関係を同時にインストールできます。

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

次のコマンドを使用して、これらのパッケージの現在のバージョンを表示できます。

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

例：

```
[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 (mariabdb) and many different client programs and
                libraries.
                : The base package contains the standard MariaDB/MySQL client programs
                and
                : utilities.
```

5. Apache ウェブサーバーを起動します。

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

6. `systemctl` コマンドを使用して、システムがブートするたびに Apache ウェブサーバーが起動するように設定します。


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

`httpd` が有効であることは、次のコマンドを実行して確認できます。

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

7. インバウンド HTTP (ポート 80) 接続をインスタンスに許可するセキュリティルールを追加していない場合には、このルールを追加します。デフォルトでは、起動時に `launch-wizard-N` セキュリティグループがインスタンス用に作成されます。その他のセキュリティグループルールを追加しなければ、このグループに含まれるのは SSH 接続を許可する単一のルールのみとなります。
 - a. Amazon EC2 コンソール (<https://console.aws.amazon.com/ec2/>) を開きます。
 - b. ナビゲーションペインで [インスタンス] を選択し、インスタンスを選択します。
 - c. [セキュリティ] タブで、インバウンドルールを表示します。次のルールが表示されます。

Port range	Protocol	Source
22	tcp	0.0.0.0/0

 Warning

0.0.0.0/0 を使用すると、すべての IPv4 アドレスからインスタンスへの、SSH によるアクセスが許可されます。これはテスト環境で短時間なら許容できますが、実稼働環境で行うのは安全ではありません。本番環境では、特定の IP アドレスまたは特定のアドレス範囲にのみ、インスタンスへのアクセスを限定します。

- d. HTTP (ポート 80) 接続を許可するインバウンドルールがないときは、この時点でルールを追加する必要があります。セキュリティグループのリンクを選択します。「[Linux インスタンスのインバウンドトラフィックを許可する](#)」の手順を使用して、次の値を持つ新しいインバウンドセキュリティルールを追加します。
 - [Type]: HTTP
 - [Protocol]: TCP
 - [Port Range]: 80
 - [Source]: Custom

8. ウェブサーバーをテストします。ウェブブラウザで、インスタンスのパブリック DNS アドレス (またはパブリック IP アドレス) を入力します。/var/www/html にコンテンツがない場合は Apache テストページが表示され、このページには「正常に動作しました」と表示されます。

インスタンスのパブリック DNS は、Amazon EC2 コンソールを使用して取得できます ([パブリック IPv4 DNS] 列を確認します。この列が表示されない場合は、[設定] (歯車のアイコン) をクリックし、[パブリック IPv4 DNS] を選択します)。

インスタンスのセキュリティグループに、ポート 80 での HTTP ラフィックを許可するルールが含まれていることを確認します。詳細については、[「セキュリティグループにルールを追加する」](#)を参照してください。

Important

Amazon Linux を使用していない場合は、それらの接続を許可するようにインスタンスのファイアウォールを設定する必要があるかもしれません。ファイアウォールの設定方法の詳細については、[デистриビューション用のドキュメント](#)を参照してください。

Apache httpd は、Apache ドキュメントルートと呼ばれるディレクトリに維持されるファイルを提供します。Amazon Linux Apache ドキュメントルートは /var/www/html であり、デフォルトでは root によって所有されます。

ec2-user アカウントがこのディレクトリで複数のファイルを操作することを許可するには、ディレクトリの所有権とアクセス許可を変更する必要があります。このタスクを行うには、複数の方法があります。このチュートリアルでは、ec2-user を apache グループに追加し、/var/www ディレクトリの所有権を apache グループに付与し、グループへの書き込み権限を割り当てます。

ファイルの許可を設定するには

1. ユーザー (この場合は ec2-user) を apache グループに追加します。

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

2. ログアウトし、再度ログインして新しいグループを選択し、メンバーシップを確認します。
 - a. ログアウトします (exit コマンドを使用するか、ターミナルウィンドウを閉じます)。

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

- b. apache グループのメンバーシップを検証するには、インスタンスに再接続して次のコマンドを実行します。

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

3. /var/www とそのコンテンツのグループ所有権を apache グループに変更します。

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

4. グループの書き込み許可を追加して、これからのサブディレクトにグループ ID を設定するには、/var/www とサブディレクトのディレクトリ許可を変更します。

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

5. グループ書き込み許可を追加するには、/var/www とサブディレクトリのファイル許可を再帰的に変更します。

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

ここで、ec2-user (および apache グループの将来のメンバー) は、Apache ドキュメントルートでファイルを追加、削除、編集できるようになります。したがって、静的ウェブサイトや PHP アプリケーションなどのコンテンツを追加できます。

ウェブサーバーを保護するには (オプション)

HTTP プロトコルを実行するウェブサーバーは、送受信したデータのトランスポートセキュリティを提供しません。ウェブブラウザを使用して HTTP サーバーに接続すると、閲覧した URL、受信したウェブページのコンテンツ、送信した HTML フォームの内容 (パスワードなど) はすべて、ネットワーク経路上のだれでも傍受できるようになります。ウェブサーバーを保護するためのベストプラクティスとして、SSL/TLS 暗号化でデータを保護する HTTPS (HTTP Secure) のサポートをインストールしてください。

サーバーで HTTPS を有効にする方法については、「[チュートリアル: AL2023 で SSL/TLS を設定する](#)」を参照してください。

ステップ 2: LAMP サーバーをテストする

サーバーがインストールおよび実行されており、ファイルのアクセス許可が正しく設定されている場合、ec2-user アカウントは、インターネットから使用できる /var/www/html ディレクトリに PHP ファイルを作成できます。

LAMP サーバーをテストするには

1. Apache ドキュメントルートで PHP ファイルを作成します。


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

このコマンドを実行しようとしたときに「許可が拒否されました」というエラーが表示された場合は、ログアウトし、再度ログインして、[ファイルの許可を設定するには](#) で設定した正しいグループ許可を取得します。

2. ウェブブラウザで、作成したファイルの URL を入力します。この URL は、インスタンスのパブリック DNS アドレスにスラッシュとファイル名を追加したものです。次に例を示します。


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

PHP 情報ページが表示されるはずですが、

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-xmldr.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



このページが表示されない場合は、前のステップで `/var/www/html/phpinfo.php` ファイルが正しく作成されたことを確認します。次のコマンドで、必要なパッケージがすべてインストールされたことを確認することもできます。

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

必要なパッケージのいずれかが出力に表示されていない場合は、`sudo yum install package` コマンドを使ってインストールします。

3. `phpinfo.php` ファイルを削除します。これは有用な情報であることもありますが、セキュリティ上の理由から、インターネット上で公表しないでください。

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

これで、完全に機能する LAMP ウェブサーバーを設定しました。`/var/www/html` の Apache ドキュメントルートにコンテンツを追加する場合、そのコンテンツはインスタンスのパブリック DNS アドレスで表示できます。

ステップ 3: データベースサーバーをセキュリティで保護する

MariaDB サーバーのデフォルトのインストールには、テストおよび開発に役立ついくつかの機能がありますが、実稼働サーバーでは無効にするか削除する必要があります。`mysql_secure_installation` コマンドを使用すると、ルートパスワードを設定し、安全でない機能をインストールから削除する手順が案内されます。MariaDB サーバーを使用する予定がない場合でも、この手順を実行することが推奨されます。

MariaDB サーバーをセキュリティで保護するには

1. MariaDB サーバーを起動します。

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

2. `mysql_secure_installation` を実行します。

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

- a. プロンプトが表示されたら、ルートアカウントのパスワードを入力します。
 - i. 現在のルートパスワードを入力します。デフォルトでは、ルートアカウントにはパスワードが設定されていません。Enter キーを押します。
 - ii. 「Y」と入力してパスワードを設定し、安全なパスワードを 2 回入力します。安全なパスワード作成の詳細については、「<https://identitysafe.norton.com/password-generator/>」を参照してください。このパスワードは必ず安全な場所に保管します。

MariaDB のルートパスワードの設定は、データベースを保護するための最も基本的な手段にすぎません。データベース駆動型アプリケーションを構築またはインストールす

る必要がある場合、通常はそのアプリケーションのデータベースサービスユーザーを作成します。ルートアカウントは、データベース管理以外には使用しないでください。

- b. 「Y」と入力して匿名ユーザーアカウントを削除します。
 - c. 「Y」と入力してリモートルートログインを無効にします。
 - d. 「Y」と入力してテストデータベースを削除します。
 - e. 「Y」と入力して権限テーブルを再ロードし、変更を保存します。
3. (オプション) MariaDB サーバーをすぐに使用する予定がない場合は、これを停止します。再び必要になったときには再起動できます。

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

4. (オプション) ブート時に毎回 MariaDB サーバーを起動させる場合は、次のコマンドを入力します。

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

ステップ 4: (オプション) をインストールする phpMyAdmin

[phpMyAdmin](#) は、EC2 インスタンスの MySQL データベースを表示および編集するために使用できるウェブベースのデータベース管理ツールです。Amazon Linux インスタンスで phpMyAdmin をインストールして設定するには、以下の手順に従ってください。

Important

Apache で SSL/TLS を有効にしていない場合、LAMP サーバーへのアクセスに phpMyAdmin を使用することは推奨されません。そのようにすると、データベース管理者のパスワードや他のデータは、インターネット上を安全ではない状態で送信されます。デベロッパーからのセキュリティに関する推奨事項については、[phpMyAdmin 「インストールのセキュリティ保護」](#)を参照してください。EC2 インスタンスでのウェブサーバーの保護に関する一般的な情報については、「[チュートリアル: AL2023 で SSL/TLS を設定する](#)」を参照してください。

をインストールするには phpMyAdmin

1. 必要な依存ファイルをインストールします。

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

2. Apache を再起動します。

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

3. php-fpm を再起動します。

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

4. /var/www/html で Apache ドキュメントルートに移動します。

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

5. <https://www.phpmyadmin.net/downloads> から最新 phpMyAdmin リリースのソースパッケージを選択します。ファイルディレクトリをインスタンスにダウンロードするには、次の例のようにリンクをコピーして wget コマンドに貼り付けます。

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

6. phpMyAdmin フォルダを作成し、次のコマンドでパッケージを展開します。

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

7. **phpMyAdmin-latest-all-languages.tar.gz** tarball を削除します。

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

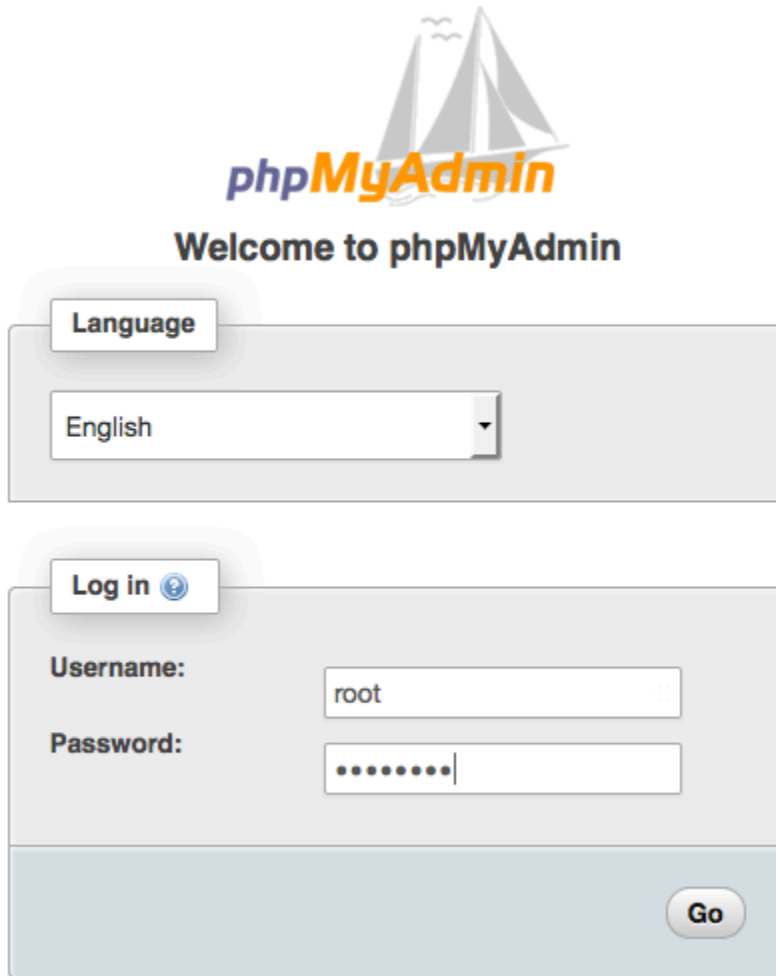
8. (オプション) MySQL サーバーが実行中ではない場合は、今すぐ起動します。

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

9. ウェブブラウザで、phpMyAdmin インストールの URL を入力します。この URL は、インスタンスのパブリック DNS アドレス (または、パブリック IP アドレス) にスラッシュとインストールディレクトリを追加してものです。例:

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

phpMyAdmin ログインページが表示されます。



The image shows the phpMyAdmin login interface. At the top, there is a logo for phpMyAdmin and the text "Welcome to phpMyAdmin". Below this, there is a "Language" dropdown menu currently set to "English". The main section is titled "Log in" and contains two input fields: "Username:" with the value "root" and "Password:" with masked characters. A "Go" button is located at the bottom right of the login form.

10. 以前に作成したrootユーザー名と MySQL ルートパスワードを使用して phpMyAdmin インストールにログインします。

インストールは、サービス開始前に設定する必要があります。次の手順に従って、設定ファイルを手動で作成することから始めるのをお勧めします。

- a. 最小の設定ファイルから開始するには、お気に入りのテキストエディタを使用して新しいファイルを作成し、`config.sample.inc.php` の内容をそのファイルにコピーします。
- b. ファイルを を含む phpMyAdmin ディレクトリ `config.inc.php` として保存します `index.php`。
- c. 追加のセットアップについては、 phpMyAdmin インストール手順の [「セットアップスクリプトの使用」](#) セクションの「ファイル作成後の手順」を参照してください。

の使用の詳細については phpMyAdmin、[「phpMyAdmin ユーザーガイド」](#)を参照してください。

トラブルシューティング

このセクションでは、新しい LAMP サーバーの設定時に発生する可能性がある一般的な問題の解決案を提供します。

ウェブブラウザを使用してサーバーに接続できません。

以下のチェックを行って、Apache ウェブサーバーが実行されていて、アクセス可能であるかどうかを確認します。

- ウェブサーバーが実行されていますか？

httpd が有効であることは、次のコマンドを実行して確認できます。

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

httpd プロセスが実行されていない場合は、[LAMP サーバーを準備するには](#)に記載されているステップを繰り返します。

- ファイアウォールは正しく設定されていますか？

インスタンスのセキュリティグループに、ポート 80 での HTTP ラフィックを許可するルールが含まれていることを確認します。詳細については、[「セキュリティグループにルールを追加する」](#)を参照してください。

HTTPS を使用してサーバーに接続できない

以下のチェックを行って、Apache ウェブサーバーが HTTPS をサポートするように設定されているかどうかを確認します。

- ウェブサーバは正しく設定されていますか？

Apache をインストールすると、サーバーは HTTP トラフィック用に設定されます。HTTPS をサポートするには、サーバーで TLS を有効にし、SSL 証明書をインストールします。詳細については、[「チュートリアル: AL2023 で SSL/TLS を設定する」](#)を参照してください。

- ファイアウォールは正しく設定されていますか？

インスタンスのセキュリティグループに、ポート 443 で HTTPS トラフィックを許可するルールが含まれていることを確認します。詳細については、[「Linux インスタンスのインバウンドトラフィックを許可する」](#)を参照してください。

関連トピック

インスタンスへのファイルの転送、またはウェブサーバーへの WordPress ブログのインストールの詳細については、次のドキュメントを参照してください。

- [「Amazon EC2 ユーザーガイド」の WinSCP を使用して Linux インスタンスにファイルを転送する](#)。Amazon EC2
- [「Amazon EC2 ユーザーガイド」の「SCP クライアントを使用して Linux インスタンスにファイルを転送する Amazon EC2」](#)。
- [チュートリアル: AL2023 で WordPress ブログをホストする](#)

このチュートリアルで使用されているコマンドおよびソフトウェアの詳細については、次のウェブページを参照してください。

- Apache ウェブサーバー: <http://httpd.apache.org/>
- MariaDB データベースサーバー: <https://mariadb.org/>
- PHP プログラミング言語: <http://php.net/>

ウェブサーバーのドメイン名の登録、または、既存のドメイン名をこのホストに移す方法についての詳細は、『Amazon Route 53 デベロッパーガイド』の[「Amazon Route 53 のドメインとサブドメインの作成と移行」](#)を参照してください。

チュートリアル: AL2023 で SSL/TLS を設定する

Secure Sockets Layer/Transport Layer Security (SSL/TLS) は、ウェブサーバーとウェブクライアントの間に、転送中のデータが傍受されないように保護する、暗号化されたチャネルを確立します。このチュートリアルでは、AL2023 および Apache ウェブサーバーを使用する EC2 インスタンスに SSL/TLS のサポートを手動で追加する方法について説明します。このチュートリアルでは、ロードバランサーを使用していないことを前提としています。Elastic Load Balancing を使用している場合

は、代わりに [AWS Certificate Manager](#) の証明書を使用して、ロードバランサーで SSL オフロードを設定できます。

歴史的経緯から、ウェブの暗号化は、単純に SSL と呼ばれることが少なくありません。ウェブブラウザでは今でも SSL がサポートされていますが、後継プロトコルである TLS プロトコルの方が攻撃を受けにくくなります。AL2023 は、デフォルトですべてのバージョンの SSL に対するサーバー側のサポートを無効にします。[セキュリティ標準化団体](#)は、TLS 1.0 は安全でないとみなしています。TLS 1.0 および TLS 1.1 は、2021 年 3 月に正式に[非推奨になりました](#)。このチュートリアルは、TLS 1.2 を有効にすることを前提としたガイダンスです。TLS 1.3 は 2018 年に確定され、基盤となる TLS ライブラリ (このチュートリアルの OpenSSL) がサポートされ、有効になっている限り、AL2 で使用できます。[クライアントは 2023 年 6 月 28 日までに TLS 1.2 以降をサポートしている必要があります](#)。最新の暗号化基準の詳細については、「[RFC 7568](#)」および「[RFC 8446](#)」を参照してください。

このチュートリアルでは、現代のウェブ暗号化を単に TLS と呼びます。

Important

これらの手順は、AL2023 で使用することを目的としています。異なるディストリビューションを実行している EC2 インスタンス、または古いバージョンの Amazon Linux を実行しているインスタンスをセットアップしようとする、このチュートリアルの一部の手順が上手いかないことがあります。Ubuntu については、[Ubuntu 上の OpenSSL](#) に関する Ubuntu コミュニティドキュメントを参照してください。Red Hat Enterprise Linux については、以下を参照してください。[Apache HTTP Web サーバーの設定](#)。その他のディストリビューションについては、それぞれのドキュメントを参照してください。

Note

または、(AWS Certificate Manager ACM) for AWS Nitro Enclaves を使用することもできます。これは、Nitro Enclaves で Amazon EC2 インスタンスで実行されているウェブアプリケーションおよびサーバーでパブリックおよびプライベート SSL/TLS 証明書を使用できるようにする AWS エンクレーブアプリケーションです。Nitro Enclaves は、SSL/TLS 証明書やプライベートキーなどの機密性の高いデータを保護し、安全に処理するために、分離されたコンピューティング環境を作成できる Amazon EC2 の機能です。

Nitro Enclaves 向け ACM では、Amazon EC2 Linux インスタンスで実行する nginx を使用することで、プライベートキーの作成、証明書とプライベートキーの配布、および証明書の更新を実行します。

Nitro Enclaves 向け ACM を使用するには、エンクレーブ対応の Linux インスタンスを使用する必要があります。

詳細については、[AWS 「Nitro Enclaves ユーザーガイド」](#)の[AWS Certificate Manager 「Nitro Enclaves とは」](#) および [「for Nitro Enclaves」](#) を参照してください。AWS

内容

- [前提条件](#)
- [ステップ 1: サーバーで TLS を有効にする](#)
- [ステップ 2: CA 署名証明書を取得する](#)
- [ステップ 3: セキュリティ設定をテストして強化する](#)
- [トラブルシューティング](#)

前提条件

このチュートリアルを開始する前に、次のステップを完了してください。

- EBS-backed AL2023 インスタンスを起動します。詳細については、「[Amazon EC2 での AL2023](#)」を参照してください。
- インスタンスが以下の TCP ポートで接続を受け付けるようにセキュリティグループを設定します。
 - SSH (ポート 22)
 - HTTP (ポート 80)
 - HTTPS (ポート 443)

詳細については、「[Amazon EC2 ユーザーガイド](#)」の「[Linux インスタンスのインバウンドトラフィックの承認](#)」を参照してください。Amazon EC2

- Apache ウェブサーバーをインストールします。step-by-step 手順については、「[」](#)を参照してください。[チュートリアル: AL2023 に LAMP サーバーをインストールする](#)。必要なのは httpd パッケージおよび対応する従属コンポーネントのみです。PHP および MariaDB に関連する手順は無視してかまいません。
- ウェブサイトの識別と認証を行うため、TLS の公開鍵基盤 (PKI) ではドメインネームシステム (DNS) を使用します。EC2 インスタンスを使用してパブリックウェブサイトをホストするには、ウェブサーバーのドメイン名を登録するか、既存のドメイン名を Amazon EC2 ホストに移す必要

があります。これについては、ドメイン登録および DNS ホスティングに関するサードパーティのサービスが多数存在します。[Amazon Route 53](#) を使用することもできます。

ステップ 1: サーバーで TLS を有効にする

この手順では、自己署名デジタル証明書を使用して AL2023 で TLS を設定するプロセスについて説明します。

Note

自己署名証明書はテスト用であり、本稼働環境では使用できません。インターネットに自己署名証明書を公開すると、サイトへの訪問者にセキュリティ警告が表示されます。

サーバーで TLS を有効にするには

1. インスタンスに接続し、Apache が実行されていることを確認します。詳細については、「[AL2023 インスタンスへの接続](#)」を参照してください。

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

返される値が「enabled」でない場合、Apache を起動し、システムブート時に毎回起動されるように設定します。

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

2. すべてのソフトウェアパッケージが最新の状態であることを確認するため、インスタンスでソフトウェアの更新を実行します。この処理には数分かかりますが、最新の更新とバグ修正を確実に適用することが重要です。

Note

-y オプションを指定すると、確認メッセージを表示せずに更新をインストールします。インストール前に更新を検査する場合は、このオプションを省略できます。

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



```
DBBTb211T3JnYW5pemF0aW9uMR8wHQYDVQQLDBZTb211T3JnYW5pemF0aW9uYWxV
bm10MRkwFwYDVQQDDDBBpcC0xNzItMzEtMjAtMjMMSQwIgyYJKoZiIhvcNAQkBFhVY
...
z5rRUE/XzxRLBZ0oWZpNWTXJkQ3uFYH6s/sBwtHpKKZMz0vDedREjNKAvk4ws6F0
CuIjvubtUysVYqoMVPQ971deakHWeRMiEJFXg6kZZ0vrGvwnKoMh3D1K44D9d1U3
WanXWehT6FiSZvB4sTEXXJN2jdw8g+sHGnZ8zC0sc1knYhHrCVD2vnB1ZJKSZvak
3ZazhBxtQSukFM0nWPP2a0DMMFGYUHOd0BQE8sBJxg==
-----END CERTIFICATE-----
```

ファイル名および拡張子は利便性のためであり、機能には影響しません。例えば、`cert.crt` または `cert.pem` などのファイル名で証明書を呼び出すことができます。ただし、`ssl.conf` ファイルの関連ディレクティブが同じ名前を使用している場合に限りです。

Note

デフォルトの TLS ファイルを独自にカスタマイズしたファイルに置き換える場合は、PEM 形式であることを確認してください。

4. Apache を再起動します。

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

Note

前述のとおり、TCP 443 番ポートが EC2 インスタンスでアクセス可能であることを確認してください。

5. Apache ウェブサーバーではポート 443 経由で HTTPS (セキュア HTTP) がサポートされるようになっています。これをテストするには、ブラウザの URL バーに、**https://** というプレフィックスを指定して、EC2 インスタンスの IP アドレスまたは完全修飾ドメイン名を入力します。

信頼されていない自己署名ホスト証明書を使用してサイトに接続しようとしているため、ブラウザには一連のセキュリティ警告が表示されることがあります。この警告を無視し、サイトに進みます。

サーバーで TLS を正しく設定できていれば、Apache のデフォルトのテストページが開きます。これで、ブラウザとサーバーの間でやり取りされるすべてのデータが暗号化されるようになります。

Note

サイト訪問者に対して警告画面が表示されないようにするには、暗号化だけではなく、サイト所有者のパブリック認証を行うための信頼された CA 署名証明書を取得する必要があります。

ステップ 2: CA 署名証明書を取得する

CA 署名証明書を取得するには、次の手順に従います。

- プライベートキーから証明書署名リクエスト (CSR) を作成します。
- 作成した CSR を認証機関 (CA) に送信します。
- 署名付きホスト証明書を入手する
- 証明書を使用するように Apache を設定します

自己署名 TLS X.509 ホスト証明書は、暗号化技術上は CA 署名証明書と同じです。これらの相違は数学的なものではなく、社会的なものです。CA では、最低でもドメイン所有権を検証してから申請者に証明書を発行することを保証しています。そのため、各ウェブブラウザには、ブラウザベンダーが信頼する CA のリストが含まれています。X.509 証明書は主に、プライベートサーバーキーに対応するパブリックキーと、このパブリックキーに暗号で関連付けられている CA による署名で構成されています。HTTPS 経由でブラウザがウェブサーバーに接続すると、サーバーは、信頼された CA のリストをブラウザが確認できるように、証明書を提示します。Signer がリストに含まれている場合や、他の信頼された署名者の信頼チェーンを通じてアクセス可能である場合、ブラウザはサーバーと、高速暗号化データチャネルのネゴシエーションを行い、ページをロードします。

証明書には、リクエストの確認作業が必要であり、一般的に費用がかかるため、各社を比較することをお勧めします。いくつかの CA では、基本レベル証明書が無料で提供されます。これらの CA で最も注目すべきは [Let's Encrypt](#) プロジェクトです。このプロジェクトでは、証明書の作成および更新プロセスの自動化もサポートしています。Let's Encrypt 証明書の使用の詳細については、「[Certbot の取得](#)」を参照してください。

商業グレードのサービスを提供する予定がある場合は、[AWS Certificate Manager](#) は良い選択肢です。

ホスト証明書の基盤にはキーがあります。2019 年時点で、[政府](#)および[業界グループ](#)は、2030 年まで、ドキュメントを保護するための RSA キーに 2048 ビットの最小キー (モジユロ) サイズを使用す

ることを推奨しています。AL2023 で OpenSSL によって生成されるデフォルトのモジュラスサイズは 2048 ビットで、CA 署名証明書での使用に適しています。次の手順では、モジュラスサイズを大きくする、別の暗号化アルゴリズムを使用するなど、キーのカスタマイズが必要な場合のオプションのステップを提供しています。

⚠ Important

CA 署名ホスト証明書を取得するための手順は、登録およびホスト済みの DNS ドメインを所有している場合を除き、使用しません。

CA 署名証明書を取得するには

1. インスタンスに接続して、`/etc/pki/tls/private/` に移動します。サーバーの TLS 用プライベートキーは、このディレクトリに格納されます。既存のホストキーを使用して CSR を生成する場合は、ステップ 3 に進みます。インスタンスへの接続の詳細については、「」を参照してください。[AL2023 インスタンスへの接続](#)
2. (オプション) 新しいプライベートキーを生成します。キー設定のいくつかのサンプルを次に示します。生成されたキーのどれもウェブサーバーで機能しますが、実装されるセキュリティの強度とタイプはそれぞれ異なります。
 - 例 1: デフォルトの RSA ホストキーを作成します。結果として生成されるファイル **custom.key** が、2048 ビットの RSA プライベートキーです。

