
AWS CloudShell

ユーザーガイド

Amazon の商標およびトレードドレスは、お客様に混乱を招く可能性が高い方法、または Amazon の評判もしくは信用を損なう方法で、Amazon が所有しない製品またはサービスと関連付けて使用することはできません。Amazon が所有しない商標はすべてそれぞれの所有者に所属します。所有者は必ずしも Amazon との提携や関連があるわけではありません。また、Amazon のサポートを受けているとはかぎりません。

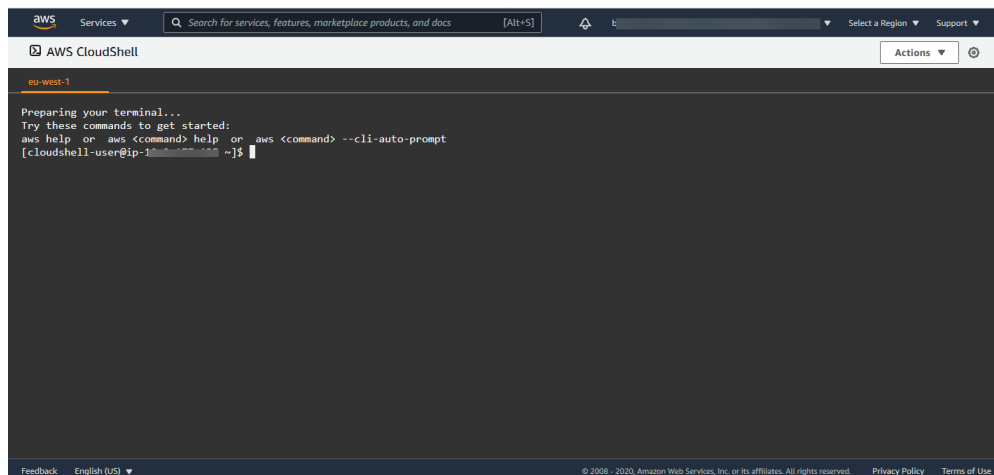
Table of Contents

| | |
|--|----|
| AWS CloudShell とは | 1 |
| AWS CloudShell の機能 | 1 |
| AWS コマンドラインインターフェイス | 1 |
| シェルと開発ツール | 1 |
| 永続的ストレージ | 2 |
| Security | 2 |
| カスタマイズオプション | 2 |
| Pricing | 2 |
| 使用を開始する方法をお知りになりたいですか? | 3 |
| キーAWS CloudShellトピック | 3 |
| よくある質問 | 3 |
| 開始する方法AWS CloudShell? | 3 |
| アクセスするにはどのような権限が必要ですか?AWS CloudShell? | 4 |
| どのAWSリージョンはAWS CloudShellはで使用可能に | 4 |
| どのタイプのシェルを使用できますかAWS CloudShell? | 4 |
| 使用できるWebブラウザはAWS CloudShell? | 4 |
| シェル環境にはどのようなソフトウェアがプリインストールされていますか? | 4 |
| シェル環境では利用できないソフトウェアをインストールできますか? | 5 |
| ユーザーがで実行可能なアクションを制限できますか?AWS CloudShell? | 5 |
| 変更したい場合、ホームディレクトリからデータを移動するにはどうすればいいですか?AWS 私が使用している地域AWS CloudShell? | 5 |
| 入門チュートリアル | 6 |
| Prerequisites | 6 |
| Contents | 6 |
| ステップ 1: にサインインするAWS Management Console | 7 |
| ステップ 2: を起動するAWS CloudShellで、[リージョン] を選択し、シェルを選択します。 | 8 |
| ステップ 3: AWS CloudShell へのファイルのアップロード | 10 |
| ステップ 4: ファイルのコードを編集し、コマンドラインから実行します。 | 11 |
| ステップ 5: を使用するAWS CLI をクリックして、ファイルを Amazon S3 バケットにオブジェクト として追加します。 | 12 |
| 関連トピック | 12 |
| チュートリアル | 14 |
| チュートリアル: 複数のファイルをコピーする | 14 |
| Amazon S3 を使用した複数のファイルのアップロードとダウンロード | 14 |
| zip 形式のフォルダを使用した複数のファイルのアップロードとダウンロード | 16 |
| チュートリアル: CodeCommit の使用方法 | 17 |
| Prerequisites | 17 |
| ステップ 1: CodeCommit リポジトリの作成とクローン作成 | 18 |
| ステップ 2: CodeCommit リポジトリにプッシュする前にファイルをステージングしてコミット する | 18 |
| チュートリアル: 署名付き URL の作成 | 19 |
| Prerequisites | 19 |
| ステップ 1: IAM ロールを作成して Amazon S3 バケットへのアクセスを許可します。 | 19 |
| 署名付き URL の生成 | 20 |
| AWS CloudShell の使用 | 22 |
| を起動するAWS CloudShell | 22 |
| のナビゲーションAWS CloudShellインターフェイスからリクエスト | 22 |
| シェルの選択 | 23 |
| の使用AWSリージョン | 23 |
| デフォルトの指定AWSのリージョンAWS CLI | 24 |
| ファイルとストレージの操作 | 24 |
| シェルセッションの開始と終了 | 27 |
| の使用AWSサービス | 28 |
| AWS CLIのコマンドライン例AWSサービス | 28 |

| | |
|---|-----|
| DynamoDB | 28 |
| AWS Cloud9 | 29 |
| Amazon EC2 | 29 |
| S3 Glacier | 29 |
| AWSElastic Beanstalk CLI | 29 |
| Amazon ECS CLI | 30 |
| AWS SAM CLI | 30 |
| AWS CloudShell のカスタマイズ | 31 |
| コマンドライン表示を複数のタブに分割する | 31 |
| フォントサイズを変更する | 31 |
| インタフェースのテーマを変更する | 31 |
| マルチテキストにセーフペーストを使用する | 32 |
| セキュリティ | 2 |
| データ保護 | 33 |
| データ暗号化 | 34 |
| Identity and access management | 34 |
| Audience | 35 |
| アイデンティティを使用した認証 | 35 |
| 管理AWS CloudShellIAM ポリシーでのアクセスと使用 | 36 |
| ロギングとモニタリング | 40 |
| CloudTrail によるアクティビティのモニタリング | 40 |
| AWS CloudShellCloudTrail では | 40 |
| コンプライアンス検証 | 41 |
| 耐障害性 | 41 |
| インフラストラクチャセキュリティ | 42 |
| 設定と脆弱性の分析 | 42 |
| セキュリティベストプラクティス | 42 |
| AWS CloudShellコンピューティング環境 | 44 |
| コンピューティング環境リソース | 44 |
| プリインストールされたソフトウェア | 44 |
| Shells | 44 |
| AWSコマンドラインインターフェイス (CLI) | 45 |
| ランタイムと AWS SDK: Node.js と Python 3 | 46 |
| 開発ツールとシェルユーティリティ | 47 |
| のインストールAWS CLIホームディレクトリに移動します | 51 |
| シェル環境にサードパーティーソフトウェアをインストールする | 52 |
| スクリプトでシェルを修正する | 52 |
| ホームディレクトリの削除 | 53 |
| トラブルシューティング | 54 |
| を起動できませんAWS CloudShell「環境を開始できません。アクセス許可がありません。IAM 管理者 にAWS CloudShell。」 | 54 |
| アクセスできませんAWS CloudShellのコマンドラインを使用します。 | 54 |
| 外部 IP アドレスに ping を実行できません。 | 54 |
| サポートされるブラウザ | 56 |
| サポートされているリージョン | 57 |
| 制限 | 58 |
| 永続的ストレージ | 58 |
| 毎月の使用状況 | 58 |
| 同時シェル数 | 58 |
| シェルセッション | 59 |
| ネットワークアクセスとデータ転送 | 59 |
| システムファイルとページのリロードに関する制限 | 59 |
| ドキュメント履歴 | 60 |
| | lxi |

AWS CloudShell とは

AWS CloudShellはブラウザベースの事前認証済みシェルで、[を直接起動する \(p. 3\)](#)からのAWS Management Console。次のようにAWS CLIに対するコマンドAWSサービスを優先するシェル (Bash、PowerShell、または Z シェル) を使用してインストールします。また、コマンドラインツールをダウンロードまたはインストールする必要なく、これを行うことができます。



を起動するとAWS CloudShell、[コンピューティング環境 \(p. 44\)](#)が作成され、Amazon Linux 2 に基づいています。この環境内では、[広範なプリインストール開発ツール \(p. 44\)](#)、[ファイルのアップロードおよびダウンロードのオプション \(p. 24\)](#)、および[セッション間で持続するファイルストレージ \(p. 2\)](#)。

今すぐ試す: [チュートリアル: AWS CloudShell の開始方法 \(p. 6\)](#)。)

AWS CloudShell の機能

AWS コマンドラインインターフェイス

あなたが起動するAWS CloudShellからのAWS Management Consoleであり、AWSクレデンシャルは、新しいシェルセッションで自動的に利用できるようになります。この事前認証AWS CloudShellユーザーを使用すると、AWS CLI バージョン 2 (シェルのコンピューティング環境にプリインストール) を使用してAWS サービスを操作するときに、[認証情報の設定をスキップ](#)できます。

との対話の詳細についてはAWSサービスをコマンドラインインターフェイスを使用して実行する方法については、[の使用AWS内のサービスAWS CloudShell \(p. 28\)](#)。

シェルと開発ツール

用に作成されたシェルでAWS CloudShellセッションでは、[優先するコマンドラインシェル間でシームレスに切り替える \(p. 23\)](#)。具体的には、Bash、PowerShell、Z シェルを切り替えることができます。また、git、make、pip、sudo、tar、tmux、vim、wget、zipなどのプリインストールされたツールやユーティリティにアクセスできます。

シェル環境は、主要なソフトウェア言語をサポートして事前構成されており、たとえば、最初にランタイムインストールを実行しなくても、Node.js プロジェクトや Python プロジェクトを実行できます。PowerShellユーザーは、.NET Coreランタイムを使用できます。

で作成された、またはアップロードされたファイルAWS CloudShellによって管理されるリモートリポジトリにプッシュされる前に、ローカルリポジトリにコミットすることもできますAWS CodeCommit。

詳細については、「[AWS CloudShellコンピューティング環境:仕様とソフトウェア \(p. 44\)](#)」を参照してください。

永続的ストレージ

を使用する場合AWS CloudShellそれぞれに 1 GB の永続ストレージがありますAWSリージョンは追加料金なしでご利用いただけます。永続ストレージは、あなたのホームディレクトリ (\$HOME) であり、あなたにはプライベートです。各シェルセッションが終了した後にリサイクルされる一時的な環境リソースとは異なり、ホームディレクトリ内のデータはセッション間で保持されます。

永続ストレージでのデータの保持の詳細については、「[永続的ストレージ \(p. 58\)](#)」。

Security

-AWS CloudShell環境とそのユーザーは、IAM アクセス許可管理、シェルセッションの制限、テキスト入力用の安全な貼り付けなどの特定のセキュリティ機能によって保護されます。

IAM によるアクセス許可管理

管理者は、AWS CloudShellユーザーの IAM ポリシーを使用します。管理者は、ユーザーがシェル環境で実行できる特定のアクションを詳細なレベルで指定するポリシーを作成することもできます。詳細については、「[管理AWS CloudShellIAM ポリシーでのアクセスと使用 \(p. 36\)](#)」を参照してください。

シェルセッション管理

非アクティブなセッションと長時間実行セッションは自動的に停止され、リサイクルされます。詳細については、「[シェルセッション \(p. 59\)](#)」を参照してください。

テキスト入力用の安全な貼り付け

既定で有効になっているセーフペーストは、シェルに貼り付けようとしているマルチテキストに悪意のあるスクリプトが含まれていないことを確認するセキュリティ機能です。詳細については、「[マルチテキストにセーフペーストを使用する \(p. 32\)](#)」を参照してください。

カスタマイズオプション

お客様のAWS CloudShellエクスペリエンスは、画面レイアウト (複数のタブ)、テキストサイズ、明暗のインターフェイステーマを変更することでカスタマイズできます。詳細については、「[カスタマイズするAWS CloudShellエクスペリエンス \(p. 31\)](#)」を参照してください。

また、シェル環境を拡張することもできます[インストール, 独自のソフトウェア \(p. 52\)](#)および[変更, スタートアップシェルスクリプト \(p. 52\)](#)。

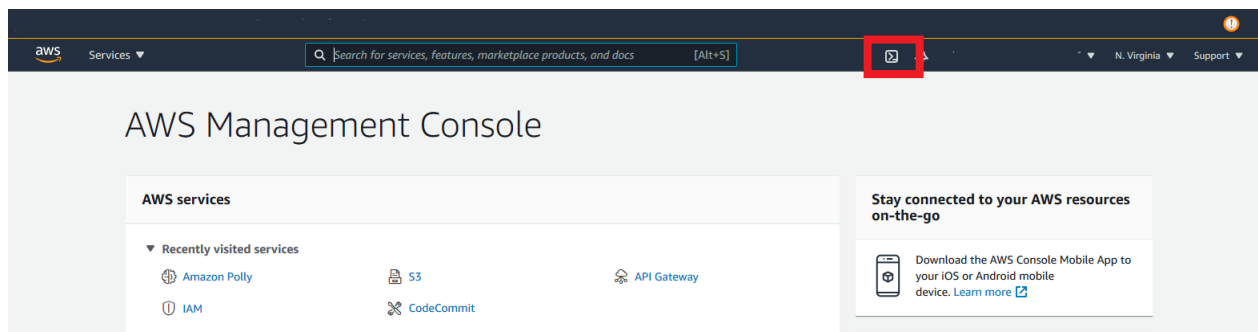
Pricing

AWS CloudShellである。AWSサービスは追加料金なしでご利用いただけます。あなたは他のもののために支払うAWSで実行するリソースAWS CloudShell。[標準データ転送速度](#)も適用されます。

詳細については、「[の制限と制約AWS CloudShell \(p. 58\)](#)」を参照してください。

使用を開始する方法をお知りになりたいですか？

シェルの操作を開始するには、AWS Management Consoleを選択し、AWS CloudShellホームページから。



へのログインのワークスルーについては、AWS Management Consoleを使用して主要なタスクを実行するAWS CloudShell「」を参照してください。チュートリアル: AWS CloudShell の開始方法 (p. 6)。

キーAWS CloudShellトピック

- チュートリアル: AWS CloudShell の開始方法 (p. 6)
- AWS CloudShell の使用 (p. 22)
- の使用AWS内のサービスAWS CloudShell (p. 28)
- カスタマイズするAWS CloudShellエクスペリエンス (p. 31)
- AWS CloudShellコンピューティング環境:仕様とソフトウェア (p. 44)

AWS CloudShellのよくある質問

これに関するよくある質問に対する回答AWSサービス。

- 開始する方法AWS CloudShell? (p. 3)
- アクセスするにはどのような権限が必要ですか?AWS CloudShell? (p. 4)
- どのAWSリージョンはAWS CloudShellはで使用可能に (p. 4)
- どのタイプのシェルを使用できますかAWS CloudShell? (p. 4)
- 使用できるWebブラウザはAWS CloudShell? (p. 4)
- シェル環境にはどのようなソフトウェアがプリインストールされていますか? (p. 4)
- シェル環境では利用できないソフトウェアをインストールできますか? (p. 5)
- ユーザーがで実行可能なアクションを制限できますか?AWS CloudShell? (p. 5)
- 変更したい場合、ホームディレクトリからデータを移動するにはどうすればいいですか?AWS私が使用している地域AWS CloudShell? (p. 5)

開始する方法AWS CloudShell?

起動することができますAWS CloudShellを 1 回クリックで開始AWS Management Console。始めるために必要なのは、コンソールを使用するAWSまたは IAM 認証情報<https://console.aws.amazon.com/console/home>。選択できるのは、AWS CloudShellリンクをクリックするか、サービスの検索ボックスに移動するとそのように表示されます。

詳細については、「チュートリアル: AWS CloudShell の開始方法 (p. 6)」を参照してください。

[よくある質問の一覧に戻る \(p. 3\)](#)

アクセスするにはどのような権限が必要ですか?AWS CloudShell?

あなたがアクセスするのでAWS CloudShellからのAWS Management Consoleを使用するには、有効なアカウントのエイリアスまたは ID、ユーザー名、パスワードを提供できる IAM ユーザーである必要があります。

を起動するにはAWS CloudShellをインストールするには、アタッチされたポリシーによって提供される IAM アクセス許可が必要です。詳細については、「[管理AWS CloudShellIAM ポリシーでのアクセスと使用 \(p. 36\)](#)」を参照してください。

[よくある質問の一覧に戻る \(p. 3\)](#)

どのAWSリージョンはAWS CloudShellはで使用可能に

サポートされているリストについては、AWSリージョンおよび関連付けられたサービスエンドポイントについては、『[AWS CloudShellページでのAmazon Web Services 全般リファレンス](#)』。

どのタイプのシェルを使用できますかAWS CloudShell?

