aws

ユーザーガイド

Amazon Linux 2

Amazon Linux 2: ユーザーガイド

Table of Contents

Amazon Linux 2 とは

Amazon Linux 2 (AL2) は、Amazon Web Services () の Linux オペレーティングシステムです AWS。AL2 は、Amazon EC2 で実行されているアプリケーションに、安定した安全で高性能な環境を提供するように設計されています。また、起動設定ツールや多くの一般的なAWSライブラリやツールなどAWS、 との効率的な統合を可能にするパッケージも含まれています。 は、AL2 を実行しているすべてのインスタンスに継続的なセキュリティおよびメンテナンスの更新AWSを提供します。CentOS で開発された多くのアプリケーション、および同様のディストリビューションは、AL2 で実行されます。AL2 は追加料金なしで提供されます。

> **Note**
>
> AL2 は Amazon Linux の最新バージョンではなくなりました。AL2023 は AL2 の後継です。詳細については、AL2023 ユーザーガイドの「AL2 と AL2023 の比較」および「AL2023 でのパッケージの変更のリスト」を参照してください。 AL2023

> **Note**
>
> AL2 はアップストリームの Firefox 延長サポートリリース (ESR) バージョンに厳密に従い、利用可能になるとすぐに次の ESR に更新されます。詳細については、Firefox ESR リリースカレンダーと Firefox リリースノートを参照してください。

Amazon Linux の入手可能性

AWSは、AL2023, AL2、Amazon Linux 1 (AL1、以前の Amazon Linux AMI) を提供します。別の Linux ディストリビューションから Amazon Linux に移行する場合、AL2023 に移行することをお勧めします。

> **Note**
>
> AL1 の標準サポートは 2020 年 12 月 31 日に終了しました。AL1 メンテナンスサポートフェーズは 2023 年 12 月 31 日に終了しました。AL1 のサポート終了および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 のサポート終了に関するアップデート」を参照してください。

Amazon Linux の詳細については、AL2023、AL2、AL1」を参照してください。

Amazon Linux コンテナイメージについては、「Amazon Elastic コンテナレジストリユーザーガイド」の「Amazon Linux コンテナイメージ」を参照してください。

AL2 の廃止された機能

以下のセクションでは、AL2 でサポートされ、AL2023 には存在しない機能について説明します。つまり、AL2 には含まれているが、AL2023 には含まれておらず、今後も AL2023 に追加されることがない機能やパッケージのことです。この機能が AL2 でサポートされている期間については、AL2 ドキュメントを参照してください。

compat- パッケージ

プレフィックスが compat- の AL2 のパッケージはすべて、パッケージの最新バージョン向けに再構築されていない古いバイナリとのバイナリ互換性を確保するために提供されています。Amazon Linux の新しい各メジャーバージョンでは、以前のリリースの compat- パッケージは一切引き継がれません。

Amazon Linux のリリース (AL2 など) のすべてのcompat-パッケージは廃止され、後続のバージョン (AL2023 など) には存在しません。ライブラリの更新バージョンに対してソフトウェアを再構築することを強くお勧めします。

AL1 で中止され AL2 で削除された廃止済みの機能

このセクションでは、AL1 で使用できたものの 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段階の一環として、一部のパッケージには AL1 のサポート終了日より早いサポート終了日が設定されています。詳細については、「AL1 パッケージのサポートステートメント」を参照してください。

> **Note**
>
> 一部の AL1 機能は、以前のリリースで廃止されています。詳細については、「AL1 リリースノート」を参照してください。

トピック

32 ビット x86 (i686) AMI

AL1 の 2014.09 リリースにおいて、これが 32 ビット AMI を提供する最後のリリースになることが発表されました。したがって、AL1 の 2015.03 リリース以降、Amazon Linux では 32 ビットモードでのシステム実行をサポートしていません。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。

`aws-apitools-*` に置き換えAWS CLI

2013 年 9 AWS CLI月に がリリースされる前は、 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、ユーザーは 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。一連のコマンドラインユーティリティには、次の `aws-apitools-*` 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はありませんが、Amazon Linux では `aws-apitools-ec2` など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提供し、ユーザーの下位互換性を維持しています。AWS CLIは、`aws-apitools-*`パッケージよりも堅牢で完全なツールであり、アクティブに保守されており、すべての AWSAPIs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、更新プログラムは提供されていません。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 には含まれていません。

AL1 では aws-apitools-iam および aws-apitools-rds パッケージも提供されていましたが、AL1 で廃止され、AL2 以降の Amazon Linux には含まれていません。

AL2 での upstart から systemd への切り替え

AL2 は、AL1 の upstart に代わって systemd init システムを採用した最初の Amazon Linux リリースでした。AL1 から Amazon Linux の新バージョンに移行する際には、upstart 固有の設定を変更する必要があります。AL1 では systemd を使用できないため、upstart から systemd への移行は、AL2 や AL2023 など、Amazon Linux のより新しいメジャーバージョンへの移行の一環としてのみ行うことができます。

AL2 で廃止され、AL2023 で削除された機能

このセクションでは、AL2 で使用でき、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-* に置き換え AWS CLI
- awslogs は廃止となり、 Amazon CloudWatch Logs エージェント を推奨
- bzr リビジョン制御システム
- 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
- mcrypt
- OpenJDK 7 (java-1.7.0-openjdk)
- Python 2.7
- rsyslog-openssl による rsyslog-gnutls の置き換え
- Network Information Service (NIS) / yp
- Amazon VPC create-dhcp-options 内の複数のドメイン名
- glibc の Sun RPC
- audit 口グの OpenSSH キーフィンガープリント

- ld.gold リンカー
- ping6
- ftp パッケージ

32 ビット x86 (i686) パッケージ

AL1 の 2014.09 リリースにおいて、これが 32 ビット AMI を提供する最後のリリースになることが発表されました。したがって、AL1 の 2015.03 リリース以降、Amazon Linux では 32 ビットモードでのシステム実行をサポートしていません。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。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。

`aws-apitools-*` に置き換え AWS CLI

2013 年 9 AWS CLI 月の のリリース以前は、 で実装された AWS 一連のコマンドラインユーティリティを利用できるようになりました。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 はユーザーに下位互換性を提供するために、これらのコマンドラインユーティリティ (`aws-apitools-ec2` など) の一部を引き続き提供しました。 AWS CLI は、`aws-apitools-*`パッケージよりも堅牢で完全なツールであり、アクティブに保守されており、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、更新プログラムは提供されていません。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 には含まれていません。

awslogs は廃止となり、 Amazon CloudWatch Logs エージェント を推奨

awslogs パッケージは AL2 では廃止され、AL2023 では存在しなくなりました。これは、amazon-cloudwatch-agent パッケージで利用可能な統合 CloudWatch Logs エージェントに置き換えられます。詳細については、「Amazon CloudWatch Logs ユーザーガイド」を参照してください。

bzr リビジョン制御システム

GNU Bazaar (bzr) リビジョン制御システムは AL2 では廃止され、AL2023 では存在しなくなりました。

bzr のユーザーは、リポジトリを git に移行することをお勧めします。

cgroup v1

AL2023 は Unified Control Group 階層 (cgroup v2) に移行しますが、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 では削除されています。

CVE-2021-44228 に対応して、Amazon Linux は AL1 と AL2 用の Apache Log4j 用ホットパッチの RPM パッケージバージョンをリリースしました。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 では廃止され、AL2023 には存在しなくなりました。

アップストリームの PHP は、2016年 12 月に最初にリリースされ、PHP 7.1 において mcrypt 拡張機能を廃止し、2019 年 10 月の最終リリースされたで終了しました。

アップストリームの mcrypt ライブラリは2007 年に最後にリリースを行いましたが、SourceForge が 2017 年に新しいコミットに必要とする cvs リビジョン制御からの移行は行われていません。最新のコミット (および 3 年前のみ) は 2011 年のものであり、プロジェクトに管理者がいるという記述は削除されています。

mcrypt を引き続き使用しているユーザーには、コードを OpenSSL に移植することが推奨されています。mcrypt は AL2023 には追加されません。

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複数のバージョンの Amazon Corretto が用意されています。OpenJDK 7 パッケージは AL2 では廃止さ

れ、AL2023 では存在しなくなりました。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 に関する詳細については、「Java AL2 の」を参照してください。

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、「Python AL2 の」を参照してください。

`rsyslog-openssl` による `rsyslog-gnutls` の置き換え

`rsyslog-gnutls` パッケージは AL2 では廃止され、AL2023 では存在しなくなりました。`rsyslog-openssl` パッケージは、`rsyslog-gnutls` パッケージの使用をドロップインに置き換える必要があります。

Network Information Service (NIS) / `yp`

Network Information Service (NIS) は、当初は Yellow Pages または YP と呼ばれていましたが、AL2 では廃止され、AL2023 では存在しなくなりました。これには、`ypbind`、`ypserv`および `yp-tools` のパッケージが含まれます。NIS と統合されている他のパッケージでは、この機能は AL2023 で削除されています。

Amazon VPC `create-dhcp-options` 内の複数のドメイン名

Amazon Linux 2 では、`domain-name` パラメータの複数のドメイン名を `create-dhcp-options` に渡すことができ、結果として、`/etc/resolv.conf` に `search foo.example.com bar.example.com` などが含まれることがありました。Amazon VPC の DHCP サーバーは、DHCP オプション 15 を使用して指定されたドメイン名のリストを送信します。オプション 15 は単一のド

メイン名のみをサポートします (RFC 2132 セクション 3.17 を参照)。AL2023 ではネットワーク構成に `systemd-networkd` を使用しており、これは RFC に準拠しているため、AL2 に存在していた偶発的な機能は AL2023 には存在しません。

AWS CLI および Amazon VPC ドキュメントには、「一部の Linux オペレーティングシステムでは、スペースで区切られた複数のドメイン名を使用できます」と記載されています。ただし、Windows や他の Linux オペレーティングシステムでは、この値は単一のドメインとして処理されるため、予期せぬ動作の原因となります。DHCP オプションセットが、単一のドメインとして値を処理するオペレーティングシステムを実行するインスタンスを持つ VPC に関連付けられている場合は、ドメイン名を 1 つだけ指定します。

AL2023 などのこれらのシステムでは、DHCP オプション 15 を使用して 2 つのドメインを指定し (1 つのみ許可)、ドメイン名でスペース文字が無効であるため、スペース文字は `032` としてエンコードされ、結果として `/etc/resolv.conf` には `search foo.exmple.com032bar.example.com` が含まれます。

複数のドメイン名をサポートするには、DHCP サーバーで DHCP オプション 119 を使用する必要があります (RFC 3397、セクション 2 を参照)。Amazon VPC DHCP サーバーでサポートされている場合は、「Amazon VPC ユーザーガイド」を参照してください。

`glibc` の Sun RPC

`glibc` での Sun RPC の実装は AL2 では廃止され、AL2023 では削除されました。Sun RPC 機能が必要な場合は、`libtirpc` ライブラリ (AL2 および AL2023 で利用可能) の使用に移行することをお勧めします。また、`libtirpc` を採用することで、アプリケーションは IPv6 をサポートできます。

この変更は、アップストリームの`glibc`による機能の削除をコミュニティ全体が採用していることを反映しています。Fedora での `glibc` からの Sun RPC インターフェイスの削除や Gentoo での同様の変更がその例です。

`audit` ログの OpenSSH キーフィンガープリント

AL2 のライフサイクルの後半で、認証に使用されるキーフィンガープリントを出力するためのパッチが OpenSSH パッケージに追加されました。この機能は AL2023 には存在しません。

`ld.gold` リンカー

`ld.gold` リンカーは AL2 で使用でき、AL2023 で削除されています。明示的に gold リンカーを参照するソフトウェアを構築するユーザーは、通常の (`ld.bfd`) リンカーに移行する必要があります。

アップストリームの「GNU Binutils バージョン 2.44 リリースノート」(2025 年 2 月リリース) には、ld.goldの削除が文書化されています:「以前の慣行を変更し、このリリースでは binutils-2.44.tar の tarball に gold リンカーのソースは含まれていません。これは、gold リンカーが廃止され、ボランティアが自発的に開発とメンテナンスを継続することを提案しない限り、最終的に削除されるためです。」

ping6

AL2023 では、通常の ping ユーティリティが IPv6 をネイティブにサポートしており、別の /bin/ping6 を必要としません。AL2023 では、/usr/sbin/ping6 は /usr/bin/ping実行ファイルへのシンボリックリンクです。

この変更は、Fedora での Ping IPv6 の変更など、新しい iputils のバージョンが提供する機能をコミュニティ全体で導入していることに応じたものです。

ftp パッケージ

AL2 の ftp パッケージは、AL2023 以降の Amazon Linux では使用できなくなりました。これは、セキュリティ、保守性、最新のソフトウェア開発プラクティスに対する継続的なコミットメントの一環として決定されたものです。AL2023 への移行にあたって (またはそれ以前に)、使用されている ftp のレガシーパッケージを代替機能の 1 つに移行することをお勧めします。

背景

ftp のレガシーパッケージは、アップストリームで何年もアクティブにメンテナンスされていません。ソースコードの最後の重要な更新は 2000 年代初頭に行われ、元のソースリポジトリは使用できなくなっています。一部の Linux ディストリビューションではセキュリティ脆弱性のパッチが適用されていますが、コードベースはほとんど保守されていません。

推奨される代替機能

AL2023 には、FTP 機能の代わりとして、アクティブに保守されている最新の機能がいくつか用意されています。

lftp (AL2 および AL2023 で利用可能)

FTP、HTTP、SFTP、およびその他のプロトコルをサポートする高度なファイル転送プログラム。従来の ftp クライアントよりも多くの機能を提供し、アクティブに保守されています。

インストールのコマンド: dnf install lftp

`curl` (AL2 および AL2023 で利用可能)

URL を用いたデータ転送のための汎用性の高いコマンドラインツール。FTP、FTPS、HTTP、HTTPSをはじめとする多数のプロトコルをサポートしています。

AL2023 では、デフォルトで `curl-minimal` パッケージを介して使用できます。より広範囲なプロトコルをサポートするには、オプションで `curl-full` を使用して dnf swap curl-minimal curl-full にアップグレードできます。

`wget` (AL2 および AL2023 で利用可能)

ウェブからファイルをダウンロードするための非インタラクティブなコマンドラインユーティリティ。HTTP、HTTPS、FTP プロトコルをサポートしています。

インストールのコマンド: dnf install wget (すべての AL2023 イメージにデフォルトでインストールされているわけではありません)

`sftp` (AL2 および AL2023 で利用可能)

SSH 経由で動作し、暗号化されたファイル転送を可能にする安全なファイル転送プロトコル。

デフォルトでは、OpenSSH パッケージの一部として提供されます。

移行に関する考慮事項

アプリケーションまたはスクリプトが `ftp` のレガシークライアントに依存している場合は、次の移行アプローチを検討してください。

1. 最新の代替機能を使用するようにスクリプトを更新する: `ftp` レガシークライアントの代わりに `lftp`、`curl`、`wget`、または `sftp` を使用するようにスクリプトを変更します。
2. パッケージの依存関係を確認する: 一部のアプリケーションでは、最新のプロトコルへの移行が内部で行われてからかなり時間が経過しているにもかかわらず、`ftp` パッケージがパッケージメタデータに依存関係として一覧表示されている場合があります。このような場合、`ftp` パッケージに `/usr/bin/ftp` が含まれていなくても、アプリケーションは AL2023 で正しく動作する可能性があります。指定された依存関係のみに依存するのではなく、アプリケーションの実際の要件を確認します。
3. アプリケーションの依存関係を更新する: `ftp` パッケージとの依存関係を宣言していものの、実際には使用していないアプリケーションについては、パッケージメタデータを更新してこの不要な依存関係を削除します。

セキュリティに関する考慮事項

FTP プロトコルは、認証情報を含むデータをプレーンテキストで送信します。セキュリティ重視のアプリケーションでは、推奨される代替ツールでサポートされている SFTP や HTTPS などの暗号化された代替機能を使用することを強くお勧めします。

AL2023 への移行を準備する

AL2023 への移行を準備できます。 AL2

トピック

AL2023 の変更のリストを確認する

AL2023 ドキュメントには、AL2 以降に実装された変更の詳細なリストが含まれています。この情報は、AL2 と AL2023 の比較」セクションにあります。また、AL2023 の「パッケージの変更」セクションにあるソフトウェアパッケージの変更の包括的なリストもあります。

AL2023 には は含まれませんamazon-linux-extras。代わりに、複数のバージョンが提供されている名前空間パッケージが提供されます。AL2023 では多くのパッケージが更新されるため、AL2023 の基本バージョンは から取得するバージョンよりも後になる可能性がありますamazon-linux-extras。

> **Note**
>
> EOL であるためamazon-linux-extras、 は実行しないことをお勧めします。

ドキュメントのこれらのセクションを確認したら、AL2023 に変更があり、移行に合わせて環境を適応させる必要があるかどうかを判断できます。たとえば、最終的に Python 2.7 スクリプトを Python 3 に移行する必要がある場合があります。

cron ジョブからsystemdタイマーに移行する

デフォルトでは、 cronは AL2023 にインストールされません。AL2023 への移行に備えて、ジョブを AL2 のcronsystemdタイマーに移行できます。 systemdには、タイマーの実行時期をより正確に制御したり、ログ記録を改善したりするなど、多くの利点があります。

AL2 の制限事項

以下のトピックでは、AL2 のさまざまな制限と、それらが新しいバージョンの Amazon Linux で解決されたかどうかについて説明します。

トピック

yum は GPG サブキーで行われた GPG 署名を検証できません

AL2 のrpmパッケージマネージャーのバージョンは、GPG サブキーで行われたパッケージ署名の検証のサポートrpmが追加される前からのものです。AL2 と互換性のあるパッケージを作成する場合は、AL2 の一部rpmである と互換性のある GPG 署名キーを使用する必要があります。

既存のユーザーの下位互換性を確保するために、AL2 rpmの のバージョンはセキュリティバックポートのみを受け取ります。

AL2023 rpmの バージョンには、GPG サブキーで作成されたパッケージ署名の検証のサポートが含まれています。

AL1 と AL2 の比較

以下のトピックでは、AL1 と AL2 の主な違いについて説明します。また、有効期間とサポート、パッケージの変更に関する情報も含まれています。

トピック

AL1 サポートと EOL

AL1 が EOL になりました。AL1 は 2020 年 12 月 31 日に標準サポートを終了し、2023 年 12 月 31 日までメンテナンスサポートフェーズでした。

最新の Amazon Linux バージョンにアップグレードすることをお勧めします。

Graviton AWSプロセッサのサポート

AL2 は Graviton プロセッサのサポートを導入しました。AL2023 は Graviton プロセッサ用にさらに最適化されています。

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

AL2 では、 を`init`システム`upstart`として`systemd`置き換えます。

Python 2.6 および 2.7 が 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、パッケージはまだインストールするリポジトリに存在していました。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 が付属しています。

AL2023 は Python 3 への移行を完了し、Python 2.x バージョンはリポジトリに含まれません。

AL1 AMI と AL2 AMIs にインストールされているパッケージの比較

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

パッケージ	AL1 AMI	AL2 AMI
basesystem	10.0	10.0
bash	4.2.46	4.2.46
bash-completion		2.1
bc	1.06.95	1.06.95
bind-export-libs		9.11.4
bind-libs	9.8.2	9.11.4
bind-libs-lite		9.11.4
bind-license		9.11.4
bind-utils	9.8.2	9.11.4
binutils	2.27	2.29.1
blktrace		1.0.5
boost-date-time		1.53.0
boost-system		1.53.0
boost-thread		1.53.0
bridge-utils		1.5
bzip2	1.0.6	1.0.6
bzip2-libs	1.0.6	1.0.6
ca-certificates	2023 年 2 月 62 日	2023 年 2 月 62 日
checkpolicy	2.1.10	
chkconfig	1.3.49.3	1.7.4

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

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

パッケージ	AL1 AMI	AL2 AMI
dracut-config-ec2		2.0
dracut-config-generic		033
dracut-modules-growroot	0.20	
dump	0.4	
dyninst		9.3.1
e2fsprogs	1.43.5	1.42.9
e2fsprogs-libs	1.43.5	1.42.9
ec2-hibinit-agent	1.0.0	1.0.2
ec2-instance-connect		1.1
ec2-instance-connect-selinux		1.1
ec2-net-utils	0.7	1.7.3
ec2-utils	0.7	1.2
ed	1.1	1.9
elfutils-default-yama-scope		0.176
elfutils-libelf	0.168	0.176
elfutils-libs		0.176
epel-release	6	
ethtool	3.15	4.8
expat	2.1.0	2.1.0
および	5.37	5.11

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

パッケージ	AL1 AMI	AL2 AMI
python-chardet		2.2.1
python-configobj		4.7.2
python-daemon		1.6
python-devel		2.7.18
python-docutils		0.12
python-enum34		1.0.4
python-idna		2.4
python-iniparse		0.4
python-ipaddress		1.0.16
python-jinja2		2.7.2
python-jsonpatch		1.2
python-jsonpointer		1.9
python-jwcrypto		0.4.2
python-kitchen		1.1.1
python-libs		2.7.18
python-lockfile		0.9.1
python-markupsafe		0.11
python-pillow		2.0.0
python-ply		3.4
python-pycparser		2.14

パッケージ	AL1 AMI	AL2 AMI
python-pycurl		7.19.0
python-repoze-lru		0.4
python-requests		2.6.0
python-simplejson		3.2.0
python-urlgrabber		3.10
python-urllib3		1.25.9
python2-botocore		1.18.6
python2-colorama		0.3.9
python2-cryptography		1.7.2
python2-dateutil		2.6.1
python2-futures		3.0.5
python2-jmespath		0.9.3
python2-jsonschema		2.5.1
python2-oauthlib		2.0.1
python2-pyasn1		0.1.9
python2-rpm		4.11.3
python2-rsa		3.4.1
python2-s3transfer		0.3.3
python2-setuptools		41.2.0
python2-six		1.11.0

パッケージ	AL1 AMI	AL2 AMI
python27	2.7.18	
python27-PyYAML	3.10	
python27-babel	0.9.4	
python27-backports	1.0	
python27-backports-ssl_match_hostname	3.4.0.2	
python27-boto	2.48.0	
python27-botocore	1.17.31	
python27-chardet	2.0.1	
python27-colorama	0.4.1	
python27-configobj	4.7.2	
python27-crypto	2.6.1	
python27-daemon	1.5.2	
python27-dateutil	2.1	
python27-devel	2.7.18	
python27-docutils	0.11	
python27-ecdsa	0.11	
python27-futures	3.0.3	
python27-imaging	1.1.6	
python27-iniparse	0.3.1	
python27-jinja2	2.7.2	

パッケージ	AL1 AMI	AL2 AMI
python27-jmespath	0.9.2	
python27-jsonpatch	1.2	
python27-jsonpointer	1.0	
python27-kitchen	1.1.1	
python27-libs	2.7.18	
python27-lockfile	0.8	
python27-markupsafe	0.11	
python27-paramiko	1.15.1	
python27-pip	9.0.3	
python27-ply	3.4	
python27-pyasn1	0.1.7	
python27-pycurl	7.19.0	
python27-pygpgme	0.3	
python27-pyliblzma	0.5.3	
python27-pystache	0.5.3	
python27-pyxattr	0.5.0	
python27-requests	1.2.3	
python27-rsa	3.4.1	
python27-setuptools	36.2.7	
python27-simplejson	3.6.5	

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

パッケージ	AL1 AMI	AL2 AMI
rng-tools	5	6.8
rootfiles	8.1	8.1
rpcbind	0.2.0	0.2.0
rpm	4.11.3	4.11.3
rpm-build-libs	4.11.3	4.11.3
rpm-libs	4.11.3	4.11.3
rpm-plugin-systemd-inhibit		4.11.3
rpm-python27	4.11.3	
rsync	3.0.6	3.1.2
rsyslog	5.8.10	8.24.0
ruby	2.0	
ruby20	2.0.0.648	
ruby20-irb	2.0.0.648	
ruby20-libs	2.0.0.648	
rubygem20-bigdecimal	1.2.0	
rubygem20-json	1.8.3	
rubygem20-psych	2.0.0	
rubygem20-rdoc	4.2.2	
rubygems20	2.0.14.1	
scl-utils		20130529