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key
```

- 例 2: これより大きなモジュラスサイズを使用して、より強力な RSA キーを作成します。結果として生成されるファイル **custom.key** が、4096 ビットの RSA プライベートキーです。

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key 4096
```

- 例 3: パスワードで保護された 4096 ビット暗号化 RSA キーを作成します。結果のファイル、**custom.key** は、AES-128 暗号で暗号化された 4096 ビットの RSA プライベートキーです。

⚠ Important

キーを暗号化するとセキュリティを強化できますが、暗号化キーにはパスワードが必要であるため、暗号化に依存するサービスを自動的に開始することはできません。こ

のキーを使用するたびに、SSH 接続でパスワード (前述の例では、"abcde12345") を指定する必要があります。

```
[ec2-user ~]$ sudo openssl genrsa -aes128 -passout pass:abcde12345 -out custom.key 4096
```

- 例 4: 非 RSA 暗号を使用してキーを作成します。RSA 暗号化は、2 つの大きな素数の積に基づくパブリックキーのサイズのために、比較的遅くなる可能性があります。ただし、非 RSA 暗号化方式を使用する TLS 用のキーを作成することも可能です。同等レベルのセキュリティを提供する場合は、楕円曲線の計算に基づいたキーのほうが小さく計算処理も高速です。

```
[ec2-user ~]$ sudo openssl ecparam -name prime256v1 -out custom.key -genkey
```

結果は、prime256v1 (OpenSSL でサポートされる "名前付き曲線") を使用した 256 ビットの楕円曲線プライベートキーです。暗号化強度は ([NIST](#) によると) 2048 ビットの RSA キーよりやや優れています。

Note

すべての CAs が RSA elliptic-curve-based キーと同じレベルのキーのサポートを提供するわけではありません。

新しいプライベートキーには、制限の厳しい所有権とアクセス権を設定します (所有者 = root、グループ = root、所有者のみの読み取り/書き込み)。コマンドは次の例のようになります。

```
[ec2-user ~]$ sudo chown root:root custom.key
[ec2-user ~]$ sudo chmod 600 custom.key
[ec2-user ~]$ ls -al custom.key
```

上記のコマンドにより、次のような結果が得られます。

```
-rw----- root root custom.key
```

適切なキーを作成し、設定できたら、CSR を作成できます。

3. 好みのキーを使用して CSR を作成します。次の例では **custom.key** を使用しています。

```
[ec2-user ~]$ sudo openssl req -new -key custom.key -out csr.pem
```

OpenSSL によりダイアログが開かれ、次の表に示されている情報の入力が必要です。基本的なドメイン検証済みホスト証明書については、[共通名] 以外のフィールドはすべてオプションです。

名前	説明	例
国名	2 文字の ISO 略称 (国名コード)。	US (= 米国)
州名	あなたが所属する組織の所在地の州または県。省略不可です。	ワシントン
市区町村	市など、組織の場所。	シアトル
組織名	組織の正式名称。組織名は、省略不可です。	Example Corp
部門名	組織に関する追加情報 (存在する場合)。	Example Dept
共通名	この値は、ユーザーがブラウザに入力する必要のあるウェブアドレスと正確に一致します。通常、これはプレフィックス付きのホスト名またはエイリアスによるドメイン名 (www.example.com の形式) を意味します。自己署名証明書を使用し、DNS 解決なしでテストを行う場合、共通名の構成要素はホスト名のみになる場合があります。CA では、 *.example.com などのワイルドカード名を許容する、よりコストの高い証明書も用意されています。	www.example.com
E メールアドレス	サーバー管理者の E メールアドレス。	someone@example.com

最後に、OpenSSL により、オプションのチャレンジパスワードが求められます。このパスワードは CSR と、ユーザーと CA の間のトランザクションのみに適用されるため、このフィールド

と、もう 1 つのオプションフィールドである、オプションの会社名については、CA の推奨事項に従ってください。CSR のチャレンジパスワードは、サーバー操作には影響しません。

結果として生成されるファイル `csr.pem` には、パブリックキー、パブリックキーのデジタル署名、入札したメタデータが含まれています。

4. CA に CSR を送信します。この作業は通常、テキストエディタで CSR ファイルを開く動作と、内容をウェブフォームにコピーする動作で構成されています。このとき、証明書に適用する 1 つ以上のサブジェクト代替名 (SAN) を指定するように求められることがあります。共通名が `www.example.com` の場合、有効な SAN は `example.com` になります (逆も同様です)。サイトへの訪問者がこれら名前のいずれかを入力すると、エラーなしの接続が提示されます。CA のウェブフォームで許可される場合は、SAN のリストに共通名を含めます 一部の CA では自動的に含められます。

リクエストが承認されると、CA によって署名された新しいホスト証明書が届きます。CA の信頼チェーンを完成するために必要な、追加の証明書が含まれている中間証明書ファイルをダウンロードするよう指示されることもあります。

Note

多様な用途向けに複数の形式のファイルを送信してくる CA もあります。このチュートリアルでは、PEM 形式の証明書ファイルのみ使用してください。PEM 形式のファイルには通常、`.pem` または `.crt` ファイル拡張子が使用されます (ただし、常にこれらの拡張子が使用されるわけではありません)。どのファイルを使用すべきかわからない場合は、テキストエディタでファイルを開き、以下の行で始まる 1 つ以上のブロックを含むファイルを見つけてください。

```
- - - - -BEGIN CERTIFICATE - - - - -
```

ファイルの末尾は次のような行になっている必要があります。

```
- - - - -END CERTIFICATE - - - - -
```

以下に示すように、コマンドラインでファイルをテストすることもできます。

```
[ec2-user certs]$ openssl x509 -in certificate.crt -text
```


これらの行がファイルに表示されていることを確認してください。 .p7b、 .p7c、または類似のファイル拡張子で終了するファイルは使用しないでください。

5. 新しい CA 署名証明書と任意の中間証明書を /etc/pki/tls/certs ディレクトリに配置します。

Note

EC2 インスタンスに新しい証明書をアップロードする方法は複数ありますが、最も簡単でわかりやすい方法は、テキストエディタ (vi、 nano、またはメモ帳など) をローカルコンピュータとインスタンスの両方で開いて、両者の間でファイルの内容をコピーして貼り付けることです。EC2 インスタンス内でこれらの操作を実行する際には、root [sudo] アクセス許可が必要です。こうすることで、許可やパスに問題があるかどうかをすぐに確認できます。ただし、内容をコピーする際に行を追加したり、内容を変更したりしないでください。

/etc/pki/tls/certs ディレクトリ内から、ファイルの所有権、グループ、およびアクセス許可の設定が、制限の厳しい AL2023 のデフォルト (所有者 = root、グループ = root、所有者のみの読み取り/書き込み) と一致していることを確認します。以下の例では、使用するコマンドを示しています。

```
[ec2-user certs]$ sudo chown root:root custom.crt
[ec2-user certs]$ sudo chmod 600 custom.crt
[ec2-user certs]$ ls -al custom.crt
```

これらのコマンドによって、次の結果が得られます。

```
-rw----- root root custom.crt
```

中間証明書ファイルのアクセス権は、比較的厳しくありません (所有者 = root、グループ = root、所有者による書き込み可、グループによる読み取り可、その他による読み取り可)。以下の例では、使用するコマンドを示しています。

```
[ec2-user certs]$ sudo chown root:root intermediate.crt
[ec2-user certs]$ sudo chmod 644 intermediate.crt
[ec2-user certs]$ ls -al intermediate.crt
```

これらのコマンドによって、次の結果が得られます。

```
-rw-r--r-- root root intermediate.crt
```

6. CSR の作成に使用したプライベートキーを `/etc/pki/tls/private/` ディレクトリに配置します。

Note

EC2 インスタンスにカスタムキーをアップロードする方法は複数ありますが、最も簡単でわかりやすい方法は、テキストエディタ (vi、nano、メモ帳など) をローカルコンピュータとインスタンスの両方で開いて、両者の間でファイルの内容をコピーして貼り付けることです。EC2 インスタンス内でこれらの操作を実行する際には、`root [sudo]` アクセス許可が必要です。こうすることで、許可やパスに問題があるかどうかをすぐに確認できます。ただし、内容をコピーする際に行を追加したり、内容を変更したりしないでください。

`/etc/pki/tls/private` ディレクトリ内から、次のコマンドを使用して、ファイルの所有権、グループ、アクセス許可の設定が制限の厳しい AL2023 のデフォルト (所有者 = root、グループ = root、所有者のみの読み取り/書き込み) と一致することを確認します。

```
[ec2-user private]$ sudo chown root:root custom.key  
[ec2-user private]$ sudo chmod 600 custom.key  
[ec2-user private]$ ls -al custom.key
```

これらのコマンドによって、次の結果が得られます。


```
-rw----- root root custom.key
```

7. 新しい証明書とキーファイルに合わせるには、`/etc/httpd/conf.d/ssl.conf` を編集します。
 - a. CA 署名のホスト証明書のパスとファイル名を Apache の `SSLCertificateFile` ディレクティブで指定します。

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

- b. 中間証明書ファイル (この例では `intermediate.crt`) を受け取ったら、Apache の `SSLCACertificateFile` ディレクティブを使用して、次のファイルのパスとファイル名を指定します。

```
SSLCACertificateFile /etc/pki/tls/certs/intermediate.crt
```

 Note

一部の CA では、ホスト証明書と中間証明書を組み合わせて 1 つのファイルを作成するため、この `SSLCACertificateFile` ディレクティブは必要ありません。CA が提供している手順を参照してください。

- c. プライベートキー (この例では `custom.key`) のパスとファイル名を Apache の `SSLCertificateKeyFile` ディレクティブで指定します。

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


8. `/etc/httpd/conf.d/ssl.conf` を保存して、Apache を再起動します。

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

9. サーバーをテストするには、ブラウザの URL バーにドメイン名を入力し、プレフィックス `https://` を指定します。ブラウザによって、エラーが生成されることなく、HTTPS 経由でテストページがロードされます。

ステップ 3: セキュリティ設定をテストして強化する

TLS が運用可能になりパブリックに公開されたら、実際の安全性をテストする必要があります。セキュリティセットアップの詳細な分析を無料で行うことのできる [Qualys SSL Labs](#) などのオンラインサービスを使用すると簡単です。その結果に基づき、受け入れるプロトコル、優先する暗号化方式、除外する暗号化方式を制御することによって、デフォルトのセキュリティ設定を強化するかどうかを決定できます。詳細については、「[Qualys のスコアの計算方法](#)」を参照してください。

 Important

サーバーのセキュリティを確保するには、実際のテストが非常に重要です。小さな設定エラーによって、深刻なセキュリティ侵害やデータの損失が生じる可能性があります。調査

や新たな脅威に応じて、推奨されるセキュリティ管理方法は常に変化するため、適切なサーバー管理を行うには、定期的なセキュリティ監査が不可欠です。

[Qualys SSL Labs](#) のサイトで、サーバーの完全修飾ドメイン名を `www.example.com` という形式で入力します。約 2 分後に、サイトに関するグレード (A から F) と、結果の詳細な内訳が届きます。次の表は、AL2023 のデフォルトの Apache 設定と同じ設定で、デフォルトの Certbot 証明書を持つドメインのレポートをまとめたものです。

総合評価	B
証明書	100%
プロトコルサポート	95%
キー交換	70%
暗号強度	90%

概要は設定がほとんど正常であることを示していますが、詳細レポートでは、いくつかの潜在的な問題が指摘されています。重大度の高い順に以下に示します。

RC4 暗号は、特定の古いブラウザでの使用がサポートされています。暗号は、暗号化アルゴリズムの計算の中核です。TLS データストリームの暗号化に使用される高速の暗号化方式である RC4 は、いくつかの**重大な脆弱性**を持つことで知られています。従来のブラウザをサポートするもっともな理由がない限り、この暗号化方式を無効にする必要があります。

x旧バージョンの TLS がサポートされています。設定では TLS 1.0 (すでに廃止されています) と TLS 1.1 (廃止予定) がサポートされています。2018 年以降は、TLS 1.2 のみ推奨されています。

前方秘匿性は完全にサポートされていません。**前方秘匿性**は、プライベートキーから派生した一時 (エフェメラル) セッションキーを使用して暗号化を行う、アルゴリズムの機能です。これは、攻撃者がウェブサーバーの長期的なプライベートキーを所有していても、HTTPS データを復号できないことを意味します。

TLS 設定を修正し、将来への対応性を確保するには

1. 設定ファイル `/etc/httpd/conf.d/ssl.conf` を開き、行頭に `#` を付けて以下の行をコメントアウトしてください。

```
#SSLProtocol all -SSLv3
```

2. 次のディレクティブを追加します。

```
#SSLProtocol all -SSLv3  
SSLProtocol -SSLv2 -SSLv3 -TLSv1 -TLSv1.1 +TLSv1.2
```

このディレクティブにより、SSL バージョン 2、3、および TLS バージョン 1.0、1.1 が明示的に無効化されます。これで、サーバーでは、TLS 1.2 以外を使用した、クライアントとの暗号化された接続の受け入れが拒否されます。ディレクティブに含める指定が多くなるほど、サーバーの動作に対する設定内容が明確に伝わります。

Note

このようにして、TLS バージョン 1.0 および 1.1 を無効にすると、ごく一部の古くなったウェブブラウザによるサイトへのアクセスがブロックされるようになります。

許可された暗号のリストを変更するには

1. 設定ファイル `/etc/httpd/conf.d/ssl.conf` で、**SSLCipherSuite** ディレクティブを含むセクションを探し、行頭に `#` を付けて既存の行をコメントアウトします。

```
#SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5
```

2. 明示的な暗号スイートと、前方秘匿性を優先し、安全でない暗号を禁止する暗号順序を指定します。ここで使用される `SSLCipherSuite` ディレクティブは、[Mozilla SSL Configuration Generator](#) の出力に基づいています。これは、お客様のサーバーで実行されている特定のソフトウェアに合わせて TLS 設定を調整します。(詳細については、Mozilla の有益なリソース「[Security/Server Side TLS](#)」を参照してください。) まず、以下のコマンドの出力を使用して、Apache と OpenSSL のバージョンを確認します。

```
[ec2-user ~]$ yum list installed | grep httpd
```

```
[ec2-user ~]$ yum list installed | grep openssl
```

例えば、返された情報が Apache 2.4.34 および OpenSSL 1.0.2 である場合、これをジェネレーターに入力します。"最新" 互換性モデルを選択すると、`SSLCipherSuite` ディレクティブが作

成されます。このディレクティブは、積極的にセキュリティを適用しますが、ほとんどのブラウザで使用できます。ソフトウェアで最新互換性モデルがサポートされていない場合は、ソフトウェアを更新するか、"中間"の構成を選択します。

```
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
```

選択された暗号化方式の名前には、ECDHE が含まれています (Elliptic Curve Diffie-Hellman Ephemeral の略語です)。ephemeral は前方秘匿性を示します。また、これらの暗号化方式では、RC4 はサポートされていません。

デフォルトや、内容が見えない簡単なディレクティブに依存するのではなく、暗号化方式の明示的なリストを使用することをお勧めします。

生成されたディレクティブを `/etc/httpd/conf.d/ssl.conf` にコピーします。

Note

ここでは読みやすくするために数行に分けて示していますが、このディレクティブは、`/etc/httpd/conf.d/ssl.conf` にコピーする際に、暗号化方式名の間をコロンのみ (スペースなし) で区切り、1 行に指定する必要があります。

- 最後に、次の行について、行頭の `#` を削除してコメント解除します。

```
#SSLHonorCipherOrder on
```

このディレクティブは、(この場合) 前方秘匿性をサポートするものも含めて、ランクの高い暗号化方式を優先するようサーバーに強制します。このディレクティブが有効になると、サーバーは、セキュリティの弱い暗号化方式に戻る前に、セキュリティが強力な接続を確立しようとしています。

これらの手順がいずれも完了したら、変更内容を `/etc/httpd/conf.d/ssl.conf` に保存し、Apache を再起動します。

[Qualys SSL Labs](#) でドメインをもう一度テストすると、RC4 脆弱性やその他の警告は解決し、次のようなサマリレポートが出力されます。

総合評価	A
証明書	100%
プロトコルサポート	100%
キー交換	90%
暗号強度	90%

OpenSSL の更新ごとに、新しい暗号化方式が導入され古い暗号化方式のサポートが削除されます。EC2 AL2023 インスタンスを保管し up-to-date、[OpenSSL](#)からのセキュリティに関する発表を監視し、テクニカルメディアで新しいセキュリティの悪用に関するレポートに注意してください。

トラブルシューティング

- パスワードを指定しないと Apache ウェブサーバーが起動しません

これは、パスワードで保護された暗号化プライベート サーバー キーをインストールした場合は正常な動作です。

暗号化とパスワードの要件をキーから削除できます。デフォルトディレクトリに `custom.key` という暗号化プライベート RSA キーがあり、そのパスワードが `abcde12345` であるとする、EC2 インスタンスで次のコマンドを実行し、このキーの非暗号化バージョンを生成してください。

```
[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
```

パスワードが求められずに Apache が起動するようになります。

- `sudo dnf install -y mod_ssl` を実行するとエラーが発生します。

SSL に必要なパッケージをインストールすると、次のようなエラーが表示されることがあります。

```
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
```

これは通常、EC2 インスタンスが AL2023 を実行していないことを意味します。このチュートリアルでは、公式の AL2023 AMI から新しく作成されたインスタンスのみをサポートします。

チュートリアル: AL2023 で WordPress ブログをホストする

次の手順は、AL2023 インスタンスに WordPress ブログをインストール、設定、保護するのに役立ちます。このチュートリアルは、WordPress ブログをホストするウェブサーバーを完全に制御できるという点で Amazon EC2 を使用する方法を紹介するのに役立ちます。これは、従来のホスティングサービスでは一般的ではありません。

サーバーに対するソフトウェアパッケージの更新とセキュリティパッチの維持は、お客様の責任となります。ウェブサーバー設定と直接やり取りする必要のない、より自動化された WordPress インストールのために、この AWS CloudFormation サービスでは WordPress、すぐに開始できるテンプレートも用意されています。詳細については、AWS CloudFormation ユーザーガイドの「[開始方法](#)」を参照してください。Windows インスタンスで WordPress ブログをホストする場合は、[Amazon EC2 ユーザーガイド](#)の「[Amazon EC2 Windows インスタンスに WordPress ブログをデプロイする](#)」を参照してください。Amazon EC2 分離されたデータベースで高可用性ソリューションが必要な場合は、「[デベロAWS Elastic Beanstalk ツパーガイド](#)」の「[高可用性 WordPress ウェブサイトのデプロイ](#)」を参照してください。

Important

これらの手順は、AL2023 での使用を目的としています。その他のディストリビューションの情報については、各ドキュメントを参照してください。このチュートリアルの多くの手順は、Ubuntu インスタンスには使用できません。Ubuntu インスタンス WordPress へのインストールについては、Ubuntu ドキュメント [WordPress](#) の「」を参照してください。を使用して [CodeDeploy](#)、Amazon Linux、macOS、または Unix システムでこのタスクを実行することもできます。

トピック

- [前提条件](#)
- [のインストール WordPress](#)
- [次のステップ](#)
- [ヘルプ! パブリック DNS 名が変更されたため、ブログが壊れました](#)

前提条件

WordPress ブログのホストに使用しているインスタンスに Elastic IP アドレス (EIP) を関連付けることを強くお勧めします。これにより、インスタンスのパブリック DNS アドレスが変更されて、インストールが破損することを防止できます。ドメイン名を所有していてそのドメインをブログに使用する場合、EIP アドレスをポイントするようにドメイン名の DNS レコードを更新できます (これを行うには、ドメイン名レジストラにお問い合わせください)。実行中のインスタンスに関連付けられた EIP アドレスを無料で 1 つ取得できます。詳細については、「Amazon EC2 ユーザーガイド」の「[Elastic IP アドレス](#)」を参照してください。[チュートリアル: AL2023 に LAMP サーバーをインストールする](#) チュートリアルでは、セキュリティグループで HTTP および HTTPS トラフィックを許可するように設定する手順や、ウェブサーバー用にファイル許可が正しく設定されていることを確認する手順も示します。セキュリティグループにルールを追加する方法については、「セキュリティグループに[ルールを追加する](#)」を参照してください。

ブログのドメイン名がまだない場合は、Route 53 にドメイン名を登録し、そのドメイン名にインスタンスの EIP アドレスを関連付けることができます。詳細については、Amazon Route 53 デベロッパーガイドの「[Amazon Route 53 を使用したドメイン名の登録](#)」を参照してください。

のインストール WordPress

インスタンスに接続し、WordPress インストールパッケージをダウンロードします。インスタンスへの接続の詳細については、[AL2023 インスタンスへの接続](#)を参照してください。

1. 次のコマンドを使用してこれらのパッケージをダウンロードしてインストールします。

```
dnf install wget php-mysqlnd httpd php-fpm php-mysqlcli mariadb105-server php-json
php php-devel -y
```

2. 出力に同様の警告文が表示されることがあります (バージョンは表示されるタイミングによって異なる場合があります)。

```
WARNING:
```

```
A newer release of "Amazon Linux" is available.
```

```
Available Versions:
```

```
dnf update --releasever=2023.0.20230202
```

```
Release notes:
```

```
https://aws.amazon.com
```

```
Version 2023.0.20230204:
```

```
Run the following command to update to 2023.0.20230204:
```

```
dnf update --releasever=2023.0.20230204 ... etc
```

ベストプラクティスとして、OS を up-to-date 可能な限り維持することをお勧めしますが、環境内で競合が発生しないように、各バージョンを繰り返し実行することをお勧めします。ステップ 1 に記載されている前述のパッケージのインストールが失敗した場合は、リストされているいずれかの新しいリリースに更新して、再試行する必要がある場合があります。

3. `wget` コマンドを使用して最新の WordPress インストールパッケージをダウンロードします。次のコマンドを実行すると、最新リリースが必ずダウンロードされます。

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

4. インストールパッケージを解凍します。インストールフォルダは、`wordpress` という名前のフォルダに解凍されます。

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

WordPress インストール用のデータベースユーザーとデータベースを作成するには

WordPress インストールでは、ブログ投稿やユーザーコメントなどの情報をデータベースに保存する必要があります。この手順を実行すると、ブログのデータベースを作成するのに役立ち、このデータベースに対して情報の読み取りや保存を許可されたユーザーにも有用です。

1. データベースおよびウェブサーバーを起動します。

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

2. データベースサーバーに root ユーザーとしてログインします。メッセージが表示されたら、データベース root パスワードを入力します。これは通常の root システムパスワードと異なることもあれば、データベースサーバーのセキュリティ確保を実行していない場合は、空のときもあります。

データベースサーバーのセキュリティを確保していない場合、セキュリティ確保を行うことは重要です。詳細については、[ステップ 3: データベースサーバーをセキュリティで保護する](#)「(AL2023)」を参照してください。

```
[ec2-user ~]$ mysql -u root -p
```

3. MySQL データベースのユーザーとパスワードを作成します。WordPress インストールでは、これらの値を使用して MySQL データベースと通信します。一意のユーザー名とパスワードを入力して、次のコマンドを入力します。

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

ユーザー用に強力なパスワードを作成してください。パスワードに一重引用符 (') を使用しないでください。この文字は前述のコマンドを中断させるためです。既存のパスワードを再利用しないでください。また、このパスワードは必ず安全な場所に保管してください。

4. データベースを作成します。wordpress-db など、データベースにはわかりやすい名前を使用します。

Note

次のコマンドのデータベース名を囲む区切り記号は、「バックティック」と呼ばれています。バックティック (`) キーは通常、標準キーボードの Tab キーの上に配置されています。バックティックは必ずしも必要ではありませんが、データベース名では使用できない文字 (ハイフンなど) の代わりに使用できます。

```
CREATE DATABASE `wordpress-db`;
```

5. 前に作成した WordPress ユーザーにデータベースのフル権限を付与します。

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

6. すべての変更を有効にするため、データベース権限をフラッシュします。

```
FLUSH PRIVILEGES;
```

7. mysql クライアントを終了します。

```
exit
```

wp-config.php ファイルの作成と編集を行うには

WordPress インストールフォルダには、というサンプル設定ファイルが含まれていますwp-config-sample.php。この手順では、このファイルをコピーして、特定の構成に合うように編集します。

1. wp-config-sample.php ファイルを wp-config.php という名前でコピーします。この操作を実行すると、新しい構成ファイルが作成され、元のファイルがバックアップとしてそのまま保持されます。

```
[ec2-user ~]$ cp wordpress/wp-config-sample.php wordpress/wp-config.php
```

2. お好みのテキストエディタ (wp-config.php、nano など) を使って vim ファイルを編集し、インストール用の値を入力します。お好みのテキストエディタがない場合、nano が初心者に適しています。

```
[ec2-user ~]$ nano wordpress/wp-config.php
```

- a. DB_NAME を定義する行を探して、database_name_here を [Step 4 の WordPress インストール用のデータベースユーザーとデータベースを作成するには](#) で作成したデータベース名に変更します。

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

- b. DB_USER を定義する行を探して、username_here を [Step 3 の WordPress インストール用のデータベースユーザーとデータベースを作成するには](#) で作成したデータベースユーザーに変更します。

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

- c. DB_PASSWORD を定義する行を探して、password_here を [Step 3 の WordPress インストール用のデータベースユーザーとデータベースを作成するには](#) で作成した強力なパスワードに変更します。

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

- d. Authentication Unique Keys and Salts というセクションを見つけます。これらの KEY との SALT 値は、WordPress ユーザーがローカルマシンに保存するブラウザ Cookie に暗号化レイヤーを提供します。基本的に、ここで長くてランダムな値を指定すると、サイトのセキュリティが向上します。<https://api.wordpress.org/secret-key/1.1/salt/> にアクセスして、ランダムに生成されるキーセット値を取得し、wp-config.php ファイルにコピーして貼り付けることができます。PuTTY 端末にテキストを貼り付けるには、テキストを貼り付ける場所にカーソルを置き、PuTTY 端末内でマウスを右クリックします。

セキュリティキーの詳細については、<https://wordpress.org/support/article/editing-wp-config-php/#security-keys> を参照してください。

Note

次の値はサンプル専用です。これらの値を実際のインストールには使わないでください。

```
define('AUTH_KEY',          ' #U$$+[RXN8:b^-L 0(WU_+ c+WFkI~c]o]-bHw+)/Aj[wTwSiZ<Qb[mghEXcRh-');
define('SECURE_AUTH_KEY',  'Zsz._P=l/|y.Lq)XjlkwS1y5NJ76E6EJ.AV0pCKZZB,*~*r ?60P$eJT@;+(ndLg');
define('LOGGED_IN_KEY',    'ju}qwre3V**+8f_z0Wf?{LLGsQ]Ye@2Jh^,8x>)Y |;(^[Iw]Pi+LG#A4R?7N`YB3');
define('NONCE_KEY',        'P(g62HeZxEes|LnI^i=H,[XwK9I&[2s|:~?0N}VJM%?;v2v]v+;+^9eXUahg@:~:~Cj');
define('AUTH_SALT',        'C$DpB4Hj[JK:~?{qL`sRVa:~:7yShy(9A@5wg+`JJVb1fk%_-Bx*M4(qc[Qg%JT!h');
define('SECURE_AUTH_SALT', 'd!uRu#}+q#{f$Z?Z9uFPG.$~+S{n~1M&%@~gL>U>NV<zpD-@2-Es7Q10-bp28EKv');
define('LOGGED_IN_SALT',   ';j{00P*owZf)kVD+FVLn-~ >.|Y%Ug4#I^*LVd9QeZ^&XmK|e(76miC+&W&+^0P/');
define('NONCE_SALT',       '-97r*V/cgxLmp?Zy4zUU4r99QQ_rGs2LTd%P;|_e1tS)8_B/,~.6[=UK<J_y9?JWG');
```