コマンドの実行は、Bash シェル、PowerShell、または Z シェルから選択できます。特定のシェルに切り替えるには、コマンドプロンプトでシェルプログラムの名前を入力します。

- bash : Bashシェルを使用する
- pwshPowerShell を使用する
- zsh: Z シェルを使用します

[よくある質問の一覧に戻る \(p. 3\)](#)

使用できるWebブラウザはAWS CloudShell?

AWS CloudShellは、Google Chrome、Mozilla Firefox と Microsoft Edge、Apple Safari の3つの最新バージョンをサポートしています。

[よくある質問の一覧に戻る \(p. 3\)](#)

シェル環境にはどのようなソフトウェアがプリインストールされていますか？

AWS CloudShell セッション用に作成されたシェルを使用して、[優先するコマンドラインシェル間でシームレスに切り替える \(p. 23\)](#) (Bash、PowerShell、Z シェル)。また、Make、pip、sudo、tar、tmux、Vim、Wget、Zipなどのプリインストールされたツールやユーティリティにアクセスできます。

シェル環境は、主要なソフトウェア言語をサポートして事前に構成されています。たとえば、ランタイムインストールを実行しなくても、Node.js プロジェクトや Python プロジェクトを実行できます。PowerShellユーザーは、.NET Coreランタイムを使用できます。

シェルを使用して作成されたファイル、またはシェルインターフェイスでアップロードされたファイルは、事前にインストールされたバージョンの Git を使用して管理されるバージョン管理リポジトリに追加できます。

詳細については、「[プリインストールされたソフトウェア \(p. 44\)](#)」を参照してください。

[よくある質問の一覧に戻る \(p. 3\)](#)

シェル環境では利用できないソフトウェアをインストールできますか？

はい。AWS CloudShellユーザーは sudo 権限を持っているので、コマンドラインからソフトウェアをインストールするための管理者権限を持っています。詳細については、「[シェル環境にサードパーティソフトウェアをインストールする \(p. 52\)](#)」を参照してください。

[よくある質問の一覧に戻る \(p. 3\)](#)

ユーザーがで実行可能なアクションを制限できますか？AWS CloudShell?

はい。たとえば、ユーザーがアクセスできるようにすることができますAWS CloudShellを使用しますが、シェル環境内でファイルをアップロードまたはダウンロードすることはできません。また、それらにアクセスすることを完全に防ぐこともできますAWS CloudShell。詳細については、「[管理AWS CloudShellIAM ポリシーでのアクセスと使用 \(p. 36\)](#)」を参照してください。

[よくある質問の一覧に戻る \(p. 3\)](#)

変更したい場合、ホームディレクトリからデータを移動するにはどうすればいいですか？AWS私が使用している地域AWS CloudShell?

を移動するにはAWS CloudShell1 つのデータからAWSリージョンから別のリージョンに移動する場合は、リージョン A のホームディレクトリの内容をローカルマシンにダウンロードし、ローカルマシンからリージョン B のホームディレクトリにアップロードできます。利用可能なオプションの詳細については、「[チュートリアル: ローカルマシンととの間で複数のファイルをコピーするAWS CloudShell \(p. 14\)](#)」。

[よくある質問の一覧に戻る \(p. 3\)](#)

チュートリアル: AWS CloudShell の開始方法

この入門チュートリアルでは、の起動方法を示します。AWS CloudShell シェルコマンドラインインターフェイスを使用して主要なタスクを実行します。

まず、にサインインしますAWS Management Consoleを起動するAWS CloudShell新しいブラウザウィンドウに表示されます。次に、AWS使用するリージョンとシェルタイプ。

次に、ホームディレクトリに新しいフォルダを作成し、ローカルマシンからフォルダにファイルをアップロードします。コマンドラインからプログラムとして実行する前に、プリインストールされたエディタを使用してそのファイルを操作します。最後に、あなたは電話しますAWS CLIAmazon S3 バケットを作成し、ファイルをオブジェクトとして追加します。

Prerequisites

IAM アクセス許可

の権限を取得する最速の方法AWS CloudShell以下を添付することでAWSIAM アイデンティティ (ユーザー、ロール、またはグループ) に対する管理ポリシー:

- AWS CloudShellFullAccess: ユーザーへのフルアクセスを提供します。AWS CloudShellとその特徴。

このチュートリアルでは、AWSサービス (この場合は、Amazon S3 バケットを作成し、そのバケットにオブジェクトを追加します)。したがって、IAM アイデンティティには、最低限、s3:CreateBucketおよびs3:PutObjectアクセス許可。

詳細については、「」を参照してください。 [Amazon S3 のアクション](#)のAmazon Simple Storage Service ユーザーガイド。

演習ファイル

この演習では、コマンドラインインターフェイスからプログラムとして実行されるファイルをアップロードして編集します。ローカルマシンでテキストエディタを開き、次のコードスニペットを追加します。

```
import sys
x=int(sys.argv[1])
y=int(sys.argv[2])
sum=x+y
print("The sum is",sum)
```

ファイルを名前を付けて保存します。add_prog.py。

Contents

- [ステップ 1: にサインインするAWS Management Console \(p. 7\)](#)
- [ステップ 2: を起動するAWS CloudShellで、\[リージョン\] を選択し、シェルを選択します。 \(p. 8\)](#)
- [ステップ 3: AWS CloudShell へのファイルのアップロード \(p. 10\)](#)

- ステップ 4: ファイルのコードを編集し、コマンドラインから実行します。(p. 11)
- ステップ 5: を使用するAWS CLI] をクリックして、ファイルを Amazon S3 バケットにオブジェクトとして追加します。(p. 12)

ステップ 1: にサインインするAWS Management Console

この手順では、IAM ユーザー情報を入力して、AWS Management Console。コンソールにすでに入っている場合は、「」に進みます。[ステップ 2 \(p. 8\)](#)。

- のにはからアクセスできます。AWS Management ConsoleIAM ユーザーのサインイン URL を使用するか、メインのサインインページに移動します。

IAM user sign-in URL

- ブラウザを開き、次のサインイン URL を入力し、`account_alias_or_id`管理者から提供されたアカウントエイリアスまたはアカウント ID を使用して、次の作業を行います。

```
https://account_alias_or_id.signin.aws.amazon.com/console/
```

- IAM ユーザー名とパスワードを入力し、[Sign In (サインイン)] を選択します。

Sign in as IAM user

Account ID (12 digits) or account alias

IAM user name

Password

Sign in

[Sign in using root user email](#)

[Forgot password?](#)

Main sign-in page

- オープン<https://aws.amazon.com/console/>。

- このブラウザを使用して以前にサインインしたことがない場合は、メインのサインインページが表示されます。[IAM ユーザー] を選択し、アカウントのエイリアスまたはアカウント ID を入力して、次。
- IAM ユーザーとして以前にサインインしている場合。お使いのブラウザには、AWS アカウントのアカウントエイリアスまたはアカウント ID が記憶されている可能性があります。その場合は、IAM ユーザー名とパスワードを入力し、サインイン。

Sign in as IAM user

Account ID (12 digits) or account alias

IAM user name

Password

Sign in

[Sign in using root user email](#)

[Forgot password?](#)

Note

また、としてサインインすることもできます。[root ユーザー](#)。このアイデンティティでは、すべての人に完全にアクセスできます。AWSアカウントのサービスとリソース。日常的なタスクには (それが管理タスクであっても)、ルートユーザーを使用しないよう強くお勧めします。代わりに、最初の IAM ユーザーを作成するためにのみ、ルートユーザーを使用するというベストプラクティスに従います。

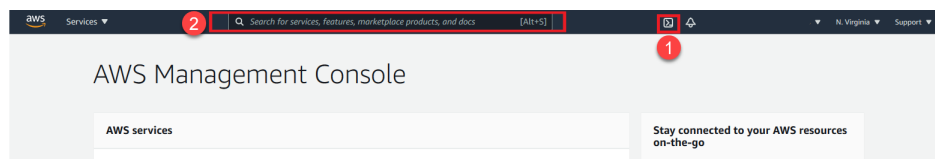
ステップ 2: を起動するAWS CloudShellで、[リージョン] を選択し、シェルを選択します。

このステップでは、を起動します。AWS CloudShellコンソールインターフェイスから、使用可能なものを選択します。AWSリージョンを選択し、好みのシェル (Bash、PowerShell、または Z シェル) に切り替えます。

1. [] からAWS Management Consoleで起動できるAWS CloudShellナビゲーションバーで使用できる次のオプションを選択します。

AWS CloudShell ユーザーガイド
ステップ 2: を起動するAWS CloudShellで、
[リージョン] を選択し、シェルを選択します。

- [AWS CloudShellアイコン]。
- に「cloudshell」と入力し始める検索ボックスをクリックし、CloudShellオプション。



メトリックAWS CloudShellが新しいブラウザウィンドウで初めて起動すると、ウェルカムパネルが表示され、主要な機能が一覧表示されます。このパネルを閉じると、シェルがコンソール認証情報を構成および転送する間、ステータスの更新が提供されます。コマンドプロンプトが表示されたら、シェルは対話の準備が整います。

Note

正常に起動または操作できない問題が発生した場合AWS CloudShellで、これらの問題を特定して対処するための情報を確認してください。[AWS CloudShell のトラブルシューティング \(p. 54\)](#)。

2. 選択するにはAWS作業する地域は、リージョンの選択メニューを選択し、[サポート対象AWSリージョン \(p. 57\)](#)で働く。(使用可能なリージョンは強調表示されます)。

Important

Regions を切り替えると、インターフェイスが更新され、選択したの名前が更新されます。AWSRegion はコマンドラインテキストの上に表示されます。永続ストレージに追加したファイルは、これでのみ使用できます。AWSリージョン。リージョンを変更すると、異なるストレージ (およびファイル) にアクセスできます。

3. 使用するプリインストールされたシェルを選択するには、コマンドラインプロンプトでプログラム名を入力します。

Bash

bash

Bash に切り替えると、コマンドプロンプトの記号が\$。

Note

Bash は、起動時に実行されるデフォルトのシェルです。AWS CloudShell。

PowerShell

pwsh

PowerShell に切り替えると、コマンドプロンプトの記号がPS>。

Z shell

zsh

Zシェルに切り替えると、コマンドプロンプトの記号が%。

ステップ 3: AWS CloudShell へのファイルのアップロード

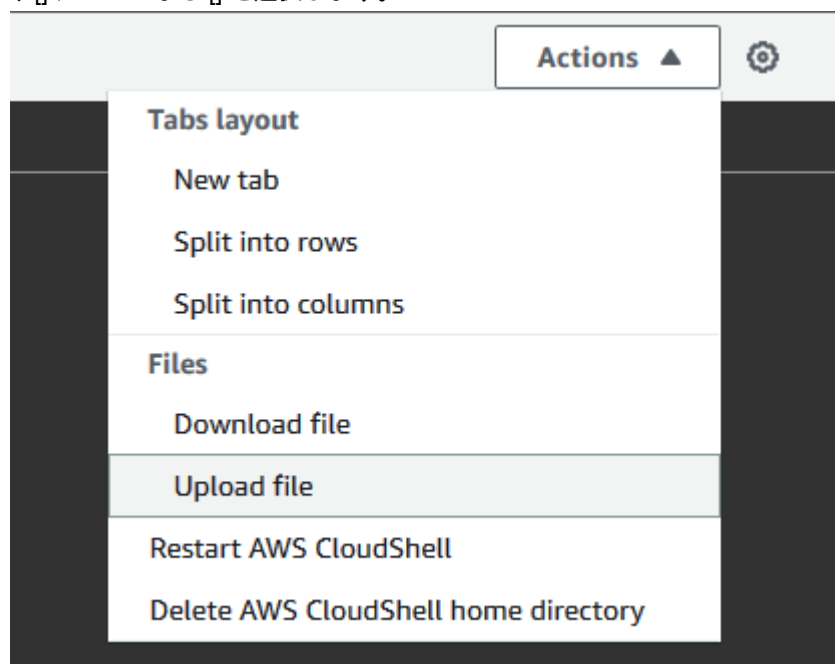
この手順では、ファイルをアップロードし、ホームディレクトリ内の新しいディレクトリに移動する手順を説明します。

1. 現在の作業ディレクトリを確認するには、プロンプトで次のコマンドを入力します。

```
pwd
```

を押すとEnterの場合、シェルは現在の作業ディレクトリを返します。例えば、`/home/cloudshell-user` などです。

2. このディレクトリにファイルをアップロードするには、アクションを選択し、ファイルのアップロード[]メニューから[]を選択します。



-ファイルのアップロード[]ダイアログボックスが表示されます。

3. [Browse (参照)] を選択します。
4. お使いのシステムのファイルのアップロードダイアログボックスで、このチュートリアル用に作成したテキストファイルを選択します (`add_prog.py`) を選択し、オープン。
5. 左ファイルのアップロード[]ダイアログボックスで、[]を選択しますアップロード。

プログレスバーはアップロードを追跡します。アップロードが成功すると、次のことが確認されるメッセージが表示されます。 `add_prog.py`がホームディレクトリのルートに追加されました。

6. ファイルのディレクトリを作成するには、`make directories` コマンドを入力します。 `mkdir mysub_dir`。
7. アップロードしたファイルをホームディレクトリのルートから新しいディレクトリに移動するには、`mv`コマンド:

```
mv add_prog.py mysub_dir.
```

8. 作業ディレクトリを新しいディレクトリに変更するには、次のように入力します。 `cd mysub_dir`。

コマンドプロンプトが更新され、作業ディレクトリが変更されたことを示します。

9. 現在のディレクトリの内容を表示するには、`mysub_dir 「」`と入力します。`ls`コマンド。

アップロードしたファイルを含め、作業ディレクトリの内容が一覧表示されます。

ステップ 4: ファイルのコードを編集し、コマンドラインから実行します。

この手順では、プリインストールされた Vim エディタを使用してファイル进行操作する方法を説明します。その後、そのファイルをコマンドラインからプログラムとして実行します。

1. 前のステップでアップロードしたファイルを編集するには、次のコマンドを入力します。

```
vim add_prog.py
```

シェルインターフェイスが更新され、Vim エディタが表示されます。

2. Vim でファイルを編集するには、`l`key。次に、プログラムが 2 つではなく 3 つの数字を加算するように内容を編集します。

```
import sys
x=int(sys.argv[1])
y=int(sys.argv[2])
z=int(sys.argv[3])
sum=x+y+z
print("The sum is",sum)
```

Note

テキストをエディタに貼り付けて、[セーフペースト機能 \(p. 32\)](#)有効にすると、警告が表示されます。コピーされたマルチテキストには、悪意のあるスクリプトが含まれている可能性があります。[セーフペースト]を使用すると、貼り付け前にテキスト全体を検証できます。テキストが安全であることが満足したら、Paste (貼り付ける)。

3. プログラムを編集したら、`Esc`をクリックして Vim コマンドモードに入ります。次に、次のコマンドを入力してファイルを保存し、エディタを終了します。

```
:wq
```

Note

Vim を初めて使用する場合は、コマンドモード (ファイルの保存とアプリケーションの終了時に使用) と挿入モード (新しいテキストの挿入時に使用) を切り替えるのが最初難しいかもしれません。わかりやすい基準点として、挿入モードに入るには、`l`、コマンドモードを開始するには、`Esc`。Vim およびその他のツールの詳細については、AWS CloudShell 「」を参照してください。[開発ツールとシェルユーティリティ \(p. 47\)](#)。

4. メインコマンドラインインターフェイスに戻り、プログラムを実行し、入力に次の 3 つの数値を指定します。

```
python3 add_prog.py 4 5 6
```

コマンドラインにプログラムの出力が表示されます。The sum is 15。

ステップ 5: を使用するAWS CLI をクリックして、ファイルを Amazon S3 バケットにオブジェクトとして追加します。

このステップでは、Amazon S3 バケットを作成し、PutObjectメソッドを使用して、コードファイルをオブジェクトとしてバケットに追加します。

Note

ほとんどの場合、次の作業を行うことができます。[CodeCommit などのサービスを使用する \(p. 17\)](#) ソフトウェアファイルをバージョン管理されたリポジトリにコミットします。この入門チュートリアルでは、ストレージに Amazon S3 を使用して、使いやすさを示します。AWS CLIがAWS CloudShell。この方法では、追加リソースをダウンロードしてインストールする必要はありません。さらに、シェル内で既に認証されているため、呼び出しを行う前に資格情報を設定する必要はありません。

1. 指定した AWS リージョンにバケットを作成するには、以下のコマンドを入力します。

```
aws s3api create-bucket --bucket insert-unique-bucket-name-here --region us-east-1
```