パッケージ	AL1 AMI	AL2 AMI
screen	4.0.3	4.1.0
sed	4.2.1	4.2.2
selinux-policy		3.13.1
selinux-policy-targeted		3.13.1
sendmail	8.14.4	
setserial	2.17	2.17
セットアップ	2.8.14	2.8.71
setuptool		1.19.11
sgpio	1.2.0.10	1.2.0.10
shadow-utils	4.1.4.2	4.1.5.1
shared-mime-info	1.1	1.8
slang	2.2.1	2.2.4
sqlite	3.7.17	3.7.17
sssd-client		1.16.5
strace		4.26
sudo	1.8.23	1.8.23
sysctl-defaults	1.0	1.0
sysfsutils	2.1.0	
sysstat		10.1.5
system-release	2018 年 3 月	2

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

パッケージ	AL1 AMI	AL2 AMI
update-motd	1.0.1	1.1.2
upstart	0.6.5	
usermode		1.111
ustr	1.0.4	1.0.4
util-linux	2.23.2	2.30.2
vim-common	9.0.1712	9.0.2081
vim-data	9.0.1712	9.0.2081
vim-enhanced	9.0.1712	9.0.2081
vim-filesystem	9.0.1712	9.0.2081
vim-minimal	9.0.1712	9.0.2081
virt-what		1.18
wget	1.18	1.14
which	2.19	2.20
words	3.0	3.0
xfsdump		3.1.8
xfsprogs		5.0.0
xorg-x11-font-utils	7.2	
xorg-x11-fonts-Type1	7.2	
xxd	9.0.1712	9.0.2081
xz	5.2.2	5.2.2

パッケージ	AL1 AMI	AL2 AMI
xz-libs	5.2.2	5.2.2
yajl		2.0.4
yum	3.4.3	3.4.3
yum-langpacks		0.4.2
yum-metadata-parser	1.1.4	1.1.4
yum-plugin-priorities	1.1.31	1.1.31
yum-plugin-upgrade-helper	1.1.31	
yum-utils	1.1.31	1.1.31
zip	3.0	3.0
zlib	1.2.8	1.2.7

AL1 ベースコンテナイメージと AL2 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

パッケージ	AL1 コンテナ	AL2 コンテナ
amazon-linux-extras		2.0.3
basesystem	10.0	10.0
bash	4.2.46	4.2.46
bzip2-libs	1.0.6	1.0.6
ca-certificates	2023 年 2 月 62 日	2023 年 2 月 62 日
chkconfig	1.3.49.3	1.7.4

パッケージ	AL1 コンテナ	AL2 コンテナ
coreutils	8.22	8.22
cpio		2.12
curl	7.61.1	8.3.0
cyrus-sasl-lib	2.1.23	2.1.26
db4	4.7.25	
db4-utils	4.7.25	
diffutils		3.3
elfutils-libelf	0.168	0.176
expat	2.1.0	2.1.0
file-libs	5.37	5.11
filesystem	2.4.30	3.2
findutils		4.5.11
gawk	3.1.7	4.0.2
gdbm	1.8.0	1.13
glib2	2.36.3	2.56.1
glibc	2.17	2.26
glibc-common	2.17	2.26
glibc-langpack-en		2.26
glibc-minimal-langpack		2.26
gmp	6.0.0	6.0.0

パッケージ	AL1 コンテナ	AL2 コンテナ
gnupg2	2.0.28	2.0.22
gpgme	1.4.3	1.3.2
grep	2.20	2.20
gzip	1.5	
情報	5.1	5.1
keyutils-libs	1.5.8	1.5.8
krb5-libs	1.15.1	1.15.1
libacl	2.2.49	2.2.51
libassuan	2.0.3	2.1.0
libattr	2.4.46	2.4.46
libblkid		2.30.2
libcap	2.16	2.54
libcom_err	1.43.5	1.42.9
libcrypt		2.26
libcurl	7.61.1	8.3.0
libdb		5.3.21
libdb-utils		5.3.21
libffi	3.0.13	3.0.13
libgcc		7.3.1
libgcc72	7.2.1	

パッケージ	AL1 コンテナ	AL2 コンテナ
libgcrypt	1.5.3	1.5.3
libgpg-error	1.11	1.12
libicu	50.2	
libidn2	2.3.0	2.3.0
libmetalink		0.1.3
libmount		2.30.2
libnghttp2	1.33.0	1.41.0
libpsl	0.6.2	
libselinux	2.1.10	2.5
libsepol	2.1.7	2.5
libssh2	1.4.2	1.4.3
libstdc++		7.3.1
libstdc++72	7.2.1	
libtasn1	2.3	4.10
libunistring	0.9.3	0.9.3
libuuid		2.30.2
libverto	0.2.5	0.2.5
libxml2	2.9.1	2.9.1
libxml2-python27	2.9.1	
lua	5.1.4	5.1.4

パッケージ	AL1 コンテナ	AL2 コンテナ
make	3.82	
ncurses	5.7	6.0
ncurses-base	5.7	6.0
ncurses-libs	5.7	6.0
nspr	4.25.0	4.35.0
nss	3.53.1	3.90.0
nss-pem	1.0.3	1.0.3
nss-softokn	3.53.1	3.90.0
nss-softokn-freebl	3.53.1	3.90.0
nss-sysinit	3.53.1	3.90.0
nss-tools	3.53.1	3.90.0
nss-util	3.53.1	3.90.0
openldap	2.4.40	2.4.44
openssl	1.0.2k	
openssl-libs		1.0.2k
p11-kit	0.18.5	0.23.22
p11-kit-trust	0.18.5	0.23.22
pcre	8.21	8.32
pinentry	0.7.6	0.8.1
pkgconfig	0.27.1	

パッケージ	AL1 コンテナ	AL2 コンテナ
popt	1.13	1.13
pth	2.0.7	2.0.7
pygpgme		0.3
pyliblzma		0.5.3
python		2.7.18
python-iniparse		0.4
python-libs		2.7.18
python-pycurl		7.19.0
python-urlgrabber		3.10
python2-rpm		4.11.3
python27	2.7.18	
python27-chardet	2.0.1	
python27-iniparse	0.3.1	
python27-kitchen	1.1.1	
python27-libs	2.7.18	
python27-pycurl	7.19.0	
python27-pygpgme	0.3	
python27-pyliblzma	0.5.3	
python27-pyxattr	0.5.0	
python27-urlgrabber	3.10	

パッケージ	AL1 コンテナ	AL2 コンテナ
pyxattr		0.5.1
readline	6.2	6.2
rpm	4.11.3	4.11.3
rpm-build-libs	4.11.3	4.11.3
rpm-libs	4.11.3	4.11.3
rpm-python27	4.11.3	
sed	4.2.1	4.2.2
セットアップ	2.8.14	2.8.71
shared-mime-info	1.1	1.8
sqlite	3.7.17	3.7.17
sysctl-defaults	1.0	
system-release	2018 年 3 月	2
tar	1.26	
tzdata	2023c	2023c
vim-data		9.0.2081
vim-minimal		9.0.2081
xz-libs	5.2.2	5.2.2
yum	3.4.3	3.4.3
yum-metadata-parser	1.1.4	1.1.4
yum-plugin-ovl	1.1.31	1.1.31

パッケージ	AL1 コンテナ	AL2 コンテナ
yum-plugin-priorities	1.1.31	1.1.31
yum-utils	1.1.31	
zlib	1.2.8	1.2.7

Amazon EC2 の AL2 Amazon EC2

> **Note**
>
> AL2 は Amazon Linux の最新バージョンではなくなりました。AL2023 は AL2 の後継です。詳細については、AL2023 ユーザーガイドの「AL2 と AL2023 の比較」および「AL2023 でのパッケージの変更のリスト」を参照してください。 AL2023

トピック

AL2 AMI を使用して Amazon EC2 インスタンスを起動する AL2

AL2 AMI を使用して Amazon EC2 インスタンスを起動できます。 AL2 詳細については、「ステップ 1: インスタンスを起動する」を参照してください。

Systems Manager を使用して最新の AL2 AMI を検索する

Amazon EC2 は、インスタンスの起動時に使用できる、 によって管理 AWS される AWS Systems Manager パブリック AMIs のパブリックパラメータを提供します。例えば、EC2-provided パラメータ `/aws/service/ami-amazon-linux-latest/amzn2-ami-kernel-default-hvm-x86_64-`

gp2はすべてのリージョンで使用でき、常に特定のリージョンの AL2 AMI の最新バージョンを指します。

を使用して最新の AL2023 AMI を検索するには AWS Systems Manager、AL2023 の開始方法」を参照してください。

Amazon EC2 AMI のパブリックパラメータは、次のパスから使用できます。

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

次の AWS CLI コマンドを実行すると、現在の AWS リージョン内のすべての Amazon Linux AMIs のリストを表示できます。

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

パブリックパラメータを使用してインスタンスを作成するには

次の例では、EC2-providedパブリックパラメータを使用して、最新の AL2 AMI を使用して`m5.xlarge`インスタンスを起動します。

このパラメータをコマンドで指定するには、`resolve:ssm:`*`public-parameter`* 構文を使用します。`resolve:ssm` は標準のプレフィクス、`public-parameter` はパブリックパラメータのパスと名前です。

この例では、`--count` パラメータと `--security-group` パラメータは含まれていません。`--count` はデフォルトで 1 になります。デフォルトの VPC とデフォルトのセキュリティグループがある場合は、これらが使用されます。

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

詳細については、「 ユーザーガイド」の「パブリックパラメータの使用」を参照してください。
AWS Systems Manager

Amazon Linux 2 AMI 名について

Amazon Linux 2 AMI 名は、次の命名スキームを使用します。

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

- 最小 AMIs、イメージサイズを減らすために、プリインストールされたパッケージのセットが最小限に抑えられています。
- kernel-VERSION は、それぞれの AMI にプリインストールされているカーネルバージョンを決定します。
  - `kernel-5.10` は Linux カーネルバージョン 5.10 を選択します。これは AL2 に推奨されるカーネルバージョンです。
  - `kernel-default` は、AL2 に推奨されるデフォルトカーネルを選択します。これは kernel-5.10 のエイリアスです。
  - `kernel-4.14` は Linux カーネルバージョン 4.14 を選択します。これは、古い AMI リリースとの互換性のためにのみ提供されています。新しいインスタンスの起動には、このバージョンを使用しないでください。この AMI がサポートされなくなることが予想されます。
  - AMI 名の特別なセットは、特定のカーネルを参照せずに存在します。これらの AMIs は kernel-4.14 のエイリアスです。これらの AMIsは、古い AMI リリースとの互換性のためにのみ提供されています。この AMI 名を新しいインスタンスの起動に使用しないでください。これらの AMIsることを期待します。
- x86_64/aarch64 は、AMI を実行する CPU プラットフォームを決定します。Intel および AMD ベースの EC2 インスタンスには x86_64 を選択します。EC2 Graviton インスタンスの aarch64 を選択します。
- ebs/gp2 は、それぞれの AMI を供給するために使用される EBS ボリュームタイプを決定します。リファレンスについては、「EBS ボリュームタイプ」を参照してください。常に gp2 を選択します。

Amazon EC2 インスタンスに接続する

Amazon Linux インスタンスに接続するには、SSH、EC2 Instance Connect など AWS Systems Manager Session Manager、いくつかの方法があります。詳細については、「Amazon EC2 ユーザーガイド」の「Linux インスタンスへの接続」を参照してください。

SSH ユーザーと sudo

Amazon Linux では、デフォルトでリモート`root`セキュアシェル (SSH) は許可されません。また、ブルートフォース攻撃を防ぐためにパスワード認証は無効になっています。Amazon Linux インスタンスへの SSH ログインを有効にするには、起動時にキーペアをインスタンスに提供する必要があり

ます。インスタンスを起動するときに使用するセキュリティグループで、SSH アクセスを許可するよう設定する必要もあります。デフォルトでは、SSH を使用してリモートでログインできるアカウントは のみですec2-user。このアカウントには sudo権限もあります。リモートrootログインを有効にする場合、キーペアとセカンダリユーザーに依存するよりも安全性が低いことに注意してください。

AL2 AMI ブートモード

AL2 AMIs にはブートモードパラメータが設定されていません。AL2 AMIs から起動されたインスタンスは、インスタンスタイプのデフォルトのブートモード値に従います。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。

パッケージリポジトリ

この情報は AL2 に適用されます。AL2023 の詳細については、Amazon Linux 2AL2023の「AL2023 でのパッケージとオペレーティングシステムの更新の管理」を参照してください。

AL2 および AL1 は、各 Amazon EC2 AWS Region でホストされているオンラインパッケージリポジトリで使用するように設計されています。リポジトリはすべてのリージョンに存在し、yum 更新ツールを使用してアクセスできます。各リージョンでリポジトリをホストしているため、データ転送料金なしで、更新を迅速にデプロイできます。

> **Important**
>
> AL1 の最新バージョンは 2023 年 12 月 31 日に EOL に達し、2024 年 1 月 1 日以降、セキュリティアップデートやバグ修正は行われません。詳細については「Amazon Linux AMI のサポート終了」を参照してください。

インスタンスのデータやカスタマイズを保持する必要がない場合は、現在の AL2 AMI を使用して新しいインスタンスを起動できます。インスタンスのデータまたはカスタマイズを保持する必要がある場合は、Amazon Linux パッケージリポジトリを使用してそれらのインスタンスを維持できます。これらのリポジトリには、更新されたすべてのパッケージが含まれます。実行中のインスタンスにこれらの更新を適用するよう選択できます。AMI および更新パッケージの以前のバージョンは、新しいバージョンがリリースされても引き続き使用できます。

> **Note**
>
> Amazon EC2 インスタンスでインターネットアクセスなしでパッケージを更新およびインストールするには、「ALAL1, AL2または AL2023 を実行している Amazon EC2 インスタンスでインターネットアクセスなしで yum を更新するか、パッケージをインストールするにはどうすればよいですか？」を参照してください。

パッケージを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 yum install package
```

Amazon Linux に必要なアプリケーションが含まれていない場合は、Amazon Linux インスタンスにアプリケーションを直接インストールできます。Amazon Linux はパッケージ管理yumに RPMs とを使用します。これは新しいアプリケーションをインストールする最も直接的な方法です。多くのアプリケーションが中央の Amazon Linux リポジトリで利用可能なので、最初にアプリケーションがそのリポジトリで利用できるかどうかを確認する必要があります。そこから、これらのアプリケーションを Amazon Linux インスタンスに追加できます。

実行中の Amazon Linux インスタンスにアプリケーションをアップロードするには、scp または sftp を使用し、インスタンスにログインしてアプリケーションを設定します。組み込みの cloud-init パッケージから PACKAGE_SETUP アクションを使用して、インスタンスの起動時にアプリケーションをアップロードすることもできます。詳細については、AL2 での cloud-init の使用 を参照してください。

セキュリティ更新

セキュリティ更新は、 パッケージリポジトリを使用して提供されます。セキュリティ更新と更新された AMI セキュリティアラートの両方が Amazon Linux セキュリティセンターで公開されます。AWS セキュリティポリシーの詳細については、またはセキュリティの問題を報告するには、「AWS クラウドのセキュリティ」を参照してください。

AL1 および AL2 は、起動時に重要または重要なセキュリティ更新プログラムをダウンロードしてインストールするように設定されています。この設定にはカーネルの更新は含まれません。

AL2023 では、AL1 および AL2 と比較してこの設定が変更されました。AL2023 のセキュリティアップデートの詳細については、「Amazon Linux 2023 ユーザーガイド」の「セキュリティアップデートと機能」を参照してください。

起動後にユースケースに必要な更新を行うことをお勧めします。たとえば、起動時にすべての更新プログラム (セキュリティ更新プログラムだけでなく) を適用したり、各更新プログラムを評価してシステムに適用可能な更新プログラムのみを適用したりできます。これは、cloud-init 設定 `repo_upgrade` を使用して制御されます。次の cloud-init 設定のスニペットは、インスタンス初期化に渡すユーザーデータテキストで設定を変更する方法を示しています。

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

`repo_upgrade` の有効な値は次のとおりです。

`critical`

まだ適用されていない緊急のセキュリティ更新プログラムを適用します。

`important`

まだ適用されていない緊急および重要なセキュリティ更新プログラムを適用します。

`medium`

まだ適用されていない緊急、重要、中レベルのセキュリティ更新プログラムを適用します。

`low`

低レベルのセキュリティ更新プログラムを含む、まだ適用されていないセキュリティ更新プログラムをすべて適用します。

`security`

Amazon によってセキュリティ更新としてマークされた保留中のクリティカルまたは重要な更新を適用します。

`bugfix`

Amazon によってバグフィックスとしてマークされた更新を適用します。バグフィックスは大きなサイズの更新セットで、セキュリティ更新および他のさまざまな小さなバグに対する修正が含まれます。

`all`

分類に関係なく、使用できる適切な更新すべてを適用します。

`none`

起動時に更新をインスタンスに適用しません。

> **メモ**
>
> Amazon Linux は、更新を としてマークしませんbugfix。Amazon Linux からセキュリティに関連しない更新を適用するには、 を使用しますrepo_upgrade: all。

repo_upgrade のデフォルトの設定は security です。つまり、ユーザーデータに異なる値を指定しない場合、デフォルトでは、Amazon Linux はその時点でインストールされているパッケージの起動時に、セキュリティ更新を実行します。Amazon Linux は、インストール済みのパッケージに更新がある場合も、/etc/motd ファイルを使用して、ログイン時に利用可能な更新の数を一覧表示して通知します。これらの更新をインストールするには、インスタンスで sudo yum upgrade を実行する必要があります。

リポジトリの設定

AL1 および AL2 AMIs はセキュリティ更新を除き、AMI の作成時に利用可能なパッケージのスナップショットです。元の AMI にはないが、ランタイムにインストールされたパッケージは、利用可能な最新バージョンになります。AL2 で利用可能な最新のパッケージを取得するには、 を実行しますyum update -y。

> **トラブルシューティングのヒント**
>
> t3.nano などのナノインスタンスタイプで yum update の実行中に cannot allocate memory エラーが発生した場合は、更新を有効にするためにスワップ領域を割り当てる必要がある場合があります。

AL2023 では、AL1 および AL2 と比較してリポジトリ設定が変更されました。AL2023 リポジトリの詳細については、「パッケージおよびオペレーションシステムアップデートの管理」を参照してください。

AL2023 までのバージョンは、Amazon Linux の 1 つのマイナーバージョンから次のバージョンにローリングする更新の継続的なフロー (ローリングリリースとも呼ばれます) を提供するように設定されていました。ベストプラクティスとして、古い AMI を起動して更新を適用するのではなく、AMI を利用可能な最新の AMIs に更新することをお勧めします。

AL1 から AL2 へ、または AL2 から AL2AL2023 へなど、主要な Amazon Linux バージョン間でインプレースアップグレードはサポートされていません。詳細については、「Amazon Linux の入手可能性」を参照してください。

AL2 での cloud-init の使用

cloud-init パッケージは、Canonical によって構築されたオープンソースアプリケーションであり、Amazon EC2 などの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ときに使用されます。Amazon Linux には、カスタマイズされたバージョンの cloud-init が含まれています。これにより、起動時にインスタンスに発生するアクションを指定できます。インスタンスの起動時に、ユーザーデータフィールドを使用して必要なアクションを cloud-init に渡すことができます。つまり、さまざまなユースケースに対して共通の AMI を使用し、起動時にその AMI を動的に設定できます。Amazon Linux は、ec2 ユーザーアカウントの初期設定を実行するためにも cloud-init を使用します。

詳細については、「cloud-init ドキュメント」を参照してください。

Amazon Linux 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` に作成することができます。このディレクトリ内のすべてのファイルは、cloud-init で読み取られます。それらは辞書と同じ順序に読み取られ、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 パッケージは、起動時にインスタンスのこれらの (およびその他の) 共通の設定タスクを実行します。

- デフォルトのロケールを設定。
- ホスト名を設定。
- ユーザーデータの解析と処理。
- 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` に追加する。
- パッケージ管理のためにリポジトリを準備する
- ユーザーデータで定義されたパッケージアクションの処理。
- ユーザーデータにあるユーザースクリプトを実行します。
- インスタンスストアボリュームをマウントする (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ある場合は `/media/ephemeral0` にマウントされ、有効なファイルシステムが含まれます。それ以外の場合は、マウントされません。
  - デフォルトでは、インスタンスに関連付けられたスワップボリュームがマウントされます (`m1.small`および`c1.medium`インスタンスタイプの場合のみ)。

- 次の 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することができます。

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

  マウントをより詳細にコントロールするには、cloud-init ドキュメントの「マウント」を参照してください。

- TRIM をサポートするインスタンスストアボリュームは、インスタンスの起動時にはフォーマットされないため、マウントする前にパーティション化してフォーマットする必要があります。詳細については、「インスタンスストアボリュームTRIMのサポート」を参照してください。disk_setup モジュールを使用して、起動時にインスタンスストアボリュームをパーティションおよびフォーマット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「Disk Setup」を参照してください。

サポートされているユーザーデータ形式

cloud-init パッケージは、さまざまな形式のユーザーデータ処理をサポートしています。

- Gzip
  - ユーザーデータが gzip 圧縮されている場合、cloud-init はデータを解凍し、適切に処理します。
- MIME マルチパート
  - MIME マルチパートファイルを使用して、複数のデータタイプを指定できます。たとえば、ユーザーデータスクリプトとクラウド設定タイプの両方を指定できます。マルチパートファイルのパートの形式が、サポートされている形式のいずれかの場合、そのパートは cloud-init で処理できます。
- Base64 デコード
  - ユーザーデータが base64 でエンコードされている場合、cloud-init は、デコードされたデータをサポートされているタイプの 1 つとして理解できるかどうかを決定します。デコードされたデータを認識できる場合、データをデコードし、適切に処理します。認識できない場合、base64 データは変更されません。
- ユーザーデータスクリプト
  - 「#!」または「Content-Type: text/x-shellscript」で始まります。

  - このスクリプトは、初回の起動サイクル時に /etc/init.d/cloud-init-user-scripts によって実行されます。これは起動プロセスの後半 (初期設定アクションが実行された後) に実行されます。
- インクルードファイル
  - 「#include」または「Content-Type: text/x-include-url」で始まります。
  - このコンテンツはインクルードファイルです。ファイルには URL の一覧 (1行に1つの URL) が含まれます。各 URL が読み取られ、そのコンテンツが同じルールセットを使用して渡されます。URL から読み取られたコンテンツは gzip 圧縮され、MIME マルチパート、またはプレーンテキスト形式になります。
- クラウド設定データ
  - 「#cloud-config」または「Content-Type: text/cloud-config」で始まります。
  - このコンテンツはクラウド設定データです。
- アップスタートジョブ (AL2 ではサポートされていません)
  - 「#upstart-job」または「Content-Type: text/upstart-job」で始まります。
  - このコンテンツは のファイルに保存され/etc/init、アップスタートは他のアップスタートジョブと同様にコンテンツを消費します。
- クラウドブートフック
  - 「#cloud-boothook」または「Content-Type: text/cloud-boothook」で始まります。
  - このコンテンツはブートフックデータです。このデータは /var/lib/cloud にあるファイルに保存され、すぐに実行されます。
  - これは最初に使用可能な [hook] (フック) です。1 回だけ実行するためのメカニズムはありません。ブートフックは自身でこの点に対処する必要があります。環境変数 INSTANCE_ID でインスタンス ID が指定されています。この変数を使用して、インスタンスあたり1つのブートフックデータのセットを提供します。

AL2 インスタンスを設定する

AL2 インスタンスを正常に起動してログインしたら、そのインスタンスを変更できます。特定のアプリケーションのニーズを満たすためにインスタンスを設定するには、多くの方法があります。ここでは、初めて作業する場合の一般的なタスクについて説明します。

コンテンツ

- AL2 インスタンスでソフトウェアを管理する
- Amazon EC2 AL2 インスタンスのプロセッサ状態制御
- AL2 の I/O スケジューラ
- AL2 インスタンスのホスト名を変更する
- AL2 インスタンスで動的 DNS を設定する
- AL2 の ec2-net-utils を使用してネットワークインターフェイスを設定する

一般的な設定シナリオ

Amazon Linux のベースのディストリビューションには、基本的なサーバー操作に必要なソフトウェアパッケージとユーティリティが含まれています。ただし、さまざまなソフトウェアリポジトリでさらに多くのソフトウェアパッケージを利用できます。また、ソースコードから、さらに多くのパッケージソースコードを作成できます。これらの場所からソフトウェアをインストールし、作成する方法についての詳細は、AL2 インスタンスでソフトウェアを管理する を参照してください。

Amazon Linux インスタンスには、`ec2-user` が事前設定されていますが、スーパーユーザー権限を持たない他のユーザーを追加することがあります。ユーザーの追加と削除の詳細については、Amazon EC2 ユーザーガイド」のLinux「インスタンスでユーザーを管理する」を参照してください。

お客様がネットワークを所有し、それにドメイン名を登録している場合、インスタンスのホスト名を変更して、そのドメインに含まれる一部としてインスタンスを識別できます。また、システムプロンプトを変更して、より意味のある名前を表示することもできます。ホスト名設定を変更する必要はありません。詳細については、AL2 インスタンスのホスト名を変更するを参照してください。動的 DNS サービスプロバイダを使用するようにインスタンスを設定できます。詳細については、AL2 インスタンスで動的 DNS を設定するを参照してください。

Amazon EC2 でインスタンスを起動するとき、起動後にそのインスタンスにユーザーデータを渡し、一般的な設定タスクを実行したり、スクリプトを実行したりできます。2 つのタイプのユーザーデータを Amazon EC2 に渡すことができます。cloud-init ディレクティブとシェルスクリプトです。詳細については、Amazon EC2 ユーザーガイド」の「起動時にLinuxインスタンスでコマンドを実行する」を参照してください。

AL2 インスタンスでソフトウェアを管理する

Amazon Linux のベースのディストリビューションには、基本的なサーバー操作に必要なソフトウェアパッケージとユーティリティが含まれています。

この情報は AL2 に適用されます。AL2023 の詳細については、Amazon Linux 2AL2023の「AL2023 でのパッケージとオペレーティングシステムの更新の管理」を参照してください。

ソフトウェアは、最新の状態に維持することが重要です。Linux ディストリビューションの多くのパッケージは頻繁に更新されます。これにより、バグが修正され、機能が追加されて、セキュリティ上の弱点に対する防御措置が行われます。詳細については、「AL2 インスタンスのインスタンスソフトウェアを更新する」を参照してください。

デフォルトでは、AL2 インスタンスは、次のリポジトリを有効にして起動します。

- `amzn2-core`
- `amzn2extra-docker`

これらのリポジトリには、 によって更新される多くのパッケージがありますが AWS、インストールするパッケージが別のリポジトリに含まれている可能性があります。詳細については、「AL2 インスタンスにリポジトリを追加する」を参照してください。有効なリポジトリでパッケージを検索してインストールする方法については、「AL2 インスタンスでソフトウェアパッケージを検索してインストールする」を参照してください。

リポジトリに保管されているソフトウェアパッケージで、すべてのソフトウェアが利用できるわけではありません。一部のソフトウェアは、そのソースコードからインスタンスでコンパイルする必要があります。詳細については、「AL2 インスタンスでソフトウェアをコンパイルする準備をする」を参照してください。

AL2 インスタンスは、yum パッケージマネージャーを使用してソフトウェアを管理します。yum パッケージマネージャはソフトウェアをインストール、削除、更新し、各パッケージのすべての依存関係を管理できます。

内容

AL2 インスタンスのインスタンスソフトウェアを更新する

ソフトウェアは、最新の状態に維持することが重要です。Linux ディストリビューションのパッケージは頻繁に更新されます。これにより、バグが修正され、機能が追加されて、セキュリティ上の弱点

に対する防御措置が行われます。最初に Amazon Linux インスタンスを起動して接続する際、セキュリティ上の目的から、ソフトウェアパッケージを更新するように促すメッセージが表示される場合があります。このセクションでは、システム全体またはパッケージを 1 つだけ更新する方法を紹介します。

この情報は AL2 に適用されます。AL2023 の詳細については、Amazon Linux 2AL2023の「AL2023 でのパッケージとオペレーティングシステムの更新の管理」を参照してください。

AL2 の変更と更新の詳細については、AL2 リリースノート」を参照してください。

AL2023 への変更と更新の詳細については、「AL2023 リリースノート」を参照してください。

> **Important**
>
> Amazon Linux 2 AMI を使用する EC2 インスタンスを IPv6 専用サブネットに起動した場合は、インスタンスに接続して `sudo amazon-linux-https disable` を実行する必要があります。これにより、AL2 インスタンスが http パッチサービスを使用して、IPv6 経由で S3 の yum リポジトリに接続できるようになります。

AL2 インスタンスのすべてのパッケージを更新するには

1. (オプション) シェルウィンドウで screen セッションを開始します。時折、ネットワークが遮断され、インスタンスへの SSH 接続が切断されることがあります。この状態が長時間におよぶソフトウェア更新中に発生した場合、インスタンスは混乱した状態になりますが、その復元は可能です。screen セッションを開始しておけば、接続が遮断された場合でも更新を続行でき、後で問題なくセッションに再接続できます。

   a. セッションを開始するには screen コマンドを実行します。

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

   b. セッションが中断された場合、インスタンスにログインし直し、利用できる画面を表示します。

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

   c. screen -r コマンドと前のコマンドのプロセス ID を使用して、画面に再接続します。

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

d. screen の使用が終わったら、exit コマンドを使用してセッションを閉じます。

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

2. yum update コマンドを実行します。オプションで、--security フラグを追加すれば、セキュリティ更新のみを適用できます。

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

3. 表示されたパッケージを確認したら、「**y**」と入力して Enter キーを押し、この更新を受け入れます。システムのパッケージをすべて更新するには数分かかります。実行中、yum 出力には更新のステータスが表示されます。
4. (オプション) インスタンスを再起動して、更新から最新のパッケージとライブラリを使用していることを確認します。カーネルの更新は、再起動が発生するまでロードされません。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、更新を取得するにはサービスの再起動で十分かもしれませんが、システムを再起動することで、それ以前のすべてのパッケージとライブラリの更新を確実に完了できます。

AL2 インスタンスで 1 つのパッケージを更新するには

システム全体ではなく、1 つのパッケージ (とその依存関係) を更新するには、この手順を使用します。

1. 更新するパッケージの名前を指定した yum update コマンドを実行します。

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

2. 表示されたパッケージ情報を確認したら、「**y**」と入力して Enter キーを押し、この更新を受け入れます。時折、解決する必要があるパッケージ依存関係ある場合、リストには複数のパッケージがあります。実行中、yum 出力には更新のステータスが表示されます。
3. (オプション) インスタンスを再起動して、更新から最新のパッケージとライブラリを使用していることを確認します。カーネルの更新は、再起動が発生するまでロードされません。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、更新を取得するにはサービスの再起動で十分かもしれませんが、システムを再起動することで、それ以前のすべてのパッケージとライブラリの更新を確実に完了できます。

AL2 インスタンスにリポジトリを追加する

この情報は AL2 に適用されます。AL2023 の詳細については、「Amazon Linux 2023 ユーザーガイド」の「AL2023 のバージョニングされたリポジトリによる確定的なアップグレード」を参照してください。

デフォルトでは、AL2 インスタンスは、次のリポジトリを有効にして起動します。

- amzn2-core
- amzn2extra-docker

これらのリポジトリには、Amazon Web Services が更新するさまざまなパッケージが用意されていますが、別のリポジトリで、インストールしたいパッケージが見つかる可能性もあります。

yum 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/yum.conf ファイル、または *repository*.repo ディレクトリにあるお客様の /etc/yum.repos.d ファイルに、リポジトリ情報を追加する必要があります。これは手動で行えますが、ほとんどの yum 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。

既にインストールされている yum レポジトリを調べるには

次のコマンドで、インストール済みの yum リポジトリを表示します。

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

生成される出力には、インストール済みのリポジトリが一覧表示され、それぞれのステータスが報告されます。有効なリポジトリには、そこに含まれているパッケージの数が表示されます。

yum リポジトリを /etc/yum.repos.d に追加するには

1. .repo ファイルの場所を検索します。場所は、追加しているリポジトリによって異なります。この例では、.repo ファイルは、https://www.*example*.com/*repository*.repo にあります。
2. yum-config-manager コマンドを使用してリポジトリを追加します。