- e. ファイルを保存し、テキストエディタを終了します。

Apache ドキュメントルートの下に WordPress ファイルをインストールするには

- インストールフォルダを解凍し、MySQL データベースとユーザーを作成し、WordPress 設定ファイルをカスタマイズしたので、インストールファイルをウェブサーバドキュメントのルートにコピーして、インストールを完了するインストールスクリプトを実行できるようになります。これらのファイルの場所は、WordPress ブログをウェブサーバーの実際のルート (など *my.public.dns.amazonaws.com*) で使用するか、ルートの下の子ディレクトリまたはフォルダ (など) で使用するかによって異なります *my.public.dns.amazonaws.com/blog*。
- ドキュメントルートで WordPress を実行する場合は、wordpress インストールディレクトリの内容を (ディレクトリ自体ではなく) 次のようにコピーします。

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

- ドキュメントルートの下にある代替ディレクトリで WordPress を実行する場合は、まずそのディレクトリを作成し、そのディレクトリにファイルをコピーします。この例では、WordPress はディレクトリ から実行されます *blog*。

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

Important

セキュリティ上の理由から、次の手順にすぐに進まない場合は、Apache ウェブサーバー (httpd) を直ちに停止してください。インストールを Apache ドキュメントルートの下に移動すると、WordPress インストールスクリプトは保護されず、Apache ウェブサーバーが実行されていれば攻撃者がブログにアクセスできる可能性があります。Apache ウェブサーバーを停止するには、`sudo service httpd stop` コマンドを入力します。次の手順に移動する場合、Apache ウェブサーバーを停止する必要はありません。

WordPress が permalink を使用できるようにするには

WordPress permalinks は、正しく動作するために Apache `.htaccess` ファイルを使用する必要がありますが、これは Amazon Linux ではデフォルトで有効になっていません。Apache ドキュメントルートですべての上書きできるようにするには、次の手順を使用します。

1. お好みのテキストエディタ (httpd.conf や nano など) で、vim ファイルを開きます。お好みのテキストエディタがない場合、nano が初心者に適しています。

```
[ec2-user ~]$ sudo vim /etc/httpd/conf/httpd.conf
```

2. <Directory "/var/www/html"> で始まるセクションを見つけます。

```
<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. 上のセクションの AllowOverride None 行を AllowOverride **All** に変更します。

Note

このファイルには複数の AllowOverride 行があります。必ず <Directory "/var/www/html"> セクションの行を変更してください。

```
AllowOverride ALL
```

4. ファイルを保存し、テキストエディタを終了します。

AL2023 に PHP グラフィック描画ライブラリをインストールするには

PHP 用の GD ライブラリを使用すると、イメージを変更することができます。ブログのヘッダーイメージをトリミングする必要がある場合は、このライブラリをインストールします。インストール phpMyAdmin する のバージョンには、このライブラリの特定の最小バージョン (バージョン 8.1 など) が必要になる場合があります。

次のコマンドを使用して、AL2023 に PHP グラフィック描画ライブラリをインストールします。例えば、LAMP スタックをインストールする一環としてソースから php8.1 をインストールした場合、このコマンドは PHP グラフィック描画ライブラリのバージョン 8.1 をインストールします。

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

インストールしたバージョンを検証するには、次のコマンドを使用します。

```
[ec2-user ~]$ sudo dnf list installed | grep php-gd
```

出力例を次に示します。

```
php-gd.x86_64                8.1.30-1.amzn2                @amazonlinux
```

PHP グラフィック描画ライブラリを Amazon Linux AMI にインストールするには

PHP 用の GD ライブラリを使用すると、イメージを変更することができます。ブログのヘッダーイメージをトリミングする必要がある場合は、このライブラリをインストールします。インストール phpMyAdmin する のバージョンには、このライブラリの特定の最小バージョン (バージョン 8.1 など) が必要になる場合があります。

使用可能なバージョンを確認するには、次のコマンドを使用します。

```
[ec2-user ~]$ dnf list | grep php
```


PHP グラフィック描画ライブラリ (バージョン 8.1) の出力例を次に示します。

```
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
```

PHP グラフィック描画ライブラリの特定のバージョン (バージョン php8.1 など) を Amazon Linux AMI にインストールするには、次のコマンドを使用します。

```
[ec2-user ~]$ sudo dnf install -y php8.1-gd
```

Apache ウェブサーバーのファイル許可を修正するには

で利用できる機能の中には、Apache ドキュメントルートへの書き込みアクセス (管理画面を介したメディアのアップロードなど) WordPress を必要とするものがあります。まだ設定していない場合は、次のグループのメンバーシップおよびアクセス許可を適用します (プロセスの詳細は「[LAMP ウェブサーバーチュートリアル](#)」を参照)。

1. /var/www とそのコンテンツのファイル所有権を apache ユーザーに付与します。

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

2. /var/www とそのコンテンツのグループ所有権を apache グループに付与します。

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

3. /var/www およびそのサブディレクトリのディレクトリ許可を変更してグループの書き込み許可を設定し、将来のサブディレクトリにグループ ID を設定します。

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

4. `/var/www` およびそのサブディレクトリのファイル許可を繰り返し変更します。

```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod 0644 {} \;
```

Note

を FTP サーバー WordPress としても使用する場合、ここではより許容度の高いグループ設定が必要になります。これを行うには、[の推奨手順とセキュリティ設定 WordPress](#)を確認してください。

5. Apache ウェブサーバーを再起動して、新しいグループと許可を有効にします。

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

AL2023 で WordPress インストールスクリプトを実行するには

をインストールする準備ができました WordPress。使用するコマンドは、オペレーティングシステムによって異なります。この手順のコマンドは、AL2023 で使用するためのものです。AL2023 AMI では、この手順に従って実行します。

1. `systemctl` コマンドを使って、`httpd` サービスとデータベースサービスがシステムブート時に起動することを確認します。

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

2. データベースサーバーが実行中であることを確認します。

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

データベースサービスが実行されていない場合は、起動します。

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

3. Apache ウェブサーバー (`httpd`) が実行中であることを確認します。

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

`httpd` サービスが実行されていない場合は、起動します。

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

4. ウェブブラウザで、WordPress ブログの URL (インスタンスのパブリック DNS アドレス、またはそのアドレスの後に blog フォルダ) を入力します。WordPress インストールスクリプトが表示されます。WordPress インストールに必要な情報を入力します。インストール WordPress を選択してインストールを完了します。詳細については、WordPress ウェブサイトの [「ステップ 5: インストールスクリプトを実行する」](#) を参照してください。

AL2023 AMI で WordPress インストールスクリプトを実行するには

1. chkconfig コマンドを使って、httpd サービスとデータベースサービスがシステムブート時に起動することを確認します。

```
[ec2-user ~]$ sudo chkconfig httpd on && sudo chkconfig mariadb on
```

2. データベースサーバーが実行中であることを確認します。

```
[ec2-user ~]$ sudo service mariadb status
```

データベースサービスが実行されていない場合は、起動します。

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

3. Apache ウェブサーバー (httpd) が実行中であることを確認します。

```
[ec2-user ~]$ sudo service httpd status
```

httpd サービスが実行されていない場合は、起動します。

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

4. ウェブブラウザで、WordPress ブログの URL (インスタンスのパブリック DNS アドレス、またはそのアドレスの後に blog フォルダ) を入力します。WordPress インストールスクリプトが表示されます。WordPress インストールに必要な情報を入力します。インストール WordPress を選択してインストールを完了します。詳細については、WordPress ウェブサイトの [「ステップ 5: インストールスクリプトを実行する」](#) を参照してください。

次のステップ

WordPress ブログをテストしたら、設定の更新を検討してください。

カスタムドメイン名を使用する

EC2 インスタンスの EIP アドレスに関連付けられたドメイン名がある場合、EC2 パブリック DNS アドレスの代わりにその名前を使用するようにブログを設定できます。詳細については、WordPress ウェブサイトの「[サイト URL の変更](#)」を参照してください。

ブログを設定する

読者にパーソナライズされた体験を提供するため、さまざまな[テーマ](#)や[プラグイン](#)を使用するようにブログを設定できます。ただし、インストールプロセスで問題が発生してブログ全体が失われることがあります。インストール中に問題が発生した場合もブログを復元できるように、テーマやプラグインをインストールする前にインスタンスのバックアップ Amazon マシンイメージ (AMI) を作成しておくことを強くお勧めします。詳細については、「Amazon EC2 [ユーザーガイド](#)」の「[独自の AMI を作成する](#)」を参照してください。 Amazon EC2

容量を増やす

WordPress ブログが人気が高くなり、より多くのコンピューティング能力やストレージが必要な場合は、次のステップを検討してください。

- インスタンスストレージ領域を拡張する。詳細については、「[Amazon EBS Elastic Volumes](#)」を参照してください。
- MySQL データベースを [Amazon RDS](#) に移動して、サービスが持つ容易にスケールする機能を活用する。

インターネットトラフィックのネットワークパフォーマンスを向上させる

ブログにより世界中のユーザーからのトラフィックが増加すると予想される場合は、[AWS Global Accelerator](#) をご検討ください。Global Accelerator は、ユーザーのクライアントデバイスと実行されている WordPress アプリケーション間のインターネットトラフィックのパフォーマンスを向上させることで、レイテンシーの短縮に役立ちます AWS。Global Accelerator は、[AWS グローバルネットワーク](#)を使用して、クライアントに最も近い AWS リージョン内の正常なアプリケーションエンドポイントにトラフィックを誘導します。

の詳細 WordPress

次のリンクには、に関する詳細情報が含まれています WordPress。

- の詳細については WordPress、WordPress 「Codex のヘルプドキュメント <http://codex.wordpress.org/>」を参照してください。
- インストールのトラブルシューティングの詳細については、[「インストールに関する一般的な問題」](#)を参照してください。
- WordPress ブログのセキュリティを強化する方法については、[「の強化 WordPress」](#)を参照してください。
- WordPress ブログの の保持については up-to-date、[「の更新 WordPress」](#)を参照してください。

ヘルプ! パブリック DNS 名が変更されたため、ブログが壊れました

WordPress インストールは、EC2 インスタンスのパブリック DNS アドレスを使用して自動的に設定されます。インスタンスを停止および再開した場合、パブリック DNS アドレスが変更され (Elastic IP アドレスに関連付けられている場合を除く)、ブログが存在しなくなった (または別の EC2 インスタンスに割り当てられた) アドレスにあるリソースを参照することになるため、ブログは機能しなくなります。問題のより詳細な説明と考えられる解決策については、<https://wordpress.org/support/article/changing-the-site-url/> で説明しています。

WordPress インストールにこのような問題が発生した場合は、 のwp-cliコマンドラインインターフェイスを使用する以下の手順でブログを復元できる場合があります WordPress。

を使用して WordPress サイト URL を変更するには wp-cli

1. SSH を使って EC2 インスタンスに接続します。
2. インスタンスの古いサイト URL と新しいサイト URL を書き留めます。古いサイト URL は、をインストールしたときの EC2 インスタンスのパブリック DNS 名である可能性があります WordPress。新しいサイト URL は、EC2 インスタンスの現在のパブリック DNS 名です。古いサイト URL が不明な場合、次のコマンドで curl を使用して調べることができます。

```
[ec2-user ~]$ curl localhost | grep wp-content
```

古いパブリック DNS 名への参照が出力に表示されます。次に例を示します (古いサイト URL は赤色になっています)。

```
<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. 次のコマンドを使って wp-cli をダウンロードします。

```
[ec2-user ~]$ curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar
```

4. 次のコマンドを使用して、WordPress インストール内の古いサイト URL を検索して置き換えます。EC2 インスタンスの古いサイト URL と新しいサイト URLs、および WordPress インストールへのパス (通常は /var/www/html または) を置き換えます /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. ウェブブラウザで、WordPress ブログの新しいサイト URL を入力して、サイトが正常に動作していることを確認します。そうでない場合は、[「サイト URL の変更」](#)と [「一般的なインストールの問題」](#)を参照してください。

Amazon EC2 以外で Amazon Linux 2023 を使用する

Amazon Linux 2023 コンテナイメージは、互換性のあるコンテナランタイム環境で実行できます。Amazon Linux 2023 をコンテナ内で使用方法の詳細については、「[コンテナでの AL2023](#)」を参照してください。

Amazon Linux 2023 (AL2023) は、Amazon EC2 で直接実行する以外に、仮想化ゲストとして実行することもできます。現在、KVM (qcow2)、VMware (OVA)、Hyper-V (vhdx) のイメージが利用可能です。

Note

Amazon Linux 2023 イメージの設定は Amazon Linux 2 とは異なります。
[Amazon Linux 2 をオンプレミスの仮想化マシンとして実行する](#)場合は、AL2023 と互換性があるように設定を調整する必要があります。

KVM、VMware、Hyper-V で使用する Amazon Linux 2023 イメージをダウンロードする

KVM、VMware、および Hyper-V で使用する Amazon Linux 2023 ディスクイメージは、cdn.amazonlinux.com からダウンロードできます。

Amazon EC2 以外の仮想化環境での使用がサポートされている Amazon Linux 2023 の設定

このセクションでは、KVM、VMware、Hyper-V などの Amazon EC2 以外の仮想化環境で Amazon Linux 2023 を実行するための要件について説明します。

基本の [AL2023 システム要件](#) は Amazon EC2 以外のすべての仮想化環境に適用されます。各ハイパーバイザー環境でサポートされるデバイスモデルのリストは、以下のトピックで詳しく説明されています。

KVM、VMwareHyper-V には多くの設定オプションがあり、セキュリティ、パフォーマンス、信頼性のニーズに合わせて設定するには注意が必要です。詳細については、ハイパーバイザーから提供されたドキュメントを参照してください。

トピック

- [KVM で AL2023 を実行するための要件](#)
- [AL2023 を実行するための要件 VMware](#)
- [Hyper-V で Amazon Linux 2023 を実行するための要件](#)

KVM で AL2023 を実行するための要件

このセクションでは、KVM で AL2023 を実行するための要件について説明します。AL2023 の KVM イメージは、aarch64 および x86-64 アーキテクチャの両方で使用できます。これらの要件は、KVM イメージのベースに追加[AL2023 システム要件](#)されます。

トピック

- [KVM で AL2023 を実行するための KVM ホスト要件](#)
- [KVM での AL2023 のデバイスサポート](#)
- [KVM での AL2023 のブートモード \(UEFI および BIOS\) サポート](#)
- [KVM で AL2023 を実行する制限事項](#)

KVM で AL2023 を実行するための KVM ホスト要件

KVM イメージは現在、 のマシンタイプ x86-64 と の q35 マシン virt タイプを使用して 6.2+dfsg-2ubuntu6.15、この Ubuntu バージョン で Ubuntu qemu 22.04.3 LTS を実行しているホストで認定されています aarch64。

KVM での AL2023 のデバイスサポート

AL2023 KVM イメージ (aarch64 と x86-64 の両方) での使用がテストされた **qemu** デバイスマodel は以下のとおりです。

- virtio-blk (virtio ブロックデバイス)
- virtio-scsi (ディスクデバイスを搭載した virtio SCSI コントローラ)
- virtio-net (virtio ネットワークデバイス)
- ahci (仮想化 CD-ROM ドライブ用)
- usb-storage (xhci 以上)

AL2023 KVM イメージ認定で有効になっているが、あまり実用化されていない追加の `qemu` デバイスモデルは次のとおりです。

- x86-64 のみでの VGA (qemu VGA)
- `virtio-rng` (仮想化乱数ジェネレーター)
- 従来の AT キーボードおよび PS/2 マウスデバイス
- 従来のシリアルデバイス

KVM での AL2023 のブートモード (UEFI および BIOS) サポート

x86-64 イメージは従来の BIOS および UEFI ブートモードの両方でテストされます。aarch64 イメージは UEFI ブートモードでテストされます。

Note

デフォルトでは、UEFIブートモードを使用する場合、一部の仮想マシンマネージャーは、セキュアブートを有効にする Microsoft Secure Boot キーを使用して VM をプロビジョニングします。この設定では AL2023 は起動しません。

AL2023 ブートローダーは Microsoft によって署名されていないため、VM は UEFI キーを使用しないか、セキュアブート用の AL2023 キーを使用してプロビジョニングする必要があります。

Important

KVMイメージのセキュアブートサポートはまだ検証されていません。

KVM で AL2023 を実行する制限事項

KVM で AL2023 を実行するには、既知の制限がいくつかあります。

Note

リストされているサポートされていない機能の一部を実装するコードは、AL2023 に存在し、正しく機能する可能性があります。サポートされていない機能のリストは、今日の作業

に何に依存するか、および Amazon Linux チームが将来の更新の一環として動作する資格について、情報に基づいた意思決定を行うために存在します。

KVM で AL2023 を実行する場合の既知の制限事項

- KVM ゲストエージェントは現在パッケージ化されておらず、サポートもされていません。
- CPU、メモリ、またはその他の種類のデバイスのホットプラグとアンプラグはサポートされていません。
- VM の休止はサポートされていません。
- VM の移行はサポートされていません。
- PCI パススルーや USB パススルーなどによるデバイスのパススルーはサポートされていません。

AL2023 を実行するための要件 VMware

このセクションでは、AL2023 を実行するための要件について説明します。VMwareAL2023 VMware のイメージはアーキテクチャでのみ使用できます。x86-64 VMwareaarch64のイメージは利用できず、サポートもされていません。これらの要件は、[AL2023 システム要件](#) VMware 画像の基本要件に加えて追加されます。

トピック

- [VMwareAL2023 を実行するためのホスト要件 VMware](#)
- [AL2023 のデバイスサポート VMware](#)
- [AL2023 のブートモード \(UEFIおよびBIOS\) サポート VMware](#)
- [で AL2023 を実行する際の制限事項 VMware](#)

VMwareAL2023 を実行するためのホスト要件 VMware

AL2023 VMware OVA イメージは現在、以下の条件を満たしています。

- VMwareインテル (R) Xeon (R) Platinum 8124M プロセッサを搭載したホスト上で動作するワークステーション 17.5.0
- VMwareインテル (R) Xeon (R) プラチナ 8275CL プロセッサを搭載した vSphere 8.0

AL2023 VMware OVA イメージでは、マシンハードウェアバージョン 13 が指定されています。

VMwareマシンハードウェアバージョン 13 は以下でサポートされています。

- ESXi 6.5 以降
- VMwareワークステーション 14 以降

AL2023 のデバイスサポート VMware

VMware次のデバイスモデルが AL2023 VMware OVA イメージ (x86-64のみ) での使用がテストされました。

- vmw_pvscsi (VMware準仮想化コントローラSCSI)
- vmxnet3 (VMware準仮想化ネットワークデバイス)
- ata_piix (レガシー IDE は仮想化 CD-ROM ドライブでのみ使用)

AL2023 VMware のイメージクオリフィケーションでは有効になっているが、VMwareあまり活用されていないデバイスモデルが追加されました。

- vmw_vmcisock および関連インターフェース (ゲストエージェント用の仮想ソケットトランスポート) VMware
- vmw_balloon メモリバルーンデバイス
- VMwareSVGAコントローラー
- 従来の AT キーボードおよび PS/2 マウスデバイス

VMwareゲストエージェントパッケージ (open-vm-tools) は AL2023 VMware OVA イメージにデフォルトでインストールされています。

AL2023 のブートモード (UEFIおよびBIOS) サポート VMware

2023.3.20231211 リリースの時点で、AL2023 VMware OVA イメージはレガシーモードとブートモードの両方で検証されています。BIOS UEFI OVA のデフォルト設定はまだレガシーですが、ユーザーが変更できます。BIOS

Important

UEFIセキュアブートサポートには必要ですが、これを実行している AL2023 については検証されていません。VMware

で AL2023 を実行する際の制限事項 VMware

AL2023 を上で実行することには、いくつかの既知の制限事項があります。VMware

Note

記載されているサポートされていない機能の一部を実装するコードが AL2023 に存在しますが、正しく機能する可能性があります。サポートされていない機能のリストは、お客様が現在の作業で何に頼るべきか、また Amazon Linux チームが今後の更新の一環として動作しているとみなすものについて、情報に基づいた決定を下せるようにするためのものです。

AL2023 をで実行する場合の既知の制限事項 VMware

- UEFI 現在、AL2023 をオンにした状態でのセキュアブートは検証されていません。VMware
- CPU、メモリ、またはその他の種類のデバイスのホットプラグとアンプラグはサポートされていません。
- VM のハイパネーションはサポートされていません。
- VM の移行はサポートされていません。
- PCI パススルーや USB パススルーなどによるデバイスのパススルーはサポートされていません。

Hyper-V で Amazon Linux 2023 を実行するための要件

このセクションでは、Hyper-V で Amazon Linux 2023 を実行するための要件について説明します。AL2023 の Hyper-V イメージは、そのアーキテクチャでのみ使用できます。x86-64 の Hyper-V イメージは、現時点では使用できず aarch64、サポートもされていません。

このセクションでは、Hyper-V [AL2023 システム要件](#) イメージの基本要件に加えて追加の要件について説明します。

トピック

- [Hyper-V で Amazon Linux 2023 を実行するための Hyper-V ホスト要件](#)
- [Hyper-V での Amazon Linux 2023 のデバイスサポート](#)
- [Hyper-V で Amazon Linux 2023 を実行する制限事項](#)

Hyper-V で Amazon Linux 2023 を実行するための Hyper-V ホスト要件

Hyper-V での Amazon Linux 2023 の主な条件は、EC2 インスタンスで実行されている Windows Server 2022 に関するものです。c5.metal

Hyper-V での Amazon Linux 2023 のデバイスサポート

Amazon Linux 2023 は、以下の仮想ハードウェアセットを使用して、第 1 世代と第 2 世代の Hyper-V 仮想マシンの両方でテストされています。

- 第 1 世代 (レガシー BIOS ブート) 仮想マシン
- 第 2 世代 (UEFI ブート-セキュアブートなし) VM
- 以下のデバイスモデルが AL2023 Hyper-V イメージでの使用がテストされています。
 - 第 2 世代 VM のルートディスクとエミュレート CD-ROM *hv_storvsc* ドライブ用の Hyper-V 仮想ストレージ
 - 第 1 世代仮想マシンの仮想 CD-ROM *ata_piix* ドライブ用のエミュレート PIIX IDE
 - Hyper-V 仮想イーサネット *hv_netvsc*
- 以下のデバイスモデルは有効ですが、ほとんどテストされていません。
 - 第 1 世代 VM のレガシー VGA テキストモード
 - 第 2 世代 VM 上の UEFI *simpldrmfb* ファームウェアベースのフレームバッファ
 - Hyper-V バルーン *hv_balloon*
 - Hyper-V バルーン *hv_balloon*
 - Hyper-V HID/マウス *hid_hyperv*
- 現時点では AL2023 では以下のデバイスモードが有効になっていません。
 - Hyper-V PCI パススルー
 - Hyper-V DRM グラフィックス

Important

第 2 世代の仮想マシンでは、セキュアブートはサポートされていないため、Amazon Linux 2023 を正常に起動するには、仮想マシンを起動する前に無効にする必要があります。Hyper-V は現在、マイクロソフト独自のキーで署名されたソフトウェアコンポーネントによるセキュアブートのみをサポートしていますが、Amazon Linux ブートローダーは

Amazon プライベートキーで署名されています。現時点では、Hyper-V はサードパーティキーのインポートをサポートしていません。

Hyper-V で Amazon Linux 2023 を実行する制限事項

Hyper-V で Amazon Linux 2023 を実行する場合の既知の制限事項は次のとおりです。

Note

記載されているサポートされていない機能の一部を実装するコードが AL2023 に存在しますが、正しく機能する可能性があります。サポートされていない機能のリストは、お客様が現在の作業で何に頼るべきか、また Amazon Linux チームが今後の更新の一環として動作しているとみなすものについて、情報に基づいた決定を下せるようにするためのものです。

Hyper-V で AL2023 を実行する場合の既知の制限事項

- UEFI セキュアブートモードは、Hyper-V 上の AL2023 では現在サポートされておらず、機能もしていません。
- CPU、メモリ、またはその他の種類のデバイスのホットプラグとアンプラグはサポートされていません。
- 仮想マシン (VM) のハイバネーションはサポートされていません。
- 仮想マシン (VM) の移行はサポートされていません。
- PCI パススルーや USB パススルーなどによるデバイスのパススルーはサポートされていません。

Amazon EC2 以外で使用する場合の Amazon Linux 2023 の設定および **cloud-init** 設定

このセクションでは、KVM、VMware、または Hyper-V など、Amazon EC2 で直接実行されない場合に Amazon Linux 2023 仮想マシンをセットアップおよび設定する方法について説明します。

デフォルトでは、Amazon Linux 2023 の仮想化マシンイメージにはユーザーパスワードや SSH キーはプロビジョニングされず、DHCP を経由して最初に検出されたネットワークインターフェイスからネットワーク設定を取得します。つまり、デフォルトでは、追加の設定を行わないと、生成された仮想化マシンに接続する方法はありません。

そのため、何らかの構成を仮想化マシンに提供する必要があります。Amazon Linux でこれを行う標準的なメカニズムは、cloud-init データソース経由です。

Amazon Linux 2023 は以下のデータソースの認定を受けています。

NoCloud

これは、cloud-init 設定ファイルを含むシード ISO9660 イメージを含む仮想化 CD-ROM を使用してオンプレミスイメージを設定する従来の方法です。

VMware

Amazon Linux 2023 では、VMware guestinfo.userdata および guestinfo.metadata 経由の VMware 固有のデータソースを介して vSphere 上で実行されている VMware イメージを設定することもサポートされています。

Note

データソースの設定は Amazon Linux 2 とは異なる場合があります。具体的には、Amazon Linux 2023 は設定に systemd-networkd を使用しており、「[cloud-init ネットワーク設定ドキュメント](#)」に記載されている cloud-init 「ネットワーク設定バージョン 2」を使用する必要があります。

Amazon Linux 2023 の cloud-init パッケージのバージョンの cloud-init 設定メカニズムに関するすべてのドキュメントは、「[cloud-init アップストリームのドキュメント](#)」にあります。

NoCloud (**seed.iso**) KVM と VMware での Amazon Linux 2023 **cloud-init** の設定

このセクションでは、seed.iso イメージを作成して使用して、KVM/VMware またはで実行される Amazon Linux 2023 を設定する方法について説明します。KVM/VMware および環境には [Amazon EC2 インスタンスメタデータサービス \(IMDS\)](#) がないため、Amazon Linux 2023 を設定する別の方法が必要であり、seed.iso イメージの提供もその方法の 1 つです。

seed.iso ブートイメージには、ネットワーク設定、ホスト名、ユーザーデータなどの、新しい仮想化マシンの起動に必要な初期設定情報が含まれます。

Note

seed.iso イメージには、VM の起動に必要な設定情報のみ含まれています。Amazon Linux 2023 オペレーティングシステムファイルは含まれていません。

seed.iso イメージを生成するには、2 つ (もしくは 3 つ) の設定ファイルが必要です。

meta-data

このファイルには通常、仮想化マシンのホスト名が含まれます。

user-data

このファイルでは、ユーザーアカウント、パスワード、ssh キーペア、およびアクセスメカニズムを設定します。デフォルトでは、Amazon Linux 2023 KVM および VMware イメージは、ec2-user ユーザーアカウントを作成します。user-data 設定ファイルを使用して、デフォルトユーザーアカウントのパスワードおよび SSH キーを設定します。

network-config (オプション)