コールが成功すると、コマンドラインに次のようなサービスからの応答が表示されます。

```
{
  "Location": "/insert-unique-bucket-name-here"
}
```

Note

君が守らないと [バケットの名前付け規則](#) (たとえば、小文字のみを使用)、次のエラーが表示されます。CreateBucket オペレーションを呼び出すときにエラーが発生しました (InvalidBucketName): 指定されたバケットは有効ではありません。

2. ファイルをアップロードし、作成したばかりのバケットにオブジェクトとして追加するには、PutObject方法:

```
aws s3api put-object --bucket insert-unique-bucket-name-here --key add_prog --body add_prog.py
```

オブジェクトが Amazon S3 バケットに正常にアップロードされると、コマンドラインに次のようなサービスからの応答が表示されます。

```
{
  "ETag": "\"ab123c1:w:wad4a567d8bfd9a1234ebee56\""
}
```

-ETag格納されているオブジェクトのハッシュです。そのために使用できます。[Amazon S3 にアップロードされたオブジェクトの整合性をチェックする](#)。

関連トピック

- [の使用AWS内のサービスAWS CloudShell \(p. 28\)](#)

- [チュートリアル: ローカルマシンととの間で複数のファイルをコピーするAWS CloudShell \(p. 14\)](#)
- [チュートリアル: CodeCommit の使用の詳細については、AWS CloudShell \(p. 17\)](#)
- [AWS CloudShell の使用 \(p. 22\)](#)
- [カスタマイズするAWS CloudShellエクスペリエンス \(p. 31\)](#)

AWS CloudShell のチュートリアル

以下のチュートリアルでは、tasksを使用したAWS CloudShell。

トピック

- チュートリアル: ローカルマシンととの間で複数のファイルをコピーするAWS CloudShell (p. 14)
- チュートリアル: CodeCommit の使用の詳細については、AWS CloudShell (p. 17)
- チュートリアル: を使用して Amazon S3 オブジェクトの署名付き URL を作成するAWS CloudShell (p. 19)

チュートリアル: ローカルマシンととの間で複数のファイルをコピーするAWS CloudShell

CloudShell インターフェイスを使用して、ローカルマシンとシェル環境の間で一度に 1 つのファイルをアップロードまたはダウンロードできます。CloudShell とローカルマシン間で複数のファイルを同時にコピーするには、次のいずれかのオプションを使用します。

- Amazon S3: ローカルマシンと CloudShell 間でファイルをコピーするときは、S3 バケットを仲介として使用します。
- Zip ファイル: CloudShell インターフェイスを使用してアップロードまたはダウンロードできる 1 つの zip フォルダーに複数のファイルを圧縮します。

Note

なぜならAWS CloudShellは着信インターネットトラフィックを許可しません。現在、次のようなコマンドは使用できませんscpまたはrsyncローカルマシンと CloudShell コンピューティング環境間で複数のファイルをコピーします。

Amazon S3 を使用した複数のファイルのアップロードとダウンロード

Prerequisites

バケットとオブジェクトを操作するには、次の Amazon S3 API アクションを実行するアクセス権限を付与する IAM ポリシーが必要です。

- s3:CreateBucket
- s3:PutObject
- s3:GetObject

Amazon S3 のアクションの一覧については、「」を参照してください。[アクション](#)のAmazon Simple Storage Service API リファレンス。

に複数のファイルをアップロードするAWS CloudShellAmazon S3 の使用

1. EclipseAWS CloudShellで、以下を実行して S3 バケットを作成します。s3コマンド:

```
aws s3api create-bucket --bucket your-bucket-name --region us-east-1
```

コールが成功すると、コマンドラインに S3 サービスからの応答が表示されます。

```
{  
  "Location": "/your-bucket-name"  
}
```

- 次に、ローカルマシンからバケットにディレクトリ内のファイルをアップロードする必要があります。ファイルのアップロードには、次の 2 つのオプションがあります。
 - AWS Management Console: ドラッグアンドドロップを使用して、ファイルおよびフォルダをバケットにアップロードします。
 - AWS CLI: ローカルマシンにインストールされているツールのバージョンで、コマンドラインを使用してファイルとフォルダをバケットにアップロードします。

Using the console

- <https://console.aws.amazon.com/s3/> で Amazon S3 コンソールを開きます。

(使用しているAWS CloudShellコンソールに既にログインしている必要があります。)

- [Buckets (バケット)] リストで、フォルダやファイルのアップロード先のバケットの名前を選択します。
- コンソールウィンドウ以外のウィンドウで、アップロードするファイルとフォルダを選択します。次に、送信先バケット内のオブジェクトを一覧表示するコンソールウィンドウに選択内容をドラッグアンドドロップします。

選択したファイルは、[Upload (アップロード)] ページに一覧表示されます。

- チェックボックスを選択して、追加するファイルを指定します。
- 選択アップロード をクリックして、バケットに選択したファイルを追加します。

Note

コンソール使用時の設定オプションについては、「」を参照してください。[S3 バケットにファイルとフォルダをアップロードする方法](#)の Amazon Simple Storage Service ユーザーガイド。

Using AWS CLI

Note

このオプションでは、AWS CLI ツールがローカルマシンにインストールされ、への呼び出しに対して資格情報が構成されているAWSのサービス。詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

- を起動します。AWS CLI ツールを実行し、以下を実行します。aws s3 コマンドを実行して、指定したバケットをローカルマシン上の現在のディレクトリの内容と同期します。

```
aws s3 sync . s3://your-bucket-name
```

同期が成功すると、バケットに追加されたすべてのオブジェクトについてアップロードメッセージが表示されます。

- 次に、AWS CloudShell コマンドラインで次のコマンドを入力して、シェル環境のディレクトリを S3 バケットの内容と同期させます。

```
aws s3 sync s3://your-bucket-name .
```

同期が成功すると、バケットからディレクトリにダウンロードされたすべてのファイルについて、ダウンロードメッセージが表示されます。

Note

sync コマンドを使用すると、新しいファイルおよび更新されたファイルのみが送信元ディレクトリから送信先に再帰的にコピーされます。

複数のファイルをダウンロードするAWS CloudShellAmazon S3 の使用

1. の使用AWS CloudShellコマンドラインで、以下を入力します。aws s3シエル環境内の現在のディレクトリの内容に S3 バケットを同期させるコマンドは次のとおりです。

```
aws s3 sync . s3://your-bucket-name
```

同期が成功すると、バケットに追加されたすべてのオブジェクトについてアップロードメッセージが表示されます。

2. これで、バケットの内容をローカルマシンにダウンロードする必要があります。Amazon S3 コンソールは複数のオブジェクトのダウンロードをサポートしていないため、AWS CLIローカルマシンにインストールされているツール。

のコマンドラインからAWS CLIツールで、次のコマンドを実行します。

```
aws s3 sync s3://your-bucket-name .
```

同期が成功すると、宛先ディレクトリに更新または追加された各ファイルのダウンロードメッセージがコマンドラインに表示されます。

Note

このオプションでは、AWS CLIツールがローカルマシンにインストールされ、への呼び出しに対して資格情報が構成されているAWSのサービス。詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

zip 形式のフォルダを使用した複数のファイルのアップロードとダウンロード

zip/unzip ユーティリティを使用すると、単一のファイルとして扱うことができるアーカイブ内の複数のファイルを圧縮できます。ユーティリティは CloudShell コンピューティング環境にプリインストールされています。

プリインストールされたツールの詳細については、「」を参照してください。[開発ツールとシェルユーティリティ \(p. 47\)](#)。

に複数のファイルをアップロードするAWS CloudShellZip フォルダの使用

1. ローカルマシンで、アップロードするファイルを zip 形式で追加します。
2. を起動するAWS CloudShell[] の順に選択します。アクション,ファイルのアップロード。
3. 左ファイルのアップロード] ダイアログボックスで、[] を選択します。ファイルを選択先ほど作成したフォルダの Zip ファイルを選択します。

- 次に、[ファイルのアップロード] ダイアログボックスで、 を選択します。アップロードを選択して、選択したファイルをシェル環境に追加します。
- 次に、CloudShell コマンドラインで次のコマンドを実行して、zip アーカイブの内容を指定されたディレクトリに解凍します。

```
unzip zipped-files.zip -d my-unzipped-folder
```

複数のファイルをダウンロードするAWS CloudShellZip フォルダの使用

- CloudShell コマンドラインで次のコマンドを実行して、現在のディレクトリ内のすべてのファイルを zip 形式のフォルダーに追加します。

```
zip -r zipped-archive.zip *
```

- 次に を選択します。アクション、ファイルのダウンロード。
- 左[ファイルのダウンロード] ダイアログボックスで、圧縮フォルダのパスを入力します。(/home/cloudshell-user/zip-folder/zipped-archive.zip など) を選択し、のダウンロード。

パスが正しい場合は、ブラウザのダイアログで zip 形式のフォルダを開くか、ローカルマシンに保存するかを選択できます。
- ローカルマシンで、ダウンロードした zip フォルダの内容を解凍できるようになりました。

チュートリアル: CodeCommit の使用の詳細については、AWS CloudShell

CodeCommit は、セキュアでスケーラブル性が高いマネージド型ソースコントロールサービスで、プライベート Git リポジトリをホストします。を使用するAWS CloudShellを使用すると、コマンドラインで CodeCommit を操作できます。git-remote-codecommit。このユーティリティは、AWS CloudShellコンピューティング環境をサポートし、CodeCommit リポジトリからコードをプッシュおよびプルするための簡単な方法を提供します。これは、Gitを拡張することによって行います。詳細については、[AWS CodeCommit ユーザーガイド](#)を参照してください。

このチュートリアルでは、CodeCommit リポジトリを作成し、それをAWS CloudShellコンピューティング環境。このチュートリアルでは、ファイルをステージングしてクローンされたリポジトリにコミットしてから、AWSクラウド。

Prerequisites

IAM ユーザーが使用するために必要なアクセス許可の詳細については、AWS CloudShell詳細については、『[「はじめに」チュートリアルの「前提条件」セクション \(p. 6\)](#)。また、IAM アクセス許可を使用して CodeCommit を操作します。

さらに、開始する前に、次の項目が格納されているはずで。

- Gitコマンドとバージョン管理の概念の基本的な理解
- ローカルリポジトリとリモートリポジトリにコミットできる、シェルのホームディレクトリにあるファイル。このチュートリアルでは、my-git-file。

ステップ 1: CodeCommit リポジトリの作成とクローン作成

1. AWS CloudShell コマンドラインインターフェイスで、次のように入力します。codecommitコマンドを使用すると、CodeCommit リポジトリを作成できます。MyDemoRepo:

```
aws codecommit create-repository --repository-name MyDemoRepo --repository-description "My demonstration repository"
```

リポジトリが正常に作成されると、コマンドラインにサービスの応答が表示されます。

```
{
  "repositoryMetadata": {
    "accountId": "111122223333",
    "repositoryId": "0dcd29a8-941a-1111-1111-11111111111a",
    "repositoryName": "MyDemoRepo",
    "repositoryDescription": "My demonstration repository",
    "lastModifiedDate": "2020-11-23T20:38:23.068000+00:00",
    "creationDate": "2020-11-23T20:38:23.068000+00:00",
    "cloneUrlHttp": "https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/MyDemoRepo",
    "cloneUrlSsh": "ssh://git-codecommit.eu-west-1.amazonaws.com/v1/repos/MyDemoRepo",
    "Arn": "arn:aws:codecommit:eu-west-1:111111111111:MyDemoRepo"
  }
}
```

2. コマンドラインを使用して、ローカルリポジトリ用の新しいディレクトリを作成し、それを作業ディレクトリにします。

```
mkdir my-shell-repo
cd my-shell-repo
```

3. 次に、リモートリポジトリをgit cloneコマンド。(あなたが作業しているようにgit-remote-codecommitの場合は、HTTPS (GRC) URL スタイルを使用します)。

```
git clone codecommit::eu-west-1://MyDemoRepo
```

リポジトリが正常にクローンされると、コマンドラインにサービスの応答が表示されます。

```
Cloning into 'MyDemoRepo'...
warning: You appear to have cloned an empty repository.
```

4. クローンされたリポジトリに移動するには、cdコマンド:

```
cd MyDemoRepo
```

ステップ 2: CodeCommit リポジトリにプッシュする前にファイルをステージングしてコミットする

1. という名前のファイルを追加するmy-git-fileをMyDemoRepoフォルダに、Vim エディタまたはAWS CloudShell。両方の使用方法については、[入門チュートリアル \(p. 6\)](#)。
2. リポジトリにファイルをステージングするには、gitaddコマンド:

```
git add my-git-file
```

3. ファイルがステージングされ、コミットする準備ができていることを確認するには、`gitstatus`コマンド:

```
git status
```

`my-git-file`は新しいファイルとしてリストされ、緑色のテキストで表示され、コミットする準備ができていることを示します。

4. 次に、ステージングされたファイルのこのバージョンをリポジトリにコミットします。

```
git commit -m "first commit to repo"
```

Note

コミットを完了するための設定情報の入力を求められた場合は、次の形式を使用します。

```
$ git config --global user.name "Jane Doe"  
$ git config --global user.email janedoe@example.com
```

5. 最後に、リモートリポジトリをローカルリポジトリで行った変更と同期するには、変更をアップストリームブランチにプッシュします。

```
git push
```

チュートリアル: を使用して Amazon S3 オブジェクトの署名付き URL を作成するAWS CloudShell

このトピックでは、署名付き URL を作成して Amazon S3 オブジェクトを他のユーザーと共有する方法を示します。オブジェクトの所有者は、共有時に独自のセキュリティ認証情報を指定するため、署名済み URL を受信したユーザーは誰でも期間限定でオブジェクトにアクセスできます。

Prerequisites

- によって提供されるアクセス権限を持つ IAM ユーザー `AWS CloudShellFullAccess` ポリシー。
- 署名付き URL の作成に必要な IAM 権限については、「」を参照してください。 [他ユーザーとのオブジェクトの共有](#)の Amazon Simple Storage Service ユーザーガイド。

ステップ 1: IAM ロールを作成して Amazon S3 バケットへのアクセスを許可します。

1. 共有できる IAM の詳細を取得するには、`get-caller-identity`コマンドからAWS CloudShell:

```
aws sts get-caller-identity
```

そのファイルにコールは正常に終了すると、コマンドラインに次のようなレスポンスが表示されます。

```
{
  "Account": "123456789012",
  "UserId": "AROAXXOZUUOTWDCVIDZ:redirect_session",
  "Arn": "arn:aws:sts::531421766567:assumed-role/Feder08/redirect_session"
}
```

2. 前の手順で取得したユーザー情報を取得し、あなAWS CloudFormationテンプレート。このテンプレートにより IAM ロールが作成されます。このロールは、共有リソースの最小権限をコラボレーターに付与します。

```
Resources:
  CollaboratorRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              AWS: "arn:aws:iam::531421766567:role/Feder08"
            Action: "sts:AssumeRole"
        Description: Role used by my collaborators
        MaxSessionDuration: 7200
  CollaboratorPolicy:
    Type: AWS::IAM::Policy
    Properties:
      PolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Action:
              - 's3:*'
            Resource: 'arn:aws:s3:::shared-bucket-for-my-cool-startup'
            Condition:
              StringEquals:
                s3:prefix:
                  - "myfolder/*"
        PolicyName: S3ReadSpecificFolder
    Roles:
      - !Ref CollaboratorRole
Outputs:
  CollaboratorRoleArn:
    Description: Arn for the Collaborator's Role
    Value: !GetAtt CollaboratorRole.Arn
```

3. を保存します。AWS CloudFormationという名前のファイル内のテンプレートtemplate.yaml。
4. 次に、テンプレートを使用してスタックをデプロイし、deployコマンド:

```
aws cloudformation deploy --template-file ./template.yaml --stack-name CollaboratorRole
--capabilities CAPABILITY_IAM
```