```
[ec2-user ~]$ sudo yum-config-manager --add-repo https://
www.example.com/repository.repo
Loaded plugins: priorities, update-motd, upgrade-helper
adding repo from: https://www.example.com/repository.repo
```

```
grabbing file https://www.example.com/repository.repo to /etc/
yum.repos.d/repository.repo
repository.repo                                          | 4.0 kB     00:00
repo saved to /etc/yum.repos.d/repository.repo
```

リポジトリをインストールしたら、次の手順で説明するように有効にする必要があります。

yum リポジトリを /etc/yum.repos.d で有効にするには

yum-config-manager フラグを付けて `--enable` *repository* コマンドを使用します。次のコマンドを使用すると、Fedora プロジェクトの Extra Packages for Enterprise Linux (EPEL) リポジトリが有効になります。デフォルトでは、このリポジトリは Amazon Linux AMI インスタンスの `/etc/yum.repos.d` にありますが、有効になっていません。

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

詳細、およびこのパッケージの最新バージョンをダウンロードするには、https://fedoraproject.org/wiki/EPEL を参照してください。

AL2 インスタンスでソフトウェアパッケージを検索してインストールする

パッケージ管理ツールを使用して、ソフトウェアパッケージを検索してインストールできます。Amazon Linux 2 では、デフォルトのソフトウェアパッケージ管理ツールは ですYUM。AL2023 では、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。詳細については、「Amazon Linux 2023 ユーザーガイド」の「パッケージ管理ツール」を参照してください。

AL2 インスタンスでソフトウェアパッケージを検索する

yum search コマンドを使用すると、設定したリポジトリで利用できるパッケージの説明を検索できます。これは特に、インストールするパッケージの正確な名前がわからない場合に便利です。キーワード検索をコマンドに追加します。複数の単語を検索するには、引用符で検索クエリを囲みます。

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

以下は出力の例です。

```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
============================== N/S matched: find ===============================
findutils.x86_64 : The GNU versions of find utilities (find and xargs)
gedit-plugin-findinfiles.x86_64 : gedit findinfiles plugin
```

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

引用符で囲まれた複数の単語検索クエリは、正確なクエリに一致する結果のみを返します。予想されたパッケージが表示されない場合、キーワードを 1 つに絞って検索し、結果をスキャンします。キーワードの同義語を試して、検索の幅を広げることもできます。

AL2 のパッケージの詳細については、以下を参照してください。

- AL2 Extras ライブラリ
- パッケージリポジトリ

AL2 インスタンスにソフトウェアパッケージをインストールする

AL2 では、yum パッケージ管理ツールは、有効なすべてのリポジトリでさまざまなソフトウェアパッケージを検索し、ソフトウェアのインストールプロセスの依存関係を処理します。AL2023 にソフトウェアパッケージをインストールする方法については、「Amazon Linux 2023 ユーザーガイド」の「パッケージとオペレーティングシステムの更新の管理」を参照してください。

リポジトリからパッケージをインストールするには

yum install ***package*** コマンドを使用します。この際、*package* はインストールするソフトウェアの名前に置き換えます。例えば、links テキストベースウェブブラウザをインストールするには、次のコマンドを入力します。

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

ダウンロードした RPM パッケージファイルをインストールするには

また、yum install を使用して、インターネットからダウンロードした RPM パッケージファイルをインストールすることもできます。その場合には、リポジトリのパッケージ名の代わりに、RPM ファイルのパス名をインストールコマンドに追加します。

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

インストールされているパッケージを一覧表示するには

インスタンスにインストールされているパッケージを一覧表示するには、次のコマンドを使用します。

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

AL2 インスタンスでソフトウェアをコンパイルする準備をする

オープンソースのソフトウェアは、事前コンパイルされていないインターネットで使用できます。これらは、パッケージリポジトリからダウンロードできます。入手したソフトウェアパッケージがソースコードであり、自分でコンパイルする必要があると判明することがあります。システムが AL2 および Amazon Linux でソフトウェアをコンパイルできるようにするには、、make、 gccなどのいくつかの開発ツールをインストールする必要がありますautoconf。

ソフトウェアのコンパイルはすべての Amazon EC2 インスタンスで必要なタスクではないため、そのようなツールはデフォルトでインストールされていません。ただし、「Development Tools」という名前のパッケージグループで利用でき、yum groupinstall コマンドでインスタンスに簡単に追加されます。

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

ソフトウェアのソースコードパッケージは、多くの場合、tarball と呼ばれる圧縮アーカイブファイルの形で (https://github.com/ や http://sourceforge.net/ などのウェブサイトから) ダウンロードできます。通常、これらの tarball には `.tar.gz` というファイル拡張子が付いています。これらのアーカイブは tar コマンドで解凍できます。

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

ソースコードパッケージを解凍したら、ソースコードディレクトリで README ファイルまたは INSTALL ファイルを探します。これらのファイルに、ソースコードのコンパイルとインストールに関する詳細な指示があります。

Amazon Linux パッケージのソースコードを取得するには

Amazon Web Services は、保守管理されているパッケージのソースコードを提供します。yumdownloader --source コマンドを使用して、インストールされているパッケージのソースコードをダウンロードできます。

yumdownloader --source ***package*** コマンドを実行して、*package* のソースコードをダウンロードします。例えば、htop パッケージのソースコードをダウンロードするには、次のコマンドを入力します。

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

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

ソース RPM の場所は、コマンドを実行したディレクトリにあります。

Amazon EC2 AL2 インスタンスのプロセッサ状態制御

C ステートはアイドル時のコアのスリープレベルを制御します。C ステートはC0 (コアがアクティブで、命令を実行している最も浅い状態) から始まる番号が付けられ、C6 (コアの電源がオフになっている最も深いアイドル状態) まで移行します。

P ステートはコアに希望するパフォーマンス (CPU 周波数) を制御します。P ステートはP0 (コアが Intel Turbo Boost Technology を使用して可能であれば周波数を上げることができる最高パフォーマンスの設定) から始まる番号が付けられ、P1 (最大限のベースライン周波数をリクエストする P ステート) から P15 (最小限の周波数) まで移行します。

プロセッサのパフォーマンスの安定性を向上させたり、レイテンシーを減らしたり、インスタンスを特定のワークロード用に調整するために、C ステートまたは P ステートの設定を変更したいと思う場合があるかもしれません。デフォルトの C ステートおよび P ステートの設定はほとんどの作業負荷に対して最適なパフォーマンスを提供します。ただし、アプリケーションにおいて、より高いシングルコアまたはデュアルコアの周波数でレイテンシーを軽減したい場合、またはバースト的な Turbo Boost 周波数よりも低い周波数でより安定したパフォーマンスを維持することを優先する場合、これらのインスタンスで利用可能な C ステートまたは P ステートを試みることを考慮してください。

オペレーティングシステムがプロセッサの C ステートと P ステートを制御する機能を提供する Amazon EC2 インスタンスタイプについては、Amazon EC2 ユーザーガイド」の「Amazon EC2 インスタンスのプロセッサステートコントロール」を参照してください。 Amazon EC2

以下のセクションでは、プロセッサのさまざまなステート設定と、設定の効果をモニタリングする方法について説明します。これらの手順は、Amazon Linux 用に記述され、Amazon Linux に適用されますが、Linux カーネルバージョン 3.9 以降の他の Linux ディストリビューションでも機能する可能性があります。

> **Note**
>
> このページの例では、以下を使用しています。
>
> - プロセッサの周波数と C ステート情報を表示する turbostat ユーティリティ。turbostat ユーティリティは Amazon Linux でデフォルトで使用できます。
> - ワークロードをシミュレートする stress コマンド。stress をインストールするには、まず sudo amazon-linux-extras install epel を実行して EPEL リポジトリを有効にし、次に sudo yum install -y stress を実行します。
>
> 出力に C ステート情報が表示されない場合は、コマンド (--debug) に sudo turbostat --debug stress ***<options>*** オプションを含めます。

コンテンツ

最大 Turbo Boost 周波数による最高パフォーマンス

これは、Amazon Linux AMI のデフォルトのプロセッサのステート制御設定であり、ほとんどのワークロードにお勧めします。この設定では、変動性を抑え、最高パフォーマンスを実現します。非アクティブなコアは深いスリープ状態になることができるため、シングルまたはデュアルコアプロセスが Turbo Boost の潜在能力を最大限に引き出すために必要な発熱量の余裕を実現できます。

次の例は、2 個のコアでアクティブに処理を実行する `c4.8xlarge` インスタンスが Turbo Boost の最大周波数に到達した状況を示しています。

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

この例では、vCPU 21 と 28 が最大 Turbo Boost 周波数で実行され、他のコアは C6 スリープ状態になることで電力を節約し、実行中のコアに電力と発熱の余裕を持たせています。vCPU 3 と 10 (それぞれ vCPU 21 および 28 とプロセッサコアを共有する) は C1 ステートであり、命令を待っています。

以下の例では、18 個のコアはすべてアクティブに処理を実行しているため、Turbo Boost の最大周波数のための余裕はありませんが、すべてのコアが「all core Turbo Boost」の速度である 3.2 GHz で実行されています。

```
[ec2-user ~]$ sudo turbostat stress -c 36 -t 10
```

```
stress: info: [30685] dispatching hogs: 36 cpu, 0 io, 0 vm, 0 hdd
stress: info: [30685] successful run completed in 10s
pk cor CPU    %c0  GHz  TSC SMI    %c1    %c3    %c6    %c7   %pc2   %pc3   %pc6   %pc7
  Pkg_W RAM_W PKG_% RAM_%
           99.27 3.20 2.90   0   0.26   0.00   0.47   0.00   0.00   0.00   0.00   0.00
 228.59 31.33 199.26  0.00
 0   0   0  99.08 3.20 2.90   0   0.27   0.01   0.64   0.00   0.00   0.00   0.00   0.00
 114.69 18.55 99.32  0.00
 0   0  18  98.74 3.20 2.90   0   0.62
 0   1   1  99.14 3.20 2.90   0   0.09   0.00   0.76   0.00
 0   1  19  98.75 3.20 2.90   0   0.49
 0   2   2  99.07 3.20 2.90   0   0.10   0.02   0.81   0.00
 0   2  20  98.73 3.20 2.90   0   0.44
 0   3   3  99.02 3.20 2.90   0   0.24   0.00   0.74   0.00
 0   3  21  99.13 3.20 2.90   0   0.13
 0   4   4  99.26 3.20 2.90   0   0.09   0.00   0.65   0.00
 0   4  22  98.68 3.20 2.90   0   0.67
 0   5   5  99.19 3.20 2.90   0   0.08   0.00   0.73   0.00
 0   5  23  98.58 3.20 2.90   0   0.69
 0   6   6  99.01 3.20 2.90   0   0.11   0.00   0.89   0.00
 0   6  24  98.72 3.20 2.90   0   0.39
...
```

深い C ステートの制限による高パフォーマンスと低レイテンシー

C ステートは非アクティブ時のコアのスリープレベルを制御します。C ステートを制御して、システムのレイテンシーとパフォーマンスを調整することができます。コアをスリープ状態にするには時間がかかります。また、スリープ状態のコアによって、別のコアが高い周波数で動作するための余裕が生まれますが、そのスリープ状態にあるコアが再び稼働し処理を実行するのにも時間がかかります。例えば、ネットワークパケットの中断を処理するように割り当てられたコアがスリープ状態である場合、その中断の処理に遅延が生じる可能性があります。より深い C ステートを使用しないようにシステムを設定できます。これにより、プロセッサの応答のレイテンシーは減少しますが、他のコアの Turbo Boost 用の余裕も減少します。

深いスリープ状態を無効にする一般的なシナリオとして、Redis データベースアプリケーションがあります。このアプリケーションは、最速のクエリ応答時間を実現するために、データベースをシステムメモリ内に格納します。

AL2 で深いスリープ状態を制限するには

1. 適切なエディタで `/etc/default/grub` ファイルを開きます。

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

2. GRUB_CMDLINE_LINUX_DEFAULT 行を編集し intel_idle.max_cstate=1 と processor.max_cstate=1 のオプションを追加します。これにより、アイドル状態にあるコアの最も深い C ステートとして C1 を設定します。

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

   intel_idle.max_cstate=1 オプションでは、インテルベースのインスタンスで C ステート制限が設定され、また、processor.max_cstate=1 オプションでは、AMD ベースのインスタンスで C ステート制限が設定されます。両方のオプションを設定に追加しておくと安心です。これにより、インテルと AMD のインスタンスに対し、共通の方法で目的の動作を設定することができます。

3. ファイルを保存し、エディタを終了します。
4. 起動設定を再構築するには、次のコマンドを実行します。

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

5. 新しい kernel オプションを有効にするためにインスタンスを再起動します。

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

Amazon Linux AMI で深いスリープ状態を制限するには

1. 適切なエディタで /boot/grub/grub.conf ファイルを開きます。

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

2. 最初のエントリの kernel 行を編集して intel_idle.max_cstate=1 と processor.max_cstate=1 オプションを追加し、アイドル状態のコアの最も深い C ステートに C1 を設定します。

```
# created by imagebuilder
default=0
```

```
timeout=1
hiddenmenu

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

`intel_idle.max_cstate=1` オプションでは、インテルベースのインスタンスで C ステート制限が設定され、また、`processor.max_cstate=1` オプションでは、AMD ベースのインスタンスで C ステート制限が設定されます。両方のオプションを設定に追加しておくと安心です。これにより、インテルと AMD のインスタンスに対し、共通の方法で目的の動作を設定することができます。

3. ファイルを保存し、エディタを終了します。
4. 新しい kernel オプションを有効にするためにインスタンスを再起動します。

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

次の例では、2 つのコアが「all core Turbo Boost」コア周波数でアクティブに処理を実行している `c4.8xlarge` インスタンスを示します。

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

```
...
```

この例では、vCPU 19 および 28 のコアは 3.2 GHz で実行中であり、その他のコアは C1 C ステートで、命令を待機しています。稼働中のコアは Turbo Boost の最大周波数には到達していませんが、非アクティブなコアはより深い C6 C ステートにある場合と比べて、新しいリクエストに迅速に応答します。

変動性が最も低いベースラインパフォーマンス

P ステートによってプロセッサの周波数の変動性を抑制することができます。P ステートはコアに希望するパフォーマンス (CPU 周波数) を制御します。ほとんどのワークロードでは、Turbo Boost をリクエストする、P0 でパフォーマンスが向上します。ただし、Turbo Boost 周波数が有効であるときに発生する可能性があるバースト的なパフォーマンスではなく、安定したパフォーマンスになるようにシステムを調整することもできます。

Intel Advanced Vector Extensions (AVX または AVX2) のワークロードは低い周波数でもパフォーマンスに優れ、AVX 命令はより多くの処理能力を使用できます。Turbo Boost を無効にして、プロセッサをより低い周波数で実行すると、使用される処理能力が抑えられ、より安定した速度が維持されます。AVX のインスタンス設定とワークロードの最適化の詳細については、インテルのウェブサイトを参照してください。

CPU がアイドル状態のドライバは P ステートを制御します。最近の世代の CPU には、以下のようにカーネルレベルでの対応が可能な、更新された CPU アイドルドライバが必要です。

- Linux カーネルバージョン 6.1 以降 – Intel Granite Rapids (R8i など) をサポート
- Linux カーネルバージョン 5.10 以降 – AMD Milan をサポート (M6a など)
- Linux カーネルバージョン 5.6 以降 – Intel Icelake (M6i など) をサポート

実行中のシステムのカーネルが CPU を認識するかどうかを確認するには、次のコマンドを実行します。

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

このコマンドの出力により、サポートが不十分であることを確認した場合は、カーネルをアップグレードすることをお勧めします。

このセクションでは、深いスリープ状態を制限し、(P1 P ステートをリクエストすることにより) Turbo Boost を無効にすることで、これらのタイプのワークロードに対して、低レイテンシーを提供し、プロセッサ速度の変動性を最低限に抑える方法を説明します。

深いスリープ状態を制限し、AL2 で Turbo Boost を無効にするには

1. 適切なエディタで /etc/default/grub ファイルを開きます。

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

2. GRUB_CMDLINE_LINUX_DEFAULT 行を編集し intel_idle.max_cstate=1 と processor.max_cstate=1 のオプションを追加します。これにより、アイドル状態にあるコアの最も深い C ステートとして C1 を設定します。

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

   intel_idle.max_cstate=1 オプションでは、インテルベースのインスタンスで C ステート制限が設定され、また、processor.max_cstate=1 オプションでは、AMD ベースのインスタンスで C ステート制限が設定されます。両方のオプションを設定に追加しておくと安心です。これにより、インテルと AMD のインスタンスに対し、共通の方法で目的の動作を設定することができます。