このファイルには通常、デフォルトのネットワーク設定に上書きされる仮想化マシンのネットワーク設定が記載されています。デフォルト設定は、最初に使用可能なネットワークインターフェースで DHCP を使用します。

seed.iso ディスクイメージを作成する

1. Linux または macOS コンピュータでは、seedconfig という名前の新しいフォルダを作成し、そのフォルダに移動します。

Note

Windows または別のオペレーティングシステムを使用してこれらのステップを実行することも可能ですが、seed.iso イメージの作成を完了するには mkisofs と同等のツールを検索する必要があります。

2. meta-data 設定ファイルを作成します。
 - a. meta-data という名前の新しいファイルを作成します。

- b. 任意のエディタを使用して meta-data ファイルを開き、以下を追加し、*vm-hostname* を VM のホスト名に置き換えます。

```
local-hostname: vm-hostname
```

- c. meta-data 設定ファイルを保存して閉じます。
3. user-data 設定ファイルを作成します。
 - a. user-data という名前の新しいファイルを作成します。
 - b. 任意のエディタを使用して user-data ファイルを開き、以下を追加し、必要に応じて置き換えます。

```
#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. オプションで、user-data設定ファイルにユーザーアカウントをさらに追加できます。

追加ユーザーアカウント、アクセスメカニズム、パスワード、およびキーペアを指定できます。サポートされているディレクトティブの詳細については、「[アップストリーム cloud-init ドキュメント](#)」を参照してください。
 - d. user-data 設定ファイルを保存して閉じます。
4. (オプション) network-config 設定ファイルを作成します。
 - a. network-config という名前の新しいファイルを作成します。
 - b. 任意のエディタを使用して network-config ファイルを開き、以下を追加し、設定に応じてさまざまな IP アドレスを IP アドレスに置き換えます。

```
version: 2
ethernets:
```

```
enp1s0:
  addresses:
    - 192.168.122.161/24
  gateway4: 192.168.122.1
  nameservers:
    addresses: 192.168.122.1
```

Note

cloud-init ネットワーク構成は、仮想化マシンの設定によって変わる可能性のあるインターフェース名を指定する代わりに、インターフェースの MAC アドレスと照合するメカニズムを提供します。ネットワーク設定用のこの (およびその他の) cloud-init 機能については、「[アップストリームの cloud-init ネットワーク設定バージョン 2 のドキュメント](#)」で詳しく説明されています。

- c. network-config 設定ファイルを保存して閉じます。
5. 前の手順で作成した meta-data、user-data、およびオプションの network-config 設定ファイルを使用して seed.iso ディスクイメージを作成します。

seed.iso ディスクイメージを作成する OS に応じて、以下のいずれかの操作を行います。

- Linux システムでは、**mkisofs** または **genisoimage** などのツールを使用して完成した seed.iso ファイルを作成します。seedconfig フォルダに移動し、以下のコマンドを実行します。

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

- network-config を使用する場合は、それを **mkisofs** の呼び出しに含めます。

```
$ mkisofs -output seed.iso -volid cidata -joliet -rock user-data meta-data
network-config
```

- macOS システムでは、**hdiutil** などのツールを使用して完成した seed.iso ファイルを生成できます。**hdiutil** はファイルのリストではなくパス名を使用するため、network-config 設定ファイルが作成されているかどうかにかかわらず、同じ呼び出しを使用できません。

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

- 作成された `seed.iso` ファイルは、`cloud-init` の仮想化 CD-ROM ドライブを介して新しい Amazon Linux 2023 仮想化マシンに添付でき、初回起動時に検索してシステムに設定を適用できます。

VMwareAL2023 `cloud-init` のゲスト情報設定オン VMware

VMware環境には [Amazon EC2 インスタンスメタデータサービス \(IMDS\)](#) がないため、AL2023 を設定する別の方法が必要です。このセクションでは、vSphere `seed.iso` で使用できる仮想 CD-ROM ドライブに代わる構成メカニズムを使用する方法について説明します。VMware

この構成方法では、`VMwareextraconfig`構成データをに提供するメカニズムを使用します。`cloud-init`以下の各キーには、**`keyname.encoding`**対応するプロパティを指定する必要があります。

`VMwareextraconfig`メカニズムには以下のキーを指定できます。

`guestinfo.metadata`

`cloud-init` メタデータを含む JSON または YAML

`guestinfo.userdata`

`cloud-config` フォーマットの `cloud-init` ユーザーデータを含む YAML ドキュメント

`guestinfo.vendordata` (オプション)

YAML `cloud-init` ベンダーデータを含みます。

対応するエンコーディングプロパティ

(`guestinfo.metadata.encoding`、`guestinfo.userdata.encoding`、および `guestinfo.vendordata.encoding`) には以下を含めることができます。

`base64`

プロパティの内容は `base64` エンコードされます。

`gzip+base64`

プロパティの内容は `base64` エンコード後に `gzip` で圧縮されます。

Note

`seed.iso` このメソッドは別の (オプションの) `network-config` 設定ファイルをサポートします。VMware `guestinfo` ネットワーク設定の提供方法が異なります。追加情報については、次のセクションで説明します。

明示的なネットワーク設定が必要な場合は、2つの YAML または JSON プロパティのフォーマットで `metadata` に埋め込む必要があります。

network

JSON または YAML 形式でエンコードされたネットワーク設定が含まれます。

network.encoding

上記のネットワーク設定データのエンコーディングが含まれます。cloud-init でサポートされているエンコーディングは、`guestinfo` データのエンコーディングと共通です: `base64` および `gzip+base64`。

Example VMwarevSphere govc CLI ツールを使用して構成を渡す guestinfo

1. の説明に従って `meta-data`、`user-data`、`network-config` およびオプションの設定ファイルを準備します。[NoCloud \(seed.iso\) KVM と VMware での Amazon Linux 2023 cloud-init の設定](#)
2. 設定ファイルを使用して使用できる形式に変換します 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"

```

Amazon Linux 2023 標準 AMI にインストールされているパッケージと AL2023 KVM イメージの比較

AL2023 標準 AMI に存在する RPMs と AL2023 KVM イメージに存在する RPMs の比較。

パッケージ	AMI	KVM
acl	2.3.1	2.3.1
acpid	2.0.32	
alternatives	1.15	1.15
amazon-chrony-config	4.3	
amazon-ec2-net-utils	2.4.1	
amazon-linux-onprem		1.2
amazon-linux-repo-cdn		2023.4.20240513
amazon-linux-repo-s3	2023.4.20240513	
amazon-linux-sb-keys	2023.1	2023.1

パッケージ	AMI	KVM
amazon-onprem-network		1.2
amazon-rpm-config	228	228
amazon-ssm-agent	3.3.380.0	3.3.380.0
at	3.1.23	3.1.23
attr	2.5.1	2.5.1
audit	3.0.6	3.0.6
audit-libs	3.0.6	3.0.6
aws-cfn-bootstrap	2.0	
awscli-2	2.15.30	2.15.30
basesystem	11	11
bash	5.2.15	5.2.15
bash-completion	2.11	2.11
bc	1.07.1	1.07.1
bind-libs	9.16.48	9.16.48
bind-license	9.16.48	9.16.48
bind-utils	9.16.48	9.16.48
binutils	2.39	2.39
boost-filesystem	1.75.0	1.75.0
boost-system	1.75.0	1.75.0
boost-thread	1.75.0	1.75.0

パッケージ	AMI	KVM
bzip2	1.0.8	1.0.8
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023年2月64日	2023年2月64日
c-ares	1.19.0	
checkpolicy	3.4	3.4
chkconfig	1.15	1.15
chrony	4.3	4.3
cloud-init	22.2.2	22.2.2
cloud-init-cfg-ec2	22.2.2	
cloud-init-cfg-onpre		22.2.2
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

パッケージ	AMI	KVM
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

パッケージ	AMI	KVM
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

パッケージ	AMI	KVM
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
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

パッケージ	AMI	KVM
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-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 (ノーク)
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

パッケージ	AMI	KVM
gzip	1.12	1.12
hostname	3.23	3.23
hunspell	1.7.0	1.7.0
hunspell-en	0.20140811.1	0.20140811.1
hunspell-en-GB	0.20140811.1	0.20140811.1
hunspell-en-US	0.20140811.1	0.20140811.1
hunspell-filesystem	1.7.0	1.7.0
hwdata	0.353	0.353
info	6.7	6.7
inih	49	49
initscripts	10.09	10.09
iproute	5.10.0	5.10.0
iputils	20210202	20210202
irqbalance	1.9.0	1.9.0
jansson	2.14	2.14
jitterentropy	3.4.1	3.4.1
jq	1.7.1	
json-c	0.14	0.14
kbd	2.4.0	2.4.0
kbd-misc	2.4.0	2.4.0

パッケージ	AMI	KVM
kernel	6.1.90	6.1.90
kernel-livepatch-r epo-cdn		2023.4.20240513
kernel-livepatch-r epo-s3	2023.4.20240513	
kernel-modules-extra		6.1.90
kernel-modules-ext ra-common		6.1.90
kernel-srpm-macros	1.0	1.0
kernel-tools	6.1.90	6.1.90
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

パッケージ	AMI	KVM
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	0.1.20	0.1.20
libconfig	1.7.2	1.7.2
libcurl-minimal	8.5.0	8.5.0
libdb	5.3.28	5.3.28
libdhash	0.5.0	
libdnf	0.69.0	0.69.0
libeconf	0.4.0	0.4.0
libedit	3.1	3.1
libev	4.33	4.33
libevent	2.1.12	2.1.12
libfdisk	2.37.4	2.37.4

パッケージ	AMI	KVM
libffi	3.4.4	3.4.4
libfido2	1.10.0	1.10.0
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	11.4.1
libgpg-error	1.42	1.42
libibverbs	48.0	48.0
libidn2	2.3.2	2.3.2
libini_config	1.3.1	1.3.1
libkcapi	1.4.0	1.4.0
libkcapi-hmaccalc	1.4.0	1.4.0
libldb	2.6.2	
libmaxminddb	1.5.2	1.5.2
libmetalink	0.1.3	0.1.3
libmnl	1.0.4	1.0.4
libmodulemd	2.13.0	2.13.0
libmount	2.37.4	2.37.4
libnfsidmap	2.5.4	2.5.4
libnghttp2	1.59.0	1.59.0
libnl3	3.5.0	3.5.0

パッケージ	AMI	KVM
libpath_utils	0.2.1	0.2.1
libpcap	1.10.1	1.10.1
libpipeline	1.5.3	1.5.3
libpkgconf	1.8.0	1.8.0
libpsl	0.21.1	0.21.1
libpwquality	1.4.4	1.4.4
libref_array	0.1.5	0.1.5
librepo	1.14.5	1.14.5
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

パッケージ	AMI	KVM
libsss_nss_idmap	2.9.4	2.9.4
libsss_sudo	2.9.4	
libstdc++	11.4.1	11.4.1
libstoragegmt	1.9.4	1.9.4
libtalloc	2.3.4	
libtasn1	4.19.0	4.19.0
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
libyam1	0.2.5	0.2.5

パッケージ	AMI	KVM
libzstd	1.5.5	1.5.5
lm_sensors-libs	3.6.0	3.6.0
lmdb-libs	0.9.29	0.9.29
logrotate	3.20.1	3.20.1
lsof	4.94.0	4.94.0
lua-libs	5.4.4	5.4.4
lua-srpm-macros	1	1
lz4-libs	1.9.4	1.9.4
man-db	2.9.3	2.9.3
man-pages	5.10	5.10
microcode_ctl	2.1 (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

パッケージ	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	8.7p1
openssh-clients	8.7p1	8.7p1
openssh-server	8.7p1	8.7p1
openssl	3.0.8	3.0.8
openssl-libs	3.0.8	3.0.8
openssl-pkcs11	0.4.12	0.4.12
os-prober	1.77	1.77

パッケージ	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

パッケージ	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

パッケージ	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-subst	1.03	1.03
perl-Symbol	1.08	1.08
perl-Term-ANSIColor	5.01	5.01
perl-Term-Cap	1.17	1.17
perl-Text-ParseWords	3.30	3.30
perl-Text-Tabs+Wrap	2021.0726	2021.0726
perl-Time-Local	1.300	1.300
perl-vars	1.05	1.05
pkgconf	1.8.0	1.8.0
pkgconf-m4	1.8.0	1.8.0
pkgconf-pkg-config	1.8.0	1.8.0
policycoreutils	3.4	3.4

パッケージ	AMI	KVM
polycoreutils-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-awscli	0.19.19	0.19.19
python3-babel	2.9.1	2.9.1
python3-cffi	1.14.5	1.14.5
python3-chardet	4.0.0	4.0.0
python3-colorama	0.4.4	0.4.4
python3-configobj	5.0.6	5.0.6
python3-cryptography	36.0.1	36.0.1
python3-daemon	2.3.0	
python3-dateutil	2.8.1	2.8.1

パッケージ	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-jjsonschema	3.2.0	3.2.0
python3-libcomps	0.1.20	0.1.20
python3-libdnf	0.69.0	0.69.0
python3-libs	3.9.16	3.9.16
python3-libselenium	3.4	3.4
python3-libsemanage	3.4	3.4
python3-libstorage mgmt	1.9.4	1.9.4

パッケージ	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-policycore utils	3.4	3.4
python3-prettytable	0.7.2	0.7.2
python3-prompt-too lkit	3.0.24	3.0.24
python3-pycparser	2.20	2.20
python3-pyrsistent	0.17.3	0.17.3
python3-pyserial	3.4	3.4
python3-pysocks	1.7.1	1.7.1
python3-pytz	2022.7.1	2022.7.1
python3-pyyaml	5.4.1	5.4.1
python3-requests	2.25.1	2.25.1
python3-rpm	4.16.1.3	4.16.1.3
python3-ruamel-yaml	0.16.6	0.16.6

パッケージ	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

パッケージ	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	37.22	37.22
selinux-policy-targeted	37.22	37.22
setup	2.13.7	2.13.7
shadow-utils	4.9	4.9
slang	2.3.2	2.3.2
sqlite-libs	3.40.0	3.40.0
sssd-client	2.9.4	2.9.4
sssd-common	2.9.4	
sssd-kcm	2.9.4	
sssd-nfs-idmap	2.9.4	
strace	6.8	6.8

パッケージ	AMI	KVM
sudo	1.9.15	1.9.15
sysctl-defaults	1.0	1.0
sysstat	12.5.6	12.5.6
systemd	252.16	252.16
systemd-libs	252.16	252.16
systemd-networkd	252.16	252.16
systemd-pam	252.16	252.16
systemd-resolved	252.16	252.16
systemd-udev	252.16	252.16
system-release	2023.4.20240513	2023.4.20240513
systemtap-runtime	4.8	4.8
tar	1.34	1.34
tbb	2020年3月	2020年3月
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	2024年a	2024年a
unzip	6.0	6.0
update-motd	2.2	2.2

パッケージ	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
xfspgrog	5.18.0	5.18.0
xxd	9.0.2153	9.0.2153
xxhash-libs	0.8.0	0.8.0
xz	5.2.5	5.2.5
xz-libs	5.2.5	5.2.5
yum	4.14.0	4.14.0
zip	3.0	3.0
zlib	1.2.11	1.2.11

パッケージ	AMI	KVM
zram-generator	1.1.2	
zram-generator-def aults	1.1.2	
zstd	1.5.5	1.5.5

Amazon Linux 2023 標準 AMI にインストールされているパッケージと AL2023 VMware OVA イメージの比較

AL2023 標準 AMI に存在する RPMs と AL2023 VMware OVA イメージに存在する RPMs の比較。

パッケージ	AMI	VMware OVA
acl	2.3.1	2.3.1
acpid	2.0.32	
alternatives	1.15	1.15
amazon-chroney-config	4.3	
amazon-ec2-net-utils	2.4.1	
amazon-linux-onprem		1.2
amazon-linux-repo- cdn		2023.4.20240513
amazon-linux-repo-s3	2023.4.20240513	
amazon-linux-sb-keys	2023.1	2023.1
amazon-onprem-netw ork		1.2
amazon-rpm-config	228	228

パッケージ	AMI	VMware OVA
amazon-ssm-agent	3.3.380.0	3.3.380.0
at	3.1.23	3.1.23
attr	2.5.1	2.5.1
audit	3.0.6	3.0.6
audit-libs	3.0.6	3.0.6
aws-cfn-bootstrap	2.0	
awscli-2	2.15.30	2.15.30
basesystem	11	11
bash	5.2.15	5.2.15
bash-completion	2.11	2.11
bc	1.07.1	1.07.1
bind-libs	9.16.48	9.16.48
bind-license	9.16.48	9.16.48
bind-utils	9.16.48	9.16.48
binutils	2.39	2.39
boost-filesystem	1.75.0	1.75.0
boost-system	1.75.0	1.75.0
boost-thread	1.75.0	1.75.0
bzip2	1.0.8	1.0.8
bzip2-libs	1.0.8	1.0.8

パッケージ	AMI	VMware OVA
ca-certificates	2023 年 2 月 64 日	2023 年 2 月 64 日
c-ares	1.19.0	
checkpolicy	3.4	3.4
chkconfig	1.15	1.15
chrony	4.3	4.3
cloud-init	22.2.2	22.2.2
cloud-init-cfg-ec2	22.2.2	
cloud-init-cfg-onpre m		22.2.2
cloud-utils-growpart	0.31	0.31
coreutils	8.32	8.32
coreutils-common	8.32	8.32
cpio	2.13	2.13
cracklib	2.9.6	2.9.6
cracklib-dicts	2.9.6	2.9.6
crontabs	1.11	1.11
crypto-policies	20220428	20220428
crypto-policies-sc ripts	20220428	20220428
cryptsetup	2.6.1	2.6.1
cryptsetup-libs	2.6.1	2.6.1

パッケージ	AMI	VMware OVA
curl-minimal	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27
cyrus-sasl-plain	2.1.27	2.1.27
dbus	1.12.28	1.12.28
dbus-broker	32	32
dbus-common	1.12.28	1.12.28
dbus-libs	1.12.28	1.12.28
device-mapper	1.02.185	1.02.185
device-mapper-libs	1.02.185	1.02.185
diffutils	3.8	3.8
dnf	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2
dnf-plugins-core	4.3.0	4.3.0
dnf-plugin-support-info	1.2	1.2
dnf-utils	4.3.0	4.3.0
dosfstools	4.2	4.2
dracut	055	055
dracut-config-ec2	3.0	

パッケージ	AMI	VMware OVA
dracut-config-generic	055	055
dwz	0.14	0.14
dyninst	10.2.1	10.2.1
e2fsprogs	1.46.5	1.46.5
e2fsprogs-libs	1.46.5	1.46.5
ec2-hibinit-agent	1.0.8	
ec2-instance-connect	1.1	
ec2-instance-connect-selinux	1.1	
ec2-utils	2.2.0	
ed	1.14.2	1.14.2
efi-filesystem	5	5
efi-srpm-macros	5	5
efivar	38	38
efivar-libs	38	38
elfutils-debuginfod-client	0.188	0.188
elfutils-default-yama-scope	0.188	0.188
elfutils-libelf	0.188	0.188
elfutils-libs	0.188	0.188

パッケージ	AMI	VMware OVA
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
glib2	2.74.7	2.74.7
glibc	2.34	2.34

パッケージ	AMI	VMware OVA
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
grubby	8.40	8.40
gssproxy	0.8.4	0.8.4

パッケージ	AMI	VMware OVA
gzip	1.12	1.12
hostname	3.23	3.23
hunspell	1.7.0	1.7.0
hunspell-en	0.20140811.1	0.20140811.1
hunspell-en-GB	0.20140811.1	0.20140811.1
hunspell-en-US	0.20140811.1	0.20140811.1
hunspell-filesystem	1.7.0	1.7.0
hwdata	0.353	0.353
info	6.7	6.7
inih	49	49
initscripts	10.09	10.09
iproute	5.10.0	5.10.0
iputils	20210202	20210202
irqbalance	1.9.0	1.9.0
jansson	2.14	2.14
jitterentropy	3.4.1	3.4.1
jq	1.7.1	
json-c	0.14	0.14
kbd	2.4.0	2.4.0
kbd-misc	2.4.0	2.4.0

パッケージ	AMI	VMware OVA
kernel	6.1.90	6.1.90
kernel-livepatch-r epo-cdn		2023.4.20240513
kernel-livepatch-r epo-s3	2023.4.20240513	
kernel-modules-extra		6.1.90
kernel-modules-ext ra-common		6.1.90
kernel-srpm-macros	1.0	1.0
kernel-tools	6.1.90	6.1.90
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

パッケージ	AMI	VMware OVA
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	0.1.20	0.1.20
libconfig	1.7.2	1.7.2
libcurl-minimal	8.5.0	8.5.0
libdb	5.3.28	5.3.28
libdhash	0.5.0	
libdnf	0.69.0	0.69.0
libeconf	0.4.0	0.4.0
libedit	3.1	3.1
libev	4.33	4.33
libevent	2.1.12	2.1.12
libfdisk	2.37.4	2.37.4

パッケージ	AMI	VMware OVA
libffi	3.4.4	3.4.4
libfido2	1.10.0	1.10.0
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	11.4.1
libgpg-error	1.42	1.42
libibverbs	48.0	48.0
libidn2	2.3.2	2.3.2
libini_config	1.3.1	1.3.1
libkcapi	1.4.0	1.4.0
libkcapi-hmaccalc	1.4.0	1.4.0
libldb	2.6.2	
libmaxminddb	1.5.2	1.5.2
libmetalink	0.1.3	0.1.3
libmnl	1.0.4	1.0.4
libmodulemd	2.13.0	2.13.0
libmount	2.37.4	2.37.4
libmspack		0.10.1
libnfsidmap	2.5.4	2.5.4
libnghttp2	1.59.0	1.59.0

パッケージ	AMI	VMware OVA
libnl3	3.5.0	3.5.0
libpath_utils	0.2.1	0.2.1
libpcap	1.10.1	1.10.1
libpipeline	1.5.3	1.5.3
libpkgconf	1.8.0	1.8.0
libpsl	0.21.1	0.21.1
libpwquality	1.4.4	1.4.4
libref_array	0.1.5	0.1.5
librepo	1.14.5	1.14.5
libreport-filessystem	2.15.2	2.15.2
libseccomp	2.5.3	2.5.3
libselinux	3.4	3.4
libselinux-utils	3.4	3.4
libsemanage	3.4	3.4
libsepol	3.4	3.4
libsigsegv	2.13	2.13
libsmartcols	2.37.4	2.37.4
libsolv	0.7.22	0.7.22
libss	1.46.5	1.46.5
libsss_certmap	2.9.4	

パッケージ	AMI	VMware OVA
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
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

パッケージ	AMI	VMware OVA
libxml2	2.10.4	2.10.4
libxslt		1.1.34
libyaml	0.2.5	0.2.5
libzstd	1.5.5	1.5.5
lm_sensors-libs	3.6.0	3.6.0
lmdb-libs	0.9.29	0.9.29
logrotate	3.20.1	3.20.1
lsof	4.94.0	4.94.0
lua-libs	5.4.4	5.4.4
lua-srpm-macros	1	1
lz4-libs	1.9.4	1.9.4
man-db	2.9.3	2.9.3
man-pages	5.10	5.10
microcode_ctl	2.1	2.1
mpfr	4.1.0	4.1.0
nano	5.8	5.8
ncurses	6.2	6.2
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
nettle	3.8	3.8

パッケージ	AMI	VMware OVA
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
openldap	2.4.57	2.4.57
openssh	8.7p1	8.7p1
openssh-clients	8.7p1	8.7p1
openssh-server	8.7p1	8.7p1
openssl	3.0.8	3.0.8

パッケージ	AMI	VMware OVA
openssl-lib	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-lib	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

パッケージ	AMI	VMware OVA
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-I0	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

パッケージ	AMI	VMware OVA
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-subst	1.03	1.03
perl-Symbol	1.08	1.08
perl-Term-ANSIColor	5.01	5.01
perl-Term-Cap	1.17	1.17
perl-Text-ParseWords	3.30	3.30
perl-Text-Tabs+Wrap	2021.0726	2021.0726
perl-Time-Local	1.300	1.300
perl-vars	1.05	1.05
pkgconf	1.8.0	1.8.0

パッケージ	AMI	VMware OVA
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
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

パッケージ	AMI	VMware OVA
python3-cryptography	36.0.1	36.0.1
python3-daemon	2.3.0	
python3-dateutil	2.8.1	2.8.1
python3-dbus	1.2.18	1.2.18
python3-distro	1.5.0	1.5.0
python3-dnf	4.14.0	4.14.0
python3-dnf-plugins-core	4.3.0	4.3.0
python3-docutils	0.16	0.16
python3-gpg	1.15.1	1.15.1
python3-hawkey	0.69.0	0.69.0
python3-idna	2.10	2.10
python3-jinja2	2.11.3	2.11.3
python3-jmespath	0.10.0	0.10.0
python3-jsonpatch	1.21	1.21
python3-jsonpointer	2.0	2.0
python3-jjsonschema	3.2.0	3.2.0
python3-libcomps	0.1.20	0.1.20
python3-libdnf	0.69.0	0.69.0
python3-libs	3.9.16	3.9.16
python3-libselenium	3.4	3.4

パッケージ	AMI	VMware OVA
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
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

パッケージ	AMI	VMware OVA
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
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

パッケージ	AMI	VMware OVA
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
selinux-policy-targeted	37.22	37.22
setup	2.13.7	2.13.7
shadow-utils	4.9	4.9
slang	2.3.2	2.3.2
sqlite-libs	3.40.0	3.40.0
sssd-client	2.9.4	2.9.4
sssd-common	2.9.4	
sssd-kcm	2.9.4	

パッケージ	AMI	VMware OVA
sssd-nfs-idmap	2.9.4	
strace	6.8	6.8
sudo	1.9.15	1.9.15
sysctl-defaults	1.0	1.0
sysstat	12.5.6	12.5.6
systemd	252.16	252.16
systemd-libs	252.16	252.16
systemd-networkd	252.16	252.16
systemd-pam	252.16	252.16
systemd-resolved	252.16	252.16
systemd-udev	252.16	252.16
system-release	2023.4.20240513	2023.4.20240513
systemtap-runtime	4.8	4.8
tar	1.34	1.34
tbb	2020 年 3 月	2020 年 3 月
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	2024年a	2024年a

パッケージ	AMI	VMware OVA
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
xfspgrog	5.18.0	5.18.0
xmlsec1		1.2.33
xmlsec1-openssl		1.2.33
xxd	9.0.2153	9.0.2153
xxhash-libs	0.8.0	0.8.0
xz	5.2.5	5.2.5

パッケージ	AMI	VMware OVA
xz-libs	5.2.5	5.2.5
yum	4.14.0	4.14.0
zip	3.0	3.0
zlib	1.2.11	1.2.11
zram-generator	1.1.2	
zram-generator-def aults	1.1.2	
zstd	1.5.5	1.5.5

Amazon Linux 2023 標準 AMI にインストールされているパッケージと AL2023 Hyper-V イメージの比較

AL2023 標準 AMI に存在する RPMs と AL2023 Hyper-V イメージに存在する RPMs の比較。

パッケージ	AMI	Hyper-V VHDX
acl	2.3.1	2.3.1
acpid	2.0.32	
alternatives	1.15	1.15
amazon-chrony-config	4.3	
amazon-ec2-net-utils	2.4.1	
amazon-linux-onprem		1.2
amazon-linux-repo- cdn		2023.4.20240319