署名付き URL の生成

1. でエディタを使うAWS CloudShellで、次のコードを追加します。このコードは、フェデレーテッドユーザーに直接アクセスできる URL を作成します。AWS Management Console。

```
import urllib, json, sys
import requests
```



```
import boto3
import os

def main():
    sts_client = boto3.client('sts')
    assume_role_response = sts_client.assume_role(
        RoleArn=os.environ.get(ROLE_ARN),
        RoleSessionName="collaborator-session"
    )
    credentials = assume_role_response['Credentials']
    url_credentials = {}
    url_credentials['sessionId'] = credentials.get('AccessKeyId')
    url_credentials['sessionKey'] = credentials.get('SecretAccessKey')
    url_credentials['sessionToken'] = credentials.get('SessionToken')
    json_string_with_temp_credentials = json.dumps(url_credentials)
    print(f"json string {json_string_with_temp_credentials}")

    request_parameters = f"?
Action=getSignInToken&Session={urllib.parse.quote(json_string_with_temp_credentials)}"
    request_url = "https://signin.aws.amazon.com/federation" + request_parameters
    r = requests.get(request_url)
    signin_token = json.loads(r.text)
    request_parameters = "?Action=login"
    request_parameters += "&Issuer=Example.org"
    request_parameters += "&Destination=" + urllib.parse.quote("https://us-
west-2.console.aws.amazon.com/cloudshell")
    request_parameters += "&SignInToken=" + signin_token["SignInToken"]
    request_url = "https://signin.aws.amazon.com/federation" + request_parameters

    # Send final URL to stdout
    print (request_url)

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

2. という名前のファイルにコードを保存します。share.py。
3. Last,コマンドラインから次のコマンドを実行して、IAM ロールの ARN をAWS CloudFormation。次に、これを Python スクリプトで使用して、一時的なセキュリティ認証情報を取得します。

```
ROLE_ARN=$(aws cloudformation describe-stacks --stack-name CollaboratorRole --query
"Stacks[*].Outputs[?OutputKey=='CollaboratorRoleArn'].OutputValue" --output text)
python3 ./share.py
```

スクリプトは、コラボレーターがクリックして移動できる URL を返します。AWS CloudShellがAWS Management Console。コラボレーターは、myfolder/次の 3,600 秒 (1 時間) の間 Amazon S3 バケット内のフォルダ。認証情報は 1 時間後に無効になります。。この時間が過ぎると、コラボレーターはバケットにアクセスできなくなります。

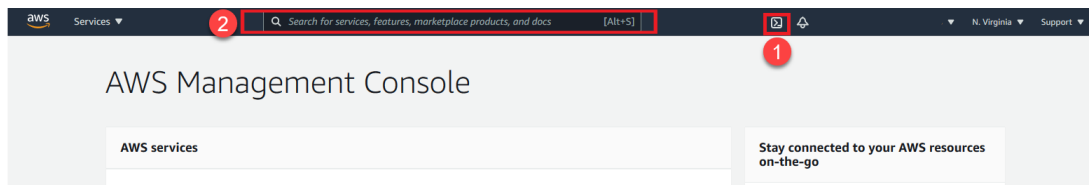
AWS CloudShell の使用

このトピックでは、AWS CloudShellを開き、シェルインターフェースのナビゲート、シェルタイプの選択、ファイルのアップロードとダウンロードの操作、シェルセッションの管理を行います。

を起動するAWS CloudShell

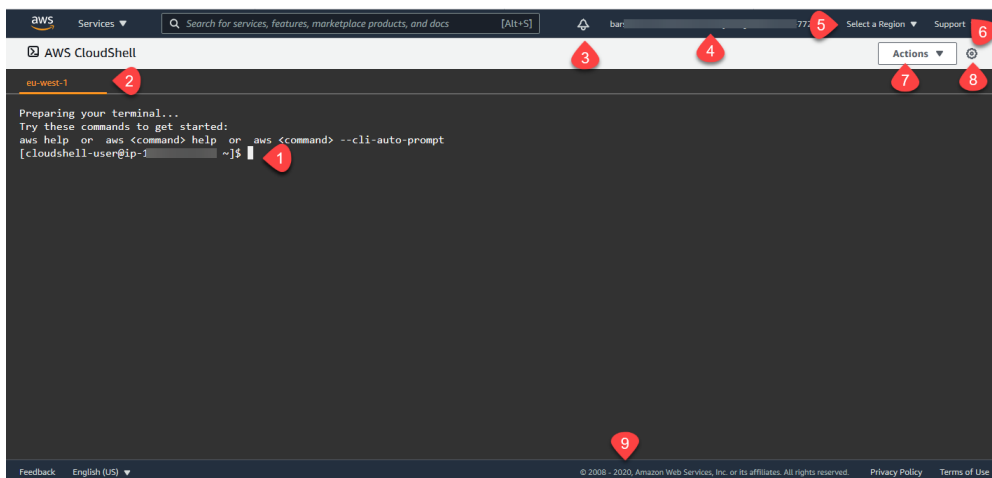
起動することができますAWS CloudShellからのAWS Management Console次の 2 つの方法のうちの 1 つを使用します。

1. [AWS CloudShellアイコンをクリックします。
2. 「cloudshell」と入力し始めるサービスの検索ボックスを選択し、CloudShellオプションを使用します。



のナビゲーションAWS CloudShellインターフェイスからリクエスト

次のスクリーンショットは、いくつかのキーを示していますAWS CloudShellインターフェイスで展開します。これらの機能は、コマンドラインの使用時、ファイルのアップロード/ダウンロードツールの使用、シェル環境のカスタマイズ、およびユーザーアシスタンスへのアクセス時に使用します。



1. AWS CloudShellコマンド回線インターフェイスを実行するために使用するを使用するコマンド**お好みのシェル**で (p. 23)。現在のシェルタイプは、コマンドプロンプトで示されます。
2. -AWSリージョンAWS CloudShellは現在実行中です。

3. 「通知」リンク。Personal Health Dashboard とイベントログにアクセスできます。
4. 現在のの名前とアカウント IDAWS CloudShell ユーザー (ここでもやっています)。このメニューを使用して、アカウント情報の表示、役割の切り替え、AWS。
5. リージョンセレクターでは、どのリージョンを選択できるかを選択できます。AWSシェル環境が実行されているリージョン。詳細については、「[サポート対象AWSのリージョンAWS CloudShell \(p. 57\)](#)」を参照してください。
6. -サポートメニューが表示されます。このメニューを使用して、ドキュメントやトレーニングなどのユーザーアシスタンスリソースにアクセスできます。
7. -アクションメニューが表示されます。このメニューには、[変更](#)、[画面レイアウト \(p. 31\)](#)、[ファイルのダウンロードおよびアップロード \(p. 24\)](#)、および[を再起動するAWS CloudShell \(p. 27\)](#)。
8. -Preferences (設定)オプションを使用します。これを使用して[シェルをカスタマイズする \(p. 31\)](#)。これを行うには、テキストサイズを変更し、インターフェイスのカラーテーマを切り替えます。
9. 下のバーには、フィードバックの提供、インターフェイスのローカライズ版へのアクセス、プライバシーポリシーと利用規約の確認を行うオプションがあります。

シェルの選択

以下のシェルはプリインストールされており、AWS CloudShell : Bashシェル、PowerShell、Zシェルです。シェル環境にプレインストールされているバージョンについては、[シェルテーブル \(p. 44\)](#)のAWS CloudShellコンピューティング環境 (p. 44)セクションに追加します。

現在使用しているシェルは、コマンドプロンプトで識別できます。

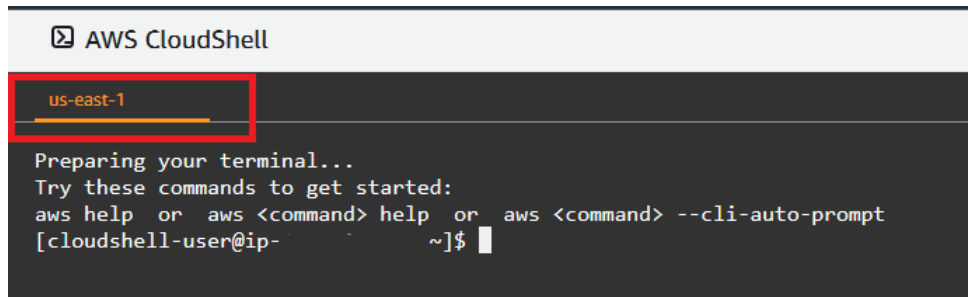
- \$Bash shell
- PS>: PowerShell
- %: Zシェル

新しいシェルに切り替えるには、コマンド・ライン・プロンプトでシェルのプログラム名を入力します。

- bash: Bash
- pwsh: PowerShell
- zsh: Zシェル

の使用AWSリージョン

現在のAWS実行中のリージョンは、コマンドラインインターフェイスの上に表示されます。



□を選択することもできます。AWSリージョンの選択から1つを選択することで、作業するリージョン。リージョンを変更すると、シェルセッションが選択したリージョンで実行されているコンピューティング環境に接続すると、インターフェイスが更新されます。

Important

それぞれに 1 GB の永続ストレージがありますAWSリージョン。永続ストレージは、ホームディレクトリ (\$HOME)。したがって、個人用ファイル、ディレクトリ、プログラム、およびスクリプトは、ホームディレクトリに格納されている 1 つのAWSリージョン。さらに、それらはホームディレクトリにあり、別のリージョンに格納されているものとは異なります。永続ストレージ内のファイルの長期保存は、リージョンごとに管理されます。詳細については、「[永続的ストレージ \(p. 58\)](#)」を参照してください。

デフォルトの指定AWSのリージョンAWS CLI

以下を使用できます。[環境変数](#)にアクセスするために必要な設定オプションと認証情報を指定しますAWSを使用するサービスAWS CLI。デフォルトを指定する環境変数AWSシェルセッションのリージョンは、AWS CloudShell内の特定のリージョンからAWS Management Consoleを選択するか、リージョンセクタでオプションを選択したときに表示されます。

なぜなら[環境変数はAWS CLI認証情報ファイル](#)によって更新されるaws configureに設定されている場合、そのコマンドを実行して環境変数で指定されたリージョンを変更することはできません。デフォルトのリージョンを変更するにはAWS CLIコマンドで、値を割り当てるAWS_REGION環境変数で展開します。以下の例では、us-east-1地域のあなたの選択と。

Bash or Zsh

```
$ export AWS_REGION=us-east-1
```

環境変数を設定すると、シェルセッションの終了時まで、または変数に別の値を設定するまで、使用する値が変更されます。変数をシェルのスタートアップスクリプトで設定することで、変数をこれからのセッションで永続的にすることができます。

PowerShell

```
PS C:\> $Env:AWS_REGION="us-east-1"
```

PowerShell プロンプトで環境変数を設定した場合は、現在のセッションの期間だけ値が保存されます。または、変数を PowerShell プロファイルに追加すると、その変数を今後のすべての PowerShell セッションに設定できます。環境変数の保存やそれをセッション間で永続的に維持する詳細については、[PowerShell ドキュメント](#)を参照してください。

デフォルトのリージョンが変更されたことを確認するには、現在のAWS CLI次のコマンドを実行して、設定データを展開します。

```
aws configure list
```

Note

具体的なAWS CLIコマンドラインオプションを使用してデフォルトのリージョンを上書きするには--region。詳細については、「」を参照してください。[コマンドラインオプションのAWS Command Line Interfaceユーザーガイド](#)。

ファイルとストレージの操作

を使用するAWS CloudShellインターフェイスからファイルをアップロードしたり、シェル環境からファイルをダウンロードしたりできます。セッション終了後に追加したファイルを使用できるようにするには、永続ストレージと一時ストレージの違いを理解しておく必要があります。

- 永続的ストレージ: それぞれに 1 GB の永続ストレージがありますAWSリージョン。永続的ストレージは、ホームディレクトリにあります。
- 一時ストレージ: 一時ストレージは、ホームディレクトリ外のディレクトリに存在し、セッションの最後にリサイクルされます。

Important

将来のシェルセッションで保持して使用するファイルは、常にホームディレクトリに残す必要があります。ファイルをホームディレクトリから移動すると (mvコマンドなど)、現在のシェルセッションが終了するとリサイクルされます。

からファイルをダウンロードするにはAWS CloudShell

1. 選択アクション,ファイルのダウンロード。
2. 左ファイルのダウンロード[] ダイアログボックスで、ダウンロードするファイルのパスを入力します。

Download file

Download files from your AWS CloudShell to your local desktop. Folders are not supported.

Individual file path

You can copy the file path from the command-line and paste it below.

/home/cloudshell-user/subfolder/mydownloadfile.txt

myfile.txt or /folder/myfile.txt.

Cancel

Download

Note

ダウンロードするファイルを指定するときは、絶対パスまたは相対パスを使用できます。相対パス名では、 /home/cloudshell-user/はデフォルトで自動的にスタートに追加されます。というファイルをダウンロードするにはmydownload-fileの場合、次の両方が有効なパスです。

- 絶対パス: /home/cloudshell-user/subfolder/mydownloadfile.txt
- 相対パス: subfolder/mydownloadfile.txt

3. [Download] を選択します。

ファイルパスが正しい場合は、ダイアログボックスが表示されます。このダイアログボックスを使用して、デフォルトのアプリケーションでファイルを開くことができます。ファイルをローカルマシン上のフォルダに保存することもできます。

AWS CloudShell にファイルをアップロードするには

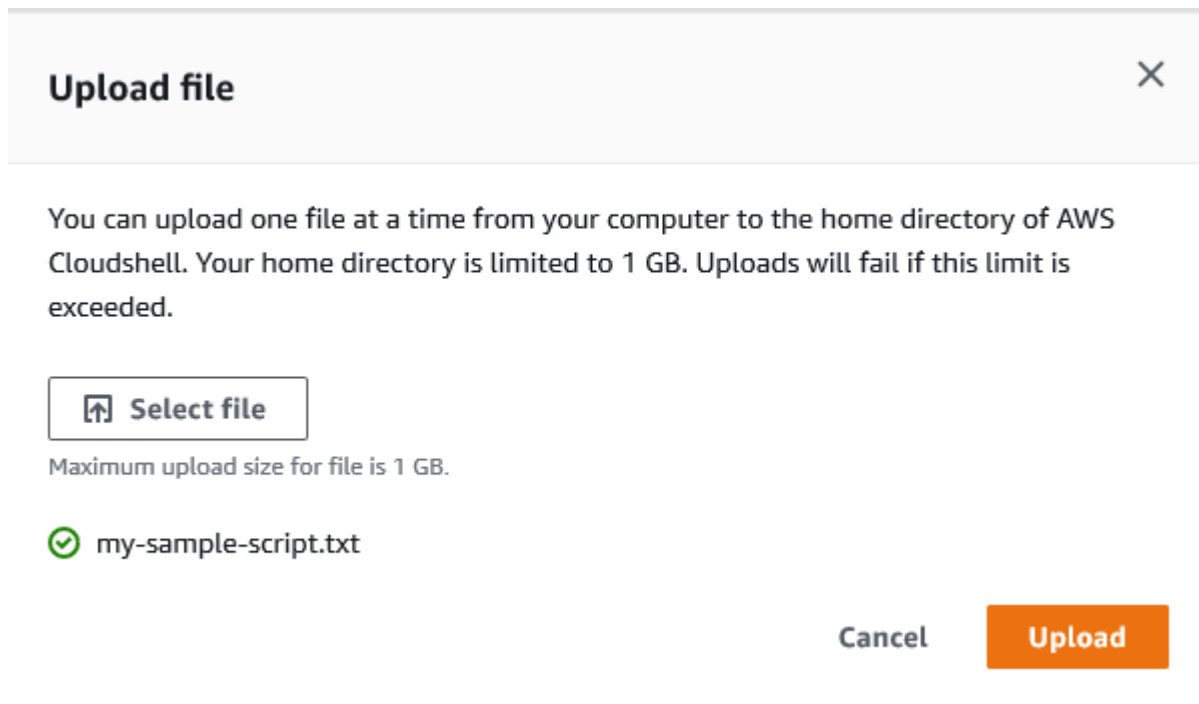
1. 選択アクション,ファイルのアップロード。

Note

デフォルトでは、ファイルはホームディレクトリのルートにアップロードされます。 /home/cloudshell-user。ファイルを移動する場合 (mvコマンドなど)、ファイルがホームディ

レクタリに残っていることを確認します。この場所には、1 GB の永続ストレージがあります。

2. 左ファイルのアップロードダイアログボックスで、参照ローカルマシン上のファイルを選択します。（最大アップロードサイズは 1 GB です。）
3. ご使用のシステム内ファイルのアップロードダイアログボックスで、アップロードするファイルを選択し、オープン。
4. 選択アップロードをクリックして、選択したファイルをシェル環境に追加します。



リージョンごとの制限を超過するファイルをアップロードしようとする、アップロードを続行できないことが通知されます。たとえば、ホームディレクトリに 900 MB のデータを格納しているときに 200 MB のファイルをアップロードしようとする、アップロードを完了できないというメッセージが表示されます。

からファイルを削除するにはAWS CloudShell

1. からファイルを削除するにはAWS CloudShellなどの標準シェルコマンドを使用します。rm(削除):

```
rm my-file-for-removal
```
2. また、指定した条件を満たす複数のファイルを削除するには、findコマンドを実行します。次の例では、名前に「.pdf」という語句が含まれるすべてのファイルを削除します。