3. ファイルを保存し、エディタを終了します。
4. 起動設定を再構築するには、次のコマンドを実行します。

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

5. 新しい kernel オプションを有効にするためにインスタンスを再起動します。

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

6. P1 P ステートによってプロセッサ速度の変動性を抑える必要がある場合は、次のコマンドを実行して Turbo Boost を無効にします。

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

7. ワークロードが終了したら、次のコマンドで Turbo Boost を再度有効にすることができます。

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

Amazon Linux AMI で深いスリープ状態を制限し、Turbo Boost を無効にするには

1. 適切なエディタで /boot/grub/grub.conf ファイルを開きます。

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

2. 最初のエントリの kernel 行を編集して intel_idle.max_cstate=1 と processor.max_cstate=1 オプションを追加し、アイドル状態のコアの最も深い C ステートに C1 を設定します。

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

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

   intel_idle.max_cstate=1 オプションでは、インテルベースのインスタンスで C ステート制限が設定され、また、processor.max_cstate=1 オプションでは、AMD ベースのインスタンスで C ステート制限が設定されます。両方のオプションを設定に追加しておくと安心です。これにより、インテルと AMD のインスタンスに対し、共通の方法で目的の動作を設定することができます。

3. ファイルを保存し、エディタを終了します。
4. 新しい kernel オプションを有効にするためにインスタンスを再起動します。

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

5. P1 P ステートによってプロセッサ速度の変動性を抑える必要がある場合は、次のコマンドを実行して Turbo Boost を無効にします。

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

6. ワークロードが終了したら、次のコマンドで Turbo Boost を再度有効にすることができます。

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

次の例では、2 つの vCPU が、Turbo Boost を使用せずに、ベースラインコア周波数でアクティブに処理を実行している c4.8xlarge インスタンスを示します。

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [5389] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [5389] successful run completed in 10s
pk cor CPU    %c0  GHz  TSC SMI    %c1    %c3    %c6    %c7   %pc2   %pc3   %pc6   %pc7
  Pkg_W RAM_W PKG_% RAM_%
            5.59 2.90 2.90   0  94.41   0.00   0.00   0.00   0.00   0.00   0.00   0.00
 128.48 33.54 200.00  0.00
 0   0   0   0.04 2.90 2.90   0  99.96   0.00   0.00   0.00   0.00   0.00   0.00   0.00
  65.33 19.02 100.00  0.00
 0   0  18   0.04 2.90 2.90   0  99.96
 0   1   1   0.05 2.90 2.90   0  99.95   0.00   0.00   0.00
 0   1  19   0.04 2.90 2.90   0  99.96
 0   2   2   0.04 2.90 2.90   0  99.96   0.00   0.00   0.00
 0   2  20   0.04 2.90 2.90   0  99.96
 0   3   3   0.05 2.90 2.90   0  99.95   0.00   0.00   0.00
 0   3  21  99.95 2.90 2.90   0   0.05
...
 1   1  28  99.92 2.90 2.90   0   0.08
 1   2  11   0.06 2.90 2.90   0  99.94   0.00   0.00   0.00
 1   2  29   0.05 2.90 2.90   0  99.95
```

vCPU 21 および 28 のコアは、ベースラインプロセッサ速度の 2.9 GHz でアクティブに処理を実行し、すべての非アクティブなコアも、C1 C ステートのベースライン速度で実行され、すぐに命令を受け付けることができます。

AL2 の I/O スケジューラ

I/O スケジューラは、I/O リクエストをソートおよびマージし、処理される順序を決定する Linux オペレーティングシステムの一部です。

I/O スケジューラは、シーク時間が長くなる可能性があり、コロケーションされた要求をマージするのが最適である磁気ハードドライブなどのデバイスにとって特に有益です。I/O スケジューラは、ソリッドステートデバイスや仮想化環境ではあまり影響しません。これは、ソリッドステートデバイス

の場合、シーケンシャルアクセスとランダムアクセスが異なることがなく、仮想化環境ではホストが独自のスケジューリング層を提供するためです。

このトピックでは、Amazon Linux I/O スケジューラについて説明します。他の Linux ディストリビューションで使用される I/O スケジューラの詳細については、それぞれのドキュメントを参照してください。

トピック

サポートされているスケジューラ

Amazon Linux では、次の I/O スケジューラがサポートされています。

- `deadline` — 期限日 I/O スケジューラは I/O リクエストをソートし、最も効率的な順序で処理します。これにより、各 I/O リクエストの開始時間が保証されます。また、長い間保留中であった I/O リクエストの優先度も高くなります。
- `cfq` — 完全公平キュー(CFQ) I/O スケジューラは、プロセス間で I/O リソースを公平に割り当てようとします。I/O リクエストをソートし、プロセスごとのキューに挿入します。
- noop — オペレーションなし (noop) I/O スケジューラは、すべての I/O リクエストを FIFO キューに挿入し、それらを単一のリクエストにマージします。このスケジューラは、リクエストの並べ替えを行いません。

デフォルトスケジューラ

[オペレーションなし (noop)] は Amazon Linux のデフォルトの I/O スケジューラです。このスケジューラは、次の理由で使用されます。

- 多くのインスタンスタイプは、基盤となるホストがインスタンスのスケジュールを実行する仮想化デバイスを使用します。
- ソリッドステートデバイスは、I/O スケジューラの利点の影響が少ない多くのインスタンスタイプで使用されます。
- これは侵襲性の低い I/O スケジューラであり、必要に応じてカスタマイズできます。

スケジューラを変更する

I/O スケジューラを変更すると、スケジューラによって一定時間内に完了する I/O リクエストが増減するかどうかに基づいて、パフォーマンスが向上または低下することがあります。これは、ワークロード、使用されているインスタンスタイプの生成、アクセスされるデバイスのタイプに大きく依存します。使用する I/O スケジューラを変更する場合は、iotop などのツールを使用し、I/O パフォーマンスを測定し、その変更がユースケースにとって有益かどうかを判断することをお勧めします。

デバイスの I/O スケジューラは、nvme0n1 を例にした次のコマンドで確認できます。インスタンスの /sys/block に表示されているデバイスを使って、次のコマンドの nvme0n1 を置き換えます。

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

デバイスの I/O スケジューラを設定するには、次のコマンドを使用します。

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

例えば、*xvda* デバイスの I/O スケジューラを noop から cfq に設定するには、次のコマンドを使用します。

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

AL2 インスタンスのホスト名を変更する

プライベート VPC 内でインスタンスを起動すると、Amazon EC2 によってゲスト OS ホスト名が割り当てられます。Amazon EC2 によって割り当てられるホスト名のタイプは、サブネット設定によって異なります。EC2 ホスト名の詳細については、Amazon EC2 ユーザーガイド」の「Amazon EC2 インスタンスのホスト名タイプ」を参照してください。 Amazon EC2

IPv4 アドレスで IP ベースの命名を使用するように構成された EC2 インスタンスの典型的な Amazon EC2 のプライベート DNS 名は、ip-12-34-56-78.us-west-2.compute.internal のような形式になります。この名前は内部ドメイン、サービス (この例では、compute)、リージョン、そしてプライベート IPv4 アドレスで構成されます。インスタンスにログインしたとき、このホスト名の一部がシェルプロンプトで表示されます (ip-12-34-56-78 など)。Amazon EC2 インスタンスを停止し、再起動するたびに (Elastic IP アドレスを使用していない限り)、パブリック IPv4 アドレスが変わり、パブリック DNS 名、システムホスト名、シェルプロンプトも変わります。

> **Important**
>
> この情報は、Amazon Linux に適用されます。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。

システムホスト名の変更

インスタンスの IP アドレスにパブリック DNS 名を登録している場合 (`webserver.mydomain.com` など)、インスタンスがそのドメインに含まれているものとして識別されるように、システムホスト名を設定できます。また、シェルプロンプトが変更され、 によって指定されたホスト名ではなく、この名前の最初の部分が表示されます AWS (例: `ip-12-34-56-78`)。パブリック DNS 名を登録していない場合でもホスト名は変更できますが、プロセスが少し違います。

ホスト名の更新内容を維持するには、`preserve_hostname` cloud-init 設定が `true` に設定されていることを確認してください。この設定を編集または追加するには、次のコマンドを実行します。

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

`preserve_hostname` 設定が一覧表示されない場合は、ファイルの末尾に次のテキスト行を追加します。

```
preserve_hostname: true
```

システムホスト名をパブリック DNS 名に変更するには

パブリック DNS 名を登録している場合、この手順を行います。

1. - AL2 の場合: hostnamectl コマンドを使用して、完全修飾ドメイン名 (など) を反映するようにホスト名を設定します**webserver.mydomain.com**。

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

   - Amazon Linux AMI の場合: インスタンスで、お好みのテキストエディタを使用して `/etc/sysconfig/network` 設定ファイルを開き、`HOSTNAME` エントリを変更して、完全修飾ドメイン名 (**webserver.mydomain.com** など) を反映させます。

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

2. インスタンスを再起動し、新しいホスト名を取得します。

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

または、Amazon EC2 コンソールを使用して再起動することもできます ([Instances (インスタンス)] ページでインスタンスを選択し、[Instance state (インスタンスの状態)]、[Reboot instance (インスタンスの再起動)] の順に選択します)。

3. インスタンスにログインして、ホスト名が更新されていることを確認します。メッセージには、新しいホスト名が表示されるはずです (最初の「.」まで)。hostname コマンドで完全修飾ドメイン名が表示されます。

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

パブリック DNS 名なしでシステムホスト名を変更するには

1. • AL2 の場合: hostnamectl コマンドを使用して、目的のシステムホスト名 (など) を反映するようにホスト名を設定します**webserver**。

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

   • Amazon Linux AMI の場合: インスタンスで、お好みのテキストエディタで /etc/sysconfig/network 設定ファイルを開き、HOSTNAME エントリを変更して、希望するシステムホスト名を反映させます (例: **webserver**)。

```
HOSTNAME=webserver.localdomain
```

2. お好みのテキストエディタで /etc/hosts ファイルを開き、下の例と一致する **127.0.0.1** で始まるエントリを変更します。ホスト名は自分のホスト名に置換します。

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

3. インスタンスを再起動し、新しいホスト名を取得します。

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

または、Amazon EC2 コンソールを使用して再起動することもできます ([Instances (インスタンス)] ページでインスタンスを選択し、[Instance state (インスタンスの状態)]、[Reboot instance (インスタンスの再起動)] の順に選択します)。

4. インスタンスにログインして、ホスト名が更新されていることを確認します。メッセージには、新しいホスト名が表示されるはずです (最初の「.」まで)。hostname コマンドで完全修飾ドメイン名が表示されます。

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

また、ユーザーデータを指定してインスタンスを設定するなど、よりプログラム的なソリューションを実装することもできます。インスタンスが Auto Scaling グループの一部である場合、ライフサイクルフックを使用してユーザーデータを定義できます。詳細については、「AWS CloudFormation ユーザーガイド」の「Run commands on your Linux instance at launch」(起動時に Linux インスタンスでコマンドを実行する) および「Lifecycle hook for instance launch」(インスタンス起動のライフサイクルフック) を参照してください。

ホスト名に影響を与えずにシェルプロンプトを変更する

インスタンスのホスト名を変更せずに、 が提供するプライベート名 (など**webserver**) よりも便利なシステム名 AWS (などip-12-34-56-78) を表示する場合は、シェルプロンプト設定ファイルを編集して、ホスト名の代わりにシステムニックネームを表示できます。

シェルプロンプトをホストニックネームに変更するには

1. /etc/profile.d で、NICKNAME と呼ばれる環境変数を設定するファイルを作成して、シェルプロンプトに表示する値を設定します。例えば、システムニックネームを **webserver** に設定するには、次のコマンドを実行します。

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

2. お好みのテキストエディタ (/etc/bashrc や /etc/bash.bashrc など) で、vim (Red Hat) または nano (Debian/Ubuntu) ファイルを開きます。エディタのコマンドで sudo を使用する必要があります。/etc/bashrc および /etc/bash.bashrc は root が所有するためです。
3. ファイルを編集し、ホスト名の代わりにニックネームを表示するようにシェルプロンプト変数 (PS1) を変更します。/etc/bashrc または /etc/bash.bashrc でシェルプロンプトを設定する次の行を見つけます (以下には、コンテキストを示すため前後の行も表示されています。["$PS1" で始まる行を探してください)。

```
  # Turn on checkwinsize
```

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

その行の \h (hostname を表す記号) を NICKNAME 変数の値に変更します。

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

4. (オプション) シェルウィンドウのタイトルを新しいニックネームに設定するには、次の手順を完了します。

   a. /etc/sysconfig/bash-prompt-xterm という名前のファイルを作成します。

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

   b. 次のコマンドを使用して、ファイルを実行可能にします。

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

   c. お好みのテキストエディタ (/etc/sysconfig/bash-prompt-xterm や vim など) で、nano ファイルを開きます。エディタのコマンドで sudo を使用する必要があります。/etc/sysconfig/bash-prompt-xterm は root が所有するためです。

   d. 次の行をファイルに追加します。

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

5. ログアウトしてから再度ログインし、新しいニックネーム値を取得します。

他の Linux ディストリビューションのホスト名の変更

このページの手順は、Amazon Linux のみで使用するためのものです。他の Linux ディストリビューションの詳細については、各ドキュメントおよび次の記事を参照してください。

- RHEL 7 または Centos 7 を実行するプライベート Amazon EC2 インスタンスに静的ホスト名を割り当てる方法を教えてください。

AL2 インスタンスで動的 DNS を設定する

EC2 インスタンスを起動すると、パブリック IP アドレスとパブリックドメインネームシステム (DNS) が割り当てられます。それらを使用してインターネットからインスタンスにアクセスできます。Amazon Web Services ドメインには数多くのホストが存在するため、これらのパブリック名はそれぞれの名前を一意にするために、かなり長くする必要があります。一般的な Amazon EC2 パブリック DNS 名は`ec2-12-34-56-78.us-west-2.compute.amazonaws.com`、Amazon Web Services ドメイン、サービス (この場合は `compute`)、 AWS リージョン、およびパブリック IP アドレスの形式で構成される のようになります。

動的 DNS サービスはそのドメイン領域内でカスタムの DNS ホスト名を提供します。この名前は覚えやすく、ホストのユースケースとの関連性が高くなっています。また、これらのサービスは一部は無料で提供されています。Amazon EC2 では動的 DNS プロバイダを利用できます。また、インスタンスを起動するたびに、パブリック DNS 名に関連付けられている IP アドレスを更新するようにインスタンスを設定できます。プロバイダは数多く存在します。また、プロバイダを選択し、それぞれのプロバイダで名前を登録する方法については本ガイドの範囲外です。

Amazon EC2 で動的 DNS を使用するには

1. 動的 DNS サービスプロバイダにサインアップし、そのサービスでパブリック DNS 名を登録します。この手順では、noip.com/free の無料サービスを例として使用します。
2. 動的 DNS 更新クライアントを設定します。動的 DNS サービスプロバイダを選び、そのサービスでパブリック DNS 名を登録したら、その DNS 名をインスタンスの IP アドレスにポイントします。多くのプロバイダ (noip.com を含む) では、この操作をウェブサイトのアカウントページから手動で実行できるようにしています。ただし、ソフトウェア更新クライアントもサポートしています。EC2 インスタンスで更新クライアントが動作している場合は、シャットダウン後に再起動したときに IP アドレスが変わるたびに動的 DNS レコードが更新されます。この例では、noip2 クライアントをインストールします。このクライアントは、noip.comが提供するサービスで動作します。

    a. Extra Packages for Enterprise Linux (EPEL) リポジトリを有効にして、`noip2`クライアントにアクセスします。

    > **Note**
    >
    > AL2 インスタンスには、EPEL リポジトリの GPG キーとリポジトリ情報がデフォルトでインストールされています。詳細、およびこのパッケージの最新バージョ

ンをダウンロードするには、https://fedoraproject.org/wiki/EPEL を参照してください。

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

b. noip パッケージをインストールします。

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

c. 設定ファイルを作成します。メッセージが表示されたら、ログインおよびパスワード情報を入力して、その後に続く質問に答え、クライアントを設定します。

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

3. noip サービスを有効にします。

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

4. noip サービスを開始します。

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

このコマンドを使用すると、クライアントが起動します。クライアントは前に作成した設定ファイル (`/etc/no-ip2.conf`) を読み、選択したパブリック DNS 名の IP アドレスを更新します。

5. 更新クライアントが動的 DNS 名に正しい IP アドレスを設定したことを確認します。DNS レコードの更新には数分かかります。その後、この手順で設定したパブリック DNS 名により、SSH を使用してインスタンスに接続します。

AL2 の ec2-net-utils を使用してネットワークインターフェイスを設定する

Amazon Linux 2 AMIs には、ec2-net-utils と呼ばれる AWSによってインストールされた追加のスクリプトが含まれている場合があります。これらのスクリプトはオプションで、ネットワークインターフェイスの設定を自動化します。これらのスクリプトは AL2 でのみ使用できます。

> **Note**
>
> Amazon Linux 2023 の場合、 `amazon-ec2-net-utils`パッケージは `/run/systemd/network` ディレクトリにインターフェイス固有の設定を生成します。詳細については、「 Amazon Linux 2023 ユーザーガイド」の「ネットワーキングサービス」を参照してください。

まだインストールされていない場合は、次のコマンドを使用してパッケージを AL2 にインストールするか、インストール済みで追加の更新が利用可能な場合は更新します。

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

ec2-net-utils には、以下のコンポーネントが含まれます。

udev ルール (`/etc/udev/rules.d`)

実行中のインスタンスにネットワークインターフェイスがアタッチ、デタッチ、または再アタッチされたときに、そのネットワークインターフェイスを特定し、ホットプラグスクリプトが実行されることを確認します (`53-ec2-network-interfaces.rules`)。MAC アドレスをデバイス名にマッピングします (`75-persistent-net-generator.rules` を生成する `70-persistent-net.rules`)。

ホットプラグスクリプト

DHCP での使用に適したインターフェイス設定ファイルを生成します (`/etc/sysconfig/network-scripts/ifcfg-ethN`)。また、ルート設定ファイルも生成します (`/etc/sysconfig/network-scripts/route-ethN`)。

DHCP スクリプト

ネットワークインターフェイスが新しい DHCP リースを受け取るたびに、このスクリプトがインスタンスメタデータに対し、Elastic IP アドレスを求めるクエリを実行します。これにより、各 Elastic IP アドレスごとに、そのアドレスからのアウトバンドトラフィックが正しいネットワークインターフェイスを使用するよう、ルーティングポリシーデータベースにルールが追加されます。また、各プライベート IP アドレスを、セカンダリアドレスとしてネットワークインターフェイスに追加します。

ec2ifup ethN (/usr/sbin/)

標準の ifup の機能を拡張します。このスクリプトが設定ファイル ifcfg-ethN および route-ethN を書き換えた後、ifup を実行します。

ec2ifdown ethN (/usr/sbin/)

標準の ifdown の機能を拡張します。このスクリプトがルーティングポリシーデータベースからネットワークインターフェイスのルールをすべて削除した後、ifdown を実行します。

ec2ifscan (/usr/sbin/)

まだ設定されていないネットワークインターフェイスを探して、それらを設定します。

このスクリプトは、ec2-net-utils の初期リリースでは提供されていません。

ec2-net-utils によって生成された設定ファイルをリストするには、以下のコマンドを使用します。

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

オートメーションを無効にするには、対応する ifcfg-ethN ファイルに EC2SYNC=no を追加します。例えば、eth1 インターフェイスの自動化を無効にするには、以下のコマンドを使用します。

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

オートメーションを完全に無効にするには、次のコマンドを使用してパッケージを削除できます。

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

ユーザー提供カーネル

Amazon EC2 インスタンスでカスタムカーネルが必要な場合は、必要としているものに近い AMI を使用して開始し、インスタンスでカスタムカーネルをコンパイルして、新しいカーネルを参照するようにブートローダーを更新します。このプロセスは AMI が使用する仮想化タイプによって異なります。詳細については、Amazon EC2 ユーザーガイド」の「Linux AMI 仮想化タイプ」を参照してください。

内容

- AMIs の準仮想化 (PV-GRUB)

HVM AMIs (GRUB)

HVM インスタンスボリュームは実際の物理ディスクのように扱われます。起動プロセスは、パーティション分割ディスクとブートローダーを備えるベアメタルオペレーティングシステムの起動プロセスに似ています。ブートローダーでは、現在サポートされているすべての Linux ディストリビューションを使用できます。最も一般的なブートローダーは GRUB または GRUB2 です。

起動が遅くなるため、デフォルトでは GRUB はインスタンスのコンソールに出力を送信しません。詳細については、「Amazon EC2 ユーザーガイド」の「インスタンスコンソール出力」を参照してください。カスタムカーネルをインストールする場合は、GRUB 出力を有効にすることを検討してください。

フォールバックカーネルを指定する必要はありません。ただし、新しいカーネルをテストする場合は、フォールバックを設定することをお勧めします。GRUB では、新しいカーネルにエラーがあった場合に別のカーネルにフォールバックできます。フォールバックカーネルを設定すると、新しいカーネルが見つからない場合でも、インスタンスを起動できます。

Amazon Linux 用のレガシー GRUB では `/boot/grub/menu.lst` を使用します。GRUB2 for AL2 は を使用します`/etc/default/grub`。ブートローダーでデフォルトカーネルを更新する方法の詳細については、ご使用の Linux ディストリビューションのドキュメントを参照してください。

AMIs の準仮想化 (PV-GRUB)

準仮想化 (PV) 仮想化を使用する AMIs、起動プロセス中に PV-GRUB と呼ばれるシステムを使用します。PV-GRUB は、パッチが適用されたバージョンの GNU GRUB 0.97 を実行する準仮想化ブートローダーです。インスタンスを起動すると、PV-GRUB では起動プロセスが開始され、お客様のイメージの `menu.lst` ファイルが指定するカーネルがチェーンロードされます。

PV-GRUB は標準の `grub.conf` または `menu.lst` コマンドを認識しますこれにより、現在サポートされているすべての Linux ディストリビューションとともに利用できます。Ubuntu 10.04 LTS、Oracle Enterprise Linux、CentOS 5.x など、古いディストリビューションでは特別な「ec2」や「xen」カーネルパッケージが必要です。新しいディストリビューションでは、デフォルトのカーネルパッケージに必要なドライバーが含まれています。

最新の準仮想 AMI では、デフォルトで PV-GRUB AKI を使用します (Amazon EC2 Launch Wizard Quick Start メニューで利用できるすべての準仮想 Linux AMI が含まれています)。そのため、使用す

るカーネルにディストリビューションとの互換性がある場合、インスタンスで別のカーネルを使用するために必要な追加の手順はありません。インスタンスでカスタムカーネルを実行する最適な方法としては、必要としているものに近い AMI を使用して開始し、インタンスでカスタムカーネルをコンパイルして、そのカーネルを使用して起動するように `menu.lst` ファイルを変更します。

AMI のカーネルイメージが PV-GRUB AKI であることを確認できます。次の describe-images コマンドを実行して (ご使用のカーネルイメージ ID に置き換えます)、`Name` フィールドが `pv-grub` で始まっているかどうかを確認します。

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

コンテンツ

PV-GRUB の制約事項

PV-GRUB には次の制約事項があります。

- PV-GRUB の 64 ビットバージョンを使用して 32 ビットカーネルを起動したり、PV-GRUB の 32 ビットバージョンを使用して 64 ビットカーネルを起動したりすることはできません。
- PV-GRUB AKI の使用時には、Amazon ラムディスクイメージ (ARI) を指定できません。
- AWS は、PV-GRUB が EXT2, EXT3, EXT4、XFS、ReiserFS のファイルシステム形式で動作することをテストおよび検証しました。その他のファイルシステム形式では動作しない場合があります。
- PV-GRUB は、gzip、bzip2、lzo、xz 圧縮形式を利用して圧縮されたカーネルを起動できます。
- Cluster AMI は PV-GRUB をサポートせず、また、必要としません。完全ハードウェア仮想化 (HVM) が使用されるためです。準仮想インスタンスは PV-GRUB を使用して起動します。一方、HVM インスタンスボリュームは実際のディスクのように扱われ、その起動プロセスはパーティション分割ディスクとブートローダーを備えるベアメタルオペレーティングシステムの起動プロセスに似ています。
- PV-GRUB バージョン 1.03 以前では、GPT パーティショニングをサポートしません。MBR パーティショニングがサポートされています。

- Amazon Elastic Block Store (Amazon EBS) で Logical Volume Manager (LVM) を使用する場合、LVM の外側に別の起動パーティションが必要です。その場合、LVM で論理ボリュームを作成できます。

準仮想化 AMIs 向けの GRUB の設定

PV-GRUB を起動するには、GRUB `menu.lst` ファイルがイメージに含まれている必要があります。このファイルの最も一般的な場所は `/boot/grub/menu.lst` です。

次の例は、PV-GRUB AKI を使用して AMI を起動する `menu.lst` 設定ファイルです。この例では、Amazon Linux 2018.03 (この AMI の元のカーネル) と Vanilla Linux 4.16.4 (https://www.kernel.org/ の Vanilla Linux カーネルの新しいバージョン) の 2 つのカーネルエントリが選択できます。Vanilla エントリは、この AMI の元々のエントリからコピーされました。`kernel` と `initrd` パスは新しい場所に更新されました。`default 0` パラメータは、ブートローダーをそれが検出した最初のエントリ (この場合、Vanilla エントリ) にポイントします。`fallback 1` パラメータは、最初のエントリの起動に問題が発生した場合、次のエントリにブートローダーをポイントします。