パッケージ	AMI	Hyper-V VHDX
amazon-linux-repo-s3	2023.4.20240319	
amazon-linux-sb-keys	2023.1	2023.1
amazon-onprem-netw ork		1.2
amazon-rpm-config	228	228
amazon-ssm-agent	3.2.2303.0	3.2.2303.0
at	3.1.23	3.1.23
attr	2.5.1	2.5.1
audit	3.0.6	3.0.6
audit-libs	3.0.6	3.0.6
aws-cfn-bootstrap	2.0	
awscli-2	2.14.5	2.14.5
basesystem	11	11
bash	5.2.15	5.2.15
bash-completion	2.11	2.11
bc	1.07.1	1.07.1
bind-libs	9.16.48	9.16.48
bind-license	9.16.48	9.16.48
bind-utils	9.16.48	9.16.48
binutils	2.39	2.39
boost-filesystem	1.75.0	1.75.0

パッケージ	AMI	Hyper-V VHDX
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年2月64日	2023年2月64日
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-onp rem		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

パッケージ	AMI	Hyper-V VHDX
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
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

パッケージ	AMI	Hyper-V VHDX
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
efi-filesystem	5	5
efi-srpm-macros	5	5
efivar	38	38
efivar-libs	38	38

パッケージ	AMI	Hyper-V VHDX
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
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

パッケージ	AMI	Hyper-V VHDX
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

パッケージ	AMI	Hyper-V VHDX
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-lic ense		0
hypervfcopyd		0
hypervkvpd		0
hyperv-tools		0
hypervvssd		0
info	6.7	6.7
inih	49	49
initscripts	10.09	10.09
iproute	5.10.0	5.10.0

パッケージ	AMI	Hyper-V VHDX
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-r epo-cdn		2023.4.20240319
kernel-livepatch-r epo-s3	2023.4.20240319	
kernel-modules-extra		6.1.79
kernel-modules-ext ra-common		6.1.79
kernel-srpm-macros	1.0	1.0
kernel-tools	6.1.79	6.1.79
keyutils	1.6.3	1.6.3
keyutils-libs	1.6.3	1.6.3
kmod	29	29

パッケージ	AMI	Hyper-V VHDX
kmod-libs	29	29
kpatch-runtime	0.9.7	0.9.7
krb5-libs	1.21	1.21
less	608	608
libacl	2.3.1	2.3.1
libaio	0.3.111	0.3.111
libarchive	3.5.3	3.5.3
libargon2	20171227	20171227
libassuan	2.5.5	2.5.5
libattr	2.5.1	2.5.1
libbasicobjects	0.1.1	0.1.1
libblkid	2.37.4	2.37.4
libcap	2.48	2.48
libcap-ng	0.8.2	0.8.2
libcbor	0.7.0	0.7.0
libcollection	0.7.0	0.7.0
libcom_err	1.46.5	1.46.5
libcomps	0.1.20	0.1.20
libconfig	1.7.2	1.7.2
libcurl-minimal	8.5.0	8.5.0

パッケージ	AMI	Hyper-V VHDX
libdb	5.3.28	5.3.28
libdhash	0.5.0	
libdnf	0.69.0	0.69.0
libeconf	0.4.0	0.4.0
libedit	3.1	3.1
libev	4.33	4.33
libevent	2.1.12	2.1.12
libfdisk	2.37.4	2.37.4
libffi	3.4.4	3.4.4
libfido2	1.10.0	1.10.0
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	11.4.1
libgpg-error	1.42	1.42
libibverbs	48.0	48.0
libidn2	2.3.2	2.3.2
libini_config	1.3.1	1.3.1
libkcapi	1.4.0	1.4.0
libkcapi-hmacalc	1.4.0	1.4.0
libldb	2.6.2	

パッケージ	AMI	Hyper-V VHDX
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	0.1.5	0.1.5
librepo	1.14.5	1.14.5
libreport-filessystem	2.15.2	2.15.2
libseccomp	2.5.3	2.5.3
libselinux	3.4	3.4
libselinux-utils	3.4	3.4

パッケージ	AMI	Hyper-V VHDX
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	
libtextstyle	0.21	0.21
libtirpc	1.3.3	1.3.3
libunistring	0.9.10	0.9.10
libuser	0.63	0.63

パッケージ	AMI	Hyper-V VHDX
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
lmbd-libs	0.9.29	0.9.29
logrotate	3.20.1	3.20.1
lsyf	4.94.0	4.94.0
lua-libs	5.4.4	5.4.4
lua-srpm-macros	1	1
lz4-libs	1.9.4	1.9.4
man-db	2.9.3	2.9.3
man-pages	5.10	5.10
microcode_ctl	2.1	2.1
mpfr	4.1.0	4.1.0

パッケージ	AMI	Hyper-V VHDX
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	6.9.7.1
openblas-srpm-macros	2	2

パッケージ	AMI	Hyper-V VHDX
openldap	2.4.57	2.4.57
openssh	8.7p1	8.7p1
openssh-clients	8.7p1	8.7p1
openssh-server	8.7p1	8.7p1
openssl	3.0.8	3.0.8
openssl-lib	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-lib	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

パッケージ	AMI	Hyper-V VHDX
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
perl-mro	1.23	1.23

パッケージ	AMI	Hyper-V VHDX
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-subst	1.03	1.03
perl-Symbol	1.08	1.08
perl-Term-ANSIColor	5.01	5.01
perl-Term-Cap	1.17	1.17
perl-Text-ParseWords	3.30	3.30

パッケージ	AMI	Hyper-V VHDX
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
python3-babel	2.9.1	2.9.1

パッケージ	AMI	Hyper-V VHDX
python3-cffi	1.14.5	1.14.5
python3-chardet	4.0.0	4.0.0
python3-colorama	0.4.4	0.4.4
python3-configobj	5.0.6	5.0.6
python3-cryptography	36.0.1	36.0.1
python3-daemon	2.3.0	
python3-dateutil	2.8.1	2.8.1
python3-dbus	1.2.18	1.2.18
python3-distro	1.5.0	1.5.0
python3-dnf	4.14.0	4.14.0
python3-dnf-plugins-core	4.3.0	4.3.0
python3-docutils	0.16	0.16
python3-gpg	1.15.1	1.15.1
python3-hawkey	0.69.0	0.69.0
python3-idna	2.10	2.10
python3-jinja2	2.11.3	2.11.3
python3-jmespath	0.10.0	0.10.0
python3-jsonpatch	1.21	1.21
python3-jsonpointer	2.0	2.0
python3-jjsonschema	3.2.0	3.2.0

パッケージ	AMI	Hyper-V VHDX
python3-libcomps	0.1.20	0.1.20
python3-libdnf	0.69.0	0.69.0
python3-libs	3.9.16	3.9.16
python3-libselenium	3.4	3.4
python3-libsemanage	3.4	3.4
python3-libstorage mgmt	1.9.4	1.9.4
python3-lockfile	0.12.2	
python3-markupsafe	1.1.1	1.1.1
python3-netifaces	0.10.6	0.10.6
python3-oauthlib	3.0.2	3.0.2
python3-pip-wheel	21.3.1	21.3.1
python3-ply	3.11	3.11
python3-policycore utils	3.4	3.4
python3-prettytable	0.7.2	0.7.2
python3-prompt-too lkit	3.0.24	3.0.24
python3-pycparser	2.20	2.20
python3-pyrsistent	0.17.3	0.17.3
python3-pyserial	3.4	3.4

パッケージ	AMI	Hyper-V VHDX
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
readline	8.1	8.1

パッケージ	AMI	Hyper-V VHDX
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
selinux-policy-targeted	37.22	37.22
setup	2.13.7	2.13.7
shadow-utils	4.9	4.9
slang	2.3.2	2.3.2

パッケージ	AMI	Hyper-V VHDX
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
systemtap-runtime	4.8	4.8
tar	1.34	1.34
tbb	2020 年 3 月	2020 年 3 月
tcpdump	4.99.1	4.99.1

パッケージ	AMI	Hyper-V VHDX
tcsh	6.24.07	6.24.07
time	1.9	1.9
traceroute	2.1.3	2.1.3
tzdata	2024年a	2024年a
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
xfspgrog	5.18.0	5.18.0
xxd	9.0.2153	9.0.2153

パッケージ	AMI	Hyper-V VHDX
xxhash-libs	0.8.0	0.8.0
xz	5.2.5	5.2.5
xz-libs	5.2.5	5.2.5
yum	4.14.0	4.14.0
zip	3.0	3.0
zlib	1.2.11	1.2.11
zram-generator	1.1.2	
zram-generator-def aults	1.1.2	
zstd	1.5.5	1.5.5

AL2023 の更新

セキュリティアップデートや新機能を活用できるように、AL2023リリースを常に最新の状態に保つことが重要です。AL2023 を使用すると、[AL2023のバージョン管理されたリポジトリによる確定的アップグレードの使用](#) を使用して、環境全体でパッケージバージョンおよび更新の一貫性を確保できます。

トピック

- [新しいアップデートの通知を受け取る](#)
- [AL2023 でパッケージとオペレーティングシステムの更新を管理する](#)
- [AL2023のバージョン管理されたリポジトリによる確定的アップグレードの使用](#)
- [AL2023 でのカーネルライブパッチ](#)

新しいアップデートの通知を受け取る

新しい AL2023 AMI がリリースされるたびに通知を受け取ることができます。通知は、以下のトピックを使用して [Amazon SNS](#) で公開されます。

```
arn:aws:sns:us-east-1:137112412989:amazon-linux-2023-ami-updates
```

新しい AL2023 AMI が公開されると、メッセージがここに投稿されます。AMI のバージョンはメッセージに含まれます。

これらのメッセージは複数の方法で受信できます。以下の方法を使用することをおすすめします。

1. [\[Amazon SNS コンソール\]](#) を開きます。
2. ナビゲーションバーで、AWS リージョン 必要に応じてを米国東部 (バージニア北部) に変更します。サブスクライブする SNS 通知が作成されたリージョンを選択する必要があります。
3. ナビゲーションペインで、[Subscriptions]、[Create subscription] の順に選択します。
4. [サブスクリプションの作成] ダイアログボックスで、次の操作を行います。
 - a. トピック ARN には、次の Amazon リソースネーム (ARN) をコピーして貼り付けます。 **arn:aws:sns:us-east-1:137112412989:amazon-linux-2023-ami-updates**
 - b. [プロトコル] で [E メール] を選択します。
 - c. [エンドポイント] に、通知を受信するために使用できる E メールアドレスを入力します。

- d. [サブスクリプションを作成] を選択します。
5. 「AWS 通知-購読確認」という件名の確認メールが届きます。メールを開いて [サブスクリプションを確定] を選択して受信登録を完了します。

AL2023 でパッケージとオペレーティングシステムの更新を管理する

以前のバージョンの Amazon Linux とは異なり、AL2023 AMIs は特定のバージョンの Amazon Linux リポジトリにロックされます。AL2023 インスタンスにセキュリティ修正とバグ修正の両方を適用するには、DNF 設定を更新します。または、新しい AL2023 インスタンスを起動します。

このセクションでは、実行中のインスタンスの DNF パッケージとリポジトリを管理する方法について説明します。また、ユーザーデータスクリプトから、使用可能な最新の Amazon Linux リポジトリを起動時に有効にするように DNF を設定する方法についても説明します。詳細については、「[DNF コマンドリファレンス](#)」を参照してください。

トピック

- [使用可能なパッケージ更新の確認](#)
- [DNF およびリポジトリバージョンを使用してセキュリティ更新を適用します。](#)
- [\(セキュリティ\) 更新後の自動サービス再起動](#)
- [最新のリポジトリバージョンを有効にしたインスタンスの起動](#)
- [パッケージサポート情報の取得](#)
- [新しいリポジトリバージョンの確認](#)
- [新しいリポジトリの追加、有効化、無効化](#)
- [cloud-init によるリポジトリの追加](#)

使用可能なパッケージ更新の確認

`dnf check-update` コマンドを使用して、システムの更新を確認できます。AL2023 の場合は、コマンドに `--releasever=version-number` オプションを追加することをお勧めします。

このオプションを追加すると、DNF は新しいバージョンのリポジトリの更新も確認します。例えば、`dnf check-update` コマンドを実行した後は、返された最新のバージョンを `version-number` の値として使用します。

最新バージョンのリポジトリを使用するようにインスタンスが更新された場合、出力には更新するすべてのパッケージのリストが含まれます。

Note

`dnf check-update` コマンドにオプションフラグを付けてリリースバージョンを指定しない場合、現在設定されているリポジトリバージョンのみが確認されます。つまり、新しいバージョンのリポジトリにあるパッケージは確認されません。

```
$ sudo dnf check-update --releasever=2023.0.20230210
```

```
Last metadata expiration check: 0:06:13 ago on Mon 13 Feb 2023 10:39:32 PM UTC.
```

```
bind-libs.x86_64                32:9.16.27-1.amzn2023      amazonlinux
bind-license.noarch             32:9.16.27-1.amzn2023      amazonlinux
bind-utils.x86_64              32:9.16.27-1.amzn2023      amazonlinux
cloud-init.noarch              22.2.2-1.amzn2023.1.4      amazonlinux
dnf.noarch                      4.12.0-2.amzn2023.0.1      amazonlinux
dnf-data.noarch                4.12.0-2.amzn2023.0.1      amazonlinux
dracut.x86_64                  055-6.amzn2023.0.4         amazonlinux
dracut-config-generic.x86_64   055-6.amzn2023.0.4         amazonlinux
glib2.x86_64                   2.73.2-678.amzn2023        amazonlinux
gmp.x86_64                     1:6.2.1-2.amzn2023         amazonlinux
grep.x86_64                    3.8-1.amzn2023.0.1        amazonlinux
kpatch-runtime.noarch         0.9.4-7.amzn2023          amazonlinux
libgcc.x86_64                  11.3.1-2.amzn2023.0.6      amazonlinux
libgomp.x86_64                 11.3.1-2.amzn2023.0.6      amazonlinux
libpkgconf.x86_64              1.7.3-7.amzn2023.0.1       amazonlinux
libstdc++.x86_64               11.3.1-2.amzn2023.0.6      amazonlinux
lz4-libs.x86_64                1.9.4-1.amzn2023          amazonlinux
pkgconf.x86_64                 1.7.3-7.amzn2023.0.1       amazonlinux
pkgconf-m4.noarch              1.7.3-7.amzn2023.0.1       amazonlinux
pkgconf-pkg-config.x86_64     1.7.3-7.amzn2023.0.1       amazonlinux
python3-dnf.noarch             4.12.0-2.amzn2023.0.1      amazonlinux
python3-rpm.x86_64             4.16.1.3-12.amzn2023.0.2   amazonlinux
rpm.x86_64                     4.16.1.3-12.amzn2023.0.2   amazonlinux
rpm-build-libs.x86_64         4.16.1.3-12.amzn2023.0.2   amazonlinux
rpm-libs.x86_64                4.16.1.3-12.amzn2023.0.2   amazonlinux
rpm-plugin-selinux.x86_64     4.16.1.3-12.amzn2023.0.2   amazonlinux
rpm-plugin-systemd-inhibit.x86_64 4.16.1.3-12.amzn2023.0.2   amazonlinux
rpm-sign-libs.x86_64          4.16.1.3-12.amzn2023.0.2   amazonlinux
slang.x86_64                   2.3.2-9.amzn2023.0.1       amazonlinux
```

system-release.noarch	2023.0.20230210-0.amzn2023	amazonlinux
systemd.x86_64	250.8-1.amzn2023.0.1	amazonlinux
systemd-libs.x86_64	250.8-1.amzn2023.0.1	amazonlinux
systemd-networkd.x86_64	250.8-1.amzn2023.0.1	amazonlinux
systemd-pam.x86_64	250.8-1.amzn2023.0.1	amazonlinux
systemd-resolved.x86_64	250.8-1.amzn2023.0.1	amazonlinux
systemd-udev.x86_64	250.8-1.amzn2023.0.1	amazonlinux
vim-common.x86_64	2:9.0.327-1.amzn2023.0.1	amazonlinux
vim-data.noarch	2:9.0.327-1.amzn2023.0.1	amazonlinux
vim-enhanced.x86_64	2:9.0.327-1.amzn2023.0.1	amazonlinux
vim-filesystem.noarch	2:9.0.327-1.amzn2023.0.1	amazonlinux
vim-minimal.x86_64	2:9.0.327-1.amzn2023.0.1	amazonlinux
wget.x86_64	1.21.3-1.amzn2023	amazonlinux
yum.noarch	4.12.0-2.amzn2023.0.1	amazonlinux

このコマンドでは、新しいパッケージが使用可能な場合、返されるコードは 100 です。新しいパッケージがない場合、返されるコードは 0 です。さらに、出力には更新予定のパッケージもすべて一覧表示されます。

DNF およびリポジトリバージョンを使用してセキュリティ更新を適用します。

新しいパッケージ更新およびセキュリティ更新は、新しいリポジトリバージョンでのみ使用可能です。以前の AL2023 AMI バージョンから起動したインスタンスの場合、セキュリティ更新をインストールする前にリポジトリバージョンを更新する必要があります。dnf check-release-update コマンドには、システムにインストールされているすべてのパッケージを新しいリポジトリのバージョンに更新する更新コマンドの例が含まれています。

```
$ sudo dnf update --releasever=2023.0.20230210
Last metadata expiration check: 0:01:40 ago on Mon 13 Feb 2023 10:39:32 PM UTC.
Dependencies resolved.
=====
Package                Arch   Version                                Repository   Size
=====
Upgrading:
bind-libs               x86_64 32:9.16.27-1.amzn2023                amazonlinux 1.2 M
bind-license            noarch 32:9.16.27-1.amzn2023                amazonlinux 16 k
bind-utils              x86_64 32:9.16.27-1.amzn2023                amazonlinux 202 k
cloud-init              noarch 22.2.2-1.amzn2023.1.4                amazonlinux 1.1 M
dnf                     noarch 4.12.0-2.amzn2023.0.1                amazonlinux 454 k
dnf-data                noarch 4.12.0-2.amzn2023.0.1                amazonlinux 42 k
```

dracut	x86_64	055-6.amzn2023.0.4	amazonlinux	345 k
dracut-config-generic	x86_64	055-6.amzn2023.0.4	amazonlinux	8.5 k
glib2	x86_64	2.73.2-678.amzn2023	amazonlinux	2.7 M
gmp	x86_64	1:6.2.1-2.amzn2023	amazonlinux	324 k
grep	x86_64	3.8-1.amzn2023.0.1	amazonlinux	316 k
kpatch-runtime	noarch	0.9.4-7.amzn2023	amazonlinux	30 k
libgcc	x86_64	11.3.1-2.amzn2023.0.6	amazonlinux	121 k
libgomp	x86_64	11.3.1-2.amzn2023.0.6	amazonlinux	296 k
libpkgconf	x86_64	1.7.3-7.amzn2023.0.1	amazonlinux	37 k
libstdc++	x86_64	11.3.1-2.amzn2023.0.6	amazonlinux	758 k
lz4-libs	x86_64	1.9.4-1.amzn2023	amazonlinux	81 k
pkgconf	x86_64	1.7.3-7.amzn2023.0.1	amazonlinux	41 k
pkgconf-m4	noarch	1.7.3-7.amzn2023.0.1	amazonlinux	15 k
pkgconf-pkg-config	x86_64	1.7.3-7.amzn2023.0.1	amazonlinux	11 k
python3-dnf	noarch	4.12.0-2.amzn2023.0.1	amazonlinux	415 k
python3-rpm	x86_64	4.16.1.3-12.amzn2023.0.2	amazonlinux	89 k
rpm	x86_64	4.16.1.3-12.amzn2023.0.2	amazonlinux	487 k
rpm-build-libs	x86_64	4.16.1.3-12.amzn2023.0.2	amazonlinux	92 k
rpm-libs	x86_64	4.16.1.3-12.amzn2023.0.2	amazonlinux	311 k
rpm-plugin-selinux	x86_64	4.16.1.3-12.amzn2023.0.2	amazonlinux	18 k
rpm-plugin-systemd-inhibit	x86_64	4.16.1.3-12.amzn2023.0.2	amazonlinux	19 k
rpm-sign-libs	x86_64	4.16.1.3-12.amzn2023.0.2	amazonlinux	22 k
slang	x86_64	2.3.2-9.amzn2023.0.1	amazonlinux	410 k
system-release	noarch	2023.0.20230210-0.amzn2023	amazonlinux	25 k
systemd	x86_64	250.8-1.amzn2023.0.1	amazonlinux	4.2 M
systemd-libs	x86_64	250.8-1.amzn2023.0.1	amazonlinux	615 k
systemd-networkd	x86_64	250.8-1.amzn2023.0.1	amazonlinux	614 k
systemd-pam	x86_64	250.8-1.amzn2023.0.1	amazonlinux	335 k
systemd-resolved	x86_64	250.8-1.amzn2023.0.1	amazonlinux	277 k
systemd-udev	x86_64	250.8-1.amzn2023.0.1	amazonlinux	1.9 M
vim-common	x86_64	2:9.0.327-1.amzn2023.0.1	amazonlinux	7.2 M
vim-data	noarch	2:9.0.327-1.amzn2023.0.1	amazonlinux	27 k
vim-enhanced	x86_64	2:9.0.327-1.amzn2023.0.1	amazonlinux	1.8 M
vim-filesystem	noarch	2:9.0.327-1.amzn2023.0.1	amazonlinux	21 k
vim-minimal	x86_64	2:9.0.327-1.amzn2023.0.1	amazonlinux	764 k
wget	x86_64	1.21.3-1.amzn2023	amazonlinux	813 k
yum	noarch	4.12.0-2.amzn2023.0.1	amazonlinux	39 k

Transaction Summary

```
=====
Upgrade 43 Packages
```

```
...
```

パッケージをセキュリティ機能のみで更新する `--security` オプションを追加できます。

```
$ sudo dnf update --releasever=2023.0.20230210 --security
Amazon Linux 2023 repository          18 MB/s | 11 MB    00:00
Last metadata expiration check: 0:00:02 ago on Mon 13 Feb 2023 10:39:32 PM UTC.
Dependencies resolved.
=====
Package                Arch      Version                                Repository      Size
=====
Upgrading:
bind-libs              x86_64   32:9.16.27-1.amzn2023                amazonlinux    1.2 M
bind-license           noarch   32:9.16.27-1.amzn2023                amazonlinux    16 k
bind-utils             x86_64   32:9.16.27-1.amzn2023                amazonlinux    202 k
gmp                    x86_64   1:6.2.1-2.amzn2023                   amazonlinux    324 k
lz4-libs               x86_64   1.9.4-1.amzn2023                     amazonlinux    81 k
vim-common             x86_64   2:9.0.327-1.amzn2023.0.1            amazonlinux    7.2 M
vim-data               noarch   2:9.0.327-1.amzn2023.0.1            amazonlinux    27 k
vim-enhanced           x86_64   2:9.0.327-1.amzn2023.0.1            amazonlinux    1.8 M
vim-filesystem         noarch   2:9.0.327-1.amzn2023.0.1            amazonlinux    21 k
vim-minimal            x86_64   2:9.0.327-1.amzn2023.0.1            amazonlinux    764 k
wget                   x86_64   1.21.3-1.amzn2023                    amazonlinux    813 k

Transaction Summary
=====
Upgrade 11 Packages
...
```

AL2023 パッケージのバージョンを確認するには、以下のうち 1 つ以上を実行します。

- `dnf check-update` コマンドを実行します。
- Amazon Linux リポジトリの更新 SNS トピック (`arn:aws:sns:us-east-1:137112412989:amazon-linux-2023-ami-updates`) をサブスクライブします。詳細については、「[Amazon Simple 通知サービス デベロッパーガイド](#)」の「Amazon SNS トピックのサブスクライブ」を参照してください。
- 「[AL2023 リリースノート](#)」を定期的に参照してください。

⚠ Important

実行中のインスタンスにセキュリティ更新を適用するときは、DNF が最新のリポジトリバージョンを指していることを確認します。

(セキュリティ) 更新後の自動サービス再起動

Amazon Linux には、[スマート再起動](#)パッケージが付属するようになりました。は、システムパッケージマネージャーを使用してパッケージがインストールまたは削除されるたびに、システム更新時に systemd サービス Smart-restart を再起動します。これは `dnf (update|upgrade|downgrade)`、 が実行されるたびに発生します。

Smart-restart は、からの `needs-restarting` パッケージ `dnf-utils` とカスタム拒否リストメカニズムを使用して、再起動する必要があるサービスや、システムの再起動が推奨されるかどうかを判断します。システムの再起動が推奨されると、再起動ヒントマーカーファイルが生成されます (`/run/smart-restart/reboot-hint-marker`)。

smart-restart をインストールするには

次のDNFコマンドを実行します (他のパッケージの場合と同様)。

```
$ sudo dnf install smart-restart
```

インストール後、後続のトランザクションによって `smart-restart` ロジックがトリガーされます。

拒否リスト

Smart-restart は、特定のサービスの再起動をブロックするように指示できます。ブロックされたサービスは、再起動が必要かどうかの判断には影響しません。追加のサービスをブロックするには、次の例 `/etc/smart-restart-conf.d/` に示すように、`-denylist` にサフィックスが付いたファイルを追加します。

```
$ cat /etc/smart-restart-conf.d/custom-denylist
# Some comments
myservice.service
```

Note

再起動が必要かどうかを判断する際に、すべての*-denylistファイルが読み取られ、評価されます。

カスタムフック

拒否リストに加え、smart-restartはサービスの再起動の前後にカスタムスクリプトを実行するメカニズムを提供します。カスタムスクリプトを使用して、手動で準備ステップを実行したり、他のコンポーネントに残りの再起動や完了した再起動を通知したりできます。

サフィックス -pre-restartまたは /etc/smart-restart-conf.d/を持つ内のすべてのスクリプト-post-restartが実行されます。順序が重要な場合は、次の例に示すように、すべてのスクリプトの前に数字を付けて実行順序を確認します。

```
$ ls /etc/smart-restart-conf.d/*-pre-restart
001-my-script-pre-restart
002-some-other-script-pre-restart
```

最新のリポジトリバージョンを有効にしたインスタンスの起動

ユーザーデータスクリプトに DNF コマンドを追加して、Amazon Linux AMI の起動時にどの RPM パッケージをインストールするかを制御できます。以下の例では、ユーザーデータスクリプトを使用して、ユーザーデータスクリプトを使用して起動されるすべてのインスタンスに同じパッケージ更新がインストールされていることを確認します。

```
#!/bin/bash
dnf update --releasever=2023.0.20230210
# Additional setup and install commands below
dnf install httpd php7.4 mysql80
```

このスクリプトは superuser (root) として実行する必要があります。これを行うには、以下のコマンドを実行します。

```
$ sudo sh -c "bash nameofscript.sh"
```

詳細については、「Amazon EC2 ユーザーガイド」の「[ユーザーデータとシェルスクリプト](#)」を参照してください。

Note

ユーザーデータスクリプトを使用する代わりに、最新の Amazon Linux AMI を起動するか、Amazon Linux AMI をベースにしたカスタム AMI を起動します。最新の Amazon Linux AMI には必要な更新がすべてインストールされており、特定のリポジトリバージョンを指すように設定されています。

パッケージサポート情報の取得

AL2023 には、さまざまなオープンソースソフトウェアプロジェクトが組み込まれています。これらのプロジェクトはそれぞれ Amazon Linux とは独立して管理され、リリースと end-of-support スケジュールは異なります。これらのさまざまなパッケージに関する Amazon Linux 固有の情報を提供するために、DNF `supportinfo` プラグインはパッケージに関するメタデータを提供します。以下の例では、`dnf supportinfo` コマンドは `glibc` パッケージのメタデータを返します。

```
$ 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.
```

新しいリポジトリバージョンの確認

AL2023 インスタンスでは、DNF ユーティリティを使用してリポジトリを管理し、更新された RPM パッケージを適用できます。これらのパッケージは Amazon Linux リポジトリにあります。DNF コマンド `dnf check-release-update` を使用して、DNF リポジトリの新しいバージョンを確認できます。

```
$ sudo dnf check-release-update
WARNING:
```

A newer release of "Amazon Linux" is available.