```
find -type f -name '*pdf*' -delete
```

Note

使用を停止した場合AWS CloudShellでAWS「リージョン」を選択すると、そのリージョンの永続ストレージ内のデータは、定義された期間が経過すると自動的に削除されます。詳細については、「[永続的ストレージ \(p. 58\)](#)」を参照してください。

シェルセッションの開始と終了

Note

セキュリティ対策として、キーボードやポインタを使用してシェルを長時間操作しないと、セッションは自動的に停止します。非常に長時間実行されているセッションも自動的に停止されます。詳細については、「[シェルセッション \(p. 59\)](#)」を参照してください。

シェルセッションの再起動

1. シェルセッションを再開するには、アクション,CloudShell を再起動する。

再起動すると通知されますAWS CloudShellは、現在のアクティブなセッションをすべて停止します。AWSリージョン。

2. 選択Restart確定します。

インターフェイスは、AWS CloudShellコンピューティング環境が停止しています。環境が停止して再起動したら、新しいセッションでコマンドラインを使用して作業を開始できます。

Note

環境を再起動するまでには数分かかる場合があります。

シェルセッションを手動で終了する

コマンドラインを使用すると、シェルセッションを終了し、exitコマンドを実行します。その後、任意のキーを押して再接続し、AWS CloudShell。

の使用AWS内のサービスAWS CloudShell

主な利点AWS CloudShellの利点は、それを使用してAWSコマンドラインインターフェイスからサービスを使用します。つまり、事前にツールをダウンロードしてインストールしたり、認証情報をローカルで設定したりする必要はありません。を起動すると、AWS CloudShellの場合、コンピューティング環境が作成され、次のAWSコマンドラインツールが既にインストールされています。

- [AWS CLI \(p. 28\)](#)
- [AWS Elastic Beanstalk CLI \(p. 29\)](#)
- [Amazon ECS CLI \(p. 30\)](#)
- [AWS SAM \(p. 30\)](#)

そして、あなたはすでにAWSを使用する場合、サービスを使用する前に資格情報をローカルに設定する必要はありません。へのサインインに使用した認証情報AWS Management Consoleに転送されます。AWS CloudShell。

デフォルトを変更する場合AWS使用地域AWS CLIに設定されている場合、[AWS_REGION環境変数。(詳しくは、[デフォルトの指定AWSのリージョンAWS CLI \(p. 24\)](#) を参照してください)。

このトピックの残りの部分では、使用を開始する方法を示します。AWS CloudShellを選択して、選択したAWSコマンドラインからサービスを使用します。

AWS CLIのコマンドライン例AWSサービス

以下の例は、多数のAWSから利用可能なコマンドを使用して操作できるサービスAWS CLIバージョン 2。詳細なリストについては、[AWS CLI コマンドリファレンス](#)。

- [DynamoDB \(p. 28\)](#)
- [AWS Cloud9 \(p. 29\)](#)
- [Amazon EC2 \(p. 29\)](#)
- [S3 Glacier \(p. 29\)](#)

DynamoDB

DynamoDB は、高速で予測可能なパフォーマンスとシームレスな拡張性を特長とするフルマネージド NoSQL データベースサービスです。このサービスの NoSQL モードの実装では、キー値およびドキュメントデータ構造がサポートされます。

以下ようになりますcreate-tableコマンドは、名前の NoSQL スタイルのテーブルを作成します。MusicCollection側AWSアカウント。

```
aws dynamodb create-table \  
  --table-name MusicCollection \  
  --attribute-definitions AttributeName=Artist,AttributeType=S \  
  AttributeName=SongTitle,AttributeType=S \  
  --key-schema AttributeName=Artist,KeyType=HASH AttributeName=SongTitle,KeyType=RANGE \  
  --provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=10
```



```
--provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5 \  
--tags Key=Owner,Value=blueTeam
```

詳細については、「」を参照してください。[DynamoDB を AWS CLI\(\)AWS Command Line Interface ユーザーガイド](#)。

AWS Cloud9

AWS Cloud9は、ブラウザウィンドウでコードの記述、実行、およびデバッグに使用できるクラウドベースの統合開発環境 (IDE) です。環境には、コードエディタ、デバッガー、ターミナルがあります。

以下のようになります `create-environment-ec2` コマンドは、AWS Cloud9 指定された設定の EC2 開発環境。Amazon EC2 インスタンスを起動し、インスタンスから環境に接続します。

```
aws cloud9 create-environment-ec2 --name my-demo-env --description "My demonstration  
development environment." --instance-type t2.micro --subnet-id subnet-1fab8aEX --  
automatic-stop-time-minutes 60 --owner-arn arn:aws:iam::123456789012:user/MyDemoUser
```

詳細については、「」を参照してください。[AWS Cloud9 コマンドラインリファレンス](#)。

Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) は、クラウド内で安全で規模を変更可能なコンピューティング性能を提供するウェブサービスです。ウェブスケールのクラウドコンピューティングをより簡単にアクセスできるように設計されています。

以下のようになります `run-instances` コマンドは、VPC の指定されたサブネットで `t2.micro` インスタンスを起動します。

```
aws ec2 run-instances --image-id ami-xxxxxxx --count 1 --instance-type t2.micro --key-name  
MyKeyPair --security-group-ids sg-903004f8 --subnet-id subnet-6e7f829e
```

詳細については、「」を参照してください。[での Amazon EC2 の使用 AWS CLI\(\)AWS Command Line Interface ユーザーガイド](#)。

S3 Glacier

S3 Glacier および S3 Glacier Deep Archive は、データのアーカイブおよび長期バックアップを行うための、安全で耐久性が高く、非常に低コストの Amazon S3 クラウドストレージクラスです。

以下のようになります `create-vault` コマンドは、アーカイブを格納するコンテナである Vault を作成します。

```
aws glacier create-vault --vault-name my-vault --account-id -
```

詳細については、「」を参照してください。[での Amazon S3 Glacier の使用 AWS CLI\(\)AWS Command Line Interface ユーザーガイド](#)。

AWS Elastic Beanstalk CLI

-AWS Elastic Beanstalk CLI には、ローカルリポジトリからの環境の作成、更新、モニタリングを簡素化するコマンドラインインターフェイスが用意されています。この文脈では、環境のコレクションを参照し、AWS アプリケーションバージョンを実行するリソース。

以下のようになります。createコマンドを実行すると、カスタム Amazon Virtual Private Cloud (VPC) に新しい環境が作成されます。

```
$ eb create dev-vpc --vpc.id vpc-0ce8dd99 --vpc.elbsubnets subnet-b356d7c6,subnet-02f74b0c  
--vpc.ec2subnets subnet-0bb7f0cd,subnet-3b6697c1 --vpc.securitygroup sg-70cff265
```

詳細については、「」を参照してください。[EB CLI コマンドリファレンス\(\)](#)AWS Elastic Beanstalk開発者ガイド。

Amazon ECS CLI

Amazon Elastic Container Service (Amazon ECS) コマンドラインインターフェイス (CLI) には、いくつかの高レベルコマンドが用意されています。ローカル開発環境からのクラスターおよびタスクの作成、更新、モニタリングプロセスを簡素化するように設計されています。Amazon ECS クラスターは、タスクまたはサービスの論理グループです。

以下のようになります。configureコマンドは、Amazon ECS CLI を設定して、ecs-cli-demo。このクラスター設定ではFARGATEのデフォルトの起動タイプとしてecs-cli-demo内のクラスターus-east-1 region。

```
ecs-cli configure --region us-east-1 --cluster ecs-cli-demo --default-launch-type FARGATE  
--config-name ecs-cli-demo
```

詳細については、「」を参照してください。[Amazon ECS コマンドラインリファレンス\(\)](#)Amazon Elastic Container Service。

AWS SAM CLI

AWS SAMCLI は、コマンドラインツールで、AWS Serverless Application Modelテンプレートとアプリケーションコード。これを使用して複数のタスクを実行できます。これには、ローカルでLambda 関数を呼び出す、サーバーレスアプリケーションのデプロイパッケージを作成する、サーバーレスアプリケーションをAWSクラウド。

以下のようになります。initコマンドは、パラメータとして渡された必須パラメータで新しい SAM プロジェクトを初期化します。

```
sam init --runtime python3.7 --dependency-manager pip --app-template hello-world --name  
sam-app
```

詳細については、「」を参照してください。[AWS SAMCLI コマンドリファレンス\(\)](#)AWS Serverless Application Model開発者ガイド。

カスタマイズするAWS CloudShellエクスペリエンス

次の側面をカスタマイズできます。AWS CloudShellExperience:

- [タブのレイアウト \(p. 31\)](#): コマンドラインインターフェイスを複数の列と行に分割します。
- [Font size \(フォントサイズ\) \(p. 31\)](#): コマンドラインテキストのサイズを調整します。
- [カラーテーマ \(p. 31\)](#): 明るいテーマと暗いテーマを切り替えます。
- [セーフペースト \(p. 31\)](#): 貼り付け前にマルチテキストを確認する必要がある機能を有効または無効にします。

また、シェル環境を拡張することもできます [インストール, 独自のソフトウェア \(p. 52\)](#) および [変更, スタートアップシェルスクリプト \(p. 52\)](#)。

コマンドライン表示を複数のタブに分割する

この生産性機能を使用すると、コマンドラインインターフェイスを複数のペインに分割して、複数のコマンドを実行できます。

Note

複数のタブ付きペインが開いている場合は、ペイン内の任意の場所をクリックして作業するタブを選択できます。タブを閉じるには、x記号をクリックします。

- 選択アクション次のいずれかのオプションからのオプションタブのレイアウト:
 - [新規作成] タブ: 現在アクティブなタブの横に新しいタブを追加します。
 - 行に分割する: 現在アクティブなタブの下に新しいタブを追加します。
 - 列に分割する: 現在アクティブなタブの横にある列に新しいタブを追加します。

各タブを完全に表示するための十分なスペースがない場合は、スクロールバーを使用してインターフェイスをナビゲートできます。また、ペインを分割する分割バーを選択し、ポインタを使用してドラッグして、ペインのサイズを拡大または縮小することもできます。

フォントサイズを変更する

コマンドラインインターフェイスに表示されるテキストのサイズは、増減できます。

1. 選択Preferences (設定), フォントのサイズ。
2. テキストサイズを選択最小値および最大。

インターフェイスのテーマを変更する

コマンドラインインターフェイスの明るいテーマと暗いテーマを切り替えることができます。

1. 選択Preferences (設定),カラーテーマ。
2. 選択[Light (ライト)]またはダーク。

マルチテキストにセーフペーストを使用する

安全な貼り付けは、シェルに貼り付けようとしている複数行のテキストに悪意のあるスクリプトが含まれていないことを確認するセキュリティ機能です。サードパーティのサイトからコピーされたテキストには、シェル環境で予期しない動作を引き起こす隠しコードが含まれている可能性があります。

[セーフペースト] ダイアログボックスには、クリップボードにコピーしたテキスト全体が表示されます。セキュリティ上のリスクがないことに満足できる場合は、Paste (貼り付ける)。

Warning: Pasting multiline text into AWS CloudShell



Text that's copied from external sources can contain malicious scripts. Verify the text below before pasting.