```
default 0
fallback 1
timeout 0
hiddenmenu

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

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

`menu.lst` ファイルにフォールバックカーネルを指定する必要はありません。ただし、新しいカーネルをテストするときは、フォールバックを設定することをお勧めします。PV-GRUB では、新しいカーネルにエラーがあった場合に別のカーネルにフォールバックできます。フォールバックカーネルを設定すると、新しいカーネルが見つからない場合でもインスタンスを起動できます。

PV-GRUB は、次の場所で `menu.lst` をチェックします。その際、それが検出した最初の場所が利用されます。

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

PV-GRUB 1.03 以前では、このリストの最初の 2 つの場所うちの 1 つのみがチェックされることに注意してください。

Amazon PV-GRUB カーネルイメージ ID

PV-GRUB AKI は、アジアパシフィック (大阪) を除くすべての Amazon EC2 リージョンで利用できます。32 ビットと 64 ビットの両方のアーキテクチャタイプに AKI があります。最新の AMI では、デフォルトで PV-GRUB AKI が使用されます。

すべてのバージョンの PV-GRUB AKI がすべてのインスタンスタイプと互換性があるとは限らないため、常に最新バージョンの PV-GRUB AKI を使用することをお勧めします。次の describe-images コマンドを使用し、現在のリージョンの PV-GRUB AKI のリストを取得します。

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

PV-GRUB は、ap-southeast-2 リージョンで利用できる唯一の AKI です。このリージョンにコピーする AMI が、このリージョンで利用できる PV-GRUB のバージョンを使用していることを確認してください。

各リージョンの現在の AKI ID は次のとおりです。新しい AMI は、hd0 AKI を使用して登録します。

> **Note**
>
> hd00 AKI は、以前に利用可能であったリージョンでの下位互換性のために引き続き提供されます。

ap-northeast-1、アジアパシフィック (東京)

イメージ ID	イメージ名
aki-f975a998	pv-grub-hd0_1.05-i386.gz
aki-7077ab11	pv-grub-hd0_1.05-x86_64.gz

ap-southeast-1、アジアパシフィック (シンガポール) リージョン

イメージ ID	イメージ名
aki-17a40074	pv-grub-hd0_1.05-i386.gz
aki-73a50110	pv-grub-hd0_1.05-x86_64.gz

ap-southeast-2、アジアパシフィック (シドニー)

イメージ ID	イメージ名
aki-ba5665d9	pv-grub-hd0_1.05-i386.gz
aki-66506305	pv-grub-hd0_1.05-x86_64.gz

eu-central-1、欧州 (フランクフルト)

イメージ ID	イメージ名
aki-1419e57b	pv-grub-hd0_1.05-i386.gz
aki-931fe3fc	pv-grub-hd0_1.05-x86_64.gz

eu-west-1、欧州 (アイルランド)

イメージ ID	イメージ名
aki-1c9fd86f	pv-grub-hd0_1.05-i386.gz

イメージ ID	イメージ名
aki-dc9ed9af	pv-grub-hd0_1.05-x86_64.gz

sa-east-1、南米 (サンパウロ)

イメージ ID	イメージ名
aki-7cd34110	pv-grub-hd0_1.05-i386.gz
aki-912fbcfd	pv-grub-hd0_1.05-x86_64.gz

us-east-1、US East (N. Virginia)

イメージ ID	イメージ名
aki-04206613	pv-grub-hd0_1.05-i386.gz
aki-5c21674b	pv-grub-hd0_1.05-x86_64.gz

us-gov-west-1、 AWS GovCloud (米国西部)

イメージ ID	イメージ名
aki-5ee9573f	pv-grub-hd0_1.05-i386.gz
aki-9ee55bff	pv-grub-hd0_1.05-x86_64.gz

us-west-1、米国西部 (北カリフォルニア)

イメージ ID	イメージ名
aki-43cf8123	pv-grub-hd0_1.05-i386.gz
aki-59cc8239	pv-grub-hd0_1.05-x86_64.gz

us-west-2、米国西部 (オレゴン)

イメージ ID	イメージ名
aki-7a69931a	pv-grub-hd0_1.05-i386.gz
aki-70cb0e10	pv-grub-hd0_1.05-x86_64.gz

PV-GRUB の更新

すべてのバージョンの PV-GRUB AKI がすべてのインスタンスタイプと互換性があるとは限らないため、常に最新バージョンの PV-GRUB AKI を使用することをお勧めします。また、古いバージョンの PV-GRUB はすべてのリージョンで使用できるわけではないため、旧バージョンを使用する AMI を、そのバージョンをサポートしないリージョンにコピーした場合、カーネルのイメージを更新するまで、その AMI から起動されたインスタンスを起動できなくなります。次の手順を使用してインスタンスの PV-GRUB のバージョンを確認し、必要に応じて更新します。

PV-GRUB のバージョンを確認するには

1. インスタンスのカーネル ID を見つけます。

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

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

   このインスタンスのカーネル ID は、aki-70cb0e10 です。

2. このカーネル ID のバージョン情報を表示します。

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

{
    "Images": [
        {
            "VirtualizationType": "paravirtual",
            "Name": "pv-grub-hd0_1.05-x86_64.gz",
            ...
```

```
            "Description": "PV-GRUB release 1.05, 64-bit"
        }
    ]
}
```

このカーネルイメージは PV-GRUB 1.05 です。PV-GRUB のバージョンが最新バージョン (Amazon PV-GRUB カーネルイメージ ID を参照) でない場合、次の手順を使用して更新する必要があります。

PV-GRUB のバージョンを更新するには

インスタンスが古いバージョンの PV-GRUB を使用している場合は、最新バージョンに更新する必要があります。

1. Amazon PV-GRUB カーネルイメージ ID で、使用するリージョンとプロセッサアーキテクチャーの最新の PV-GRUB AKI を特定します。
2. インスタンスを停止します。使用されるカーネルイメージを変更するには、インスタンスを停止する必要があります。

   ```
   aws ec2 stop-instances --instance-ids instance_id --region region
   ```
3. インスタンスに使用するカーネルイメージを変更します。

   ```
   aws ec2 modify-instance-attribute --instance-id instance_id --kernel kernel_id --
   region region
   ```
4. インスタンスを再起動します。

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

AL2 AMI リリース通知

新しい Amazon Linux AMI がリリースされた場合に通知を受取るには、Amazon SNS を使用して申し込みできます。

AL2023 の通知のサブスクライブの詳細については、「Amazon Linux 2023 ユーザーガイド」の「新しい更新に関する通知の受信」を参照してください。

> **Note**
>
> AL1 の標準サポートは 2020 年 12 月 31 日に終了しました。AL1 メンテナンスサポートフェーズは 2023 年 12 月 31 日に終了しました。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「Update on Amazon Linux AMI end-of-life」を参照してください。

Amazon Linux の通知をサブスクライブするには

1. Amazon SNS コンソール (https://console.aws.amazon.com/sns/v3/home) を開きます。
2. ナビゲーションバーで、必要に応じて、リージョンを [米国東部 (バージニア北部)] に変更します。購読する 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. [AL2] トピック ARN の場合は、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-2-ami-updates**。
   b. [Amazon Linux] [トピックの ARN] には、以下の Amazon リソースネーム (ARN) **arn:aws:sns:us-east-1:137112412989:amazon-linux-ami-updates** をコピーして貼り付けます。
   c. [Protocol] で [Email] を選択します。
   d. [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。
   e. [サブスクリプションを作成] を選択します。
5. AWS 「通知 - サブスクリプションの確認」という件名の確認メールが届きます。メールを開いて [Confirm subscription] を選択して受信登録を完了します。

AMI がリリースされるごとに、対応するトピックの受信者に通知が送信されます。このような通知を停止するには、以下の手順を使用してサブスクリプション解除します。

Amazon Linux の通知の受信登録を解除するには

1. Amazon SNS コンソール (https://console.aws.amazon.com/sns/v3/home) を開きます。
2. ナビゲーションバーで、必要に応じて、リージョンを [米国東部 (バージニア北部)] に変更します。SNS 通知が作成されたリージョンを使用する必要があります。
3. ナビゲーションペインで、[サブスクリプション] を選択し、サブスクリプションを選択したら、[アクション]、[サブスクリプションの削除] の順に選択します。

4. 確認を求めるメッセージが表示されたら、[削除] を選択します。

Amazon Linux AMI の SNS メッセージ形式

SNS メッセージのスキーマは次のとおりです。

```
{
    "description": "Validates output from AMI Release SNS message",
    "type": "object",
    "properties": {
        "v1": {
            "type": "object",
            "properties": {
                "ReleaseVersion": {
                    "description": "Major release (ex. 2018.03)",
                    "type": "string"
                },
                "ImageVersion": {
                    "description": "Full release (ex. 2018.03.0.20180412)",
                    "type": "string"
                },
                "ReleaseNotes": {
                    "description": "Human-readable string with extra information",
                    "type": "string"
                },
                "Regions": {
                    "type": "object",
                    "description": "Each key will be a region name (ex. us-east-1)",
                    "additionalProperties": {
                        "type": "array",
                        "items": {
                            "type": "object",
                            "properties": {
                                "Name": {
                                    "description": "AMI Name (ex. amzn-ami-
hvm-2018.03.0.20180412-x86_64-gp2)",
                                    "type": "string"
                                },
                                "ImageId": {
                                    "description": "AMI Name (ex.ami-467ca739)",
                                    "type": "string"
                                }
                            },
```

```
                            "required": [
                                "Name",
                                "ImageId"
                            ]
                        }
                    }
                }
            },
            "required": [
                "ReleaseVersion",
                "ImageVersion",
                "ReleaseNotes",
                "Regions"
            ]
        }
    },
    "required": [
        "v1"
    ]
}
```

AL2 MATE デスクトップ接続を設定する

MATE デスクトップ環境は、AMI にあらかじめインストールおよび設定されています。説明は次のとおりです。

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

環境には、コマンドラインの使用を最小限に抑えて AL2 インスタンスを管理するための直感的なグラフィカルユーザーインターフェイスが用意されています。このインタフェースでは、アイコン、ウィンドウ、ツールバー、フォルダ、壁紙、デスクトップウィジェットなどのグラフィカルな表現が使用されています。一般的なタスクを実行するために、組み込みの GUI ベースのツールを使用することができます。例えば、ソフトウェアの追加と削除、更新プログラムの適用、ファイルの整理、プログラムの起動、システムの状態のモニタリングのためのツールが用意されています。

> **Important**
>
> xrdp は、AMI にバンドルされているリモートデスクトップソフトウェアです。デフォルトでは、xrdp は、自己署名の TLS 証明書を使用してリモートデスクトップセッションを暗

号化します。xrdp もメンテナンス担当者 AWS も、本番環境で自己署名証明書を使用することを推奨していません。代わりに、適切な認証局 (CA) から証明書を取得し、インスタンスにインストールします。TLS の設定については、xrdp wiki の「TLS セキュリティレイヤー」を参照してください。

> **Note**
>
> xrdp の代わりに仮想ネットワークコンピューティング (VNC) サービスを使用する場合は、「AL2Knowledge Center を実行する Amazon EC2 インスタンスに GUI をインストールする方法 AWS 」の記事を参照してください。

前提条件

このトピックに示すコマンドを実行するには、 AWS Command Line Interface (AWS CLI) または をインストールし AWS Tools for Windows PowerShell、 AWS プロファイルを設定する必要があります。

オプション

1. のインストール AWS CLI – 詳細については、AWS Command Line Interface 「 ユーザーガイド」の「 AWS CLIのインストール」および「設定の基本」を参照してください。
2. Tools for Windows PowerShell のインストール — 詳細については、「AWS Tools for PowerShell ユーザーガイド」の「AWS Tools for Windows PowerShellのインストール」と「共有認証情報」を参照してください。

> **Tip**
>
> を完全にインストールする代わりに AWS CLI、 から直接起動するブラウザベースの事前認証済みシェルAWS CloudShellに を使用できます AWS マネジメントコンソール。サポートされている AWS リージョンをチェックして、作業しているリージョンで使用可能であることを確認します。

RDP 接続の設定

以下の手順に従って、ローカルマシンから MATE デスクトップ環境を実行している AL2 インスタンスへのリモートデスクトッププロトコル (RDP) 接続を設定します。

1. AMI 名に MATE を含む AL2 用 AMI の ID を取得するには、ローカルコマンドラインツールから describe-images コマンドを使用できます。コマンドラインツールをインストールしていない場合は、 AWS CloudShell セッションから直接次のクエリを実行できます。CloudShell からシェルセッションを起動する方法の詳細については、「[Getting started with AWS CloudShel]」(CloudShell の使用方法) を参照してください。Amazon EC2 コンソールでは、インスタンスを起動し、AMI 検索バーに MATE と入力すると、MATE が含まれている AMI を見つけることができます。MATE がプリインストールされた AL2 クイックスタートが検索結果に表示されます。

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

   用途に合った AMI を選択します。
2. 前のステップで特定した AMI を使用して EC2 インスタンスを起動します。ポート 3389 へのインバウンド TCP トラフィックを許可するようにセキュリティグループを設定します。セキュリティグループの設定の詳細については、「VPC のセキュリティグループ」を参照してください。この設定により、RDP クライアントを使用してインスタンスに接続できます。
3. SSH を使用してインスタンスに接続します。
4. インスタンス上のソフトウェアとカーネルを更新します。

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

更新が完了したら、インスタンスを再起動し、更新から最新のパッケージとライブラリが使用されていることを確認します。カーネルの更新は、再起動するまでロードされません。

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

5. インスタンスに再接続し、Linux のインスタンスで次のコマンドを実行して、ec2-user のパスワードを設定します。

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

6. 証明書ファイルとキーをインストールする

   証明書とキーがすでにある場合は、証明書とキーを以下のように /etc/xrdp/ ディレクトリにコピーします。

   - 証明書 - /etc/xrdp/cert.pem
   - キー — /etc/xrdp/key.pem

   証明書とキーがない場合は、次のコマンドを使用して/etc/xrdpディレクトリに証明書とキーを作成します。

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

   > **Note**
   >
   > このコマンドは 365 日間有効な証明書を生成します。

7. インスタンスに接続するコンピュータで RDP クライアントを開きます (Microsoft Windows を実行するコンピュータのリモートデスクトップ接続など)。ユーザー名として「ec2-user」と入力し、前のステップで設定したパスワードを入力します。

Amazon EC2 インスタンスで **xrdp** を無効にするには

Linux インスタンスで次のコマンドのいずれかを実行することにより、いつでも xrdp を無効にすることができます。次のコマンドは、X11 サーバーを使用して MATE を使用する能力に影響を及ぼすものではありません。

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

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

Amazon EC2 インスタンスで **xrdp** を有効にするには

MATE デスクトップ環境を実行している AL2 インスタンスに接続xrdpできるように を再度有効にするには、Linux インスタンスで次のいずれかのコマンドを実行します。

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

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

AL2 チュートリアル

以下のチュートリアルでは、AL2 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。動画チュートリアルについては、AWS の講習動画とラボを参照してください。

AL2023 の手順については、「Amazon Linux 2023 ユーザーガイド」の「チュートリアル」を参照してください。

チュートリアル

- チュートリアル: AL2 に LAMP サーバーをインストールする
- チュートリアル: AL2 で SSL/TLS を設定する
- チュートリアル: AL2 で WordPress ブログをホストする

チュートリアル: AL2 に LAMP サーバーをインストールする

次の手順は、PHP と MariaDB (MySQL のコミュニティ開発フォーク) サポートを備えた Apache ウェブサーバーを AL2 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をホストしたり、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**
>
> 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と、このチュートリアルの通りにはなりません。AL2023 については、AL2023 に LAMP サーバーをインストールする」を参照してください。Ubuntu については、Ubuntu コミュニティドキュメントの 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。

オプション: オートメーション を使用してこのチュートリアルを完了する

以下のタスクの代わりに AWS Systems Manager Automation を使用してこのチュートリアルを完了するには、AWS Docs-InstallALAMPServer-AL2 Automation ドキュメントを実行します。

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) phpMyAdmin をインストールする
- 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、インターネットからアクセスできるパブリック DNS 名を持つ AL2 を使用して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 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。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「セキュリティグループルール」を参照してください。
- 次の手順では、AL2 で利用可能な最新の PHP バージョンをインストールします。現在は ですphp8.2。このチュートリアルで説明されている以外の PHP アプリケーションを使用する場合は、php8.2 と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。
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。

   -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 yum update -y
   ```

3. mariadb10.5 Amazon Linux Extras リポジトリをインストールして、MariaDB パッケージの最新バージョンを取得します。

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

   sudo: amazon-linux-extras: command not found というエラーが表示された場合、インスタンスは Amazon Linux 2 AMI で起動されていません (おそらく、代わりに Amazon Linux AMI を使用しています)。次のコマンドを使用して、Amazon Linux のバージョンを表示できます。

   ```
   cat /etc/system-release
   ```

4. php8.2 Amazon Linux Extras リポジトリをインストールして、AL2 のPHPパッケージの最新バージョンを取得します。

   ```
   [ec2-user ~]$ sudo amazon-linux-extras install php8.2
   ```

5. これでインスタンスが最新状態になったので、Apache ウェブサーバー、MariaDB、および PHP ソフトウェアパッケージをインストールできます。yum インストール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 yum install -y httpd
   ```

   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。

   ```
   yum info package_name
   ```

6. Apache ウェブサーバーを起動します。

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

7. 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
```

8. 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に設定されます。このグループには SSH 接続を許可する単一のルールが含まれます。

   a. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。

   b. [インスタンス] を選択し、該当する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. セキュリティグループのリンクを選択します。「セキュリティグループにルールを追加する」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。

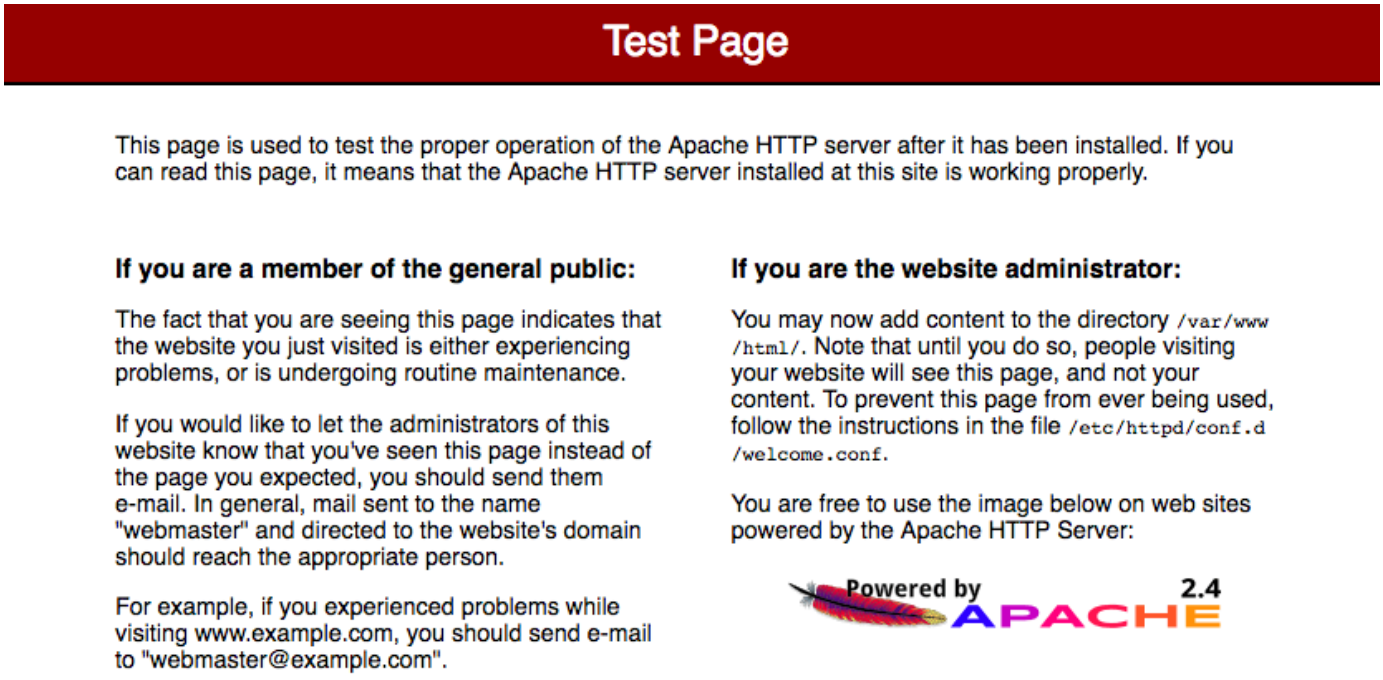
   - [Type]: HTTP
   - [Protocol]: TCP

- [Port Range]: 80
- [Source]: Custom

9. 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、Apache テストページが表示されます。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 ([Public DNS] 列を確認します。この列が表示されない場合は、[Show/Hide Columns] (歯車型のアイコン) をクリックして、[Public 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へのルールの追加」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、ディストリ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、apache ディレクトリの所有権を /var/www 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: AL2 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

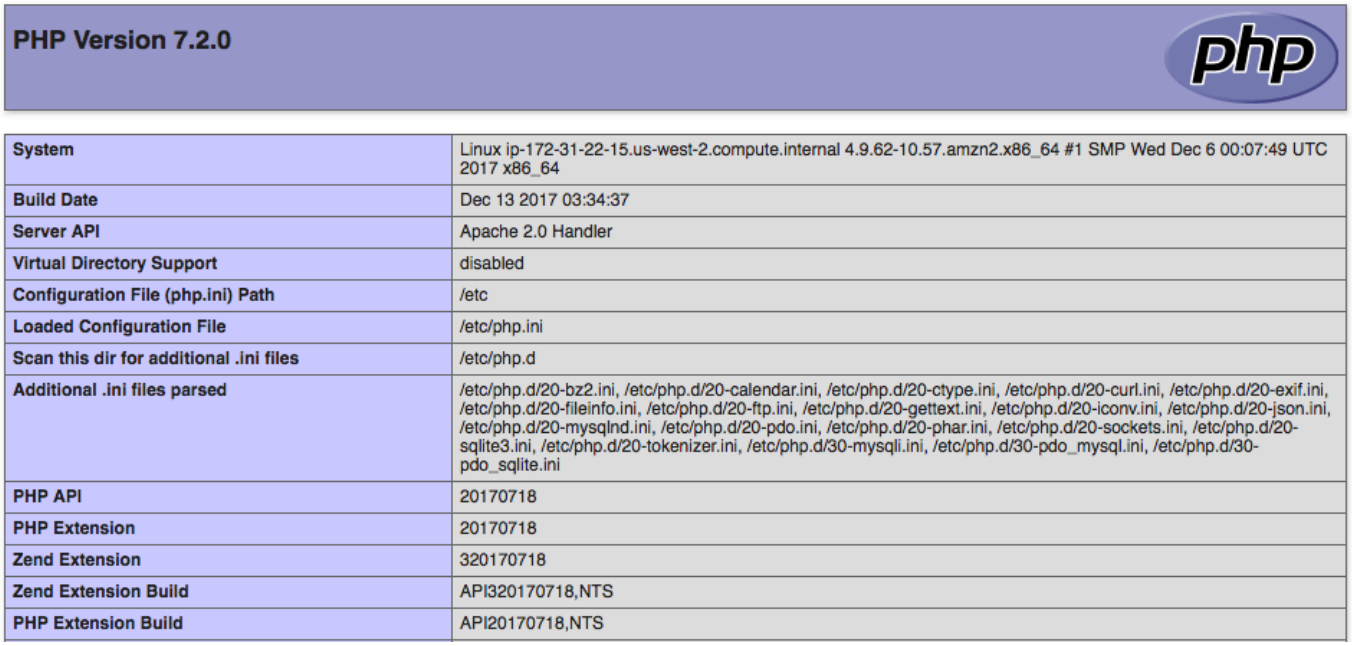
1. Apache ドキュメントルートで PHP ファイルを作成します。

   ```
   [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。



PHP Version 7.2.0

System	Linux ip-172-31-22-15.us-west-2.compute.internal 4.9.62-10.57.amzn2.x86_64 #1 SMP Wed Dec 6 00:07:49 UTC 2017 x86_64
Build Date	Dec 13 2017 03:34:37
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-json.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS

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 yum 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。また、`php7.2` と `lamp-mariadb10.2-php7.2` のエキストラが amazon-linux-extras のコマンド出力で有効になっていることを確認してください。

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) phpMyAdmin をインストールする

phpMyAdmin は、EC2 インスタンスで MySQL データベースを表示して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。開発者によるセキュリティ関連の推奨事項については、「Securing your phpMyAdmin installation」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: AL2 で SSL/TLS を設定する」を参照してください。

pMyAdmin をインストールするには

1. 必要な依存ファイルをインストールします。