Available Versions:

Version 2023.0.20230210:

Run the following command to update to 2023.0.20230210:

```
dnf update --releasever=2023.0.20230210
```

Release notes:

<https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes.html>

これにより、使用可能なすべての新しいバージョンの DNF リポジトリのすべてのリストが返されます。何も返されない場合は、DNF が現在使用可能な最新バージョンを使用するように設定されていることを意味します。現在インストールされている system-release パッケージのバージョンによって releasever DNF 変数が設定されます。現在のリポジトリバージョンを確認するには、以下のコマンドを実行します。

```
$ rpm -q system-release --qf "%{VERSION}\n"
```

DNF パッケージトランザクション (インストール、更新、削除コマンドなど) を実行すると、リポジトリの新しいバージョンを知らせる警告メッセージが表示されます。例えば、AL2023 の古いバージョンから起動されたインスタンスに httpd パッケージをインストールすると、以下の出力が返されます。

```
$ 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!

新しいリポジトリの追加、有効化、無効化

DNF パッケージで異なるリポジトリからパッケージをインストールには、`/etc/dnf/dnf.conf` ファイル、または `/etc/yum.repos.d` ディレクトリにあるお客様の `repository.repo` ファイルに、リポジトリ情報を追加します。この操作は手動で行うことができます。ただし、ほとんどの DNF リポジトリのリポジトリ URL で、独自の `repository.repo` ファイルが提供されています。

Note

現時点では、AL2023 に追加できるリポジトリはありません。これは今後変更される可能性があります。また、独自のパッケージを作成して、そのパッケージを AL2023 エンタープライズ環境で使用できるようにすることもできます。パッケージを使用する前に、パッケージが保存されているリポジトリを追加して有効にする必要があります。

現在有効になっているリポジトリを確認するには、以下のコマンドを実行します。

```
$ dnf repolist all --verbose
```

```
Loaded plugins: builddep, changelog, config-manager, copr, debug, debuginfo-install,
download, generate_completion_cache, groups-manager, needs-restarting, playground,
release-notification, repoclosure, repodiff, repograph, repomanage, reposync,
supportinfo
```

```
DNF version: 4.12.0
```

```
cachedir: /var/cache/dnf
```

```
Last metadata expiration check: 0:00:02 ago on Wed Mar 1 23:40:15 2023.
```

```
Repo-id           : amazonlinux
Repo-name         : Amazon Linux 2023 repository
Repo-status      : enabled
Repo-revision    : 1677203368
Repo-updated     : Fri Feb 24 01:49:28 2023
Repo-pkgs        : 12632
Repo-available-pkgs: 12632
Repo-size        : 12 G
Repo-mirrors     : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/mirrors/2023.0.20230222/x86_64/mirror.list
Repo-baseurl    : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/guids/
cf9296325a6c46ff40c775a8e2d632c4c3fd9d9164014ce3304715d61b90ca8e/x86_64/
                  : (0 more)
Repo-expire      : 172800 second(s) (last: Wed Mar 1 23:40:15
                  : 2023)
Repo-filename    : /etc/yum.repos.d/amazonlinux.repo
```

```
Repo-id           : amazonlinux-debuginfo
Repo-name         : Amazon Linux 2023 repository - Debug
Repo-status      : disabled
Repo-mirrors     : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/mirrors/2023.0.20230222/debuginfo/x86_64/mirror.list
Repo-expire      : 21600 second(s) (last: unknown)
```

```
Repo-filename      : /etc/yum.repos.d/amazonlinux.repo

Repo-id            : amazonlinux-source
Repo-name          : Amazon Linux 2023 repository - Source packages
Repo-status        : disabled
Repo-mirrors       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/mirrors/2023.0.20230222/SRPMS/mirror.list
Repo-expire        : 21600 second(s) (last: unknown)
Repo-filename      : /etc/yum.repos.d/amazonlinux.repo

Repo-id            : kernel-livepatch
Repo-name          : Amazon Linux 2023 Kernel Livepatch repository
Repo-status        : disabled
Repo-mirrors       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/kernel-livepatch/mirrors/al2023/x86_64/mirror.list
Repo-expire        : 172800 second(s) (last: unknown)
Repo-filename      : /etc/yum.repos.d/kernel-livepatch.repo

Repo-id            : kernel-livepatch-source
Repo-name          : Amazon Linux 2023 Kernel Livepatch repository -
                   : Source packages
Repo-status        : disabled
Repo-mirrors       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/kernel-livepatch/mirrors/al2023/SRPMS/mirror.list
Repo-expire        : 21600 second(s) (last: unknown)
Repo-filename      : /etc/yum.repos.d/kernel-livepatch.repo
Total packages: 12632
```

Note

--verbose オプションフラグを追加しない場合、出力には、Repo-id、Repo-name、および Repo-status の情報のみが含まれます。

yum リポジトリを `/etc/yum.repos.d` ディレクトリに追加する方法

1. `.repo` ファイルの場所を検索します。この例では、`.repo` ファイルは、<https://www.example.com/repository.repo> にあります。
2. `dnf config-manager` コマンドを使用してリポジトリを追加します。

```
$ 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
```

リポジトリをインストールしたら、以下の手順で説明するように有効にする必要があります。

/etc/yum.repos.d で yum リポジトリを有効にするには、`--enable` フラグと `#####` 名を指定して `dnf config-manager` コマンドを実行します。

```
$ sudo dnf config-manager --enable repository
```

Note

リポジトリを無効にするには、同じコマンド構文を使用しますが、コマンド内の `--enable` を `--disable` に置き換えます。

cloud-init によるリポジトリの追加

前の方法でリポジトリを追加するほかに、cloud-init フレームワークを使用して新しいリポジトリを追加することもできます。

新しいパッケージリポジトリを追加するには、以下のテンプレートを使用することをお勧めします。このファイルをローカルに保存することを検討してください。

```
#cloud-config
yum_repos:
  repository.repo:
    baseurl: https://www.example.com/
    enabled: true
    gpgcheck: true
    gpgkey: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EXAMPLE
    name: Example Repository
```

Note

cloud-init を使用する利点の 1 つは、設定ファイルに `packages:` セクションを追加できることです。このセクションには、インストールするパッケージの名前を含めることができます。パッケージは、デフォルトのリポジトリからも、cloud-config ファイルに追加した新しいリポジトリからもインストールできます。

YAML ファイルの構造に関するより具体的な情報については、「[cloud-init ドキュメント](#)」の「[YUM リポジトリの追加](#)」を参照してください。

YAML フォーマットファイルを設定したら、AWS CLI の cloud-init フレームワークで実行できます。必要な操作を呼び出すための `--userdata` オプションと `.yaml` ファイル名を必ず含めてください。

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

AL2023 のバージョン管理されたリポジトリによる確定的アップグレードの使用

Note

デフォルトでは、AL2023 インスタンスは起動時に追加のクリティカルかつ重要なセキュリティアップデートを自動的に受信しません。インスタンスには、AL2023 のバージョンで利用可能であった更新および選択した AMI が含まれています。

メジャーリリースとマイナーリリースから受け取る更新を管理する

AL2023 を使用すると、環境全体でパッケージバージョンと更新の一貫性を確保できます。同じ Amazon マシンイメージ (AMI) の複数のインスタンスで一貫性を保つこともできます。デフォルトで

有効になっているバージョン化されたリポジトリによる確定的なアップグレード機能を使用すると、特定の必要に応じたスケジュールに基づいて更新を適用できます。

新しいパッケージ更新をリリースするたびに、ロックする新しいバージョンと、そのバージョンにロックされる新しい AMI があります。

AL2023 はリポジトリの特定のバージョンにロックします。これはメジャーバージョンとマイナーバージョンの両方でサポートされています。SSM パラメータによって公開される AL2023 AMI は、常に最新バージョンです。重要かつ重要なセキュリティアップデートを含む、up-to-date パッケージとアップデートが最も多く含まれています。

既存の AMI からインスタンスを起動した場合、更新は自動的に適用されません。プロビジョニングの一部としてインストールされたすべての追加パッケージは、既存の AMI のリポジトリバージョンにマップされます。

この機能では、環境全体でパッケージバージョンと更新の一貫性を確保する必要があります。これは、同じ AMI から複数のインスタンスを起動する場合に特に当てはまります。必要に応じたスケジュールに基づいて更新を適用できます。特定の更新セットを特定のリポジトリバージョンにロックすることもできるので、起動時に特定の更新セットを適用することもできます。

メジャーバージョンとマイナーバージョンのアップグレードの違い

AL2023 のメジャーバージョンリリースには大規模な更新が含まれており、パッケージを追加、削除、または更新する場合があります。互換性を確保するため、新しいメジャーバージョンでアプリケーションをテストしてからインスタンスを新しいメジャーバージョンにアップグレードします。

AL2023 のマイナーバージョンリリースには機能更新とセキュリティ更新が含まれていますが、パッケージの変更は含まれていません。これにより、Linux の機能とシステムライブラリ API を新しいバージョンでも引き続き利用できます。更新前にアプリケーションをテストする必要はありません。

AL2023 リポジトリから入手できるパッケージアップデートを管理します。

AL2023 リポジトリの新しいバージョンを公開しても、以前のバージョンはすべて引き続き利用できます。デフォルトでは、リポジトリのバージョンを管理するためのプラグインは、AMI の構築に使用されたのと同じバージョンにロックされます。パッケージの更新をコントロールする場合は、以下の手順に従います。

1. 以下のコマンドを実行して、使用可能なリポジトリバージョンを確認します。

```
$ sudo dnf check-release-update
```

2. 以下のコマンドを実行して、バージョンを選択します。

```
$ sudo dnf --releasever=version update
```

このコマンドは、dnf を使用して現在の Amazon Linux リリースバージョンからコマンドラインで指定されたリリースバージョンへの更新を開始します。パッケージ更新のリストは dnf によって表示されます。更新が処理される前に、更新を確認する必要があります。更新が完了すると、新しいリリースバージョンが、今後のすべてのアクティビティに使用される dnf のデフォルトのリリースバージョンになります。

詳細については、「[AL2023 でパッケージとオペレーティングシステムの更新を管理する](#)」を参照してください。

バージョン管理されたリポジトリの使用による確定的なアップグレード

トピック

- [確定的なアップグレードシステムを使用する](#)
- [確定的なアップグレードされたシステムの選択的更新](#)
- [確定的なアップグレードでの永続的上書きの使用](#)

確定的なアップグレードシステムを使用する

dnf upgrade コマンドを実行すると、システムは releasever 変数が指定するリポジトリ内のアップグレードをチェックします。有効な releasever は、### か、*2023.4.20240513* などの日付スタンプ付きのバージョンです。

以下のいずれかの方法を使用して、releasever の値を変更できます。これらの方法は、システム優先度の降順で一覧表示されています。つまり、方法 1 は方法 2 と 3 を上書きし、方法 2 は方法 3 を上書きします。

1. コマンドラインフラグ内の値 `--releasever=latest` (使用する場合)。
2. 上書き変数ファイルで指定されている値 `/etc/dnf/vars/releasever` (設定されている場合)。
3. 現在インストールされている `system-release` パッケージのバージョン。

以下の例ではバージョンは、*2023.0.20230210* です。

```
$ rpm -q system-release
system-release-2023.0.20230210-0.amzn2023.noarch
```

新しくインストールされたシステムには、上書き変数は存在しません。システムはインストールした system-release のバージョンのみにロックされているため、アップグレードはできません。

```
$ cat /etc/dnf/vars/releasever
cat: /etc/dnf/vars/releasever: No such file or directory
```

```
$ sudo dnf upgrade
Last metadata expiration check: 0:00:02 ago on Wed 15 Feb 2023 06:14:12 PM UTC.
Dependencies resolved.
Nothing to do.
Complete!
```

releasever フラグを使用して必要なバージョンを指定することで、特定のバージョンのパッケージを取得できます。

```
$ rpm -q system-release
system-release-2023.0.20230222-0.amzn2023.noarch
```

```
$ sudo dnf upgrade --releasever=2023.0.20230329
Amazon Linux 2023 repository                26 MB/s | 12 MB    00:00
Dependencies resolved.
=====
Package                Arch    Version                                Repository    Size
=====
Installing:
kernel                  aarch64 6.1.21-1.45.amzn2023                  amazonlinux   26 M
Upgrading:
amazon-linux-repo-s3   noarch  2023.0.20230329-0.amzn2023            amazonlinux   18 k
ca-certificates        noarch  2023.2.60-1.0.amzn2023.0.1           amazonlinux   828 k
cloud-init              noarch  22.2.2-1.amzn2023.1.7                 amazonlinux   1.1 M

... [ list edited for clarity ]

system-release          noarch  2023.0.20230329-0.amzn2023            amazonlinux   29 k

... [ list edited for clarity ]

vim-data                noarch  2:9.0.1403-1.amzn2023.0.1            amazonlinux   25 k
```

```
vim-minimal          aarch64 2:9.0.1403-1.amzn2023.0.1  amazonlinux 753 k
```

Transaction Summary

```
=====
Install    1 Package
Upgrade   42 Packages
```

```
Total download size: 56 M
```

`--releasever` オプションは `system-release` と `/etc/dnf/vars/releasever` の両方を上書きするため、このアップグレードの結果は以下のようになります。

1. アップグレードにより、以前のバージョンと新しいバージョンとの間で変更されたインストール済みパッケージがすべて置き換えられます。
2. アップグレードすると、システムは新しい `system-release` のバージョンのリポジトリにロックされます。

確定的なアップグレードされたシステムの選択的更新

システムを元のリリースバージョンにロックしたまま、最近のリリースから選択したパッケージをインストールするとします。

`dnf check-update` を使用して、アップグレードするパッケージを特定します。

```
$ 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-filessystem.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

アップグレードしたいパッケージをインストールします。sudo dnf upgrade --releasever=latest とパッケージ名を使用して、system-release パッケージが変更されないようにします。

```
$ sudo dnf upgrade --releasever=latest openssl openssl-libs
Last metadata expiration check: 0:01:28 ago on Wed 15 Feb 2023 02:52:21 AM UTC.
Dependencies resolved.
=====
Package           Arch           Version                               Repository      Size
=====
Upgrading:
openssl           aarch64       1:3.0.5-1.amzn2023.0.3              amazonlinux     1.1 M
openssl-libs     aarch64       1:3.0.5-1.amzn2023.0.3              amazonlinux     2.1 M

Transaction Summary
=====
Upgrade 2 Packages
```

Total download size: 3.2 M

Note

`sudo dnf upgrade --releasever=latest` を使用すると、`system-release` を含むすべてのパッケージが更新されます。その後、永続的な上書きを設定しない限り、バージョンは新しい `system-release` にロックされたままになります。

確定的なアップグレードでの永続的上書きの使用

`--releasever=latest` を追加する代わりに、変数値を `###` 設定することで、永続的な上書きを使用してシステムのロックを解除できます。

```
$ 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

上書き変数 `/etc/dnf/vars/releasever` を使用した場合は、以下のコマンドを使用して上書き値を消去してデフォルトのロック動作に戻します。

```
$ sudo rm /etc/dnf/vars/releasever
```

AL2023 でのカーネルライブパッチ

AL2023 のカーネルライブパッチを使用すると、実行中のアプリケーションを再起動したり中断したりすることなく、実行中の Linux カーネルにセキュリティの脆弱性と重大なバグのパッチを適用できます。さらに、カーネルライブパッチを有効にすると、インフラストラクチャを安全かつ最新に保つとともに、アプリケーションの可用性を向上させることができます。

AWS AL2023 用の 2 種類のカーネルライブパッチをリリースします。

- **セキュリティ更新** – Linux の共通脆弱性とエクスポージャー (CVE) の更新プログラムが含まれます。これらの更新プログラムは、通常、Amazon Linux Security Advisory の評価で Important または Critical と評価されます。これらは、通常、共通脆弱性評価システム (CVSS) の 7 以上のスコアに該当します。AWS 場合によっては、CVE が割り当てられる前にアップデートを提供することもあります。そのような場合、パッチはバグ修正プログラムとして提供される場合があります。
- **バグ修正** – CVE に関連付けられていない重大なバグや安定性の問題の修正プログラムが含まれます。

AWS AL2023 カーネルバージョンのカーネルライブパッチを、リリース後最大 3 か月間提供します。その後、カーネルライブパッチを引き続き入手するには、新しいカーネルバージョンに更新する必要があります。

AL2023 のカーネルライブパッチは、既存の AL2023 リポジトリから署名付きの RPM パッケージとして入手できます。パッチを個別のインスタンスにインストールするには、既存の DNF パッケージ

マネージャーワークフローを使用できます。または、AWS Systems Manager を使用してマネージドインスタンスのグループにインストールすることもできます。

AL2023 のカーネルライブパッチは、追加料金なしで提供されます。

トピック

- [制限事項](#)
- [サポートされている構成と前提条件](#)
- [カーネルライブパッチを使用する](#)

制限事項

カーネルライブパッチの適用中は、休止を実行したり、高度なデバッグツール (SystemTap、kprobes、eBPF ベースのツールなど) を使用したり、カーネルライブパッチを適用したインフラストラクチャで使用されている ftrace の出力ファイルにアクセスしたりすることはできません。

サポートされている構成と前提条件

カーネルライブパッチは、Amazon EC2 インスタンスおよび AL2023 が実行されているオンプレミスの仮想化マシンでサポートされています。

AL2023 でカーネルライブパッチを使用するには、以下を使用する必要があります。

- 64 ビット x86_64 または ARM64 アーキテクチャ
- カーネルバージョン 6.1

ポリシーの要件

AL2023 リポジトリからパッケージをダウンロードするには、Amazon EC2 はサービス所有の Amazon S3 バケットにアクセスする必要があります。環境内で Amazon S3 用の Amazon Virtual Private Cloud (VPC) エンドポイントを使用している場合は、VPC エンドポイントポリシーでそれらのパブリックバケットへのアクセスが許可されていることを確認してください。次の表では、カーネルライブパッチのために Amazon EC2 がアクセスする必要のある Amazon S3 バケットについて説明しています。

S3 バケット ARN	説明
<code>arn: aws: s3:: al2023-repos-#####-de612dc2/*</code>	AL2023 リポジトリを含むAmazon S3 バケット

カーネルライブパッチを使用する

カーネルライブパッチを有効にして個別のインスタンスで使用するには、インスタンス自体でコマンドラインを使用できます。Systems Manager を使用して、カーネルライブパッチを有効にしてマネージドインスタンスのグループで使用するには、AWS Systems Manager を使用できます。

以下のセクションでは、コマンドラインを使用して、カーネルライブパッチを有効にして個別のインスタンスで使用方法について説明します。

マネージドインスタンスのグループでカーネルライブパッチを有効にして使用方法の詳細については、「AWS Systems Manager ユーザーガイド」の「[AL2023 インスタンスでカーネルライブパッチを使用する](#)」を参照してください。

トピック

- [カーネルライブパッチの有効化](#)
- [利用可能なカーネルライブパッチを表示する](#)
- [カーネルライブパッチを適用する](#)
- [適用されたカーネルライブパッチの表示](#)
- [カーネルライブパッチの無効化](#)

カーネルライブパッチの有効化

AL2023 では、カーネルライブパッチはデフォルトでは無効になっています。ライブパッチを使用するには、カーネルライブパッチの DNF プラグインをインストールして、ライブパッチ機能を有効にする必要があります。

カーネルライブパッチを有効にする方法

1. カーネルライブパッチは、カーネルバージョン 6.1 以降の AL2023 で使用できます。カーネルバージョンを確認するには、次のコマンドを実行します。

```
$ sudo dnf list kernel
```

- カーネルライブパッチの DNF プラグインをインストールします。

```
$ sudo dnf install -y kpatch-dnf
```

- カーネルライブパッチの DNF プラグインを有効にします。

```
$ sudo dnf kernel-livepatch -y auto
```

このコマンドは、設定されているリポジトリから最新バージョンのカーネルライブパッチの RPM もインストールします。

- カーネルライブパッチの DNF プラグインが正常にインストールされたことを確認するには、以下のコマンドを実行します。

カーネルライブパッチを有効にすると、空のカーネルライブパッチの RPM が自動的に適用されます。カーネルライブパッチが正常に有効になっていれば、このコマンドは最初の空のカーネルライブパッチの RPM を含むリストを返します。

```
$ sudo rpm -qa | grep kernel-livepatch
dnf-plugin-kernel-livepatch-1.0-0.11.amzn2023.noarch
kernel-livepatch-6.1.12-17.42-1.0-0.amzn2023.x86_64
```

- kpatch パッケージをインストールします。

```
$ sudo dnf install -y kpatch-runtime
```

- 既にインストール済みの場合は、kpatch サービスを更新します。

```
$ sudo dnf update kpatch-runtime
```

- kpatch サービスを起動します。このサービスは、初期化時または起動時にすべてのカーネルライブパッチをロードします。

```
$ sudo systemctl enable kpatch.service && sudo systemctl start kpatch.service
```

利用可能なカーネルライブパッチを表示する

Amazon Linux のセキュリティアラートは、Amazon Linux Security Center に公開されます。カーネルライブパッチのアラートを含む AL2023 のセキュリティアラートの詳細については、「[Amazon Linux Security Center](#)」を参照してください。カーネルライブパッチには、ALASLIVEPATCH というプレフィクスが付きます。Amazon Linux Security Center では、バグに対応するカーネルライブパッチは一覧に表示されていない場合があります。

アドバイザリおよび CVE に対する利用可能なカーネルライブパッチは、コマンドラインを使用して見つけることもできます。

アドバイザリに対する利用可能なすべてのカーネルライブパッチを一覧表示するには

以下のコマンドを使用します。

```
$ 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
```

CVE に対する利用可能なすべてのカーネルライブパッチを一覧表示するには

以下のコマンドを使用します。

```
$ 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
```

カーネルライブパッチを適用する

カーネルライブパッチは、DNF パッケージマネージャーを使用して、通常の更新プログラムを適用するのと同じ方法で適用します。カーネルライブパッチの DNF プラグインは、適用するカーネルライブパッチを管理し、再起動も必要ありません。

Tip

カーネルが安全かつ最新に保たれるよう、カーネルライブパッチを使用して定期的にカーネルを更新することをお勧めします。

特定のカーネルライブパッチを適用するか、利用可能なカーネルライブパッチを定期的なセキュリティ更新プログラムと一緒に適用するかを選択できます。

特定のカーネルライブパッチを適用するには

1. 「[利用可能なカーネルライブパッチを表示する](#)」で説明されているコマンドのいずれかを使用して、カーネルライブパッチのバージョンを取得します。
2. AL2023 カーネルのカーネルライブパッチを適用します。

```
$ sudo dnf install kernel-livepatch-kernel_version-package_version.amzn2023.x86_64
```

例えば、以下のコマンドは、AL2023 カーネルバージョン 6.1.12-17.42 のカーネルライブパッチを適用します

```
$ sudo dnf install kernel-livepatch-6.1.12-17.42-1.0-4.amzn2023.x86_64
```

利用可能なカーネルライブパッチを定期的なセキュリティ更新プログラムと一緒に適用する方法

次のコマンドを使用します。

```
$ sudo dnf update --security
```

バグ修正プログラムを含めるには、`--security` オプションを省略します。

Important

- カーネルライブパッチを適用しても、カーネルバージョンは更新されません。バージョンは、インスタンスを再起動した後でのみ新しいバージョンに更新されます。
- AL2023 カーネルは、カーネルライブパッチを 3 か月間入手できます。その後は、そのカーネルバージョンの新しいカーネルライブパッチはリリースされなくなります。
- 3 か月が過ぎた後にカーネルライブパッチを引き続き入手するには、インスタンスを再起動して新しいカーネルバージョンに移行する必要があります。インスタンスは、更新後 3 か月間は引き続きカーネルライブパッチを受け取ります。
- お使いのカーネルバージョンのサポート期間を確認するには、以下のコマンドを実行します。

```
$ sudo dnf kernel-livepatch support
```

適用されたカーネルライブパッチの表示

適用されたカーネルライブパッチを表示するには

次のコマンドを使用します。

```
$ 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)
```

このコマンドは、ロードおよびインストールされたセキュリティ更新プログラムのカーネルライブパッチのリストを返します。出力例を次に示します。

Note

1つのカーネルライブパッチには、複数のライブパッチが含まれていてインストールされる場合があります。

カーネルライブパッチの無効化

カーネルライブパッチを使用する必要がなくなった場合は、いつでも無効にできます。

- livepatches の使用の無効化:

1. プラグインの無効化:

```
$ sudo dnf kernel-livepatch manual
```

2. kpatch サービスの無効化:

```
$ sudo systemctl disable --now kpatch.service
```

- livepatch ツールの完全削除:

1. プラグインの削除:

```
$ sudo dnf remove kpatch-dnf
```

2. kpatch-runtime の削除:

```
$ sudo dnf remove kpatch-runtime
```

3. インストールされているすべての livepatches の削除:

```
$ sudo dnf remove kernel-livepatch\*
```

AL2023 でのランタイムのプログラミング入門

AL2023 は、一部の言語ランタイムの異なるバージョンを提供しています。複数のバージョンを同時にサポートする上流プロジェクトと連携しています。名前付きバージョンパッケージのインストール方法や管理方法に関する情報は、これらのパッケージを検索してインストールする `dnf` コマンドを使って検索します。

以下のトピックでは、AL2023 における各言語エコシステムの概要を説明しています。

トピック

- [AL2023 での C、C++、および Fortran](#)
- [AL2023 での Go](#)
- [AL2023 での Java](#)
- [AL2023 での Perl](#)
- [AL2023 での PHP](#)
- [AL2023 での Python](#)
- [AL2023 での Rust](#)

AL2023 での C、C++、および Fortran

AL2023 には、GNUコンパイラコレクション (GCC) と LLVM (低レベル仮想マシン) Clang のフロントエンドの両方が含まれています。

GCC のメジャーバージョンは AL2023 の存続期間を通じて変わりません。マイナーリリースはバグ修正をもたらし、AL2023 リリースに含まれる可能性があります。その他のバグ、パフォーマンス、セキュリティの修正は、AL2023 GCC に含まれているメジャーバージョンにバックポートされる可能性があります。

AL2023 には C (`gcc`)、C++ (`g++`)、および Fortran (`gfortran`) GCC フロントエンドを備えたバージョン 11 が含まれています。

AL2023 では Ada (`gnat`)、Objective-C、Go Objective-C++ のフロントエンドは有効化されていません。

AL2023 RPM のデフォルトのコンパイラフラグには、最適化フラグと強化フラグが含まれます。GCC を使用して独自のコードを構築するには、最適化フラグと強化フラグを含めることをお勧めします。

Note

`gcc --version` が呼び出されると、`gcc (GCC) 11.3.1 20221121 (Red Hat 11.3.1-4)` のようなバージョン文字列が表示されます。Red Hat は Amazon Linux GCC パッケージのベースとなる [GCC ベンダーブランチ](#) を指します。に示されているバグレポート URL によると `gcc --help`、すべてのバグレポートとサポートリクエストは Amazon Linux に送信する必要があります。

`__GNUCC_RH_RELEASE__` マクロなど、このベンダーブランチにおける長期的な変更の詳細については、「[Fedora パッケージソース](#)」を参照してください。

コアツールチェーンの詳細については、[を参照してください](#)。 [コアツールチェーンパッケージ](#)
[glibc](#)、[gcc](#)、[binutils](#)

AL2023 と他の Linux ディストリビューションとの関係について詳しくは、[を参照してください](#)。
[Fedora との関係](#)

AL2023 と AL2 のコンパイラの 3 連符の違いについて詳しくは、[を参照してください](#)。
[コンパイラ](#)
[トリプレット](#)

AL2023 での Go

Amazon Linux [Go](#) で記述された独自のコードを作成したい場合や、AL2023 に付属しているツールチェーンを使用したい場合があります。AL2 と同様に、AL2023 Go はオペレーティングシステムの存続期間を通じてツールチェーンを更新します。これは、私たちが出荷するツールチェーン内の CVE への対応である場合もあれば、四半期ごとのリリースの一部である場合もあります。

Go は比較的動きの速い言語です。GoGo 既存のアプリケーションが新しいバージョンのツールチェーンに適応しなければならない状況があるかもしれません。詳細については Go、「[Go1 Go とプログラムの未来](#)」を参照してください。

AL2023 Go はその存続期間中にツールチェーンの新しいバージョンを組み込む予定ですが、これは上流リリースと歩調を合わせるものではありません。Go そのため、AL2023 Go で提供されているツールチェーンは、Go 言語や標準ライブラリの最先端の機能を使用してコードをビルドする場合には適していない可能性があります。Go

AL2023 の有効期間中、以前のパッケージバージョンはリポジトリから削除されません。Go 以前のツールチェーンが必要な場合は、Go 新しいツールチェーンのバグ修正やセキュリティ修正を断念

し、どの RPM でも利用できるのと同じメカニズムを使用してリポジトリから以前のバージョンをインストールすることができます。

AL2023 Go で独自のコードを作成したい場合は、AL2023 Go に含まれているツールチェーンを使用できます。ただし、このツールチェーンは AL2023 の存続期間中ずっと先に進む可能性があるためです。

AL2023 で記述された Lambda 関数 Go

Go ネイティブコードにコンパイルすると、Lambda Go はカスタムランタイムとして扱われます。provided.al2023ランタイムを使用して AL2023 Go の関数を Lambda にデプロイできます。

詳細については、『AWS Lambda 開発者ガイド』の「[による Lambda 関数の構築](#)」を参照してください。Go

AL2023 での Java

AL2023 には、ベースのワークロードをサポートするために複数のバージョンの [Amazon Corretto](#) が用意されています。JavaAL2023 Java に含まれるすべてのベースパッケージはで構築されています。Amazon Corretto 17 17

Corretto はオープン Java 開発キット (OpenJDK) のビルドで、からの長期サポートを受けています。AmazonCorretto は Java SE 標準を満たし、Linux、およびで使用できることを確認するために、Windows Java テクニカル・コンパティビリティ・キット (TCK) を使用して認定を受けています。macOS

Corretto 1.8.0、Corretto 11、Corretto 17 のそれぞれに [Amazon Corretto](#) パッケージが用意されています。

AL2023 の各 Corretto バージョンは、Corretto バージョンと同じ期間、または AL2023 のサポート終了日のいずれか早い方までサポートされます。詳細については、[Amazon Linux パッケージサポートステートメントと Amazon Corretto のよくある質問を参照してください](#)。

AL2023 での Perl

AL2023はプログラミング言語のバージョン5.32を提供します。 [Perl](#)

過去数十年にわたり Perl 5 回のリリースの一部として高度な言語互換性を提供してきましたが、Amazon Linux は AL2023 リリース時に Perl 5.32 から移行する予定はありません。PerlAmazon

Linux は、[パッケージサポートステートメントに従って](#) AL2023 の存続期間中、Perl引き続きセキュリティパッチを適用します。

AL2023 の Perl モジュール

AL2023 では、Perlさまざまなモジュールが RPM としてパッケージ化されています。RPM Perl として利用できるモジュールは多数ありますが、Amazon Linux Perl ではすべてのモジュールをパッケージ化することを目指していません。RPM としてパッケージ化されたモジュールは、他のオペレーティングシステムの RPM パッケージに依存している可能性があるため、Amazon Linux はそれらのセキュリティパッチを単なる機能更新よりも優先します。

AL2023 には、CPANPerl開発者がモジュール用の慣用的なパッケージマネージャーを使用できるようにするための機能も含まれています。Perl

AL2023 での PHP

AL2023は現在、[PHP](#)2つのバージョンのプログラミング言語を提供しており、それぞれがアップストリームと同じ期間サポートされています。PHP詳細については、「[Package サポートステートメント](#)」を参照してください。

AL2023 では PHP 8.2 の新機能を使用しながら、8.1 を必要とするアプリケーションもサポートできます。PHP

古い PHP バージョンからの移行

PHPアップストリームのコミュニティは、8.1 から 8.2 [への移行に関する包括的な移行ドキュメントをまとめました](#)。PHP [PHP 8.0 から 8.1 への移行](#)に関するドキュメントもあります。

AL2 には PHP 8.0、8.1、8.2 が含まれており、AL2023 amazon-linux-extras へのアップグレードが容易になっています。

PHP 7.x バージョンからの移行

Note

[PHPこのプロジェクトでは、サポートされているバージョンのリストとスケジュール、およびサポートされていないブランチのリストを管理しています。](#)

AL2023 がリリースされたとき、の 7.x と 5.x [PHP](#)のバージョンはすべてコミュニティによってサポートされておらず、AL2023 のオプションにも含まれていませんでした。

PHPアップストリームのコミュニティは、7.4 から 8.0 [への移行に関する包括的な移行ドキュメントをまとめました](#)。PHP PHP8.1 と PHP 8.2 への移行に関する前のセクションで参照したドキュメントと組み合わせれば、ベースとなるアプリケーションを最新のアプリケーションに移行できます。PHP

Note

AL2 には PHP 7.1、7.2、7.3、7.4 インチが含まれています。amazon-linux-extrasこれらの追加機能はすべて、今後のセキュリティアップデートの提供が保証されているわけではなく end-of-life、保証されているわけでもありませんのでご注意ください。

AL2023 の PHP モジュール

AL2023 には Core PHP に含まれる多くのモジュールが含まれています。AL2023 は、[PHPすべてのパッケージを拡張コミュニティライブラリ \(PECL\) に含めることを目指しているわけでは](#)ありません。

AL2023 での Python

PythonAL2023は2.7を削除し、必要なコンポーネントはすべて3で動作するように書かれています。Python

AL2023では、AL2023に同梱されているPythonコードと同様に、`Python/usr/bin/python3`顧客コードとの互換性を維持するために3つを用意しています。Python AL2023の有効期間中は3.9のままです。

`/usr/bin/python3`参照しているPythonのバージョンはシステムPythonと見なされ、AL2023の場合は3.9ですPython。

Python3.11などの新しいバージョンはAL2023でパッケージとして提供され、アップストリームバージョンの存続期間中サポートされます。PythonPython 3.11がどのくらいの期間サポートされているかについては、[Python 3.11](#)を参照してください。

AL2023 Python では複数のバージョンを同時にインストールできます。`/usr/bin/python3`Python常に3.9になりますが、Pythonの各バージョンには名前空間があり、バージョン番号で確認できます。例えば、python3.11がインストールされている場合、`/usr/bin/python3.11`は `/usr/bin/python3.9` および `/usr/bin/python3.9` への `/usr/bin/python3` シンボリックリンクとともに存在します。

Note

`/usr/bin/python3`シンボリックリンクの内容を変更しないでください。変更すると AL2023 のコア機能が損なわれる可能性があります。

AL2023 の Python モジュール

AL2023 では、Pythonさまざまなモジュールが RPM としてパッケージ化されています。通常、Python モジュールの RPM は Python のシステムバージョンのみをターゲットに構築されます。

AL2023 での Rust

Amazon Linux [Rust](#)で作成したコードをビルドしたり、AL2023 に付属しているツールチェーンを使用したりしたい場合があります。

AL2 と同様に、AL2023 Rust はオペレーティングシステムの存続期間を通じてツールチェーンを更新します。これは、私たちが出荷するツールチェーン内の CVE への対応である場合もあれば、四半期ごとのリリースの一部である場合もあります。

[Rust](#) は比較的動きの速い言語であり、約 6 週間の間隔で新しいリリースが行われます。これらのリリースで、新しい言語や標準ライブラリ機能が追加される可能性があります。AL2023 Rust はその存続期間中にツールチェーンの新しいバージョンを組み込む予定ですが、これは上流リリースと歩調を合わせるものではありません。Rustそのため、AL2023 Rust で提供されているツールチェーンは、Rustその言語の最先端機能を使用してコードをビルドする場合には適していない可能性があります。Rust

AL2023 の存続期間中、古いパッケージバージョンはリポジトリから削除されません。Rust古いツールチェーンが必要な場合は、Rust新しいツールチェーンのバグ修正やセキュリティ修正を断念し、どの RPM でも利用できるのと同じメカニズムを使用してリポジトリから古いバージョンをインストールすることができます。

AL2023 Rust で独自のコードを作成したい場合は、AL2023 Rust に含まれているツールチェーンを使用できます。ただし、このツールチェーンは AL2023 の存続期間中ずっと先に進む可能性があるためです。

AL2023 で記述されたLambda 関数 Rust

Rustネイティブコードにコンパイルされるため、Lambda Rust はカスタムランタイムとして扱います。provided.al2023ランタイムを使用して AL2023 Rust の関数を Lambda にデプロイできます。

詳細については、『AWS Lambda 開発者ガイド』の「[による Lambda 関数の構築](#)」を参照してください。Rust

Amazon Linux 2 のセキュリティとコンプライアンス

Important

脆弱性を報告する場合、または AWS クラウドサービスやオープンソースプロジェクトに関するセキュリティ上の懸念がある場合は、[「脆弱性レポート」ページ](#)を使用して AWS セキュリティにお問い合わせください。

のクラウドセキュリティが最優先事項 AWS です。お客様は AWS、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャからメリットを得られます。

セキュリティは、AWS とユーザーの間で共有される責任です。[責任共有モデル](#)ではこれを、クラウドのセキュリティ、およびクラウド内でのセキュリティと説明しています:

- クラウドのセキュリティ – クラウドで AWS サービスを実行するインフラストラクチャを保護する責任 AWS は AWS にあります。AWS また、では、安全に使用できるサービスも提供しています。コンプライアンス [AWS プログラム](#) コンプライアンスプログラムの一環として、サードパーティーの監査者は定期的にセキュリティの有効性をテストおよび検証。AL2 に適用されるコンプライアンスプログラムの詳細については、「[コンプライアンスプログラム AWS による対象範囲内のサービスコンプライアンスプログラム](#)」を参照してください。
- クラウド内のセキュリティ – お客様の責任は使用する AWS のサービスによって決まります。また、お客様は、お客様のデータの機密性、企業の要件、および適用可能な法律および規制などの他の要因についても責任を担います。

トピック

- [AL2023 の Amazon Linux セキュリティアドバイザリ](#)
- [AL2023 の SELinux モードの設定](#)
- [AL2023 で FIPS モードを有効にする](#)
- [AL2023 カーネルハードニング](#)
- [AL2023 での UEFI セキュアブート](#)

AL2023 の Amazon Linux セキュリティアドバイザリ

Amazon Linux の安全性を保つため、懸命に取り組んでいますが、修正が必要なセキュリティ上の問題が発生することがあります。修正が利用可能になると、アドバイザリが発行されます。アドバイザリを公開する主な場所は、Amazon Linux セキュリティセンター (ALAS) です。詳細については「[Amazon Linux Security Center](#)」を参照してください。

Important

AWS クラウドサービスまたはオープンソースプロジェクトに関する脆弱性を報告する場合、またはセキュリティ上の懸念がある場合は、[脆弱性レポートページ](#)を使用して AWS セキュリティにお問い合わせください。

AL2023 に影響する問題および関連する更新に関する情報は、Amazon Linux チームによって複数の場所で公開されます。セキュリティツールでは、これらの主要な情報源から情報を取得し、その結果をユーザーに提示するのが一般的です。そのため、Amazon Linux が公開するプライマリソースを直接操作せずに、[Amazon Inspector](#) などの任意のツールによって提供されるインターフェイスとやり取りする可能性があります。

Amazon Linux セキュリティセンターの発表

Amazon Linux の発表は、アドバイザリに当てはまらないアイテムに対して提供されます。このセクションには、ALAS 自体に関する通知と、アドバイザリに収まらない情報が含まれています。詳細については、「[Amazon Linux セキュリティセンター \(ALAS\) のお知らせ](#)」を参照してください。

例えば、[2021 年 1 月 001 日 - Apache Log4j の Amazon Linux Hotpatch Announcement](#) は、アドバイザリではなく発表に含まれています。この発表では、Amazon Linux の一部ではないソフトウェアのセキュリティ問題を軽減するためのパッケージが Amazon Linux に追加されました。

[Amazon Linux Security Center CVE Explorer](#) も ALAS の発表で発表されました。詳細については、[CVEs](#)」を参照してください。

Amazon Linux セキュリティセンターに関するよくある質問

ALAS に関するよくある質問と Amazon Linux が CVEs、[「Amazon Linux セキュリティセンター \(ALAS\) に関するよくある質問 \(FAQs\)」](#)を参照してください。

AL2023 の SELinux モードの設定

デフォルトでは、セキュリティ強化 Linux (SELinux) は enabled AL2023 のモードに設定されています。permissive 許可モードでは、アクセス拒否は記録されますが、強制ではありません。SELinux は、カーネルの主要なサブシステムに強力で柔軟性のある強制アクセス制御 (MAC) アーキテクチャを提供するためのカーネル機能とユーティリティの集まりです。

SELinux は、機密性と完全性の要件に基づいて情報を分離するように強化されたメカニズムを備えています。このように情報を分離することで、アプリケーションのセキュリティメカニズムの改ざんやバイパスの脅威が軽減されます。また、悪意のあるアプリケーションや欠陥のあるアプリケーションによって引き起こされる可能性のあるダメージも回避できます。

SELinux には、日々のセキュリティの目標を満たすように設計されたサンプルセキュリティポリシー設定ファイルのセットが含まれています。

SELinux の特徴と機能の詳細については、「[SELinux ノートブック](#)」および「[ポリシー言語](#)」を参照してください。

トピック

- [AL2023 のデフォルト SELinux ステータスとモード](#)
- [enforcing モードへの変更](#)
- [AL2023 用の SELinux を無効にするオプション](#)

AL2023 のデフォルト SELinux ステータスとモード

AL2023 では、SELinux はデフォルトでモードに設定されています。enabled permissive permissive モードでは、アクセス拒否は記録されますが、強制ではありません。

getenforce または **sestatus** コマンドは、現在の SELinux のステータス、ポリシー、モードを知ることができます。

デフォルトステータスが enabled およびに permissive 設定されていると、**getenforce** コマンドは permissive を返します。

以下の例に示すように、**sestatus** このコマンドは SELinux のステータスと現在の SELinux ポリシーを返します。

```
$ 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
```

SELinux `permissive` をモードで実行すると、ユーザーがファイルに誤ってラベルを付けることがあります。SELinux を `disabled` ステータスで実行すると、ファイルにはラベルが付けられません。enforcing モードに変更すると、正しくないファイルやラベル付けされていないファイルの両方が問題を引き起こす可能性があります。

SELinux は、この問題を回避するために自動的にファイルにラベルを付け直します。SELinux は、ステータスを `enabled` に変更したときの自動再ラベル付けにより、ラベル付けの問題を防ぎます。

enforcing モードへの変更

SELinux `enforcing` モードで実行すると、SELinux `enforcing` ユーティリティは設定されたポリシーになります。SELinux ポリシーのルールに基づいてアクセスを許可または拒否することで、特定のアプリケーションの機能を管理します。

SELinux 現在のモードを確認するには、コマンドを実行します。 `getenforce`

```
getenforce
Permissive
```

設定ファイルを編集して **enforcing** モードを有効にする

モードをに変更するには `enforcing`、以下の手順に従います。

1. `/etc/selinux/config` ファイルを編集して `enforcing` モードに変更します。SELINUX 設定は次の例のようになるはずですが。

```
SELINUX=enforcing
```

2. システムを再起動して `enforcing` モードへの変更を完了します。

```
$ sudo reboot
```

次の起動時に、SELinuxシステム内のすべてのファイルとディレクトリにラベルを付け直します。SELinuxまた、SELinuxSELinuxdisabledが作成されたときに作成されたファイルとディレクトリーのコンテキストも追加されます。

enforcingモードに変更すると、SELinuxSELinuxポリシールールが不正確または欠落しているために一部のアクションが拒否されることがあります。SELinux拒否されたアクションは、以下のコマンドで確認できます。

```
$ sudo ausearch -m AVC,USER_AVC,SELINUX_ERR,USER_SELINUX_ERR -ts recent
```

cloud-init を使用して enforcing モードを有効にする

別の方法として、インスタンスを起動するときに、以下の cloud-config をユーザーデータとして渡して、enforcing モードを有効にします。

```
#cloud-config
selinux:
  mode: enforcing
```

デフォルトでは、この設定によりインスタンスが再起動されます。安定性を高めるため、インスタンスを再起動することをお勧めします。ただし、必要に応じて、いかなる cloud-config を指定して再起動をスキップすることができます。

```
#cloud-config
selinux:
  mode: enforcing
  selinux_no_reboot: 1
```

AL2023 用の SELinux を無効にするオプション

無効にするとSELinux、SELinuxポリシーが読み込まれたり適用されたりせず、Access Vector Cache (AVC) メッセージは記録されません。実行の利点はすべて失われます。SELinux

無効にする代わりにSELinux、permissiveモードを使用することをおすすめします。permissiveモードで実行する方が、SELinux完全に無効にするよりも少しだけコストがかかります。permissiveモードからモードへの移行は、enforcingenforcing無効にした後にモードに戻るよりも構成の調整がはるかに少なく済みます。SELinuxファイルにラベルを付けることができ、アクティブなポリシーで拒否された可能性のあるアクションをシステムが追跡して記録できます。

permissiveモードに変更してください。SELinux

SELinux permissiveモードで実行すると、SELinuxポリシーは適用されません。permissiveモードでは、AVC SELinux メッセージは記録されますが、操作は拒否されません。これらの AVC メッセージは、トラブルシューティング、デバッグ、SELinuxポリシーの改善に使用できます。

SELinuxパーミッシブモードに変更するには、次の手順に従います。

1. `/etc/selinux/config` ファイルを編集して permissive モードに変更します。SELINUX値は次の例のようになるはずですが。

```
SELINUX=permissive
```

2. システムを再起動して permissive モードへの変更を完了します。

```
sudo reboot
```

SELinux の無効化

無効にするとSELinux、SELinuxポリシーが読み込まれたり適用されたりせず、AVC メッセージは記録されません。実行の利点はすべて失われます。SELinux

無効にするにはSELinux、以下の手順に従います。

1. grubbyパッケージがインストールされていることを確認します。

```
rpm -q grubby  
grubby-version
```

2. ブートローダーをカーネルコマンドラインに `selinux=0` を追加するように設定します。

```
sudo grubby --update-kernel ALL --args selinux=0
```

3. システムを再起動します。

```
sudo reboot
```

4. `getenforce` SELinuxコマンドを実行してインストールされていることを確認し、`Disabled`。

```
$ getenforce
Disabled
```

詳細についてはSELinux、[「SELinuxSELinuxノートブックと設定」](#)を参照してください。

AL2023 で FIPS モードを有効にする

このセクションでは、AL2023 で連邦情報処理標準 (FIPS) を有効にする手順について説明します。FIPS の詳細については、以下を参照してください。

- [連邦情報処理標準 \(FIPS\)](#)
- [コンプライアンスのよくある質問: 連邦情報処理標準](#)

Note

このセクションでは、AL2023 で FIPS モードを有効にする方法について説明します。AL2023 暗号モジュールの認証状況についての説明はありません。

前提条件

- インターネットにアクセスして必要なパッケージをダウンロードできる既存の AL2023 (AL2023.2 以上) Amazon EC2 インスタンス。AL2023 Amazon EC2 インスタンスの起動の詳細については、[「Amazon EC2 コンソールを使用した AL2023 の起動」](#)を参照してください。
- SSH または AWS Systems Managerを使用して Amazon EC2 インスタンスに接続する必要があります。詳細については、[「AL2023 インスタンスへの接続」](#)を参照してください。

Important

ED25519 SSH ユーザーキーは FIPS モードではサポートされていません。ED25519 SSH key pair を使用して Amazon EC2 インスタンスを起動した場合、別のアルゴリズム (RSA など) を使用して新しいキーを生成する必要があります。そうでない場合は、FIPS モードを有効にした後にインスタンスにアクセスできなくなる可能性があります。詳細については、[「Amazon EC2 ユーザーガイド」の「キーペアの作成」](#)を参照してください。Amazon EC2

FIPS モードの有効化

1. SSH または AWS Systems Manager を使用して AL2023 インスタンスに接続します。
2. システムが最新であることを確認します。詳細については、「[AL2023 でパッケージとオペレーティングシステムの更新を管理する](#)」を参照してください。
3. `crypto-policies` ユーティリティがインストールされ、であることを確認します `up-to-date`。

```
sudo dnf -y install crypto-policies crypto-policies-scripts
```

4. 以下のコマンドを実行して、FIPS モードを有効にします。