```
import sys
x=int(sys.argv[1])
y=int(sys.argv[2])
z=int(sys.argv[3])
total=x+y+z
print("The total is",total)
```

Always ask before pasting multiline code

Cancel

Paste

スクリプト内の潜在的なセキュリティ上のリスクを検出するには、[セーフペースト] を有効にすることをお勧めします。ただし、この機能をオンまたはオフに切り替えるには、Preferences (設定),セーフ貼り付けを有効にする/セーフ貼り付けを無効にする。

AWS CloudShell のセキュリティ

クラウドセキュリティはアマゾン ウェブ サービス (AWS) の最優先事項です。AWS のお客様は、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ーから利点を得られます。セキュリティは、AWS とお客様の間の共有責任です。[責任共有モデル](#)では、これをクラウドのセキュリティおよびクラウド内のセキュリティとして説明しています。

クラウドのセキュリティ – AWS は、AWS クラウド内でサービスを実行するインフラストラクチャを保護する責任を担い、安全に使用できるサービスを提供します。AWS ではセキュリティに対する責任が最優先事項であり、当社のセキュリティの有効性は、[AWS コンプライアンスプログラム](#)の一環として、サードパーティーの監査人によって定期的にテストおよび検証されています。

クラウド内のセキュリティ – お客様の責任は、使用している AWS のサービスや、データの機密性、組織の要件、適用される法律や規制などのその他の要因によって決まります。

AWS CloudShell 以下になります。[責任共有モデル](#) 特定のものを通じ AWS サポートしているサービス。AWS サービスのセキュリティ情報については、[AWS のサービスのセキュリティに関するドキュメントページ](#)と、[AWS コンプライアンスプログラムによる AWS 対象範囲内のサービス](#)を参照してください。

以下のトピックでは、セキュリティおよびコンプライアンスの目的を達成するために AWS CloudShell を設定する方法を示します。

トピック

- [AWS CloudShell でのデータ保護 \(p. 33\)](#)
- [AWS CloudShell での Identity and Access Management \(p. 34\)](#)
- [でのログ記録とモニタリング AWS CloudShell \(p. 40\)](#)
- [AWS CloudShell のコンプライアンス検証 \(p. 41\)](#)
- [での耐障害性 AWS CloudShell \(p. 41\)](#)
- [AWS CloudShell でのインフラストラクチャセキュリティ \(p. 42\)](#)
- [AWS CloudShell での設定と脆弱性の分析 \(p. 42\)](#)
- [AWS CloudShell のセキュリティのベストプラクティス \(p. 42\)](#)

AWS CloudShell でのデータ保護

[AWS 責任共有モデル](#)は、AWS CloudShell でのデータ保護に適用されます。このモデルで説明されているように、AWS は、AWS クラウド のすべてを実行するグローバルインフラストラクチャを保護する責任を担います。お客様は、このインフラストラクチャでホストされているコンテンツに対する管理を維持する責任があります。このコンテンツには、使用する AWS のサービスのセキュリティ設定および管理タスクが含まれます。データプライバシーの詳細については、「[データプライバシーのよくある質問](#)」を参照してください。欧州でのデータ保護の詳細については、AWS セキュリティブログの「[The AWS Shared Responsibility Model and GDPR](#)」を参照してください。

データを保護するため、AWS アカウント の認証情報を保護し、AWS Identity and Access Management (IAM) を使用して個々のユーザーアカウントをセットアップすることをお勧めします。この方法により、それぞれの職務を遂行するために必要なアクセス許可のみを各ユーザーに付与できます。また、以下の方法でデータを保護することをお勧めします。

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 以降が推奨されています。
- AWS CloudTrail で API とユーザーアクティビティログをセットアップします。

- AWS 暗号化ソリューションを、AWS のサービス内のすべてのデフォルトのセキュリティ管理と一緒に使用します。
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これにより、Amazon S3 に保存される個人データの検出と保護が支援されます。
- コマンドラインインターフェイスまたは API を使用して AWS にアクセスするときに FIPS 140-2 検証済みの暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。使用可能な FIPS エンドポイントの詳細については、「[連邦情報処理規格 \(FIPS\) 140-2](#)」を参照してください。

顧客の E メールアドレスなどの機密情報やセンシティブ情報は、タグや [名前] フィールドなどの自由形式のフィールドに配置しないことを強くお勧めします。これは、コンソール、API、AWS CloudShell、または AWS で AWS CLI または他の AWS サービスを使用する場合も同様です。名前に使用するタグまたは自由形式のフィールドに入力したデータは、請求ログまたは診断ログに使用できます。外部サーバーへの URL を指定する場合は、そのサーバーへのリクエストを検証するための認証情報を URL に含めないことを強くお勧めします。

データ暗号化

データ暗号化とは、保管時の (保存時の) データを保護することを指します。AWS CloudShell) と輸送中 (その間を移動するとき) AWS CloudShell およびサービスエンドポイント)。

を使用した保管時の暗号化AWS KMS

保存時の暗号化とは、保存中にデータを暗号化することで、不正なアクセスからデータを保護することです。を使用するときAWS CloudShellの場合、永続ストレージはあたり 1 GB ですAWSリージョンは無料で利用できます。永続ストレージは、ホームディレクトリ (\$HOME) であり、あなたにとって非公開です。各シェルセッションが終了した後にリサイクルされるエフェメラル環境リソースとは異なり、ホームディレクトリ内のデータは保持されます。

に保存されているデータの暗号化AWS CloudShellによって提供される暗号化キーを使用して実装されます。AWS Key Management Service(AWS KMS)。これは管理対象ですAWSカスタマーマスターキー (CMK) の作成と管理サービスに保存されたカスタマーデータの暗号化に使用される暗号化キーAWS CloudShell 環境。AWS CloudShell顧客の代わりにデータを暗号化するための暗号化キーを生成および管理します。

転送時の暗号化

転送中の暗号化とは、通信エンドポイント間の移動中にデータが傍受されるのを防ぐことです。

デフォルトでは、クライアントの Web ブラウザコンピュータとクラウドベースの間のすべてのデータ通信AWS CloudShellは、HTTPS/TLS 接続を介してすべてを送信することで暗号化されています。

通信に HTTPS/TLS を使用できるようにするには、何もする必要はありません。

AWS CloudShell での Identity and Access Management

AWS Identity and Access Management(IAM) はAmazon Web Services (AWS) は、管理者がへのアクセスを安全にコントロールするのに役立つのサービスAWSリソースの使用料金を見積もることができま。IAM 管理者は、誰にできるかを制御する認証済み(サインイン)と承認済みリソースを使用するには (権限を持つ)AWSのサービス。IAM は、AWS のサービスで追加料金は発生しません。

AWS CloudShell を使用して AWS にアクセスするには、AWS アカウントと AWS 認証情報が必要です。あなたのセキュリティを高めるためにAWSアカウント、使用することを検討してくださいIAM ユーザー自分のものではなくアクセス資格情報を提供するにはAWSアカウント認証情報。

IAM の操作方法の詳細については、「」を参照してください。[AWS Identity and Access Management](#)。

IAM ユーザーの概要と、IAM ユーザーがアカウントのセキュリティにとって重要である理由については、「」を参照してください。[AWSセキュリティの認証情報のAmazon Web Services 全般リファレンス](#)。

AWS CloudShell以下ようになります。[責任共有モデル](#)特定のAmazon Web Services (AWS) サポートしているサービス。AWS サービスのセキュリティ情報については、[AWS のサービスのセキュリティに関するドキュメントページ](#)と、[AWS コンプライアンスプログラムによる AWS 対象範囲内のサービス](#)を参照してください。

トピック

- [Audience \(p. 35\)](#)
- [アイデンティティを使用した認証 \(p. 35\)](#)
- [管理AWS CloudShellIAM ポリシーでのアクセスと使用 \(p. 36\)](#)

Audience

を使用する方法AWS Identity and Access Management(IAM) は、で行う作業によって異なりますAWS CloudShell。

[サービスユーザー] - AWS CloudShell サービスを使用してジョブを実行する場合は、必要なアクセス許可と認証情報を管理者が用意します。作業を実行するためにさらに多くの AWS CloudShell 機能を使用するとき、追加のアクセス許可が必要になる場合があります。アクセスの管理方法を理解すると、管理者から適切なアクセス許可をリクエストするのに役に立ちます。AWS CloudShell の機能にアクセスできない場合は、「[AWS CloudShell のトラブルシューティング \(p. 54\)](#)」を参照してください。

[サービス管理者] - 社内の AWS CloudShell リソースを担当している場合は、通常、AWS CloudShell へのフルアクセスがあります。従業員がどの AWS CloudShell 機能とリソースアクセスする必要があるかを決定するのは管理者の仕事です。その後、サービスユーザーのアクセス許可を変更するリクエストを IAM 管理者に送信します。このページの情報を確認して、IAM の基本概念を理解してください。

IAM 管理者-IAM 管理者の場合は、サービスへのユーザーのアクセスを管理するポリシーの作成方法の詳細について確認する場合があります。特有の詳細はこちら[AWS CloudShell 「」](#)を参照してください。[管理AWS CloudShellIAM ポリシーでのアクセスと使用 \(p. 36\)](#)。

アイデンティティを使用した認証

AWS には、次のタイプのアイデンティティでアクセスできます。

AWS アカウントのルートユーザー

AWS にサインアップするときには、AWS アカウントに関連付けられた E メールアドレスとパスワードを提供します。これらはルート認証情報であり、これらの情報を使用すると、すべての AWS リソースへの完全なアクセスが可能になります。

Important

セキュリティのベストプラクティスとして、IAM の作成にのみルート認証情報を使用する必要があります。管理者グループでの IAM 管理者ユーザー。これは、ユーザーに AWS アカウントへの完全なアクセス権限を付与するグループです。その後、この管理者ユーザーを使用して、制限されたアクセス権限を持つ他の IAM ユーザーとロールを作成できます。詳細については、「」を参照してください。[個々の IAM ユーザーの作成および最初の IAM 管理者のユーザーおよびグループの作成](#)の IAM ユーザーガイド。

IAM ユーザー

AnIAM ユーザーあなたの中のアイデンティティは単にあなたの内にあるAWS特定のカスタムアクセス許可 (でファイルをダウンロードする権限など) を持つアカウントAWS CloudShell). IAM のユーザー名と

パスワードを使用して、セキュリティ保護されたサインインできますAWSのようなウェブページAWS CloudShellコンソール、AWS Management Console、AWSディスカッションフォーラム、およびAWS Supportセンター。

ユーザー名とパスワードに加えて、各ユーザーのアクセスキーを生成することもできます。これらのキーは、いずれかの AWS SDK、AWS Command Line Interface (AWS CLI)、または aws-shell を通じて、プログラムで AWS のサービスにアクセスするときに使用します。AWS SDK、AWS CLI、および aws-shell では、これらのアクセスキーを使用してリクエストが暗号で署名されます。これらのツールを使用しない場合は、リクエストを自分で署名する必要があります。AWS CloudShell では、署名バージョン 4 がサポートされています。これは、インバウンド API リクエストを認証するためのプロトコルです。リクエストの認証の詳細については、「」を参照してください。[署名バージョン 4 の署名プロセス](#)の Amazon Web Services 全般リファレンス。

IAM ロール

AnIAM ロールは、特定のアクセス権限を持ち、アカウントで作成できるもう 1 つの IAM アイデンティティです。IAM ユーザーに似ていますが、特定のユーザーに関連付けられていません。IAM ロールでは、AWS サービスおよびリソースにアクセスするために使用できる一時的なアクセスキーを取得することができます。IAM ロールと一時的な認証情報は、次の状況で役立ちます。

AWS サービスへのアクセス

アカウントで IAM ロールを使用して付与できますAWSアカウントのリソースにアクセスするためのサービス権限。たとえば、AWS Lambdaに代わって Amazon S3 バケットにアクセスし、バケットに保存されたデータを Amazon Redshift データウェアハウスにロードします。詳細については、IAM ユーザーガイドの「[AWS のサービスにアクセス権限を委任するロールの作成](#)」を参照してください。

Amazon EC2 で実行中のアプリケーション

インスタンスで実行されるアプリケーションで使用するために、アクセスキーを Amazon EC2 インスタンス内に保存する代わりにAWSAPI リクエストの場合は、IAM ロールを使用して、これらのアプリケーションの一時認証情報を管理できます。を割り当てるにはAWSロールを Amazon EC2 インスタンスに適用し、そのすべてのアプリケーションで使用できるようにするには、インスタンスプロファイルこれは、インスタンスにアタッチされています。インスタンスプロファイルにはロールが含まれ、Amazon EC2 インスタンスで実行されるプログラムは一時認証情報を取得することができます。詳細については、「」を参照してください。[インスタンスプロファイルを作成して使用し一時認証情報を管理する](#)および[IAM ロールを使用して、Amazon EC2 インスタンスで実行されるアプリケーションにアクセス許可を付与する](#)の IAM ユーザーガイド。

フェデレーティッドユーザーアクセス

IAM ユーザーを作成する代わりに、既存のユーザー ID をAWS Directory Service、エンタープライズユーザーディレクトリ、またはウェブ ID プロバイダー。このようなユーザーはフェデレーティッドユーザーと呼ばれます。では、ID プロバイダーAWSを通じてアクセスがリクエストされたとき、フェデレーティッドユーザーにロールを割り当てます。詳細については、IAM ユーザーガイドの「[フェデレーティッドユーザーとロール](#)」を参照してください。

管理AWS CloudShellIAM ポリシーでのアクセスと使用

によって提供されるアクセス管理リソースを使用してAWS Identity and Access Management(IAM)、管理者は IAM ユーザーにアクセス権限を付与して、アクセスできるようにすることができます。AWS CloudShell環境の機能を使用します。管理者は、ユーザーがシェル環境で実行できるアクションをきめ細かく指定するポリシーを作成することもできます。

管理者がユーザーにアクセス権を付与する最も簡単な方法は、AWS管理ポリシー。An[AWS 管理ポリシー](#)は、が作成および管理するスタンドアロンポリシーですAWS。以下ようになりますAWS用の管理ポリシーAWS CloudShellIAM アイデンティティにアタッチできます。

- **AWS CloudShellFullAccess**: を使用するアクセス許可を付与しますAWS CloudShellすべての機能へのフルアクセスが可能です。

-AWS CloudShellFullAccessポリシーでは、ワイルドカード (*) 文字を使用して IAM ID (ユーザー、ロール、またはグループ) にAWS CloudShellと機能。AWS CloudShellFullAccess ポリシーを、許可されたユーザーアクションに関してより限定的なカスタムポリシーのテンプレートとして使用することもできます。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "CloudShellUser",
    "Effect": "Allow",
    "Action": [
      "cloudshell:*"
    ],
    "Resource": "*"
  }]
}
```

Note

での IAM アイデンティティAWS以下に示す管理ポリシーも起動できますAWS CloudShell。しかし、これらのポリシーは非常に広範なアクセス権限を提供するため、最小権限の原則は、IAM ユーザーのジョブロールに必須である場合にのみ付与する必要があります。

- **管理者**: ユーザーにフルアクセスを提供し、のあらゆるサービスおよびリソースにアクセス許可を委任できますAWS。
- **開発者パワーユーザー**: ユーザーがアプリケーション開発タスクを実行し、をサポートするリソースとサービスを作成および設定することができますAWSアプリケーション開発を認識します。

管理ポリシーのアタッチ方法の詳細については、「」を参照してください。[IAM ID 権限の追加 \(コンソール\)](#)のIAM ユーザーガイド。

で許可されるアクションを管理するAWS CloudShellカスタムポリシーの使用

IAM ユーザーが実行できるアクションの範囲を管理するにはAWS CloudShellでは、AWS CloudShellFullAccess 管理ポリシーをテンプレートとして使用するカスタムポリシーを作成できます。または、[インラインポリシー](#)これは、関連する IAM アイデンティティ (ユーザー、グループ、またはロール) に埋め込まれています。

たとえば、ユーザーにアクセスを許可することができますAWS CloudShellただし、シェル環境内でファイルをアップロードまたはダウンロードできないようにします。へのユーザーのアクセスを明示的に拒否できますAWS CloudShell。

Important

を起動するにはAWS CloudShellからのAWS Management Consoleでは、IAM ユーザーは次のアクションに対するアクセス権限が必要です。

- `CreateEnvironment`
- `CreateSession`
- `GetEnvironmentStatus`

- StartEnvironment

これらのアクションのいずれかがアタッチされたポリシーによって明示的に許可されていない場合、CloudShell の起動時に IAM アクセス権限エラーが返されます。

AWS CloudShell アクセス権限

| 名前 | 付与されたパーミッションの説明 | CloudShell の起動に必要? |
|---------------------------------|--|--------------------|
| cloudshell:CreateEnvironment | CloudShell 環境を作成する | あり |
| cloudshell:CreateSession | から CloudShell 環境 Connect。AWS Management Console | あり |
| cloudshell:GetEnvironmentStatus | CloudShell 環境のステータスを読み取る | あり |
| cloudshell>DeleteEnvironment | CloudShell 環境を削除する | いいえ |
| cloudshell:GetFileDownloadUrls | CloudShell からローカルマシンにファイルをダウンロードする | いいえ |
| cloudshell:GetFileUploadUrls | ローカルマシンから CloudShell にファイルをアップロードする | いいえ |
| cloudshell:PutCredentials | へのログインに使用したクレデンシャルを転送します。AWS Management Console CloudShell へ | いいえ |
| cloudshell:StartEnvironment | 停止した CloudShell 環境を起動する | あり |
| cloudshell:StopEnvironment | 実行中の CloudShell 環境を停止する | いいえ |

CloudShell の IAM ポリシー例

次の例は、アクセス可能なユーザーを制限するためのポリシーの作成方法を示しています。AWS CloudShell シェル環境で実行可能なアクション。

このポリシーは、への完全なアクセス拒否を強制します。AWS CloudShell とその機能：

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "DenyCloudShell",
    "Effect": "Deny",
    "Action": [
      "cloudshell:*"
    ],
    "Resource": "*"
  }]
}
```

このポリシーでは、へのアクセスをユーザーに許可します。AWS CloudShellただし、シェル環境でのファイルのアップロードとダウンロードをブロックします。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "CloudShellUser",
    "Effect": "Allow",
    "Action": [
      "cloudshell:*"
    ],
    "Resource": "*"
  }, {
    "Sid": "DenyUploadDownload",
    "Effect": "Deny",
    "Action": [
      "cloudshell:GetFileDownloadUrls",
      "cloudshell:GetFileUploadUrls"
    ],
    "Resource": "*"
  }
]
```

次のポリシーでは、へのアクセスをユーザーに許可しますAWS CloudShellただし、ログインに使用した資格情報は禁止されますAWS Management ConsoleCloudShell 環境に転送されるからです。このポリシーを使用するユーザーは、でクレデンシャルを手動で構成する必要があります。AWS CloudShell。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "CloudShellUser",
    "Effect": "Allow",
    "Action": [
      "cloudshell:*"
    ],
    "Resource": "*"
  }, {
    "Sid": "DenyCredentialForwarding",
    "Effect": "Deny",
    "Action": [
      "cloudshell:PutCredentials"
    ],
    "Resource": "*"
  }
]
```

へのアクセス許可AWSのサービス

AWS CloudShellは、へのサインインに使用した IAM 認証情報を使用しますAWS Management Console。この事前認証機能AWS CloudShell非常に便利に使えるAWS CLI。ただし、IAM ユーザーには引き続き明示的なアクセス権限が必要です。AWSコマンドラインから呼び出されるサービス。

たとえば、IAM ユーザーが Amazon S3 バケットを作成し、ファイルをオブジェクトとしてアップロードする必要がある場合、それらのアクションを明示的に許可するポリシーを作成できます。IAM コンソールは直観的な機能を提供します[ビジュアルエディタ](#)これは、JSON 形式のポリシードキュメントを作成するプロセスについて説明します。ポリシーを作成したら、それを関連する IAM ID (ユーザー、グループ、またはロール) にアタッチできます。

管理ポリシーのアタッチ方法の詳細については、「」を参照してください。[IAM ID 権限の追加 \(コンソール\)](#)のIAM ユーザーガイド。

でのログ記録とモニタリングAWS CloudShell

CloudTrail によるアクティビティのモニタリング

AWS CloudShell は AWS CloudTrail と統合されています。このサービスは、ユーザーやロール、または AWS の AWS CloudShell のサービスによって実行されたアクションを記録するサービスです。CloudTrail は、AWS CloudShell のすべての API コールをイベントとしてキャプチャします。キャプチャされた呼び出しには、AWS CloudShell コンソールからの呼び出しと、AWS CloudShell API へのコード呼び出しが含まれます。

証跡を作成する場合は、のイベントなど、Amazon Simple Storage Service (Amazon S3) バケットへの CloudTrail イベントの継続的な配信を有効にすることができます。AWS CloudShell。

証跡を設定しない場合でも、CloudTrail コンソールの [Event history] で最新のイベントを表示できます。CloudTrail で収集された情報を使用して、AWS CloudShell に対するリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト日時などの詳細を確認できます。

AWS CloudShellCloudTrail では

AWS CloudShell は、CloudTrail ログファイルのイベントとして以下のアクションのログ付けをサポートします。

- `createEnvironment`
- `createSession`
- `deleteEnvironment`
- `getEnvironmentStatus*`
- `getFileDownloadUrls*`
- `getFileUploadUrls*`
- `putCredentials`
- `redeemCode`
- `sendHeartBeat`
- `startEnvironment`
- `stopEnvironment`

*変更なし (読み取り専用) の API 呼び出し。

名前に「Environment」という単語が含まれるイベントは、シェルエクスペリエンスをホストするコンピューティング環境のライフサイクルに関連します。

-`sendHeartBeat` イベントは、セッションがアクティブでないことを確認するために発生します。そして、`putCredentials` イベントは、ユーザーがコンソールにサインインした資格情報が転送されたときに発生します。AWS CloudShell。

の EventBridge ルールAWS CloudShellのアクション

EventBridge ルールでは、EventBridge がルールに一致するイベントを受信したときに実行するターゲットアクションを指定します。に基づいて、実行するターゲットアクションを指定するルールを定義できます。AWS CloudShellCloudTrail ログファイルにイベントとして記録されるアクション。

例えば、以下のことが可能です。[を使用して EventBridge ルールを作成するAWS CLI](#)を使用する`put-rule`コマンド。`put-rule` コールには、少なくとも `EventPattern` または `ScheduleExpression` を含める必要があります。`EventPattern` を含むルールは、一致するイベントが確認されときにトリガーされます。用のEventPatternAWS CloudShellイベント:

```
{ "source": [ "aws.cloudshell" ], "detail-type": [ "AWS API Call via CloudTrail" ],  
  "detail": { "eventSource": [ "cloudshell.amazonaws.com" ] } }
```

詳細については、Amazon EventBridge ユーザーガイドの「[Events and Event Patterns in EventBridge](#)」を参照してください。

AWS CloudShell のコンプライアンス検証

サードパーティーの監査者は、複数の AWS コンプライアンスプログラムの一部として AWS のサービスのセキュリティとコンプライアンスを評価します。

現在、AWS CloudShell は、特定のコンプライアンスプログラムの対象外です。

特定のコンプライアンスプログラムの対象となる AWS のサービスのリストについては、「[コンプライアンスプログラムによる AWS 対象範囲内のサービス](#)」を参照してください。一般的な情報については、「[AWS コンプライアンスプログラム](#)」を参照してください。

サードパーティーの監査レポートをダウンロードするには、AWS Artifact を使用します。詳細については、「[Downloading Reports in AWS](#)」および「[AWS Artifact](#)」を参照してください。

AWS CloudShell を使用する際のお客様のコンプライアンス責任は、お客様のデータの機密性や貴社のコンプライアンス目的、適用される法律および規制によって決まります。AWS では、コンプライアンスに役立つ以下のリソースを提供しています。

- [セキュリティとコンプライアンスのクイックスタートガイド](#)：これらのデプロイメントガイドでは、アーキテクチャ上の考慮事項について説明し、セキュリティ重視およびコンプライアンスに重点を置いたベースライン環境をにデプロイするための手順を説明します。AWS。
- [HIPAA セキュリティおよびコンプライアンスのためのアーキテクチャ設計ホワイトペーパー](#) - このホワイトペーパーは、企業が AWS を使用して HIPAA 準拠のアプリケーションを作成する方法を説明します。
- [AWS コンプライアンスのリソース](#) - このワークブックおよびガイドのコレクションは、ユーザーの業界や地域で使用できるかもしれません。
- 『AWS Config デベロッパーガイド』の「[ルールでのリソースの評価](#)」 - AWS Config サービスは、リソース設定が社内のプラクティス、業界のガイドライン、規制にどの程度準拠しているかを評価します。
- [AWS Security Hub](#): この AWS サービスでは、AWS 内のセキュリティ状態を包括的に表示しており、セキュリティ業界の標準およびベストプラクティスへの準拠を確認するのに役立ちます。

での耐障害性AWS CloudShell

AWS のグローバルインフラストラクチャは AWS リージョンとアベイラビリティゾーンを中心に構築されます。AWS リージョンには、低レイテンシー、高いスループット、そして高度の冗長ネットワークで接続されている複数の物理的に独立・隔離されたアベイラビリティゾーンがあります。アベイラビリティゾーンでは、ゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、耐障害性、および拡張性が優れています。

AWS リージョンとアベイラビリティゾーンの詳細については、「[AWS グローバルインフラストラクチャ](#)」を参照してください。

AWS CloudShell では、AWS グローバルインフラストラクチャに加えて、データの耐障害性とバックアップのニーズに対応するための機能をサポートしています。

- 作成して追加したファイルをコミットするAWS CodeCommit。これは、クラウド内のアセットを非公開で保存および管理するために使用できるAmazon Web Services によってホストされるバージョン管理サービスです。これらのアセットは、ドキュメント、ソースコード、およびバイナリファイルで構成できます。詳細については、「[チュートリアル: CodeCommit の使用の詳細については、AWS CloudShell \(p. 17\)](#)」を参照してください。
- を使用するAWS CLIのホームディレクトリ内のファイルを指定するための呼び出しAWS CloudShellそして、それらを Amazon S3 バケット内のオブジェクトとして追加します。例については、「」を参照してください。 [入門チュートリアル \(p. 6\)](#)。

AWS CloudShell でのインフラストラクチャセキュリティ

マネージドサービスとして、AWS CloudShellによって保護されているAWSで説明されているグローバルネットワークセキュリティ手順[Amazon Web Services: セキュリティプロセスの概要](#)ホワイトペーパー。

AWS が公開している API コールを使用して、ネットワーク経由で AWS CloudShell にアクセスします。クライアントで Transport Layer Security (TLS) 1.2 以降がサポートされている必要があります。TLS 1.2 以降が推奨されています。また、Ephemeral Diffie-Hellman (DHE) や Elliptic Curve Ephemeral Diffie-Hellman (ECDHE) などの Perfect Forward Secrecy (PFS) を使用した暗号スイートもクライアントでサポートされている必要があります。これらのモードは、Java 7 以降など、最近のほとんどのシステムでサポートされています。

また、リクエストは、アクセスキー ID と、IAM プリンシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、[AWS Security Token Service \(AWS STS\)](#) を使用して、一時的なセキュリティ認証情報を生成し、リクエストに署名することもできます。

Note

デフォルトでは、AWS CloudShellは、コンピューティング環境のシステムパッケージのセキュリティパッチを自動的にインストールします。

AWS CloudShell での設定と脆弱性の分析

それはの責任ですAWS CloudShellユーザーは、コンピューティング環境にインストールされているサードパーティ製ソフトウェアにパッチが適用され、最新であることを確認します。

AWS CloudShell のセキュリティのベストプラクティス

以下のベストプラクティスは一般的なガイドラインであり、完全なセキュリティソリューションに相当するものではありません。これらのベストプラクティスはお客様の環境に適切ではないか、十分ではない場合があるため、絶対的な解決策ではなく、役に立つ情報としてお考えください。

AWS CloudShell のセキュリティの一部のベストプラクティス

- IAM のアクセス許可とポリシーを使用して、へのアクセスをコントロールするAWS CloudShellまた、ユーザーがロールで必要な操作 (ファイルのダウンロードやアップロードなど) のみを実行できるようにします。詳細については、「[管理AWS CloudShellIAM ポリシーでのアクセスと使用 \(p. 36\)](#)」を参照してください。

- セーフペースト機能を有効にして、外部ソースからコピーしたテキスト内の潜在的なセキュリティリスクをキャッチします。デフォルトでは、「セーフペースト」は有効になっています。詳細については、「[マルチテキストにセーフペーストを使用する \(p. 32\)](#)」を参照してください。
- について理解を深めます。共有セキュリティ責任共有モデルサードパーティアプリケーションをのコンピューティング環境にインストールした場合AWS CloudShell。
- ユーザーのシェルエクスペリエンスに影響を与えるシェルスクリプトを編集する前に、ロールバックメカニズムを準備します。詳細については、「[スクリプトでシェルを修正する \(p. 52\)](#)」を参照してください。
- コードをバージョン管理システムに安全に保存します。例: [AWS CodeCommit](#)。

AWS CloudShellコンピューティング環境:仕様とソフトウェア

起動するとAWS CloudShellに基づくコンピューティング環境Amazon Linux 2シェルエクスペリエンスをホストするために作成されます。環境は、[コンピューティングリソース \(vCPU とメモリ\)](#) (p. 44)そして、の広い範囲を提供します[プリインストールされたソフトウェア](#) (p. 44)コマンドラインインターフェイスからアクセスできます。ソフトウェアをインストールし、シェルスクリプトを変更して、デフォルト環境を構成することもできます。

コンピューティング環境リソース

EACHAWS CloudShellコンピューティング環境には、次の CPU およびメモリリソースが割り当てられます。

- 1 vCPU (仮想中央処理装置)
- RAM 2 GiB

さらに、環境は次のストレージ構成でプロビジョニングされます。

- 1 GB の永続ストレージ (セッション終了後もストレージは保持されます)

詳細については、「[永続的ストレージ](#) (p. 58)」を参照してください。

Important

現在、AWS CloudShellコンピューティング環境は Docker コンテナをサポートしていません。

プリインストールされたソフトウェア

Note

なぜなら、AWS CloudShell開発環境は、最新のソフトウェアへのアクセスを提供するために定期的に更新されています。このドキュメントでは、特定のバージョン番号は提供していません。代わりに、インストールされているバージョンを確認する方法を説明します。多くの場合、これは、プログラム名の後に`--version`オプション (たとえば、`git --version`)。

Shells

プリインストールされたシェル

| 名前 | 説明 | [Version information] |
|------------|--|---------------------------------|
| Bash | Bash シェルは、のデフォルトのシェルアプリケーションです。AWS CloudShell。 | <code>bash --version</code> |
| PowerShell | コマンドラインインターフェイスとスクリプト言語のサポートを提供する PowerShell は、マイクロソフトの.NET コマンド | <code>(Get-Host).Version</code> |

| 名前 | 説明 | [Version information] |
|------------|---|-----------------------|
| | 言語ランタイムの上に構築されています。PowerShell は、という軽量コマンドを使用しています。cmdlets.NET オブジェクトを受け入れて返します。 | |
| Zシエル (zsh) | Zシエル、別名zshは、テーマとプラグインのカスタマイズサポートを強化したBourne Shellの拡張バージョンです。 | zsh --version |

AWSコマンドラインインターフェイス (CLI)

CLI

| 名前 | 説明 | [Version information] |
|----------------|---|-----------------------|
| AWS CLI | <p>-AWS CLIは、複数の管理に使用できるコマンドラインインターフェイスです。AWSサービスをコマンドラインから実行したり、スクリプトを使用して自動化したりできます。詳細については、「の使用AWS内のサービスAWS CloudShell (p. 28)」を参照してください。</p> <p>最新バージョンの使用を保証する方法については、を参照してください。AWS CLIバージョン2の場合は、を参照してください。のインストールAWS CLIホームディレクトリに移動します (p. 51)。</p> | aws --version |
| EB CLI | <p>-AWS Elastic BeanstalkCLIには、ローカルリポジトリからの環境の作成、更新、モニタリングを簡素化するコマンドラインインターフェイスが用意されています。</p> <p>詳細については、「」を参照してください。Elastic Beanstalk コマンドラインインターフェイス (EB CLI) の使用のAWS Elastic Beanstalkデベロッパーガイド。</p> | eb --version |
| Amazon ECS CLI | Amazon Elastic Container Service (Amazon ECS) コマンドラインインターフェイス (CLI)には、クラスターおよびタスクの作成、更新、モニタリングを簡素化する高レベルコマンドが用意されています。 | ecs-cli --version |

| 名前 | 説明 | [Version information] |
|-------------|---|--------------------------|
| | <p>詳細については、「」を参照してください。Amazon ECS コマンドラインインターフェイスの使用のAmazon Elastic Container Service 開発者ガイド。</p> | |
| AWS SAM CLI | <p>AWS SAMCLI は、で動作するコマンドラインツールです。AWS Serverless Application Model テンプレートとアプリケーションコード。いくつかのタスクを実行できます。これには、ローカルで Lambda 関数を呼び出す、サーバーレスアプリケーションのデプロイパッケージを作成する、サーバーレスアプリケーションをAWS.cloud</p> <p>詳細については、「」を参照してください。AWS SAMCLI コマンドリファレンスのAWS Serverless Application Model デベロッパーガイド。</p> | <pre>sam --version</pre> |

ランタイムと AWS SDK: Node.js と Python 3

ランタイムと AWS SDK

| 名前 | 説明 | [Version information] |
|-------------------------------|--|---|
| Node.js (npm付き) | <p>Node.js は、非同期プログラミング手法を簡単に適用できるように設計された JavaScript ランタイムです。詳細については、「」を参照してください。Node.js の公式サイトのドキュメント。</p> <p>npm は JavaScript モジュールのオンラインレジストリへのアクセスを提供するパッケージマネージャーです。詳細については、「」を参照してください。公式npmサイトのドキュメント。</p> | <ul style="list-style-type: none"> • Node.js:node --version • npm:npm --version |
| Node.js 内の SDK for JavaScript | <p>このソフトウェア開発キット (SDK) は、Amazon S3、Amazon EC2、DynamoDB、Amazon SWF などの AWS のサービスに JavaScript オブジェクトを提供することにより、コーディングを簡素化します。詳細については、「AWS SDK for JavaScript</p> | <pre>npm -g ls --depth 0 2>/dev/null grep aws-sdk</pre> |

| 名前 | 説明 | [Version information] |
|------------------------|---|---|
| | デベロッパーガイド 」を参照してください。 | |
| Python | <p>Python 3 と Python 2 はどちらもシェル環境で使用できます。Python 3 は現在、プログラミング言語のデフォルトバージョンと見なされます (Python 2 のサポートは 2020 年 1 月に終了しました)。詳細については、「」を参照してください。Python 公式サイトのドキュメント。</p> <p>また、Python のパッケージインストーラである pip がプリインストールされています。このコマンドラインプログラムを使用して、Python パッケージインデックスなどのオンラインインデックスから Python パッケージをインストールできます。詳細については、「」を参照してください。Python Packaging Authorityが提供するドキュメント。</p> | <ul style="list-style-type: none"> Python 2:python --version Python 3:python3 --version pip:pip3 --version |
| SDK for Python (Boto3) | <p>Boto はソフトウェア開発キット (SDK) で、Python 開発者は EC2 や S3 などの AWS サービスを作成、設定、管理するために使用します。SDK は、使いやすい、オブジェクト指向 API と AWS サービスへの低レベルのアクセスを提供します。</p> <p>詳細については、「」を参照してください。Boto3 ドキュメント。</p> | pip3 list grep boto3 |

開発ツールとシェルユーティリティ

開発ツールとシェルユーティリティ

| 名前 | 説明 | [Version information] |
|------------|--|--------------------------|
| バッシュコンプリート | <p>bash-completion (bash-completion) は、部分的に型付けされたコマンドや引数のオートコンプリートをタブkey bash-completion がサポートするパッケージは /usr/share/bash-completion/completions。</p> <p>パッケージのコマンドのオートコンプリートを設定するには、</p> | yum info bash-completion |

| 名前 | 説明 | [Version information] |
|---------------------------|--|---|
| | <p>プログラムファイルをソースする必要があります。たとえば、Git コマンドのオートコンプリートを設定するには、次の行を追加します。 .bashrc そのため、この機能はいつでも利用できます。AWS CloudShell セッションが開始されます。</p> <pre>source /usr/share/bash-completion/completions/git</pre> <p>カスタム補完スクリプトを使用する場合は、それらを永続的なホームディレクトリに追加します (\$HOME) で直接ソースする .bashrc。</p> <p>詳細については、「」を参照してください。 README GitHub のページです。</p> | |
| Git 用の CodeCommit ユーティリティ | <p>git-remote-codeCommit は Git を拡張することにより、CodeCommit リポジトリからコードをプッシュおよびプルするための簡単な方法を提供するユーティリティです。これは、フェデレーテッドアクセス、ID プロバイダー、および一時的な認証情報を使用した接続をサポートするために推奨される方法です。</p> <p>詳細については、「」を参照してください。 への HTTPS 接続をセットアップする手順 AWS CodeCommit git-remote-codecommit での AWS CodeCommit ユーザーガイド。</p> | <pre>pip3 list grep git-remote-codecommit</pre> |
| Git | <p>Git は、ブランチワークフローとコンテンツのステージングを通じて、最新のソフトウェア開発プラクティスをサポートする分散バージョン管理システムです。詳細については、「」を参照してください。 Git の公式サイトのドキュメントページ。</p> | <pre>git --version</pre> |

| 名前 | 説明 | [Version information] |
|------------|--|-------------------------|
| イプチルス | iputils パッケージには Linux ネットワーク用のユーティリティが含まれています。提供されるユーティリティの詳細については、を参照してください。 GitHub の iputils リポジトリ 。 | iputils ツールの例:arping -v |
| jq | jq ユーティリティは JSON 形式のデータを解析して、コマンドラインフィルタによって変更された出力を生成します。詳細については、「」を参照してください。 GitHub でホストされている jq マニュアル 。 | jq --version |
| make | make ユーティリティはmakefiles一連のタスクを自動化し、コードのコンパイルを整理します。詳細については、「」を参照してください。 GNU ドキュメントを作成する 。 | make --version |
| man | man コマンドは、コマンドラインユーティリティとツールのマニュアルページを提供します。たとえば、man lsのマニュアルページを返します。lsディレクトリの内容を一覧表示するコマンド。詳細については、「」を参照してください。 ウィキペディアのマニュアルページへのエントリ 。 | man --version |
| procps | procps は、現在実行中のプロセスを監視および停止するために使用できるシステム管理ユーティリティです。詳細については、「」を参照してください。 procps で実行できるプログラムをリストする README ファイル 。 | ps --version |
| SSH クライアント | SSH クライアントは、リモートコンピュータとの暗号化通信にセキュアシェルプロトコルを使用します。OpenSSH は、プリインストールされている SSH クライアントです。詳細については、「」を参照してください。 OpenSSH サイト OpenBSD によって維持されています。 | ssh -v |

| 名前 | 説明 | [Version information] |
|-------|--|-----------------------|
| sudo | sudo ユーティリティを使用すると、ユーザーは別のユーザー (通常はスーパーユーザー) のセキュリティ権限でプログラムを実行できます。Sudo は、システム管理者としてアプリケーションをインストールする必要がある場合に便利です。詳細については、「」を参照してください。 sudo マニュアル 。 | sudo --version |
| tar | tar は、複数のファイルを単一のアーカイブファイル (tarball と呼ばれることが多い) にグループ化するために使用できるコマンドラインユーティリティです。詳細については、「」を参照してください。 GNU tar ドキュメント 。 | tar --version. |
| tmux | tmux は、複数のウィンドウで異なるプログラムを同時に実行するために使用できるターミナルマルチプレクサです。実行中、tmux はウィンドウの下部にあるステータスバーに現在のセッションに関する情報を表示します。詳細については、「」を参照してください。 tmux の簡潔な紹介を提供するブログ 。 | tmux -V |
| unzip | zip/unzipを参照 | |
| vim | vim は、ユーザーがテキストベースのインターフェイスを介して対話するカスタマイズ可能なエディタです。詳細については、「」を参照してください。 vim.org で提供される ドキュメントリソース 。 | vim --version |
| wget | wget は、コマンドラインでエンドポイントによって指定された Web サーバーからコンテンツを取得するために使用されるコンピュータプログラムです。詳細については、「」を参照してください。 GNU Wget ドキュメント 。 | wget --version |

| 名前 | 説明 | [Version information] |
|------------|---|----------------------------------|
| zip /enzip | zip/unzip ユーティリティは、データを失うことなくロスレスデータ圧縮を実現するアーカイブファイル形式を使用します。zip コマンドを呼び出して、単一のアーカイブ内のファイルをグループ化して圧縮します。解凍を使用して、アーカイブから指定したディレクトリにファイルを抽出します。 | unzip --version zip --version |

のインストールAWS CLIホームディレクトリに移動します

CloudShell 環境にプリインストールされている他のソフトウェアと同様に、AWS CLIツールは、スケジュールされたアップグレードとセキュリティパッチで自動的に更新されます。の最新バージョンであることを確認するにはAWS CLIでは、シェルのホームディレクトリにツールを手動でインストールするよう選択できます。

Important

のコピーを手動でインストールする必要がありますAWS CLIホームディレクトリで、次回 CloudShell セッションを開始したときに使用できるようにします。これは、ファイルが外部のディレクトリに追加されるためです。\$HOMEシェルセッションが終了すると、削除されます。また、このコピーをインストールした後も注意してください。AWS CLIの場合、自動的に更新されません。つまり、アップデートとセキュリティパッチを管理するのはユーザーの責任です。の詳細については、「」を参照してください。AWS責任共有モデル。「」を参照してください。[AWS CloudShell でのデータ保護 \(p. 33\)](#)。

AWS CLI をインストールするには

1. まず、CloudShell コマンドラインで、curlコマンドで圧縮されたコピーを転送するAWS CLIシェルにインストールする:

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

2. zip フォルダを解凍します。