```
[ec2-user ~]$ sudo yum 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 で最新の phpMyAdmin リリース用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 のインストール 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。次に例を示します。

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

phpMyAdmin ログインページが表示されます。

10. 前に作成した `root` ユーザー名と MySQL のルートパスワードを使って、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. `index.php` を含む phpMyAdmin ディレクトリに、ファイルを `config.inc.php` として保存します。

    c. 追加のセットアップについては、phpMyAdmin のインストール手順の「セットアップスクリプトの使用」セクションにある「ファイル作成後の手順」を参照してください。

    phpMyAdmin の使用に関する情報は、「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へのルールの追加」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: AL2 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、「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。

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。

- を使用して Linux インスタンスにファイルを転送するWinSCP。
- 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。
- チュートリアル: AL2 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/
- chmod コマンド: https://en.wikipedia.org/wiki/Chmod
- chown コマンド: https://en.wikipedia.org/wiki/Chown

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: AL2 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、AL2 および Apache ウェブサーバーを使用する EC2 インスタンスで 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。AL2 は、デフォルトですべてのバージョンの SSL の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 年に最終化され、基盤となる TLS ライブラリ (このチュートリアルでは OpenSSL) がサポートされているため、有効に設定されている限り、AL2 で利用できます。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 AL2 での使用を目的としています。また、新しい Amazon EC2 インスタンスを使用して開始するものと仮定します。別の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 AL2 を実行しているインスタンスを設定しようとすると、このチュートリアルの一部の手順が機能しない場合があります。Ubuntu については、Ubuntu 上の OpenSSL に関する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 Nitro Enclaves に AWS Certificate Manager (ACM) を使用することもできます。これは、 AWS 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と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「Nitro Enclaves とはAWS AWS Certificate Manager 」および「Nitro Enclaves 用」を参照してください。

内容

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。

- Amazon EBS ベースの AL2 インスタンスを起動します。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。
- 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「セキュリティグループのルール」を参照してください。
- Apache ウェブサーバーをインストールします。step-by-stepの手順については、「チュートリアル: AL2 に LAMP ウェブサーバーをインストールする」を参照してください。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

オプション: オートメーション を使用してこのチュートリアルを完了する

以下のタスクの代わりに AWS Systems Manager 自動化を使用してこのチュートリアルを完了するには、自動化ドキュメントを実行します。

この手順では、自己署名デジタル証明書を使用して AL2 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。

   ```
   [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 yum update -y
   ```

3. これでインスタンスが最新状態になったため、Apache モジュール mod_ssl をインストールして TLS サポートを追加します。

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

次のファイルがインスタンスに作成されました。このファイルは、セキュアサーバーの設定とテスト用の証明書の作成に使用します。

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

  mod_ssl の設定ファイル。このファイルには、暗号化キーと証明書の場所、許可する TLS プロトコル、受け入れる暗号化アルゴリズムを Apache に指示するディレクティブが含まれています。

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

  サーバーホスト用の自己署名 X.509 証明書とプライベートキーを生成するためのスクリプト。この証明書は、TLS を使用するように Apache が正しくセットアップされているかどうかをテストする場合に役立ちます。アイデンティティは証明されないため、本稼働環境では使用しないでください。本稼働環境で使用すると、ウェブブラウザで警告が表示されます。

4. テスト用に自己署名のダミー証明書とキーを生成するためのスクリプトを実行します。

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

/etc/pki/tls/certs/ ディレクトリに新しいファイル localhost.crt が生成されます。指定されたファイル名は、SSLCertificateFile の /etc/httpd/conf.d/ssl.conf ディレクティブで割り当てたデフォルトの名前と一致します。

このファイルには、自己署名証明書と証明書のプライベートキーのいずれも含まれます。Apache では、証明書とキーを PEM 形式にする必要があります。これは、次の短縮化された例のように、"BEGIN" 行と "END" 行で囲まれた Base64 エンコードの ASCII 文字で構成されます。

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

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

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ます。

> **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。

5. ルートユーザーとしてお好みのテキストエディタ (vim、nanoなど) を使用して `/etc/httpd/conf.d/ssl.conf` ファイルを開き、次の行をコメントアウトします。ダミーの自己署名証明書にも同じキーが含まれているためです。次のステップに進む前にこの行をコメントアウトしないと、Apache サービスは起動に失敗します。

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

6. 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。

7. 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。これの警告を無視し、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 (モジュロ) サイズを使用することを推奨しています。AL2 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。
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 CA で、楕円曲線ベースのキーに対して RSA キーと同じレベルのサポートが提供されてい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求められます。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	www.example.com

名前	説明	例
	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。	
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 AL2 のデフォルト (所有者 = ルート、グループ = ルート、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、およびアクセス許可の設定が、制限の厳しい AL2 のデフォルト (所有者 = ルート、グループ = ルート、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、AL2 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。

✗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。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。

3. 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 AL2 インスタンスup-to-date保ち、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エクスプロイトの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

と、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 yum 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 AL2 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 AL2 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: AL2 で WordPress ブログをホストする

次の手順は、AL2 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する (これは従来のホスティングサービスでは一般的なことではありません) という点で、Amazon EC2 を使用するための優れた手引きになります。

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 インストールの場合、この CloudFormation サービスには WordPress テンプレートが用意されており、すぐに開始することもできます。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。データベースが疎結合化された高可用性のソリューションが必要な場合は、AWS Elastic Beanstalk デベロッパーガイドの「高可用性の WordPress ウェブサイトをデプロイする」を参照してください。

> **Important**
>
> これらの手順は AL2 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詳細については、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 インスタンスでの WordPress のインストールについては、Ubuntu のドキュメントの「WordPress」を参照してください。CodeDeploy を使用して、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

このチュートリアルでは、「」のすべてのステップに従って、PHP とデータベース (MySQL または MariaDB) をサポートする機能的なウェブサーバーで AL2 インスタンスを起動していることを前提としていますチュートリアル: AL2 に LAMP サーバーをインストールする。この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アクセス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へのルールの追加の詳細については、「セキュリティグループへのルールの追加」を参照してください。

Elastic IP アドレス (EIP) は、WordPress ブログのホストに使用しているインスタンスに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問い合わせてください)。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。

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。

WordPress のインストール

オプション: オートメーション を使用してこのチュートリアルを完了する

以下のタスクの代わりに AWS Systems Manager 自動化を使用してこのチュートリアルを完了するには、自動化ドキュメントを実行します。

インスタンスに接続して、WordPress インストールパッケージをダウンロードします。

WordPress インストールパッケージをダウンロードして解凍するには

1. wget コマンドを使って、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
   ```

2. 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 インストール用に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 インストールは、ブログの投稿、ユーザーコメントなどの情報をデータベースに格納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. データベースサーバーを起動します。

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

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、MariaDB サーバーをセキュリティで保護するには「(AL2)」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 インストールは、これらの値を使って、MySQL データベースと通信を行います。

   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。

   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';
```

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. データベースに対して、以前作成した WordPress ユーザーに対する完全な権限を付与します。

```
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、wp-config-sample.php という名前の構成ファイル例が格納されています。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. お好みのテキスト工ディタ (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。お好みのテキスト工ディタがない場合、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 インストール用に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 インストール用に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 インストール用に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 SALT の値は、WordPress ユーザーがローカルマシンに保存したブラウザクッキーに対する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」にアクセスしてください。

> **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 ?
6OP$eJT@;+(ndLg');
define('LOGGED_IN_KEY',    'ju}qwre3V*+8f_zOWf?{LlGsQ]Ye@2Jh^,8x>)Y |;(^[Iw]Pi
+LG#A4R?7N`YB3');
define('NONCE_KEY',        'P(g62HeZxEes|LnI^i=H,[XwK9I&[2s|:?0N}VJM%?;v2v]v+;
+^9eXUahg@::Cj');
define('AUTH_SALT',        'C$DpB4Hj[JK:?{ql`sRVa:{:7yShy(9A@5wg+`JJVb1fk%_-
Bx*M4(qc[Qg%JT!h');
define('SECURE_AUTH_SALT', 'd!uRu#}+q#{f$Z?Z9uFPG.${+S{n~1M&%@~gL>U>NV<zpD-@2-
Es7Q1O-bp28EKv');
define('LOGGED_IN_SALT',   ';j{00P*owZf)kVD+FVLn-~ >.|Y%Ug4#I^*LVd9QeZ^&XmK|
e(76miC+&W&+^0P/');
define('NONCE_SALT',       '-97r*V/cgxLmp?Zy4zUU4r99QQ_rGs2LTd%P;|
_e1tS)8_B/,.6[=UK<J_y9?JWG');
```

e. ファイルを保存し、テキストエディタを終了します。

WordPress ファイルを Apache ドキュメントルートの下にインストールするには

- インストールフォルダの解凍、MySQL データベースとユーザーの作成、WordPress 構成ファイルのカスタマイズが終了したため、インストールファイルをウェブサーバーのドキュメントルートにコピーし、インストールスクリプトを実行して、インストールを終了する準備ができました。これらのファイルの場所は、ウェブサーバーの実際のルートで WordPress ブログを使用できるようにするかどうか (*my.public.dns.amazonaws.com* など)、またはルートの下のサブ

ディレクトリやフォルダに格納するか (*my.public.dns.amazonaws.com/blog* など) によって異なります。

- WordPress をドキュメントルートで実行する場合は、WordPress のインストールディレクトリのコンテンツを次のようにコピーします (ただし、ディレクトリ自体はコピーしません)。

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

- WordPress をドキュメントルートの下の別のディレクトリで実行する場合、まず、そのディレクトリを作成してから、そこ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 blog から実行されます。

```
[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なくなり、Apache ウェブサーバーが実行している場合、攻撃者はブログへのアクセス権を取得す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 systemctl stop httpd 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 がパーマリンクを使用できるようにするには

WordPress のパーマリンクが正しく機能するには Apache の .htaccess ファイルを使用する必要がありますが、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。

PHP グラフィック描画ライブラリを AL2 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。インストールするバージョンの phpMyAdmin は、このライブラリの特定の最小バージョン (バージョン 7.2 など) を必要とする場合があります。

次のコマンドを使用して、PHP グラフィック描画ライブラリを AL2 に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 amazon-linux-extras から php7.2 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 7.2 をインストールします。

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

出力例を次に示します。

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

Apache ウェブサーバーのファイル許可を修正するには

WordPress で使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権限が必要なものがあります (管理画面を使った、メディアのアップロードなど)。まだ適用していない場合は、次のグループメンバーシップとアクセス許可を適用します (詳細については、「」を参照してくださいチュートリアル: AL2 に LAMP サーバーをインストールする)。

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**
>
> WordPress を FTP サーバーとして使用する場合も、これよりも制限の少ないグループ設定が必要になります。これを実行するには、「WordPress で推奨されている手順とセキュリティ設定」を参照してください。

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
     ```

AL2 で WordPress インストールスクリプトを実行する

WordPress をインストールする準備ができました。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、AL2 で使用するためのものです。

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 を入力します (インスタンスのパブリック DNS アドレス、または blog フォルダに続くアドレス)。WordPress インストールスクリプトが表示されます。WordPress のインストールに必要な情報を入力します。[WordPress のインストール] 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 のウェブサイトの Step 5: Run the Install Script 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ます。

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「サイトの 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員ストー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、「独自の AMI を作成する」を参照してください。

容量を増やす

WordPress ブログが人気になり処理能力やストレージを増やす必要がある場合は、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 ユーザーガイド」の「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 を使用すると、ユーザーのクライアントデバイスと AWSで実行中の WordPress アプリケーションとの間で、インターネットトラフィックのパフォーマンスを向上でき、低レイテンシーを実現できます。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。

WordPress の詳細

WordPress の詳細については、「http://codex.wordpress.org/」にある WordPress Codex ヘルプ文書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、「一般的なインストールの問題」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、「WordPress の強化」を参照してください。

WordPress ブログをup-to-date状態に保つ方法については、WordPress の更新」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 のインストールは、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。問題の詳細な説明と考えられる解決策については、「サイト URL の変更」を参照してください。

WordPress のインストールでこの問題が発生した場合は、WordPress 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。

wp-cli を使用して WordPress のサイト URL を変更するには

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、WordPress をインストールした時点での EC2 インスタンスのパブリック DNS 名と考えら

れます。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. 次のコマンドを使って、WordPress インストールの古いサイト URL を検索し、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 URL、および WordPress のインストールパス (通常は /var/www/html または /var/www/html/blog) を置き換えます。

```
[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、サイトが再び正しく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 外で Amazon Linux 2 を使用する

AL2 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。

AL2 は、Amazon EC2 で直接実行される以外の仮想化ゲストとして実行することもできます。

> **Note**
>
> AL2 イメージの設定は AL2023 とは異なります。
> AL2023 に移行するときは、Amazon EC2 の外部で Amazon Linux 2023 を使用するを確認し、AL2023 と互換性があるように設定を適応させてください。

オンプレミスで仮想マシンとして AL2 を実行する

オンプレミスの開発とテストには AL2 仮想マシン (VM) イメージを使用します。サポートされている仮想化プラットフォームごとに異なる AL2 VM イメージを提供しています。サポートされているプラットフォームのリストは、[Amazon Linux 2 virtual machine images] (Amazon Linux 2 仮想マシンイメージ) ページで確認できます。

サポートされている仮想化プラットフォームのいずれかで AL2 仮想マシンイメージを使用するには、次の手順を実行します。

- ステップ 1: seed.iso 起動イメージを準備する
- ステップ 2: AL2 VM イメージをダウンロードする
- ステップ 3: 新しい VM を起動して接続する

ステップ 1: `seed.iso` 起動イメージを準備する

`seed.iso` 起動イメージには、新しい VM の起動に必要な初期設定情報 (例: ネットワーク設定、ホスト名、ユーザーデータ) が含まれます。

> **Note**
>
> `seed.iso` 起動イメージには、VM の起動に必要な設定情報のみ含まれています。AL2 オペレーティングシステムファイルは含まれません。

seed.iso 起動イメージを生成するには、2 つの設定ファイルが必要です。

- meta-data – このファイルには、VM のホスト名と静的ネットワーク設定が含まれます。
- user-data – このファイルはユーザーアカウントを設定し、パスワード、キーペア、アクセスメカニズムを指定します。デフォルトでは、AL2 VM イメージはec2-userユーザーアカウントを作成します。デフォルトのユーザーアカウントのパスワードを設定するには、user-data 設定ファイルを使用します。

seed.iso 起動ディスクを作成するには

1. 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。
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. 任意のテキストエディタを使用して meta-data ファイルを開き、以下を追加します。

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

   vm_hostname を任意の VM ホスト名に置き換え、必要に応じてネットワーク設定を行います。

   c. meta-data 設定ファイルを保存して閉じます。

   VM ホスト名 (meta-data) を指定し、デフォルトのネットワークインターフェイス (amazonlinux.onprem) を構成し、必要なネットワークデバイスの静的 IP アドレスを指定する eth0 設定ファイルの例については、サンプルの Seed.iso ファイルを参照してください。
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. 任意のテキストエディタを使用して user-data ファイルを開き、以下を追加します。

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

plain_text_password を、デフォルトの ec2-user ユーザーアカウントの任意のパスワードに置き換えます。

c. (オプション) デフォルトでは、cloud-init は VM が起動される度にネットワーク設定に適用されます。ブート起動時の cloud-init によるネットワーク設定の適用を無効にし、最初の起動時のネットワーク設定を保持するには、以下を追加します。

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

d. user-data 設定ファイルを保存して閉じます。

また、他のユーザーアカウントを作成して、アクセスメカニズム、パスワード、およびキーペアを指定することもできます。サポートされるディレクティブについては、「モジュール参照」を確認してください。3 人のユーザーを追加で作成し、デフォルトの user-data ユーザーアカウントのカスタムパスワードを指定する ec2-user ファイルの例については、サンプル Seed.iso ファイルを参照してください。

4. seed.iso および meta-data 設定ファイルを使用して、user-data 起動イメージを作成します。

Linux の場合は、genisoimage などのツールを使用します。seedconfig フォルダに移動し、次のコマンドを実行します。

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

macOS の場合は、hdiutil などのツールを使用します。seedconfig フォルダの 1 つ上に移動し、次のコマンドを実行します。

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

ステップ 2: AL2 VM イメージをダウンロードする

サポートされている仮想化プラットフォームごとに異なる AL2 VM イメージを提供しています。サポートされているプラットフォームのリストを表示し、選択したプラットフォームに適した VM イメージを [Amazon Linux 2 virtual machine images] (Amazon Linux 2 仮想マシンイメージ) ページからダウンロードできます。

ステップ 3: 新しい VM を起動して接続する

新しい VM を起動して接続するには、seed.isoブートイメージ (ステップ 1 で作成) と AL2 VM イメージ (ステップ 2 でダウンロード) が必要です。ステップは、選択した VM プラットフォームによって異なります。

VMware vSphere

VMware の VMイメージは、OVF 形式で提供されます。

VMware vSphere を使用して VM を起動するには

1. seed.iso ファイルの新しいデータストアを作成するか、既存のデータストアに追加します。
2. OVF テンプレートをデプロイしますが、VM はまだ起動しないでください。
3. ナビゲータパネルで、新しい仮想マシンを右クリックし、[設定の編集] を選択します。
4. [Virtual Hardware (仮想ハードウェア)] タブの [New device (新しいデバイス)] で、[CD/DVD Drive (CD/DVD ドライブ)] を選択し、[追加] を選択します。

5. [New CD/DVD Drive (新しい CD/DVD ドライブ)] で、[Datastore ISO File (データストア ISO ファイル)] を選択します。`seed.iso` ファイルを追加したデータストアを選択し、`seed.iso` ファイルを参照して選択し、[OK] を選択します。
6. [新しい CD/DVD ドライブ] で [接続]、[OK] の順にクリックします。

データストアを VM に関連付けると、起動できるようになります。

KVM

KVM を使用して VM を起動するには

1. [Create new VM (新しい VM の作成)] ウィザードを開きます。
2. ステップ 1 で、[Import existing disk image (既存のディスクイメージのインポート)] を選択します。
3. ステップ 2 で、VM イメージを参照して選択します。[OS type] (OS タイプ) と [Version] (バージョン) では、[Linux] と [Red Hat Enterprise Linux 7.0] をそれぞれ選択します。
4. ステップ 3 では、RAM の容量と CPU の数を指定します。
5. ステップ 4 では、新しい VM の名前を入力し、[Customize configuration before install (インストール前に構成をカスタマイズする)] を選択し、[完了] を選択します。
6. 仮想マシンの [構成] ウィンドウで、[Add Hardware (ハードウェアの追加)] を選択します。
7. [Add New Virtual Hardware (新しい仮想ハードウェアの追加)] ウィンドウで、[ストレージ] を選択します。
8. [ストレージ] 構成で、[Select or create custom storage (カスタムストレージの選択または作成)] を選択します。[デバイスタイプ] で、[CDROM デバイス] を選択します。[管理]、[Browse Local (ローカルを参照)] の順に選択し、`seed.iso` ファイルに移動して選択します。[Finish] を選択します。
9. [Begin Installation (インストールを開始)] を選択します。

Oracle VirtualBox

Oracle VirtualBox を使用して VM を起動するには

1. Oracle VirtualBoxを開き、[New (新規)] を選択します。
2. [Name] (名前) には、仮想マシン用のわかりやすい名前を入力します。[Type] (タイプ) と [Version] (バージョン) では、[Linux] と [Red Hat (64-bit)] をそれぞれ選択します。[続行] をクリックします。

3. [Memory size (メモリサイズ)] で、仮想マシンに割り当てるメモリの量を指定し、[Continue (続行)] を選択します。
4. [Hard disk (ハードディスク)] で、[Use an existing virtual hard disk file (既存の仮想ハードディスクファイルを使用する)] を選択し、VM イメージを参照して開き、[Create (作成)] を選択します。
5. VM を起動する前に、仮想マシンの仮想オプティカルドライブに `seed.iso` ファイルをロードする必要があります。

   a. 新しい VM を選択し、[設定]、[ストレージ] の順に選択します。
   b. [Storage Devices (ストレージデバイス)] リストの [Controller: IDE (コントローラー: IDE)] で、空のオプティカルドライブを選択します。
   c. オプティカルドライブの [属性] セクションで、参照ボタンを選択し、[Choose Virtual Optical Disk File (仮想オプティカルディスクファイルを選択する)] を選択し、`seed.iso` ファイルを選択します。[OK] を選択して変更を適用し、[Settings (設定)] を閉じます。

仮想オプティカルドライブに `seed.iso` ファイルを追加すると、VM を起動できます。

Microsoft Hyper-V

Microsoft Hyper-V 用の VM イメージは zip ファイルに圧縮されます。.zip ファイルの内容を展開する必要があります。

Microsoft Hyper-V を使用して VM を起動するには

1. [New Virtual Machine Wizard (新しい仮想マシンウィザード)] を開きます。
2. 世代を選択するよう求められたら、[Generation 1 (第 1 世代)] を選択します。
3. ネットワークアダプタの構成を求めるメッセージが表示されたら、[接続] に [外部] を選択します。
4. 仮想ハードディスクを接続するかどうかを確認するメッセージが表示されたら、[Use an existing virtual hard disk (既存の仮想ハードディスクを使用する)]、[参照] の順に選択し、VM イメージに移動して選択します。[完了] を選択し、VM を作成します。
5. 新しい VM を右クリックし、[設定] を選択します。[設定] ウィンドウの [IDE Controller 1 (IDE コントローラー 1)] で、[DVD Drive (DVD ドライブ)] を選択します。
6. DVD ドライブの場合は、[Image file (イメージファイル)] を選択し、`seed.iso` ファイルを参照して選択します。

7. 変更を適用し、VM を起動します。

VM を起動したら、user-data 設定ファイルで定義されているいずれかのユーザーアカウントを使用してログインします。初回ログイン後に、VM から seed.iso 起動イメージを切断できます。

Amazon Linux のインスタンスとバージョンの識別

OS イメージまたはインスタンスが、どの Linux ディストリビューションで、どのバージョンのディストリビューションであるかを判別できることが重要になる場合があります。Amazon Linux には、Amazon Linux を他の Linux ディストリビューションから識別し、イメージがどのリリースの Amazon Linux であるかを判別するメカニズムが備わっています。

このセクションでは、使用できるさまざまな方法、各方法の制限、いくつかの使用例を紹介します。

トピック

- os-release 標準の使用
- Amazon Linux 固有
- OS 検出のコード例

`os-release` 標準の使用

Amazon Linux は、Linux ディストリビューションを識別するための `os-release` 標準に準拠しています。このファイルは、オペレーティングシステムの識別とバージョンに関する機械読み取り可能な情報を提供します。

> **Note**
>
> 標準では、最初に `/etc/os-release` を解析し、次に `/usr/lib/os-release` を解析するよう規定してします。ファイル名とパスに関しては、標準に従うよう注意する必要があります。

トピック

- 識別の主な違い
- フィールドタイプ: 機械読み取り可能なタイプと人間が読み取り可能なタイプの比較
- /etc/os-release の例
- 他のディストリビューションとの比較

識別の主な違い

os-release は /etc/os-release にあります。そこにない場合は /usr/lib/os-release にあります。詳細については、os-release 標準を参照してください。

インスタンスが Amazon Linux を実行しているかどうかを判断する最も信頼性の高い方法は、os-release の ID フィールドを確認することです。

バージョンを区別する最も信頼性の高い方法は、os-release の VERSION_ID フィールドを確認することです。

- Amazon Linux AMI: VERSION_ID は日付ベースのバージョン (例: 2018.03) を示します。
- AL2: VERSION_ID="2"
- AL2023: VERSION_ID="2023"