```
sudo fips-mode-setup --enable
```

5. 以下のコマンドを実行して、インスタンスを再起動します。

```
sudo reboot
```

6. FIPS モードが有効であることを確認するには、インスタンスに再接続し、以下のコマンドを実行します。

```
sudo fips-mode-setup --check
```

以下の出力例は、FIPS モードが有効であることを示しています。

```
FIPS mode is enabled.  
Initramfs fips module is enabled.  
The current crypto policy (FIPS) is based on the FIPS policy.
```

AL2023 カーネルハードニング

AL2023 の 6.1 Linux カーネルは、いくつかの強化オプションと機能を使用して構成および構築されています。

カーネル強化オプション (アーキテクチャに依存していません)

CONFIG オプション	AL2023/6.1/aarch64	AL2023/6.1/x86_64
<u>CONFIG_ACPI_CUSTOM_METHOD</u>	n	n
<u>CONFIG_BINFORM_MISC</u>	m	m
<u>CONFIG_BUG</u>	y	y
<u>CONFIG_BUG_ON_DATA_CORRUPTION</u>	y	y
<u>CONFIG_CFI_CLANG</u>	該当なし	該当なし
<u>CONFIG_CFI_PERMISSIVE</u>	該当なし	該当なし
<u>CONFIG_COMPAT</u>	y	y
<u>CONFIG_COMPAT_BRK</u>	n	n
<u>CONFIG_COMPAT_VDSO</u>	該当なし	n
<u>CONFIG_DEBUG_CREDENTIALS</u>	n	n
<u>CONFIG_DEBUG_LIST</u>	y	y
<u>CONFIG_DEBUG_NOTIFIERS</u>	n	n
<u>CONFIG_DEBUG_SG</u>	n	n
<u>CONFIG_DEBUG_VIRTUAL</u>	n	n
<u>CONFIG_DEBUG_WX</u>	n	n
<u>CONFIG_DEFAULT_MMAP_MIN_ADDR</u>	65536	65536
<u>CONFIG_DEVMEM</u>	該当なし	該当なし

CONFIG オプション	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_DEVMEM	n	n
CONFIG_EFI_DISABLE_PCI_DMA	n	n
CONFIG_FORTIFY_SOURCE	y	y
CONFIG_HARDENED_USERCOPY	y	y
CONFIG_HARDENED_USERCOPY_FALLBACK	該当なし	該当なし
CONFIG_HARDENED_USERCOPY_PAGESPAN	該当なし	該当なし
CONFIG_HIBERNATION	y	y
CONFIG_HW_RANDOM_TPM	該当なし	該当なし
CONFIG_INET_DIAG	m	m
CONFIG_INIT_ON_ALLOC_DEFAULT_ON	n	n
CONFIG_INIT_ON_FREE_DEFAULT_ON	n	n
CONFIG_INIT_STACK_ALL_ZERO	該当なし	該当なし
CONFIG_IOMMU_DEFAULT_DMA_STRICT	n	n
CONFIG_IOMMU_SUPPORT	y	y

CONFIG オプション	AL2023/6.1/aarch64	AL2023/6.1/x86_64
<u>CONFIG_IO_STRICT_D EVMEM</u>	該当なし	該当なし
<u>CONFIG_KEXEC</u>	y	y
<u>CONFIG_KFENCE</u>	n	n
<u>CONFIG_LDISC_AUTOL OAD</u>	n	n
<u>CONFIG_LEGACY_PTYS</u>	n	n
<u>CONFIG_LOCK_DOWN_K ERNEL_FORCE_CONFID ENTIALITY</u>	n	n
<u>CONFIG_MODULES</u>	y	y
<u>CONFIG_MODULE_SIG</u>	y	y
<u>CONFIG_MODULE_SIG_ ALL</u>	y	y
<u>CONFIG_MODULE_SIG_ FORCE</u>	n	n
<u>CONFIG_MODULE_SIG_ HASH</u>	sha512	sha512
<u>CONFIG_MODULE_SIG_ KEY</u>	certs/signing_key. pem	certs/signing_key. pem
<u>CONFIG_MODULE_SIG_ SHA512</u>	y	y
<u>CONFIG_PAGE_POISON ING</u>	n	n

CONFIG オプション	AL2023/6.1/aarch64	AL2023/6.1/x86_64
<u>CONFIG_PAGE_POISONING_NO_SANITY</u>	該当なし	該当なし
<u>CONFIG_PAGE_POISONING_ZERO</u>	該当なし	該当なし
<u>CONFIG_PANIC_ON_OOPS</u>	y	y
<u>CONFIG_PANIC_TIMEOUT</u>	0	0
<u>CONFIG_PROC_KCORE</u>	y	y
<u>CONFIG_RANDOMIZE_KSTACK_OFFSET_DEFAULT</u>	n	n
<u>CONFIG_RANDOM_TRUST_BOOTLOADER</u>	y	y
<u>CONFIG_RANDOM_TRUST_CPU</u>	y	y
<u>CONFIG_REFCOUNT_FULL</u>	該当なし	該当なし
<u>CONFIG_SCHED_CORE</u>	該当なし	y
<u>CONFIG_SCHED_STACK_END_CHECK</u>	y	y
<u>CONFIG_SECCOMP</u>	y	y
<u>CONFIG_SECCOMP_FILTER</u>	y	y
<u>CONFIG_SECURITY</u>	y	y
<u>CONFIG_SECURITY_DMESG_RESTRICT</u>	y	y

CONFIG オプション	AL2023/6.1/aarch64	AL2023/6.1/x86_64
<u>CONFIG_SECURITY_LANDLOCK</u>	n	n
<u>CONFIG_SECURITY_LOCKDOWN_LSM</u>	y	y
<u>CONFIG_SECURITY_LOCKDOWN_LSM_EARLY</u>	y	y
<u>CONFIG_SECURITY_SELINUX_BOOTPARAM</u>	y	y
<u>CONFIG_SECURITY_SELINUX_DEVELOP</u>	y	y
<u>CONFIG_SECURITY_SELINUX_DISABLE</u>	n	n
<u>CONFIG_SECURITY_WRITABLE_HOOKS</u>	該当なし	該当なし
<u>CONFIG_SECURITY_YAMA</u>	y	y
<u>CONFIG_SHUFFLE_PAGE_ALLOCATOR</u>	y	y
<u>CONFIG_SLAB_FREELIST_HARDENED</u>	y	y
<u>CONFIG_SLAB_FREELIST_RANDOM</u>	y	y
<u>CONFIG_SLUB_DEBUG</u>	y	y
<u>CONFIG_STACKPROTECTOR</u>	y	y

CONFIG オプション	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_STACKPROTECTOR_STRONG	y	y
CONFIG_STATIC_USERMODEHELPER	n	n
CONFIG_STRICT_DEVMEM	n	n
CONFIG_STRICT_KERNEL_RWX	y	y
CONFIG_STRICT_MODULE_RWX	y	y
CONFIG_SYN_COOKIES	y	y
CONFIG_VMAP_STACK	y	y
CONFIG_WERROR	n	n
CONFIG_ZERO_CALL_USED_REGS	n	n

ACPI メソッドを実行時に挿入/置換できるようになりました (CONFIG_ACPI_CUSTOM_METHOD)

Amazon Linux では、root ユーザーが任意のカーネルメモリに書き込むことができるため、このオプションは無効になっています。

このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

その他のバイナリフォーマット (**binfmt_misc**)

このオプションは[カーネルセルフプロテクションプロジェクト \(KSPP\) の推奨設定](#)の1つですが、AL2023 ではこの設定オプションを KSPP が推奨するものに設定していません。AL2023 では、この機能はオプションで、カーネルモジュールとして構築されています。

BUG() サポート

このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

カーネルメモリ構造の有効性をチェックしているときに、カーネルがデータ破損に遭遇した場合の **BUG()**

Linux カーネルの一部は、データ構造の内部整合性をチェックし、データ破損を検出した場合は **BUG()** できます。

このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

COMPAT_BRK

このオプションを無効にすると (Amazon Linux がカーネルを設定する方法)、`randomize_va_space sysctl` 設定はデフォルトで 2 になり、`mmap` ベース、スタック、`VDSO` ページのランダム化に加えてヒープのランダム化も有効になります。

1996 年以前の古い `libc.so.5` バイナリとの互換性を保つため、このオプションはカーネルに組み込まれています。

このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

COMPAT_VDSO

この設定オプションは `x86-64` に関連しますが、`aarch64` とは関係ありません。これを `n` に設定すると、Amazon Linux カーネルは 32 ビットの仮想化動的共有オブジェクト (VDSO) を予測可能なアドレスに表示しなくなります。このオプションを `n` に設定したことで障害が発生することがわかっている最新の `glibc` は 2004 年の `glibc 2.3.3` です。

このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

CONFIG_DEBUG ゲートの強化

`CONFIG_DEBUG` ゲートの Linux カーネル設定オプションは通常、デバッグ用の問題を想定して構築されたカーネルで使用するよう設計されており、パフォーマンスなどは優先されません。`CONFIG_DEBUG_LISTAL2023` は強化オプションを有効にします。

IOMMU を設定する前に EFI スタブで PCI デバイスの DMA を無効にする

このオプションは[カーネルセルフプロテクションプロジェクト \(KSPP\) の推奨設定](#)の1つですが、AL2023 ではこの設定オプションを KSPP が推奨するものに設定していません。

カーネルとユーザースペース間のメモリーコピーの強化

カーネルがユーザースペースにメモリーをコピーしたり、ユーザー空間からメモリーをコピーしたりする必要がある場合、このオプションはいくつかのチェックを有効にすることで、ある種のヒープオーバーフローの問題を防ぐことができます。

CONFIG_HARDENED_USERCOPY_FALLBACK オプションはカーネル 4.16 から 5.15 までにあって、カーネルデベロッパーが不足している許可リストエントリを WARN() 経由で見つけやすくするためのものです。AL2023 には 6.1 カーネルが同梱されているため、このオプションは AL2023 には関係ありません。

CONFIG_HARDENED_USERCOPY_PAGESPAN このオプションは主に開発者向けのデバッグオプションとしてカーネルに存在し、AL2023 の 6.1 カーネルには適用されなくなりました。

このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の 1 つです。

休止のサポート

このオプションは[カーネルセルフプロテクションプロジェクト \(KSPP\) の推奨設定](#)の 1 つですが、AL2023 ではこの設定オプションを KSPP が推奨するものに設定していません。[オンデマンドインスタンスを休止状態にしたり、中断されたスポットインスタンスを休止状態にしたりするには、このオプションを有効にする必要があります。](#)

乱数生成

AL2023 カーネルは、EC2 内で使用できる十分なエントロピーを確保するように設定されています。

CONFIG_INET_DIAG

このオプションは[カーネルセルフプロテクションプロジェクト \(KSPP\) の推奨設定](#)の 1 つですが、AL2023 ではこの設定オプションを KSPP が推奨するものに設定していません。AL2023 では、この機能はオプションで、カーネルモジュールとして構築されています。

割り当て時と割り当て解除時に、カーネルページとスラブアロケータのすべてのメモリーをゼロにする

このオプションは[カーネルセルフプロテクションプロジェクト \(KSPP\) の推奨設定](#)の 1 つですが、AL2023 ではこの設定オプションを KSPP が推奨するものに設定していません。これらのオプションは、この機能をデフォルトで有効にする

とパフォーマンスに影響する可能性があるため、AL2023 では無効になっています。CONFIG_INIT_ON_ALLOC_DEFAULT_ONinit_on_alloc=1 動作はカーネルコマンドラインに追加することで有効にでき、CONFIG_INIT_ON_FREE_DEFAULT_ON動作は init_on_free=1 に追加することで有効にできます。

すべてのスタック変数をゼロ (CONFIG_INIT_STACK_ALL_ZERO) に初期化する

このオプションは[カーネルセルフプロテクションプロジェクト \(KSP\) の推奨設定](#)の 1 つですが、AL2023 ではこの設定オプションを KSP が推奨するものに設定していません。このオプションには GCC 12 以上が必要ですが、AL2023 には GCC 11 が含まれています。

カーネルモジュール署名

AL2023 はカーネルモジュールの署名と署名を検証します。サードパーティモジュールを構築するユーザーの互換性を保つため、モジュールに有効な署名を要求する CONFIG_MODULE_SIG_FORCE オプションは有効化されていません。すべてのカーネルモジュールが署名されていることを確認したいユーザは、これを強制するように [Linux セキュリティモジュール \(LSM\) の ロックダウン](#) を設定できます。

kexec

このオプションは[カーネルセルフプロテクションプロジェクト \(KSP\) の推奨設定](#)の 1 つですが、AL2023 ではこの設定オプションを KSP が推奨するものに設定していません。このオプションは kdump 機能を使用できるように有効になっています。

IOMMU サポート

AL2023 は IOMMU サポートを有効にします。CONFIG_IOMMU_DEFAULT_DMA_STRICT オプションはデフォルトでは有効になっていませんが、この機能はカーネルコマンドラインに iommu.passthrough=0 iommu.strict=1 を追加することで設定できます。

kfence

このオプションは[カーネルセルフプロテクションプロジェクト \(KSP\) の推奨設定](#)の 1 つですが、AL2023 ではこの設定オプションを KSP が推奨するものに設定していません。

従来の pty のサポート

AL2023 は最新のインターフェイス () を使用します。PTY devpts

このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

Linux セキュリティモジュール (LSM) の ロックダウン

AL2023 は lockdown LSM をビルドし、セキュアブートを使用するとカーネルを自動的にロックダウンします。

CONFIG_LOCK_DOWN_KERNEL_FORCE_CONFIDENTIALITY オプションは有効になっていません。このオプションは[カーネルセルフプロテクションプロジェクト \(KSPP\) の推奨設定](#)の1つですが、AL2023 ではこの設定オプションを KSPP が推奨するものに設定していません。セキュアブートを使用しない場合は、必要に応じてロックダウン LSM を有効に設定できます。

ページポイズニング

このオプションは[カーネルセルフプロテクションプロジェクト \(KSPP\) の推奨設定](#)の1つですが、AL2023 ではこの設定オプションを KSPP が推奨するものに設定していません。同様に [割り当て時と割り当て解除時に、カーネルページとスラブアロケータのすべてのメモリをゼロにする](#)、AL2023 カーネルではパフォーマンスに影響する可能性があるため、この機能は無効になっています。

スタックプロテクター

AL2023 カーネルは、オプションで有効にするスタックプロテクター機能を搭載して構築されています。GCC `-fstack-protector-strong`

このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

seccomp BPF API

seccomp 強化機能は、systemd およびコンテナランタイムなどのソフトウェアがユーザースペースアプリケーションを強化するために使用されます。

このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

panic() タイムアウト

AL2023 カーネルはこの値をに設定しています。つまり、パニックになった後にカーネルは再起動しません。このオプションは[カーネルセルフプロテクションプロジェクト \(KSPP\) の推奨設定](#)の1つですが、AL2023 ではこの設定オプションを KSPP が推奨するものに設定していません。これは `sysctl`、`/proc/sys/kernel/panic`、およびカーネルコマンドラインで設定できます。

セキュリティモデル

AL2023 はデフォルトで SELinux をパーミッシブモードで有効にします。詳細については、「[AL2023 の SELinux モードの設定](#)」を参照してください。

[Linux セキュリティモジュール \(LSM\) の ロックダウン](#) および yama モジュールも有効になっています。

/proc/kcore

このオプションは[カーネルセルフプロテクションプロジェクト \(KSP\) の推奨設定](#)の 1 つですが、AL2023 ではこの設定オプションを KSP が推奨するものに設定していません。

システムコールエントリでのカーネルスタックオフセットのランダム化

このオプションは[カーネルセルフプロテクションプロジェクト \(KSP\) の推奨設定](#)の 1 つですが、AL2023 ではこの設定オプションを KSP が推奨するものに設定していません。これはカーネルコマンドラインで randomize_kstack_offset=on を設定することで有効にできます。

参照カウントチェック (CONFIG_REFCOUNT_FULL)

このオプションは[カーネルセルフプロテクションプロジェクト \(KSP\) の推奨設定](#)の 1 つですが、AL2023 ではこの設定オプションを KSP が推奨するものに設定していません。このオプションはパフォーマンスに影響する可能性があるため、現時点では有効になっていません。

スケジューラーによる SMT コアの認識 (CONFIG_SCHED_CORE)

AL2023 カーネルは、ユーザースペースのアプリケーションで使えるように構築されています。CONFIG_SCHED_CORE prctl(PR_SCHED_CORE)このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の 1 つです。

schedule()(CONFIG_SCHED_STACK_END_CHECK) 呼び出しでスタックに問題がないか確認する

AL2023 カーネルは有効化された状態でビルドされます。CONFIG_SCHED_STACK_END_CHECKこのオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の 1 つです。

メモリアロケータの強化

AL2023 カーネルでは、オプションを使用してカーネルメモリアロケータを強化できます。CONFIG_SHUFFLE_PAGE_ALLOCATOR CONFIG_SLAB_FREELIST_HARDENED

CONFIG_SLAB_FREELIST_RANDOMこのオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

SLUB デバッグのサポート

AL2023 カーネルは、CONFIG_SLUB_DEBUGこのオプションによってカーネルコマンドラインで有効にできるアロケータのデバッグ機能をオプションで有効にするため、有効になります。このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

CONFIG_STATIC_USER_MODE_HELPER

このオプションは[カーネルセルフプロテクションプロジェクト \(KSPP\) の推奨設定](#)の1つですが、AL2023 ではこの設定オプションを KSPP が推奨するものに設定していません。これは、現在 Amazon Linux にはない配布からの特別なサポートがCONFIG_STATIC_USERMODEHELPER で必要なためです。

読み取り専用カーネルテキストおよび rodata (**CONFIG_STRICT_KERNEL_RWX** および **CONFIG_STRICT_MODULE_RWX**)

AL2023 カーネルは、カーネルとカーネルモジュールのテキストとメモリを読み取り専用としてマークし、rodataテキスト以外のメモリを実行不可としてマークするように設定されています。このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

TCP syncookie のサポート (**CONFIG_SYN_COOKIES**)

AL2023 カーネルは TCP シンクッキーをサポートするように構築されています。このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

ガードページ (**CONFIG_VMAP_STACK**) 付きの仮想化マップスタック

AL2023 カーネルはで構築されているためCONFIG_VMAP_STACK、ガードページ付きの仮想マッピングされたカーネルスタックが可能になります。このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

コンパイラ警告をエラー (**CONFIG_WERROR**) として構築する

このオプションは[カーネルセルフプロテクションプロジェクト \(KSPP\) の推奨設定](#)の1つですが、AL2023 ではこの設定オプションを KSPP が推奨するものに設定していません。

関数 `exit (CONFIG_ZERO_CALL_USED_REGS)` 時にレジスタをゼロにする

このオプションは[カーネルセルフプロテクションプロジェクト \(KSPP\) の推奨設定](#)の 1 つですが、AL2023 ではこの設定オプションを KSPP が推奨するものに設定していません。

ユーザースペース割り当ての最小アドレス

この強化オプションは、カーネル NULL ポインタのバグの影響を軽減するのに役立ちます。このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の 1 つです。

clang 特定の強化オプション

AL2023 GCC カーネルはではなくでビルドされているためclang、CONFIG_CFI_CLANG強化オプションを有効にすることはできません。これも適用されません。CONFIG_CFI_PERMISSIVEこのオプションは[カーネルセルフプロテクションプロジェクト \(KSPP\) の推奨設定](#)の 1 つですが、AL2023 ではこの設定オプションを KSPP が推奨するものに設定していません。

x86-64 固有のカーネル強化オプション

CONFIG オプション	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_AMD_IOMMU	該当なし	y
CONFIG_AMD_IOMMU_V2	該当なし	y
CONFIG_IA32_EMULATION	該当なし	y
CONFIG_INTEL_IOMMU	該当なし	y
CONFIG_INTEL_IOMMU_DEFAULT_ON	該当なし	n
CONFIG_INTEL_IOMMU_SVM	該当なし	n
CONFIG_LEGACY_VSYS_CALL_NONE	該当なし	n

CONFIG オプション	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_MODIFY_LDT_SYSCALL	該当なし	n
CONFIG_PAGE_TABLE_ISOLATION	該当なし	y
CONFIG_RANDOMIZE_MEMORY	該当なし	y
CONFIG_X86_64	該当なし	y
CONFIG_X86_MSR	該当なし	y
CONFIG_X86_VSYSCALL_EMULATION	該当なし	y
CONFIG_X86_X32	該当なし	該当なし
CONFIG_X86_X32_ABI	該当なし	n

x86-64 サポート

基本的な x86-64 サポートには、物理アドレス拡張 (PAE) と実行なし (NX) ビットのサポートが含まれます。このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の 1 つです。

AMD と Intel IOMMU のサポート

AL2023 カーネルは AMD とインテルをサポートするように構築されています。IOMMUsこのオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の 1 つです。

CONFIG_INTEL_IOMMU_DEFAULT_ON オプションは設定されていませんが、カーネルコマンドラインに intel_iommu=on を渡すことで有効にできます。このオプションは[カーネルセルフプロテクションプロジェクト \(KSPP\) の推奨設定](#)の 1 つですが、AL2023 ではこの設定オプションを KSPP が推奨するものに設定していません。

CONFIG_INTEL_IOMMU_SVMこのオプションは現在 AL2023 では有効になっていません。このオプションは[カーネルセルフプロテクションプロジェクト \(KSPP\) の推奨設定](#)の 1 つですが、AL2023 ではこの設定オプションを KSPP が推奨するものに設定していません。

32 ビットユーザースペースのサポート

Important

32 ビット x86 ユーザースペースのサポートは廃止され、32 ビットユーザースペースバイナリの実行のサポートは Amazon Linux の今後のメジャーバージョンで削除される可能性があります。

Note

AL2023 には 32 ビットパッケージが含まれなくなりましたが、カーネルは 32 ビットユーザー空間の実行を引き続きサポートします。詳細については、「[32 ビット x86 \(i686\) パッケージ](#)」を参照してください。

32 ビットユーザースペースアプリケーションの実行をサポートするため、AL2023 はオプションを有効にせず、`CONFIG_X86_VSYSCALL_EMULATION` オプションを有効にします。`CONFIG_IA32_EMULATION` `CONFIG_COMPAT` `CONFIG_X86_VSYSCALL_EMULATION` このオプションは[カーネルセルフプロテクションプロジェクト \(KSPP\) の推奨設定](#)の 1 つですが、AL2023 ではこの設定オプションを KSPP が推奨するものに設定していません。

64 ビットプロセッサ用の x32 ネイティブ 32 ビット ABI は有効になっていません (`CONFIG_X86_X32` および `CONFIG_X86_X32_ABI`)。このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の 1 つです。

x86 モデル固有レジスタ (MSR) のサポート

`turbostat` をサポートするため `CONFIG_X86_MSR` オプションは有効になっています。このオプションは[カーネルセルフプロテクションプロジェクト \(KSPP\) の推奨設定](#)の 1 つですが、AL2023 ではこの設定オプションを KSPP が推奨するものに設定していません。

`modify_ldt` システムコール

AL2023 では、ユーザープログラムが `syscall` を使用して x86 ローカル記述子テーブル (LDT) を変更することはできません。`modify_ldt` この呼び出しは 16 ビットコードまたはセグメント化されたコードを実行するために必要です。この呼び出しがないと、WINE でプログラムを実行したり、非常に古いスレッドライブラリを実行したりする場合など、`dosexu` などのソフトウェアが動作しなくな

る可能性があります。このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

ユーザーモードでカーネルマッピングを削除する

AL2023 は、カーネルアドレスの大部分がユーザー空間にマップされないようにカーネルを設定します。このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

カーネルメモリセクションをランダム化する

AL2023 は、カーネルメモリセクションの基本仮想アドレスをランダム化するようにカーネルを設定します。このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

aarch64 固有のカーネル強化オプション

CONFIG オプション	AL2023/6.1/aarch64	AL2023/6.1/x86_64
CONFIG_ARM64_BTI	y	該当なし
CONFIG_ARM64_BTI_KERNEL	該当なし	該当なし
CONFIG_ARM64_PTR_AUTH	y	該当なし
CONFIG_ARM64_PTR_AUTH_KERNEL	y	該当なし
CONFIG_ARM64_SW_TTBR0_PAN	y	該当なし
CONFIG_UNMAP_KERNEL_AT_EL0	y	該当なし

ブランチターゲット識別

AL2023 カーネルはブランチターゲット識別 () をサポートします。CONFIG_ARM64_BTIこのオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の1つです。

AL2023 で GCC を使用して構築された CONFIG_ARM64_BTI_KERNEL オプションは有効になっていません。また、[gcc のバグ](#)により、このオプションを使ったカーネル構築のサポートは、[アップストリームのカーネルでは現在無効になっています](#)。このオプションは[カーネルセルフプロテクションプロジェクト \(KSP\) の推奨設定](#)の 1 つですが、AL2023 ではこの設定オプションを KSP が推奨するものに設定していません。

ポインター認証 (CONFIG_ARM64_PTR_AUTH)

AL2023 カーネルは、ポインター認証拡張 (ARMv8.3 拡張機能の一部) をサポートするように構築されており、これを使用するとリターン指向プログラミング (ROP) の手法の軽減に役立ちます。[Graviton](#) でのポインター認証に必要なハードウェアサポートは Graviton 3 で導入されました。

CONFIG_ARM64_PTR_AUTH オプションは有効になっており、ユーザースペースのポインター認証をサポートします。CONFIG_ARM64_PTR_AUTH_KERNEL このオプションも有効になっているため、AL2023 カーネルはリターン・アドレス保護を自分で利用することができます。

このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の 1 つです。

TTBR0_EL1 切り替えを決してしない特権アクセスをエミュレートする

このオプションは、ユーザーアクセスルーチンによって一時的にのみ有効な値に設定される TTBR0_EL1 で、カーネルがユーザー空間のメモリに直接アクセスすることを防ぎます。

このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の 1 つです。

ユーザースペースでの実行時にカーネルをアンマップする

AL2023 カーネルは userspace () での実行時にカーネルをアンマップするように設定されています。CONFIG_UNMAP_KERNEL_AT_EL0 このオプションは、[カーネルセルフプロテクションプロジェクト推奨設定](#)の 1 つです。

AL2023 での UEFI セキュアブート

AL2023 は、リリース 2023.1 以降の UEFI Secure Boot をサポートしています。AL2023 は UEFI と UEFI セキュアブートの両方をサポートする Amazon EC2 インスタンスで使用する必要があります。詳細については、「Amazon EC2 [ユーザーガイド](#)」の「[インスタンスの起動](#)」を参照してください。Amazon EC2

UEFI Secure Boot が有効になっている AL2023 インスタンスは、Linux カーネルおよびによって署名されたモジュールを含むカーネルレベルコードのみを受け入れる Amazon ため、インスタンスがによって署名されたカーネルレベルコードのみを実行できるようにします AWS。

Amazon EC2 インスタンスと UEFI Secure Boot の詳細については、Amazon EC2 ユーザーガイド」の「[UEFI Secure Boot](#)」を参照してください。

前提条件

- AL2023 リリース 2023.1 以降の AMI を使用している必要があります。
- インスタンスタイプは UEFI Secure Boot をサポートする必要があります。詳細については、「Amazon EC2 [ユーザーガイド](#)」の「[インスタンスの起動](#)」を参照してください。Amazon EC2

AL2023 で UEFI Secure Boot を有効にする

標準 AL2023 AMI には、ブートローダーおよびキーで署名されたカーネルが組み込まれています。UEFI セキュアブートを有効にするには、既存のインスタンスを登録するか、スナップショットからイメージを登録して UEFI セキュアブートがあらかじめ有効になっている AMI を作成します。デフォルトでは、UEFI セキュアブートは標準の AL2023 AMI で有効になっていません。

インスタンスタイプが UEFI をサポートしている場合、AL2023 AMI のブートモードは、これらの AMI で起動されたインスタンスが UEFI ファームウェアを使用するように `uefi-preferred` に設定されています。インスタンスタイプが UEFI をサポートしていない場合、インスタンスはレガシー BIOS ファームウェアで起動されます。インスタンスをレガシー BIOS モードで起動すると UEFI セキュアブートは実行されません。

Amazon EC2 インスタンスの AMI ブートモードの詳細については、「Amazon EC2 ユーザーガイド」の「[ブートモード Amazon EC2](#)」を参照してください。

トピック

- [既存のインスタンスの登録](#)
- [スナップショットからイメージを登録する](#)
- [失効の更新](#)
- [AL2023 での UEFI Secure Boot の仕組み](#)
- [独自のキーを登録する](#)

既存のインスタンスの登録

既存のインスタンスを登録するには、特定の UEFI ファームウェア変数に、ファームウェアがブートローダーを検証し、ブートローダーが次回の起動時にカーネルを検証できるようにする一連のキーを設定します。

1. Amazon Linux には、登録プロセスを簡素化するツールが用意されています。以下のコマンドを実行して、必要なキーと証明書のセットをインスタンスにプロビジョニングします。

```
sudo amazon-linux-sb enroll
```

2. 以下のコマンドを実行して、インスタンスを再起動します。インスタンスが再起動すると、UEFI セキュアブートが有効になります。

```
sudo reboot
```

Note

Amazon Linux AMI は現在 Nitro Trusted Platform Module (NitroTPM) をサポートしていません。UEFI セキュアブートに加えて NitroTPM が必要な場合は、以下のセクションの情報を使用します。

スナップショットからイメージを登録する

Amazon EC2 `register-image` API を使用して Amazon EBS ルートボリュームのスナップショットから AMI を登録する場合、UEFI 変数ストアの状態を含むバイナリブロッブを使用して AMI をプロビジョニングできます。AL2023 `UefiData` を提供することで UEFI セキュアブートが有効になり、前のセクションの手順を実行する必要がなくなります。

バイナリ BLOB の作成と使用の詳細については、Amazon EC2 [ユーザーガイド](#) の「[オプション B: 事前に入力された変数ストアを含むバイナリ BLOB を作成する](#)」を参照してください。

AL2023 には、Amazon EC2 インスタンスで直接使用できる構築済みのバイナリ BLOB が用意されています。バイナリ BLOB は実行中のインスタンスの `/usr/share/amazon-linux-sb-keys/uefi.vars` にあります。この BLOB は、リリース 2023.1 以降の AL2023 AMI にデフォルトでインストールされている `amazon-linux-sb-keys` RPM パッケージによって提供されます。

Note

キーおよび失効のバージョンが最新であることを確認するには、AMI の作成に使用したのと同じリリースの AL2023 の BLOB を使用します。

イメージの登録時には、[RegisterImage](#) API の `BootMode` パラメータを `uefi` に設定することをお勧めします。これにより、`TpmSupport` パラメータを `v2.0` に設定することで NitroTPM を有効にできます。また、UEFI をサポートしないインスタンスタイプに切り替えたとときに UEFI セキュアブートが有効になり、誤って無効にされないように `BootMode` を `uefi` に設定できます。

NitroTPM の詳細については、「Amazon EC2 ユーザーガイド」の[NitroTPM](#)」を参照してください。
Amazon EC2

失効の更新

Amazon Linux は、更新されたキーで署名された新しいバージョンのブートローダー `grub2` または Linux カーネルを配布する必要がある場合があります。その場合は、以前のバージョンのブートローダーの悪用可能なバグが UEFI セキュアブート検証プロセスをバイパスしてしまうことを防ぐために、古いキーを取り消す必要がある場合があります。

`grub2` または `kernel` パッケージにパッケージを更新すると、実行中のインスタンスの UEFI 変数ストアへの失効リストが常に自動的に更新されます。つまり、UEFI セキュアブートを有効にすると、パッケージのセキュリティ更新をインストールした後は、古いバージョンのパッケージを実行できなくなります。

AL2023 での UEFI Secure Boot の仕組み

他の Linux 配布とは異なり、Amazon Linux には第 1 ステージのブートローダーとして機能するシムと呼ばれる追加コンポーネントはありません。シムは通常、Microsoft のキーで署名されています。例えば、シム付きの Linux ディストリビューションでは、シムが `grub2` ブートローダーをロードし、シム独自のコードを使用して Linux カーネルを検証します。さらに、シムは UEFI 変数ストアにある Machine Owner Key (MOK) データベースに独自のキーセットと失効情報を保持し、`mokutil` ツールで制御します。

Amazon Linux にはシムはありません。AMI の所有者が UEFI 変数を制御するため、この中間ステップは不要で、起動時間と起動時間に悪影響を及ぼします。また、望ましくないバイナリが実行される可能性を減らすため、デフォルトではベンダーキーへの信頼を含めないようにしました。通常通り、必要に応じてバイナリを含めることができます。

Amazon Linux では、UEFI が `grub2` ブートローダーを直接ロードして検証します。`grub2` ブートローダーは、ロード後に UEFI を使用して Linux カーネルを検証するように変更されました。そのため、Linux カーネルは、通常の UEFI db 変数 (認証キーデータベース) に格納されているのと同じ証明書を使用して検証され、ブートローダーや他の UEFI バイナリと同じ `dbx` 変数 (失効データベース) に対してテストされます。`db` データベースおよび `dbx` データベースへのアクセスを制御する独

自の PK キーおよび KEK キーを提供しているため、シムなどの仲介なしで、必要に応じて署名付きの更新と失効を配布できます。

UEFI Secure Boot の詳細については、Amazon EC2 [ユーザーガイド](#) の「[UEFI Secure Boot の仕組み](#)」を参照してください。

独自のキーを登録する

前のセクションで説明したように、Amazon Linux では Amazon EC2 で shim が UEFI セキュアブートを実行する必要はありません。他の Linux 配布ドキュメントを読んでいると、AL2023 にはない、mokutil を使用した Machine Owner Key (MOK) データベースの管理に関するドキュメントを見かける可能性があります。shim および MOK 環境は、Amazon EC2 が UEFI セキュアブートを実装する方法には当てはまらない UEFI ファームウェアへのキー登録のいくつかの制限を回避します。Amazon EC2 には、UEFI 変数ストア内のキーを簡単に直接操作するメカニズムがあります。

独自のキーを登録したい場合は、既存のインスタンス内の変数ストアを操作するか ([「インスタンス内から変数ストアにキーを追加する」](#)を参照してください)、[事前入力されたバイナリ BLOB を構築](#)できます ([「事前入力された変数ストアを含むバイナリ BLOB の作成」](#)を参照してください)。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。