```
unzip awscliv2.zip
```

3. 次に、を実行します。AWS CLIインストーラーを使用して、指定したフォルダにツールを追加します。

```
sudo ./aws/install --install-dir /home/cloudshell-user/usr/local/aws-cli --bin-dir /home/cloudshell-user/usr/local/bin
```

正常にインストールされると、コマンドラインに次のメッセージが表示されます。

```
You can now run: /home/cloudshell-user/usr/local/bin/aws --version
```

4. 独自の便宜のために、更新も行っておくことをお勧めします。PATH実行時にツールのインストールへのパスを指定する必要がないように、環境変数awsコマンド:

```
export PATH=/home/cloudshell-user/usr/local/bin:$PATH
```

Note

この変更を元に戻すとPATH,aws指定したパスを使用しないコマンドは、プリインストールされたバージョンのを使用します。AWS CLIデフォルトでは。

シェル環境にサードパーティーソフトウェアをインストールする

Note

「」を確認することをお勧めします。共有セキュリティ責任共有モデルサードパーティアプリケーションをにインストールする前にAWS CloudShellのコンピューティング環境。

デフォルトでは、すべてAWS CloudShellユーザーには sudo 権限があります。したがって、sudoコマンドを実行して、シェルのコンピューティング環境でまだ利用できないソフトウェアをインストールします。たとえば、sudoYUM パッケージ管理ユーティリティを使用して GNU nano テキストエディタをインストールします。

```
sudo yum install nano
```

次に、次のように入力して、新しくインストールしたプログラムを起動できます。nano。

Important

yum インストールプログラムなどのユーティリティをディレクトリ内にパッケージ管理する (/user/binなど)。シェルセッションが終了するとリサイクルされます。つまり、セッションごとに追加のソフトウェアがインストールされ、使用されることを意味します。

スクリプトでシェルを修正する

デフォルトのシェル環境を変更する場合は、シェル環境が起動するたびに実行されるシェルスクリプトを編集できます。-.bashrcデフォルトの bash シェルが起動するたびにスクリプトが実行されます。

Warning

を誤って変更した場合.bashrcファイルを使用すると、その後シェル環境にアクセスできないことがあります。編集する前にファイルのコピーを作成することをお勧めします。編集時にシェルを2つ開くことでリスクを軽減することもできます。.bashrc。一方のシェルでアクセスできなくなった場合でも、他のシェルにログインし、変更をロールバックできます。誤って変更した後にアクセスできなくなった場合.bashrcまたは他のファイルを返すことができますAWS CloudShellデフォルト設定に[ホームディレクトリの削除 \(p. 53\)](#)。

この手順では、.bashrcシェル環境が自動的にZシェルの実行に切り替わるようにスクリプトを実行します。

1. を開く.bashrcテキストエディタ (例:Vim) を使用します。

```
vim .bashrc
```

2. エディタインターフェイスで、インスタンスキーを押して、編集を開始し、以下を追加します。


```
zsh
```

3. 編集したを終了して保存するには `.bashrc` ファイル、Esc をクリックして Vim コマンドモードに入り、以下を入力します。

```
:wq
```

4. を使用する `source` コマンドをリロードする `.bashrc` file:

```
source .bashrc
```

コマンドラインインターフェイスが再度使用可能になると、プロンプト記号が `%Z` シェルを使用していることを示します。

ホームディレクトリの削除

Warning

ホームディレクトリを削除することは、ホームディレクトリに保存されているすべてのデータが完全に削除されるという不可逆的なアクションです。ただし、次のような場合は、このオプションを検討することをお勧めします。

- ファイルを間違えて変更し、AWS CloudShell コンピューティング環境。ホームディレクトリを削除すると、AWS CloudShell をデフォルト設定にします。
- からすべてのデータを削除する必要があります AWS CloudShell すぐに。なお、使用をやめた場合 AWS CloudShell で AWS リージョン、永続ストレージは **保持期間の終了時に自動的に削除される** (p. 58) 起動しない限り AWS CloudShell リージョンでもう一度です。

ファイルに長期保存が必要な場合は、Amazon S3 や CodeCommit などのサービスを検討してください。

ホームディレクトリを削除してリセットするには AWS CloudShell

1. CloudShell インターフェイスで、アクション、削除 AWS CloudShell ホームディレクトリ。
2. □ ダイアログボックスで `[delete]` という単語を入力して、削除オプション。

Delete AWS CloudShell home directory



Deleting your home directory will delete all data currently stored in your AWS CloudShell environment. This action cannot be undone. AWS CloudShell stops all active sessions in the current AWS Region and creates a new environment immediately.

To confirm deletion, enter **delete** in the text input field.

```
delete
```

Cancel

Delete

3. [Delete] (削除) をクリックします。

新しい AWS CloudShell デフォルト設定のコンピューティング環境が作成され、起動されます。削除を確認するには、`ls` コマンドをホームディレクトリに追加します。

AWS CloudShell のトラブルシューティング

以下の情報を使用して、AWS CloudShell の問題を特定および対処します。

該当する問題が以下に示されていない場合や、追加のヘルプが必要な場合は、AWS CloudShellディスカッションフォーラム。(このフォーラムにアクセスするには AWS へのサインインが必要になることがあります)。当社に直接お問い合わせいただくこともできます。

トピック

- [を起動できませんAWS CloudShell「環境を開始できません。アクセス許可がありません。IAM 管理者にAWS CloudShell。」](#) (p. 54)
- [アクセスできませんAWS CloudShellのコマンドラインを使用します。](#) (p. 54)
- [外部 IP アドレスに ping を実行できません。](#) (p. 54)

を起動できませんAWS CloudShell「環境を開始できません。アクセス許可がありません。IAM 管理者にAWS CloudShell。」

問題: 起動しようとする AWS CloudShell からの AWS Management Console の場合、アクセスが拒否され、アクセス許可がないことが通知されます。

原因: アクセスに使用している IAM ID AWS CloudShell には、必要な IAM アクセス許可がありません。

解決策: IAM 管理者に、必要なアクセス許可を付与するようリクエストします。これは、アタッチされた AWS 管理ポリシー (AWS CloudShellFullAccess) または埋め込みインラインポリシーのいずれかです。詳細については、「[管理AWS CloudShell IAM ポリシーでのアクセスと使用](#) (p.)」を参照してください。

[\(先頭に戻る \(p. 54\)\)](#)

アクセスできませんAWS CloudShellのコマンドラインを使用します。

問題: コンピューティング環境で使用されるファイルを変更すると、AWS CloudShell。

解決策: 誤って変更した後にアクセスが失われた場合、.bashrc または他のファイルを返す場合は、AWS CloudShell をデフォルト設定に [ホームディレクトリを削除する](#) (p. 53)。

[\(先頭に戻る \(p. 54\)\)](#)

外部 IP アドレスに ping を実行できません。

問題: コマンドラインから ping コマンドを実行すると (ping amazon.com など)、次のメッセージが表示されます。

```
ping: socket: Operation not permitted
```

原因: ping ユーティリティは、インターネット制御メッセージプロトコル (ICMP) を使用して、エコー要求パケットをターゲットホストに送信します。次に、ターゲットからのエコー応答を待機します。ICMP プロトコルはAWS CloudShellの場合、ping ユーティリティはシェルのコンピューティング環境では動作しません。

[\(先頭に戻る \(p. 54\)\)](#)

のサポートされるブラウザAWS CloudShell

次の表は、AWS CloudShell でサポートされるブラウザの一覧です。

ウェブブラウザのサポート

| ブラウザ | Version |
|----------------------|------------------|
| Google Chrome | 最新の 3 つの主要なバージョン |
| Mozilla Firefox | 最新の 3 つの主要なバージョン |
| Microsoft Edge | 最新の 3 つの主要なバージョン |
| MacOS 版 Apple Safari | 最新の 2 つの主要なバージョン |

サポート対象AWSのリージョンAWS CloudShell

サポートされているのリストについてはAWSリージョンおよび関連するサービスエンドポイントの詳細については、『[AWS CloudShellページ](#)でのAmazon Web Services 全般リファレンス。

の制限と制約AWS CloudShell

AWS CloudShellは、追加料金なしで提供されるブラウザベースのシェルです。制限と制約は、以下の領域に適用されます。

- 永続的ストレージ (p. 58)
- 毎月の使用状況 (p. 58)
- 同時シェル数 (p. 58)
- シェルセッション (p. 59)
- ネットワークアクセスとデータ転送 (p. 59)
- システムファイルとページのリロード (p. 59)

永続的ストレージ

とAWS CloudShellでは、それぞれに 1 GB の永続ストレージがあります。AWS無償のリージョン。永続ストレージはあなたのホームディレクトリ (\$HOME) にあり、あなたにはプライベートです。各シェルセッションが終了した後にリサイクルされる一時的な環境リソースとは異なり、ホームディレクトリ内のデータはセッション間で保持されます。

AWS リージョンで AWS CloudShell の使用を停止した場合、データは120 日最後のセッションの終了後。アクションを実行しない限り、120 日経過すると、そのリージョンの永続ストレージからデータが自動的に削除されます。単に起動することで削除を防ぐことができますAWS CloudShellをもう一度実行します。AWSリージョン。

Note

使用シナリオ

マルシアが使用しているAWS CloudShellを使用して、ホームディレクトリにファイルを2つのAWSリージョン: 米国東部 (バージニア北部) およびヨーロッパ (アイルランド)。彼女はその後、AWS CloudShellは、米国東部 (バージニア北部) でシェルセッションを開始することを停止しました。

米国東部 (バージニア北部) でのデータの削除期限前に、Márciaはホームディレクトリがリサイクルされないようにすることを決定し、AWS CloudShellをクリックし、米国東部 (バージニア北部) リージョンをもう一度選択します。シェルセッションにヨーロッパ (アイルランド) を継続的に使用しているため、そのリージョンの永続的なストレージは影響を受けません。

毎月の使用状況

月ごとの使用制限がありますAWS CloudShell/AWSリージョンのAWSアカウント。あなたがアクセスしようとするAWS CloudShellそのリージョンの月次制限に達すると、シェル環境を起動できない理由を説明するメッセージが表示されます。

Note

月ごとの使用制限を引き上げる必要がある場合は、[AWS サポート](#)。

同時シェル数

- 同時シェル数: 最大 10 個のシェルを同時に実行することができます。AWS無料でのリージョン。

シェルセッション

- 非アクティブセッション: AWS CloudShellは対話型シェル環境です。キーボードやポインタを使用して対話しない場合、20 — 30 分を呼び出すと、シェルセッションが終了します。(実行中のプロセスはインタラクションとしてカウントされません)。
- 長時間実行セッション: 約 12 時間連続して実行されているシェルセッションは、ユーザーがその期間中に定期的に対話している場合でも、自動的に終了します。

ネットワークアクセスとデータ転送

AWS CloudShell 環境に出入りするネットワークトラフィックには、以下の制限が適用されます。

- アウトバウンド: ユーザーはパブリックインターネットにアクセスできます。
- インバウンド: ユーザーは受信ポートにアクセスできません。パブリック IP アドレスは使用できません。

Warning

公共のインターネットにアクセスすると、特定のユーザーがAWS CloudShell環境。IAM 管理者は、信頼できるAWS CloudShellユーザーに IAM ツールを使用します。特定のユーザーのアクセスを明示的に拒否する方法については、[「で許可されるアクションを管理するAWS CloudShellカスタムポリシーの使用 \(p. 37\)」](#)。

データ転送: ファイルのアップロードおよびダウンロードAWS CloudShellは、大きなファイルの場合は遅くなる可能性があります。シェルのコマンドラインインターフェイスを使用して、Amazon S3 バケットから環境にファイルを転送することもできます。

システムファイルとページのリロードに関する制限

- システムファイル: コンピューティング環境で必要なファイルを誤って変更すると、AWS CloudShell環境。このような場合は、[ホームディレクトリを削除する \(p. 53\)](#)をクリックしてアクセスを回復します。
- ページの再ロード: を再ロードするにはAWS CloudShellインターフェイスを使用する場合は、オペレーティングシステムのデフォルトのショートカットキーシーケンスの代わりに、ブラウザのリフレッシュボタンを使用する必要があります。

AWS CloudShell ユーザーガイドのドキュメント履歴

最新の更新

以下の表は、AWS CloudShell ユーザーガイドの重要な変更点をまとめたものです。

| update-history-change | update-history-description | update-history-date |
|--|--|---------------------|
| タブキーのオートコンプリート (p. 60) | bash-completion の使用方法を説明した新しいドキュメントで、部分的に型付けされたコマンドまたは引数の自動補完をタブkey。 | 2021年9月24日 |
| 指定するAWSリージョン (p. 24) | デフォルトの指定に関するドキュメントAWSのリージョンAWS CLIコマンド。 | 2021年5月11日 |
| 新規AWSのリージョンAWS CloudShell (p. 60) | AWS CloudShellはでも利用可能になりました。AWS地域: <ul style="list-style-type: none"> ・ アジアパシフィック (ムンバイ) ・ アジアパシフィック (シドニー) ・ 欧州 (フランクフルト) | 2021年4月13日 |
| PDF および Kindle 版での書式設定 (p. 60) | 表セル内の画像サイズとテキストが修正されました | 2021年3月10日 |
| の一般可用性 (GA) リリースAWS CloudShellが選択されています。AWSリージョン (p. 60) | AWS CloudShellは一般利用可能です。AWS地域: <ul style="list-style-type: none"> ・ 米国東部 (オハイオ) ・ 米国東部 (バージニア北部) ・ 米国西部 (オレゴン) ・ アジアパシフィック (東京) ・ 欧州 (アイルランド) | 2020年12月15日 |

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。