> **Note**
>
> VERSION_ID はプログラムによる使用を目的とした機械読み取り可能なフィールドであり、PRETTY_NAME はユーザーへの表示用に設計されていることに注意してください。フィールドタイプの詳細については、「the section called “フィールドタイプ”」を参照してください。

フィールドタイプ: 機械読み取り可能なタイプと人間が読み取り可能なタイプの比較

/etc/os-release ファイル (/etc/os-release が存在しない場合は /usr/lib/os-release) には、プログラムによる使用を目的とした機械読み取り可能なフィールドと、ユーザーへの表示を目的とした人間が読み取り可能なフィールドという 2 つのタイプのフィールドが含まれています。

機械読み取り可能なフィールド

標準化された形式を使用するフィールドであり、スクリプト、パッケージマネージャー、その他の自動ツールによる処理を目的としています。小文字、数字、限定された句読点 (ピリオド、アンダースコア、ハイフン) のみを使用できます。

- ID – オペレーティングシステム識別子。Amazon Linux はすべてのバージョンで amzn を使用し、Debian (debian)、Ubuntu (ubuntu)、Fedora (fedora) などの他のディストリビューションと区別しています。
- VERSION_ID – プログラムで使用するオペレーティングシステムのバージョン (例: 2023)
- ID_LIKE – 関連するディストリビューションのスペース区切りリスト (例: fedora)
- VERSION_CODENAME – スクリプトのリリースコード名 (例: karoo)
- VARIANT_ID – プログラムで決定するバリアント識別子
- BUILD_ID – システムイメージのビルド識別子
- IMAGE_ID – コンテナ化された環境のイメージ識別子
- PLATFORM_ID – プラットフォーム識別子 (例: platform:al2023)

人間が読み取り可能なフィールド

ユーザーへの表示を目的としたフィールドであり、スペース、大文字と小文字の混在、説明テキストを使用できます。ユーザーインターフェイスでオペレーティングシステム情報を表示するときに使用してください。

- NAME – 表示用のオペレーティングシステム名 (例: Amazon Linux)
- PRETTY_NAME – 表示用のバージョンを含む完全なオペレーティングシステム名 (例: Amazon Linux 2023.8.20250721)
- VERSION – ユーザーへの表示に適したバージョン情報
- VARIANT – 表示用のバリアント名または工ディション名 (例: Server Edition)

その他の情報フィールド

オペレーティングシステムに関する追加のメタデータを示すフィールドです。

- HOME_URL – プロジェクトのホームページ URL
- DOCUMENTATION_URL – ドキュメント URL
- SUPPORT_URL – サポート情報 URL
- BUG_REPORT_URL – バグレポート URL
- VENDOR_NAME – ベンダー名
- VENDOR_URL – ベンダー URL
- SUPPORT_END – サポート終了日 (YYYY-MM-DD 形式)

- CPE_NAME – 共通プラットフォーム列挙識別子
- ANSI_COLOR – ターミナル表示用の ANSI カラーコード

Amazon Linux をプログラムで識別する必要があるスクリプトやアプリケーションを作成するときは、ID や VERSION_ID などの機械読み取り可能なフィールドを使用します。ユーザーにオペレーティングシステム情報を表示するときは、PRETTY_NAME などの人間が読み取り可能なフィールドを使用します。

/etc/os-release の例

/etc/os-release ファイルの内容は Amazon Linux のバージョンによって異なります。

AL2023

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

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

AL2

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

```
NAME="Amazon Linux"
VERSION="2"
ID="amzn"
```

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

Amazon Linux AMI

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

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

他のディストリビューションとの比較

Amazon Linux が広範な Linux エコシステムにどのように適合するかを理解するには、その `/etc/os-release` 形式を他の主要なディストリビューションと比較します。

Fedora

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

```
NAME="Fedora Linux"
VERSION="42 (Container Image)"
RELEASE_TYPE=stable
ID=fedora
VERSION_ID=42
VERSION_CODENAME=""
PLATFORM_ID="platform:f42"
PRETTY_NAME="Fedora Linux 42 (Container Image)"
ANSI_COLOR="0;38;2;60;110;180"
```

```
LOGO=fedora-logo-icon
CPE_NAME="cpe:/o:fedoraproject:fedora:42"
DEFAULT_HOSTNAME="fedora"
HOME_URL="https://fedoraproject.org/"
DOCUMENTATION_URL="https://docs.fedoraproject.org/en-US/fedora/f42/system-
administrators-guide/"
SUPPORT_URL="https://ask.fedoraproject.org/"
BUG_REPORT_URL="https://bugzilla.redhat.com/"
REDHAT_BUGZILLA_PRODUCT="Fedora"
REDHAT_BUGZILLA_PRODUCT_VERSION=42
REDHAT_SUPPORT_PRODUCT="Fedora"
REDHAT_SUPPORT_PRODUCT_VERSION=42
SUPPORT_END=2026-05-13
VARIANT="Container Image"
VARIANT_ID=container
```

Debian

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

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

Ubuntu

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

```
PRETTY_NAME="Ubuntu 24.04.2 LTS"
NAME="Ubuntu"
VERSION_ID="24.04"
VERSION="24.04.2 LTS (Noble Numbat)"
VERSION_CODENAME=noble
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
```

```
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=noble
LOGO=ubuntu-logo
```

機械読み取り可能なフィールドは、ディストリビューション間を一貫した方法で識別することに注目してください。

- ID – オペレーティングシステムを一意に識別: Amazon Linux 用の amzn、Fedora 用の fedora、Debian 用の debian、Ubuntu 用の ubuntu
- ID_LIKE – ディストリビューションの関係を表示: Amazon Linux は fedora (AL2023) または centos rhel fedora (AL2) を使用、Ubuntu は Debian の伝統を示す debian を表示
- VERSION_ID – 機械解析可能なバージョン情報を提供: AL2023 用の 2023、Fedora 用の 42、Debian 用の 12、Ubuntu 用の 24.04

対照的に、人間が読み取り可能なフィールドはユーザーへの表示用に設計されています。

- NAME – ユーザーフレンドリーな OS 名: Amazon Linux、Fedora Linux、Debian GNU/Linux、Ubuntu
- PRETTY_NAME – 完全な表示名とバージョン: Amazon Linux 2023.8.20250721、Fedora Linux 42 (Container Image)、Debian GNU/Linux 12 (bookworm)、Ubuntu 24.04.2 LTS
- VERSION – コード名やリリースタイプなどの追加のコンテキストを含む、人間が読み取り可能なバージョン

クロスプラットフォームスクリプトを記述する場合、ロジックと決定には常に機械読み取り可能なフィールド (ID、VERSION_ID、ID_LIKE) を使用し、人間が読み取り可能なフィールド (PRETTY_NAME、NAME) はユーザーに情報を表示する目的にのみ使用します。

Amazon Linux 固有

一部のファイルは Amazon Linux 固有であり、Amazon Linux とそのバージョンを識別するために使用できます。新しいコードでは、クロスディストリビューションの互換性を確保するために、/etc/os-release 標準を使用する必要があります。Amazon Linux 固有のファイルの使用はお勧めしません。

トピック

/etc/system-release ファイル

Amazon Linux には、インストールされている現在のリリースを示す /etc/system-release ファイルが含まれています。このファイルはパッケージマネージャーを使用して更新され、Amazon Linux では system-release パッケージの一部となっています。このファイルは、Fedora などの他のディストリビューションにもありますが、Ubuntu などの Debian ベースのディストリビューションにはありません。

> **Note**
>
> /etc/system-release ファイルには人間が読み取れる文字列が含まれているため、OS またはリリースを識別するためにプログラムで使用しないでください。代わりに、/etc/os-release (/etc/os-release が存在しない場合は /usr/lib/os-release) で機械読み取り可能なフィールドを使用してください。

Amazon Linux には、/etc/system-release-cpe ファイルの Common Platform Enumeration (CPE) 仕様に準拠した /etc/system-release の機械読み取り可能なバージョンも含まれています。

イメージ識別ファイル

各 Amazon Linux イメージには、Amazon Linux チームが生成した元のイメージに関する追加情報を示す独自の /etc/image-id ファイルが含まれています。このファイルは Amazon Linux に固有であり、Debian、Ubuntu、Fedora などの他の Linux ディストリビューションにはありません。このファイルには、イメージに関する次の情報が含まれています。

- image_name、image_version、image_arch - イメージの構築に使用したビルドレシピからの値。
- image_stamp – イメージの作成中に生成される一意のランダムな 16 進値。
- image_date - YYYYMMDDhhmmss 形式で画像を作成した UTC 時間。

- `recipe_name`、`recipe_id` – イメージの構築に使用したビルドレシピの名前と ID。

Amazon Linux 固有のファイルの例

以下のセクションでは、Amazon Linux のメジャーバージョンごとに Amazon Linux 固有の識別ファイルの例を示します。

> **Note**
>
> 実際のコードでは、`/etc/os-release` ファイルが存在しない場合、`/usr/lib/os-release` を使用する必要があります。

AL2023

以下の例は、AL2023 の識別ファイルを示しています。

AL2023 の `/etc/image-id` の例:

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

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

AL2023 の `/etc/system-release` の例:

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

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

AL2

以下の例は、AL2 の識別ファイルを示しています。

AL2 の `/etc/image-id` の例:

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

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

AL2 の `/etc/system-release` の例:

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

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

Amazon Linux AMI

以下の例は、Amazon Linux AMI の識別ファイルを示しています。

Amazon Linux AMI の `/etc/image-id` の例:

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

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

Amazon Linux AMI の `/etc/system-release` の例:

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

```
Amazon Linux AMI release 2018.03
```

OS 検出のコード例

以下の例は、/etc/os-release (/etc/os-release が存在しない場合は /usr/lib/os-release) ファイルを使用してオペレーティングシステムとバージョンをプログラムで検出する方法を示しています。これらの例は、Amazon Linux と他のディストリビューションを区別する方法と、ID_LIKE フィールドを使用してディストリビューションファミリーを決定する方法を示しています。

次のスクリプトは、複数の異なるプログラミング言語で実装され、各実装が同じ出力を生成します。

Shell

```
#!/bin/bash

# Function to get a specific field from os-release file
get_os_release_field() {
    local field="$1"
    local os_release_file

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

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

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

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

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

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

# Function to check if this is like Fedora (includes Amazon Linux, CentOS, RHEL,
 etc.)
is_like_fedora() {
    local id="$(get_os_release_field ID)"
    local id_like="$(get_os_release_field ID_LIKE)"
    [ "$id" = "fedora" ] || [[ "$id_like" == *"fedora"* ]]
}

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

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

echo "Operating System Detection Results:"
echo "==================================="
echo "Is Amazon Linux: $(is_amazon_linux && echo YES || echo NO)"
echo "Is Fedora: $(is_fedora && echo YES || echo NO)"
echo "Is Ubuntu: $(is_ubuntu && echo YES || echo NO)"
echo "Is Debian: $(is_debian && echo YES || echo NO)"
echo "Is like Fedora: $(is_like_fedora && echo YES || echo NO)"
```

```
echo "Is like Debian: $(is_like_debian && echo YES || echo NO)"
echo
echo "Detailed OS Information:"
echo "========================"
echo "ID: $ID"
echo "VERSION_ID: $VERSION_ID"
echo "PRETTY_NAME: $PRETTY_NAME"
[ -n "$ID_LIKE" ] && echo "ID_LIKE: $ID_LIKE"

# Amazon Linux specific information
if is_amazon_linux; then
    echo ""
    echo "Amazon Linux Version Details:"
    echo "============================="
    case "$VERSION_ID" in
        2018.03)
            echo "Amazon Linux AMI (version 1)"
            ;;
        2)
            echo "Amazon Linux 2"
            ;;
        2023)
            echo "Amazon Linux 2023"
            ;;
        *)
            echo "Unknown Amazon Linux version: $VERSION_ID"
            ;;
    esac

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

Python 3.7-3.9

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

import os
import sys

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

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

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

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

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

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

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

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

def is_like_debian(os_data):
```

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

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

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

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

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

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

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

Python 3.10+

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

import os
import sys
import platform

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

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

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

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

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

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

```
    id_like = os_data.get('ID_LIKE', '')
    return 'debian' in id_like

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

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

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

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

```
            print(f"Unknown Amazon Linux version: {version_id}")

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

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

Perl

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

use strict;
use warnings;

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

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

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

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

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

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

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

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

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

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

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

```
# Main execution
my %os_data = parse_os_release();

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

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

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

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

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

```
}
```

異なるシステムで実行すると、スクリプトは次の出力を生成します。

AL2023

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

Detailed OS Information:
========================
ID: amzn
VERSION_ID: 2023
PRETTY_NAME: Amazon Linux 2023.8.20250721
ID_LIKE: fedora

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

AL2

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

Detailed OS Information:
========================
ID: amzn
VERSION_ID: 2
```

```
PRETTY_NAME: Amazon Linux 2
ID_LIKE: centos rhel fedora

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

Amazon Linux AMI

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

Detailed OS Information:
========================
ID: amzn
VERSION_ID: 2018.03
PRETTY_NAME: Amazon Linux AMI 2018.03
ID_LIKE: rhel fedora

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

Ubuntu

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

Detailed OS Information:
```

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

Debian

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

Detailed OS Information:
========================
ID: debian
VERSION_ID: 12
PRETTY_NAME: Debian GNU/Linux 12 (bookworm)
```

Fedora

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

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

AWSAL2 での統合

AWSコマンドラインツール

AWS Command Line Interface(AWS CLI) は、コマンドラインシェルのコマンドAWS のサービスを使用して とやり取りするための一貫したインターフェイスを提供するオープンソースツールです。詳細については、「 AWS Command Line Interfaceユーザーガイド」の「 とはAWS Command Line Interface」を参照してください。

AL2 および AL1 には、 のバージョン 1 がAWS CLIプリインストールされています。Amazon Linux の現在のリリースである AL2023 には、 のバージョン 2 がAWS CLIプリインストールされています。AL2023 AWS CLIでの の使用の詳細については、Amazon Linux 2023 ユーザーガイドの「AL2023 の開始方法」を参照してください。

ランタイムのプログラミングの開始方法

AL2 は、特定の言語ランタイムのさまざまなバージョンを提供します。PHP などのアップストリームプロジェクトでは、複数のバージョンを同時にサポートしています。これらの名前付きバージョンパッケージをインストールして管理する方法については、 yum コマンドを使用してこれらのパッケージを検索してインストールします。詳細については、「パッケージリポジトリ」を参照してください。

以下のトピックでは、各言語ランタイムが AL2 でどのように機能するかについて説明します。

トピック

- CAL2 Fortranの 、C++、および
- AL2 で移動
- Java AL2 の
- Perl AL2 の
- PHP AL2 の
- Python AL2 の
- AL2 の錆

CAL2 Fortranの 、C++、および

AL2 には、GNU コンパイラコレクション (GCC) と のClangフロントエンドの両方が含まれています LLVM。

のメジャーバージョンGCCは、AL2 の存続期間中は一定です。バグやセキュリティの修正は、AL2 に付属GCCする のメジャーバージョンにバックポートされる場合があります。

デフォルトでは、AL2 にはバージョン 7.3 が含まれておりGCC、ほぼすべてのパッケージがビルドされます。`gcc10` パッケージでは GCC10 を限られた範囲で利用できますが、パッケージを構築するために 10 GCC を使用することはお勧めしません。

AL2 RPMsには、最適化フラグと強化フラグが含まれます。で独自のコードを構築する場合は、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GCC。

AL2023 のデフォルトのコンパイラと最適化フラグは、AL2 に存在するものを改善します。

AL2 で移動

AL2 で提供されているツールチェーンを使用して、Amazon Linux Goで に記述された独自のコードを構築できます。

Go ツールチェーンは AL2 の存続期間を通じて更新されます。これは、出荷するツールチェーン内の任意の CVE に応答しているか、別のパッケージ内の CVE に対処するための前提条件である可能性があります。

Go は比較的高速なプログラミング言語です。Go で記述された既存のアプリケーションを Go ツールチェーンの新しいバージョンに適応させる必要が生じる場合があります。Go の詳細については、「Go 1 and the Future of Go Programs」を参照してください。

AL2 はツールGoチェーンの存続期間中に新しいバージョンを組み込む予定ですが、これはアップストリームGoリリースではロックステップになりません。したがって、AL2 で提供されているGoツールチェーンの使用は、Go言語と標準ライブラリの最先端の機能を使用してGoコードを構築する場合に適している可能性があります。

AL2 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。以前のGoツールチェーンが必要な場合は、新しいGoツールチェーンのバグとセキュリティの修正を省略し、任意の RPM で使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できます。

AL2 で独自のGoコードを構築する場合は、このGoツールチェーンが AL2 の存続期間を通じて進む可能性があることを知って、AL2 に含まれるツールチェーンを使用できます。

Java AL2 の

AL2 は、Amazon Corretto のいくつかのバージョンを提供し、Javaベースのワークロードと一部のOpenJDKバージョンをサポートします。AL2023 への移行に備えて、Amazon Corretto に移行することをお勧めします。

Corretto は、Open Java Development Kit (OpenJDK) のビルドで、Amazon による長期サポートが付属しています。Corretto は Java Technical Compatibility Kit (TCK) を使用して認定されており、SE Java 標準を満たしておりLinux、、Windows、および で利用できますmacOS。

Amazon Corretto パッケージは、Corretto 1.8.0、Corretto 11、Corretto 17 のそれぞれで使用できます。

AL2 の各 Corretto バージョンは、Corretto バージョンと同じ期間、または AL2 の有効期限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 Corretto のFAQs」を参照してください。

Perl AL2 の

AL2 はPerl、プログラミング言語のバージョン 5.16 を提供します。

Perl AL2 のモジュール

AL2 では、さまざまなPerlモジュールRPMs としてパッケージ化されています。RPMs として利用できるPerlモジュールは多数ありますが、Amazon Linux はすべての可能なPerlモジュールをパッケージ化しようとはしません。RPMs 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 パッケージに依存する可能性があるため、Amazon Linux は純粋な機能更新よりもセキュリティパッチが適用されていることを確認することを優先します。

AL2 には もCPAN含まれているため、Perl開発者はPerlモジュールにイディオマティックパッケージマネージャーを使用できます。

PHP AL2 の

AL2 は現在、 の一部として PHP プログラミング言語の 2 つの完全にサポートされているバージョンを提供していますAL2 Extras ライブラリ。各PHPバージョンは、「」の廃止日に記載されているPHPアップストリームと同じ時間枠でサポートされていますAmazon Linux 2 Extras のリスト。

AL2 Extras を使用してインスタンスにアプリケーションとソフトウェアの更新をインストールする方法については、「」を参照してくださいAL2 Extras ライブラリ。

AL2023 への移行を支援するために、8.1 PHP と 8.2 の両方が AL2 と AL2023 で利用できます。

> **Note**
>
> AL2 には、`amazon-linux-extras` に PHP 7.1、7.2、7.3、および 7.4 が含まれています。これらの Extras はすべて EOL であり、追加のセキュリティ更新プログラムを取得する保証はありません。
> AL2 で の各バージョンPHPがいつ廃止されるかを確認するには、「」を参照してくださいAmazon Linux 2 Extras のリスト。

以前の 8.x PHP バージョンからの移行

アップストリームPHPコミュニティは、 8.1 から PHP 8.2 PHP に移行するための包括的な移行ドキュメントをまとめました。8.0 から PHP 8.1 への移行に関するドキュメントも存在します。

AL2 にはPHP、AL2023 への効率的なアップグレードパスamazon-linux-extrasを可能にする 8.0、8.1、および 8.2 が含まれています。 AL2023 AL2 で の各バージョンPHPがいつ廃止されるかを確認するには、「」を参照してくださいAmazon Linux 2 Extras のリスト。

PHP 7.x バージョンからの移行

アップストリームの PHP コミュニティでは、PHP 7.4 から PHP 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ています。8.1 および PHP 8.PHP2 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ると、 PHP ベースのアプリケーションを最新の に移行するために必要なすべてのステップがありますPHP。

PHP プロジェクトは、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保持します。

> **Note**
>
> AL2023 がリリースされたとき、 のすべての 7.x および 5.x バージョンPHPはPHPコミュニティでサポートされておらず、AL2023 のオプションとして含まれていませんでした。

Python AL2 の

AL2 は、AL2 Python コアパッケージの長期サポートコミットメントの一環として、2026 年 6 月まで AL2.7 のサポートとセキュリティパッチを提供します。このサポートは、2020 年 Python 1 月の 2.7 EOL というアップストリームPythonコミュニティ宣言を超えています。

> **Note**
>
> AL2023 は 2.7 Python を完全に削除しました。を必要とするコンポーネントPythonは、3 Python で動作するように記述されるようになりました。

AL2 は 2.7 Python に強く依存する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 3 Python に移行され、2.7 Python が不要になりました。AL2023 は 3 Python に完全に移行しました。3 Python への移行を完了することをお勧めします。

AL2 の錆

AL2 で提供されているツールチェーンを使用して、AL2 Rustで に記述された独自のコードを構築できます。

Rust ツールチェーンは AL2 の存続期間を通じて更新されます。これは、出荷するツールチェーンの CVE に応答しているか、別のパッケージの CVE 更新の前提条件として指定されている可能性があります。

Rust は比較的動きの速い言語であり、新しいリリースは約 6 週間間隔で行われます。新しいリリースでは、新しい言語または標準ライブラリ機能が追加される場合があります。AL2 はツールRustチェーンの存続期間中に新しいバージョンを組み込みますが、これはアップストリームRustリリースではロックステップになりません。したがって、AL2 で提供されているRustツールチェーンの使用は、Rust言語の最先端の機能を使用してRustコードを構築する場合に適している可能性があります。

AL2 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。以前のRustツールチェーンが必要な場合は、新しい Rustツールチェーンのバグとセキュリティの修正を省略し、任意の RPM で使用できるのと同じプロセス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できます。

AL2 で独自のRustコードを構築するには、AL2 に含まれるRustツールチェーンを使用し、このツールチェーンが AL2 の存続期間を通じて進む可能性があることを知ってください。

AL2 カーネル

AL2 には当初 4.14 カーネルが付属しており、バージョン 5.10 が現在のデフォルトとなっています。まだ 4.14 カーネルを使用している場合は、 5.10 カーネルに移行することをお勧めします。

カーネルライブパッチは AL2 でサポートされています。

トピック

AL2 でサポートされているカーネル

[Supported kernel versions] (サポートされているカーネルバージョン)

現在、AL2 AMIs はカーネルバージョン 4.14 および 5.10 で使用でき、バージョン 5.10 がデフォルトです。カーネル 5.10 で AL2 AMI を使用することをお勧めします。

AL2023 AMI はカーネルバージョン 6.1 で使用できます。詳細については、Amazon Linux AL2023 ユーザーガイドの「AL2 からの AL2023 カーネルの変更」を参照してください。

[Support Timeframe] (サポート時間枠)

AL2 で利用可能な 5.10 カーネルは、AL2 AMI が標準サポートを終了するまでサポートされます。

[Live patching support] (ライブパッチサポート)

AL2 カーネルバージョン	カーネルライブパッチのサポート
4.14	はい
5.10	はい
5.15	いいえ

AL2 でのカーネルライブパッチ適用

> **Important**
>
> Amazon Linux は、AL2 カーネル 4.14 のライブパッチ適用を 2025-10-31 に終了します。お客様は、AL2 のデフォルトカーネルとしてカーネル 5.10 を使用するか (AL2 でサポートされているカーネルを参照)、カーネル 6.1 および 6.12 を使用して AL2023 に移行することをお勧めします。
> Amazon Linux は、2026-06-30 に AL2 の有効期限が切れるまで、AL2 カーネル 5.10 のライブパッチを提供します。

Kernel Live Patching for AL2 を使用すると、実行中のアプリケーションを再起動または中断することなく、実行中の Linux カーネルに特定のセキュリティ脆弱性と重大なバグパッチを適用できます。これにより、システムの再起動までこれらの修正を適用しながら、サービスと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。

AL2023 のカーネルライブパッチの詳細については、Amazon Linux 2AL2023の「AL2023 でのカーネルライブパッチ」を参照してください。

AWS は、AL2 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。場合によっては、CVE が割り当てられる前に更新を提供する AWS ことがあります。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。
- [Bug fixes] (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 は、AL2 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。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、新しいカーネルバージョンに更新する必要があります。

AL2 カーネルライブパッチは、既存の AL2 リポジトリで署名付き RPM パッケージとして利用できます。パッチは、既存の yum ワークフローを使用して個々のインスタンスにインストールすることも、 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。

AL2 で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

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 インスタンスと AL2 を実行するオンプレミス仮想マシンでサポートされています。 AL2

AL2 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。

- `x86_64` アーキテクチャのカーネルバージョン `4.14` または `5.10`
- `ARM64` アーキテクチャのカーネルバージョン `5.10`

ポリシーの要件

Amazon Linux 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 がサービス所有の Amazon S3 バケットにアクセスする必要があります。環境で Amazon S3 用に Amazon Virtual Private Cloud (VPC) エンドポイントを使用している場合、VPC エンドポイントポリシーがそれらのパブリックバケットへのアクセスを許可しているようにする必要があります。

この表は、EC2 が Kernel Live Patching のためにアクセスする必要がある可能性がある各 Amazon S3 バケットを示しています。

S3 バケット ARN	説明
arn:aws:s3:::packages.*region*.amazonaws.com/*	Amazon Linux AMI パッケージを含む Amazon S3 バケット
arn:aws:s3:::repo.*region*.amazonaws.com/*	Amazon Linux AMI リポジトリを含む Amazon S3 バケット

S3 バケット ARN	説明
arn:aws:s3:::amazonlinux.*region*.amazonaws.com/*	AL2 リポジトリを含む Amazon S3 バケット
arn:aws:s3:::amazonlinux-2-repos-*region*/*	AL2 リポジトリを含む Amazon S3 バケット

次のポリシーは、組織に属する ID とリソースへのアクセスを制限し、Kernel Live Patching に必要な Amazon S3 バケットへのアクセスを提供する方法を示しています。*region*、*principal-org-id*、*resource-org-id* を組織の値に置き換えます。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRequestsByOrgsIdentitiesToOrgsResources",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "principal-org-id",
          "aws:ResourceOrgID": "resource-org-id"
        }
      }
    },
    {
      "Sid": "AllowAccessToAmazonLinuxAMIRepositories",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "s3:GetObject"
      ],
```

```
      "Resource": [
        "arn:aws:s3:::packages.region.amazonaws.com/*",
        "arn:aws:s3:::repo.region.amazonaws.com/*",
        "arn:aws:s3:::amazonlinux.region.amazonaws.com/*",
        "arn:aws:s3:::amazonlinux-2-repos-region/*"
      ]
    }
  ]
}
```

カーネルライブパッチを使用する

インスタンス自体のコマンドラインを使用して、個々のインスタンスでカーネルライブパッチを有効にして使用したり、 AWS Systems Manager を使用して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したりできます。

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する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のカーネルライブパッチの有効化と使用の詳細については、AWS Systems Manager ユーザーガイドのAL2 インスタンスでのカーネルライブパッチの使用」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の適用
- 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

カーネルライブパッチは、AL2 ではデフォルトで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 yum 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。

前提条件

カーネルライブパッチには binutils が必要です。binutils がインストールされていない場合は、以下のコマンドを使用してインストールします。

```
$ sudo yum install binutils
```

カーネルライブパッチを有効にするには

1. カーネルライブパッチは、次の AL2 カーネルバージョンで使用できます。

   - x86_64 アーキテクチャのカーネルバージョン 4.14 または 5.10
   - ARM64 アーキテクチャのカーネルバージョン 5.10

   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 yum list kernel
   ```

2. サポートされているカーネルバージョンが既にある場合は、この手順はスキップしてください。サポートされているカーネルバージョンがない場合は、次のコマンドを実行して、カーネルを最新バージョンに更新し、インスタンスを再起動します。

   ```
   $ sudo yum install -y kernel
   ```

   ```
   $ sudo reboot
   ```

3. カーネルライブパッチの yum プラグインをインストールします。

   ```
   $ sudo yum install -y yum-plugin-kernel-livepatch
   ```

4. カーネルライブパッチの yum プラグインを有効にします。

   ```
   $ sudo yum kernel-livepatch enable -y
   ```

   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。

5. カーネルライブパッチの yum 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、次のコマンドを実行します。

```
$ rpm -qa | grep kernel-livepatch
```

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。以下は出力の例です。

```
yum-plugin-kernel-livepatch-1.0-0.11.amzn2.noarch
kernel-livepatch-5.10.102-99.473-1.0-0.amzn2.x86_64
```

6. kpatch パッケージをインストールします。

```
$ sudo yum install -y kpatch-runtime
```

7. 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 yum update kpatch-runtime
```

8. 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
```

9. AL2 Extras Library でカーネルライブパッチトピックを有効にします。このトピックには、カーネルライブパッチが含まれています。

```
$ sudo amazon-linux-extras enable livepatch
```

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 AL2 セキュリティアラートの詳細については、Amazon Linux セキュリティセンター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

次のコマンドを使用します。

```
$ yum updateinfo list
```

出力例を次に示します。

```
Loaded plugins: extras_suggestions, kernel-livepatch, langpacks, priorities, update-
motd
ALAS2LIVEPATCH-2020-002 important/Sec. kernel-
livepatch-5.10.102-99.473-1.0-3.amzn2.x86_64
ALAS2LIVEPATCH-2020-005 medium/Sec. kernel-livepatch-5.10.102-99.473-1.0-4.amzn2.x86_64
updateinfo list done
```

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

次のコマンドを使用します。

```
$ yum updateinfo list cves
```

出力例を次に示します。

```
Loaded plugins: extras_suggestions, kernel-livepatch, langpacks, priorities, update-
motdamzn2-core/2/x86_64 | 2.4 kB 00:00:00
CVE-2019-15918 important/Sec. kernel-livepatch-5.10.102-99.473-1.0-3.amzn2.x86_64
CVE-2019-20096 important/Sec. kernel-livepatch-5.10.102-99.473-1.0-3.amzn2.x86_64
CVE-2020-8648 medium/Sec. kernel-livepatch-5.10.102-99.473-1.0-4.amzn2.x86_64
updateinfo list done
```

カーネルライブパッチの適用

カーネルライブパッチは、yum 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用の yum プラグインは、適用可能なカーネルライブパッチを管理します。

> **Tip**
>
> システムを再起動できるようになるまで、重要かつ重大なセキュリティ修正を確実に適用するために、カーネルライブパッチを利用してカーネルを定期的に更新することをお勧めしま

す。ライブパッチとしてデプロイ追加の修正がネイティブのカーネルパッケージに含まれているかどうかも確認し、その場合はカーネルを更新および再起動してください。

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. AL2 カーネルにカーネルライブパッチを適用します。

```
$ sudo yum install kernel-livepatch-kernel_version.x86_64
```

たとえば、次のコマンドは AL2 カーネルバージョン のカーネルライブパッチを適用します5.10.102-99.473。

```
$ sudo yum install kernel-livepatch-5.10.102-99.473-1.0-4.amzn2.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 yum 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。
- AL2 カーネルは、カーネルライブパッチを 3 か月間受け取ります。3 か月が過ぎると、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。これにより、次の 3 か月も引き

続きカーネルライブパッチを入手することができます。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、yum kernel-livepatch supported を実行します。

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。

```
$ kpatch list
```

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。

```
Loaded patch modules:
livepatch_cifs_lease_buffer_len [enabled]
livepatch_CVE_2019_20096 [enabled]
livepatch_CVE_2020_8648 [enabled]

Installed patch modules:
livepatch_cifs_lease_buffer_len (5.10.102-99.473.amzn2.x86_64)
livepatch_CVE_2019_20096 (5.10.102-99.473.amzn2.x86_64)
livepatch_CVE_2020_8648 (5.10.102-99.473.amzn2.x86_64)
```

Note

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。

カーネルライブパッチを無効にするには

1. 適用されたカーネルライブパッチの RPM パッケージを削除します。

   ```
   $ sudo yum kernel-livepatch disable
   ```

2. カーネルライブパッチの yum プラグインをアンインストールします。

```
$ sudo yum remove yum-plugin-kernel-livepatch
```

3. インスタンスを再起動します。

```
$ sudo reboot
```

制限事項

カーネルライブパッチには以下の制限があります。

- 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。
-

> **Note**
>
> 技術的な制限により、ライブパッチの適用では一部の問題に対処できません。このため、これらの修正はカーネルライブパッチパッケージに含まれず、ネイティブカーネルパッケージの更新でのみ提供されます。ネイティブカーネルパッケージの更新をインストールし、システムを再起動して、通常どおりパッチをアクティブ化できます。

よくある質問

AL2 のカーネルライブパッチに関するよくある質問については、「Amazon Linux 2 カーネルライブパッチに関するよくある質問」を参照してください。

AL2 Extras ライブラリ

> **Warning**
>
> epel Extra はサードパーティーリポジトリを有効にしますEPEL7。2024-06-30 の時点で、サードパーティーの EPEL7 リポジトリは維持されなくなりました。
> このサードパーティーのリポジトリは 今後更新されません。つまり、EPEL リポジトリ内のパッケージのセキュリティ修正はありません。
> 一部のEPELパッケージのオプションについては、EPEL「Amazon Linux 2023 ユーザーガイド」の「」セクションを参照してください。

AL2 では、Extras Library を使用して、インスタンスにアプリケーションとソフトウェアの更新をインストールできます。このようなソフトウェア更新は、トピックと呼ばれます。特定のバージョンのトピックをインストールしたり、最新バージョンを使用するためにバージョン情報を省略したりすることができます。追加により、オペレーティングシステムの安定性と利用可能なソフトウェアの鮮度の間で妥協する必要が軽減されます。

Extras トピックの内容は、長期サポートとバイナリ互換性に関する Amazon Linux ポリシーから除外されます。Extras トピックでは、厳選されたパッケージのリストにアクセスできます。パッケージのバージョンは頻繁に更新されるか、AL2 と同じ時間サポートされない場合があります。

> **Note**
>
> 個々の Extras トピックは、AL2 が EOL に達する前に廃止される可能性があります。

使用可能なトピックを一覧表示するには、次のコマンドを使用します。

```
[ec2-user ~]$ amazon-linux-extras list
```

トピックを有効にし、そのパッケージの最新バージョンをインストールして鮮度を確保するには、次のコマンドを使用します。

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

トピックを有効にし、特定のバージョンのパッケージをインストールして安定性を確保するには、次のコマンドを使用します。

```
[ec2-user ~]$ sudo amazon-linux-extras install topic=version topic=version
```

トピックからインストールされたパッケージを削除するには、次のコマンドを使用します。

```
[ec2-user ~]$ sudo yum remove $(yum list installed | grep amzn2extra-topic | awk
 '{ print $1 }')
```

> **Note**
>
> このコマンドは、 Extra の依存関係としてインストールされたパッケージを削除しません。

トピックを無効にし、パッケージを yum パッケージマネージャーにアクセスできないようにするには、次のコマンドを使用します。

```
[ec2-user ~]$ sudo amazon-linux-extras disable topic
```

> **Important**
>
> このコマンドは、上級ユーザーを対象としています。このコマンドの使用が不適切な場合、パッケージ互換性の競合が発生する可能性があります。

Amazon Linux 2 Extras のリスト

追加名	廃止日
BCC	
GraphicsMagick1.3	
R3.4	
R4	
ansible2	2023-09-30
aws-nitro-enclaves-cli	

追加名	廃止日
awscli1	
collectd	
collectd-python3	
corretto8	
dnsmasq	
dnsmasq2.85	2025-05-01
docker	
Ecs	
emacs	2018 年 11 月 14 日
Epel	2024-06-30
Firecracker	2022-11-08
Firefox	
gimp	2018 年 11 月 14 日
golang1.11	2023-08-01
golang1.19	2023-09-30
golang1.9	2018-12-14
ハプロキシ2	
httpd_modules	
java-openjdk11	2024-09-30
kernel-5.10	

追加名	廃止日
カーネル-5.15	
kernel-5.4	
kernel-ng	2022-08-08
lamp-mariadb10.2-php7.2	2020-11-30
libreoffice	
livepatch	
lustre	
lustre2.10	
ライニス	
mariadb10.5	2025-06-24
mate-desktop1.x	
memcached1.5	
mock	
モック2	
mono	
nano	2018 年 11 月 14 日
nginx1	
nginx1.12	2019-09-20
nginx1.22.1	
php7.1	2020-01-15

追加名	廃止日
php7.2	2020-11-30
php7.3	2021-12-06
php7.4	2022-11-03
php8.0	2023-11-26
php8.1	2025-12-31
php8.2	
postgresql10	2023-09-30
postgresql11	2023-11-09
postgresql12	2024-11-14
postgresql13	2025-11-13
postgresql14	
postgresql9.6	2022-08-09
python3	2018-08-22
python3.8	2024-10-14
redis4.0	2021-05-25
redis6	2026-01-31
ruby2.4	2020-08-27
ruby2.6	2023-03-31
ruby3.0	2024-03-31
rust1	2025-05-01

追加名	廃止日
selinux-ng	
squid4	2023-09-30
テスト	
tomcat8.5	2024-03-31
tomcat9	
unbound1.13	2025-05-01
unbound1.17	
vim	2018 年 11 月 14 日

AL2 リザーブドユーザーとグループ

AL2 は、イメージのプロビジョニング中と特定のパッケージのインストール中に、特定のユーザーとグループを事前に割り当てます。競合を防ぐために、ユーザー、グループ、および関連する UID と GID の一覧をここに示します。

トピック

Amazon Linux 2 リザーブドユーザーのリスト

UID で一覧表示

ユーザー名	UID
root	0
bin (ビン)	1
daemon	2
adm	3
lp	4
sync	5
シャットダウン	6
停止	7
mail	8
uucp	10
オペレーター	11
games	12

ユーザー名	UID
ftp	14
oprofile	16
pkiuser	17
squid	23
named	25
postgres	26
MySql	27
nscd	28
nscd	28
rpcuser	29
rpc	32
amandabackup	33
ntp	38
メールマン	41
gdm	42
mailnull	47
apache	48
smmsp	51
tomcat	53
ldap	55

ユーザー名	UID
tss	59
nslcd	65
Pegasus	66
avahi	70
tcpdump	72
sshd	74
radvd	75
サイラス	76
arpwatch	77
fax	78
dbus	81
postfix	89
quagga	92
半径	95
半径	95
hsqldb	96
dovecot	97
ID	98
nobody	99
qemu	107

ユーザー名	UID
usbmuxd	113
stap-server	155
avahi-autoipd	170
pulse	171
rtkit	172
dhcpd	177
sanlock	179
haproxy	188
hacluster	189
systemd-journal-gateway	191
systemd-network	192
systemd-resolve	193
uuidd	357
tang	358
stapdev	359
stapsys	360
stapusr	361
systemd-journal-upload	362
systemd-journal-remote	363
サニッシュ済み	364

ユーザー名	UID
pesign	365
pcpqa	366
PCP	367
memcached	368
イプシロン	369
ipaapi	370
kdcproxy	371
ods	372
sssd	373
クラスター	374
フェデフ	375
dovenull	376
coroqnetd	377
クレビス	378
clamscan	379
clamilt	380
clamupdate	381
colord	382
geoclue	383
aws-kinesis-agent-user	384

ユーザー名	UID
CWAgent	385
unbound	386
polkitd	387
saslauth	388
dirsrv	389
chrony	996
ec2-instance-connect	997
rngd	998
libstoragemgmt	999
ec2-user	1,000
nfsnobody	65534

名前順に一覧表示

ユーザー名	UID
adm	3
amandabackup	33
apache	48
arpwatch	77
avahi	70
avahi-autoipd	170

ユーザー名	UID
aws-kinesis-agent-user	384
bin (ビン)	1
chrony	996
clamilt	380
clamscan	379
clamupdate	381
クレビス	378
colord	382
coroqnetd	377
CWAgent	385
サイラス	76
daemon	2
dbus	81
dhcpd	177
dirsrv	389
dovecot	97
dovenull	376
ec2-instance-connect	997
ec2-user	1,000
fax	78

ユーザー名	UID
フェデフ	375
ftp	14
games	12
gdm	42
geoclue	383
クラスター	374
hacluster	189
停止	7
haproxy	188
hsqldb	96
ID	98
ipaapi	370
イプシロン	369
kdcproxy	371
ldap	55
libstoragemgmt	999
lp	4
mail	8
メールマン	41
mailnull	47

ユーザー名	UID
memcached	368
MySql	27
named	25
nfsnobody	65534
nobody	99
nscd	28
nscd	28
nslcd	65
ntp	38
ods	372
オペレーター	11
oprofile	16
PCP	367
pcpqa	366
Pegasus	66
pesign	365
pkiuser	17
polkitd	387
postfix	89
postgres	26

ユーザー名	UID
pulse	171
qemu	107
quagga	92
半径	95
半径	95
radvd	75
rngd	998
root	0
rpc	32
rpcuser	29
rtkit	172
サニッシュ済み	364
sanlock	179
saslauth	388
シャットダウン	6
smmsp	51
squid	23
sshd	74
sssd	373
stap-server	155

ユーザー名	UID
stapdev	359
stapsys	360
stapusr	361
sync	5
systemd-journal-gateway	191
systemd-journal-remote	363
systemd-journal-upload	362
systemd-network	192
systemd-resolve	193
tang	358
tcpdump	72
tomcat	53
tss	59
unbound	386
usbmuxd	113
uucp	10
uuidd	357

Amazon Linux 2 リザーブドグループのリスト

GID で一覧表示

グループ名	GID
root	0
bin (ビン)	1
daemon	2
sys	3
adm	4
tty	5
disk	6
disk	6
lp	7
mem	8
kmem	9
wheel	10
cdrom	11
mail	12
uucp	14
man	15
oprofile	16
pkiuser	17
dialout	18
floppy	19

グループ名	GID
games	20
slocate	21
utmp	22
squid	23
named	25
postgres	26
MySql	27
nscd	28
nscd	28
rpcuser	29
rpc	32
tape	33
tape	33
utempter	35
kvm	36
ntp	38
video	39
ディップ	40
メールマン	41
gdm	42

グループ名	GID
mailnull	47
apache	48
ftp	50
smmsp	51
tomcat	53
lock	54
ldap	55
tss	59
audio	63
Pegasus	65
avahi	70
tcpdump	72
sshd	74
radvd	75
saslauth	76
saslauth	76
arpwatch	77
fax	78
dbus	81
screen	84

グループ名	GID
quaggavt	85
wbpriv	88
wbpriv	88
postfix	89
postdrop	90
quagga	92
半径	95
半径	95
hsqldb	96
dovecot	97
ID	98
nobody	99
users	100
qemu	107
usbmuxd	113
stap-server	155
stapusr	156
stapusr	156
stapsys	157
stapdev	158

グループ名	GID
avahi-autoipd	170
pulse	171
rtkit	172
dhcpd	177
sanlock	179
haproxy	188
haclient	189
systemd-journal	190
systemd-journal	190
systemd-journal-gateway	191
systemd-network	192
systemd-resolve	193
usbmon	351
wireshark	352
uuidd	353
tang	354
systemd-journal-upload	355
sfcb	356
systemd-journal-remote	356
サニッシュ済み	357

グループ名	GID
pesign	358
pcpqa	359
PCP	360
memcached	361
virtlogin	362
イプシロン	363
pkcs11	364
ipaapi	365
kdcproxy	366
ods	367
sssd	368
libvirt	369
クラスター	370
フェデフ	371
dovenull	372
docker	373
coroqnetd	374
クレビス	375
clamscan	376
clamilt	377

グループ名	GID
virusgroup	378
virusgroup	378
virusgroup	378
clamupdate	379
colord	380
geoclue	381
printadmin	382
aws-kinesis-agent-user	383
CWAgent	384
pulse-rt	385
pulse-access	386
unbound	387
polkitd	388
dirsrv	389
cgred	993
chrony	994
ec2-instance-connect	995
rngd	996
libstoragemgmt	997
ssh_keys	998

グループ名	GID
input	999
ec2-user	1,000
nfsnobody	65534

名前順に一覧表示

グループ名	GID
adm	4
apache	48
arpwatch	77
audio	63
avahi	70
avahi-autoipd	170
aws-kinesis-agent-user	383
bin (ビン)	1
cdrom	11
cgred	993
chrony	994
clamilt	377
clamscan	376
clamupdate	379

グループ名	GID
クレビス	375
colord	380
coroqnetd	374
CWAgent	384
daemon	2
dbus	81
dhcpd	177
dialout	18
ディップ	40
dirsrv	389
disk	6
disk	6
docker	373
dovecot	97
dovenull	372
ec2-instance-connect	995
ec2-user	1,000
fax	78
フェデフ	371
floppy	19

グループ名	GID
ftp	50
games	20
gdm	42
geoclue	381
クラスター	370
haclient	189
haproxy	188
hsqldb	96
ID	98
input	999
ipaapi	365
イプシロン	363
kdcproxy	366
kmem	9
kvm	36
ldap	55
libstoragemgmt	997
libvirt	369
lock	54
lp	7

グループ名	GID
mail	12
メールマン	41
mailnull	47
man	15
mem	8
memcached	361
MySql	27
named	25
nfsnobody	65534
nobody	99
nscd	28
nscd	28
ntp	38
ods	367
oprofile	16
PCP	360
pcpqa	359
Pegasus	65
pesign	358
pkcs11	364

グループ名	GID
pkiuser	17
polkitd	388
postdrop	90
postfix	89
postgres	26
printadmin	382
pulse	171
pulse-access	386
pulse-rt	385
qemu	107
quagga	92
quaggavt	85
半径	95
半径	95
radvd	75
rngd	996
root	0
rpc	32
rpcuser	29
rtkit	172

グループ名	GID
サニッシュ済み	357
sanlock	179
saslauth	76
saslauth	76
screen	84
sfcb	356
slocate	21
smmsp	51
squid	23
ssh_keys	998
sshd	74
sssd	368
stap-server	155
stapdev	158
stapsys	157
stapusr	156
stapusr	156
sys	3
systemd-journal	190
systemd-journal	190

グループ名	GID
systemd-journal-gateway	191
systemd-journal-remote	356
systemd-journal-upload	355
systemd-network	192
systemd-resolve	193
tang	354
tape	33
tape	33
tcpdump	72
tomcat	53
tss	59
tty	5
unbound	387
usbmon	351
usbmuxd	113
users	100
utempter	35
utmp	22
uucp	14
uuidd	353

グループ名	GID
video	39
virtlogin	362
virusgroup	378
virusgroup	378
virusgroup	378
wbpriv	88
wbpriv	88
wheel	10
wireshark	352

AL2 ソースパッケージ

Amazon Linux で提供されているツールを使用して、インスタンスにインストールしたパッケージソースを参照するために表示できます。ソースパッケージは、Amazon Linux およびオンラインパッケージリポジトリに含まれるすべてのパッケージで利用できます。インストールするソースパッケージのパッケージ名を決定し、 yumdownloader --source コマンドを使用して実行中のインスタンス内のソースを表示します。例えば、次のようになります。

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

ソース RPM は解凍でき、参照用に標準の RPM ツールを使用してソースツリーを表示できます。デバッグが完了したら、パッケージを利用できます。

AL2 のセキュリティとコンプライアンス

で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を活用できます。

セキュリティは、 AWSとお客様の間の責任共有です。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 – AWS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AWS責任があります。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。サードパーティーの監査者は、AWSコンプライアンスプログラムコンプライアンスプログラムの一環として、当社のセキュリティの有効性を定期的にテストおよび検証。AL2023 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。

AL2 で FIPS モードを有効にする

このセクションでは、AL2 で連邦情報処理標準 (FIPS) を有効にする方法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

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 AL2 Amazon EC2 インスタンス。AL2 Amazon EC2 インスタンスの起動の詳細については、「」を参照してくださいAmazon EC2 の AL2 Amazon EC2。
- 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。

> **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「キーペアを作成する」を参照してください。

FIPS モードの有効化

1. SSH または を使用して AL2 インスタンスに接続しますAWS Systems Manager。
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、「パッケージリポジトリ」を参照してください。
3. 次のコマンドを実行して、 `dracut-fips`モジュールをインストールして有効にします。

   ```
   sudo yum -y install dracut-fips
   sudo dracut -f
   ```

4. 次のコマンドを使用して、Linux カーネルコマンドラインで FIPS モードを有効にします。これにより、AL2 FAQ に記載されているモジュールに対してシステム全体で FIPS モードが有効になります。

   ```
   sudo /sbin/grubby --update-kernel=ALL --args="fips=1"
   ```

5. AL2 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。

   ```
   sysctl crypto.fips_enabled
   ```

   以下の出力が表示されます。

   ```
   crypto.fips_enabled = 1
   ```

次のコマンドを実行して、OpenSSH が FIPS モードであることを確認します。

```
ssh localhost 2>&1 | grep FIPS
```

以下の出力が表示されます。

```
FIPS mode initialized
```

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。