入門ガイド

AWS Management Console

Version 1.0

AWS Management Console: 入門ガイド

Table of Contents

とは AWS Management Console

AWS Management Console は、 AWS リソースを管理するための幅広いサービスコンソールのコレクションで構成され、参照するウェブアプリケーションです。最初にサインインすると、コンソールのホームページが表示されます。各サービスコンソールにアクセスできるホームページは、 AWS に関連するタスクを実行するために必要な情報にアクセスするための単一の場所として機能します。また、最近アクセスした 、AWS ヘルス などのウィジェットを追加、削除、再配置することで、コンソールホームエクスペリエンスをカスタマイズすることもできます。

> **Note**
>
> 言語選択オプションが新しい [Unified Settings] (統合設定) ページに移動しました。詳細については、AWS Management Consoleの言語の変更を参照してください。

その一方で、個々のサービスコンソールは、クラウドコンピューティングのための様々なツールと、お客様のアカウントおよび請求に関する情報を提供しています。

任意のデバイスを使用する

AWS Management Console はタブレットおよび他のデバイスで使用できるように設計されています。

- 横および縦のスペースは画面により多くの情報を表示するよう最大化できます。

- ボタンとセレクタはより大きく、タッチしやすくなっています。

AWS Management Console は Android および iOS 用のアプリとしても利用できます。このアプリケーションは完全なウェブ体験の補助として、モバイル関連の作業を行うのに適しています。例えば、電話から既存の Amazon EC2 インスタンスと Amazon CloudWatch アラームを簡単に表示および管理できます。

AWS コンソールモバイルアプリは、Amazon Appstore、Google Play、または iTunes からダウンロードできます。

の設定 AWS Management Console

このトピックでは、 を設定する方法 AWS Management Console と、統合設定ページを使用して、すべてのサービスコンソールに適用されるデフォルトを設定する方法について説明します。また、コンソールホームダッシュボードの機能であるウィジェットについても説明します。ウィジェットを使用すると、 AWS サービスとリソースに関する情報を追跡するカスタムコンポーネントを追加できます。

トピック

ウィジェットの操作

コンソールホームダッシュボードには、 AWS 環境に関する重要な情報を表示し、サービスへのショートカットを提供するウィジェットが含まれています。ウィジェットの追加と削除、再配置、またはサイズの変更により、エクスペリエンスをカスタマイズできます。

ウィジェットを追加するには

1. コンソールホームダッシュボードの右上または右下にある [ウィジェットの追加] ボタンを選択します。
2. ウィジェットのタイトルバーの左上にある 6 つの縦のドットが示すドラッグインジケーターを選択し、コンソールホームダッシュボードまでドラッグします。

ウィジェットを削除するには

1. ウィジェットタイトルバーの右上にある 3 つの縦のドットが示す省略記号を選択します。
2. [Remove widget] (ウィジェットの削除) を選択します。

ウィジェットを並べ替えるには

- ウィジェットのタイトルバーの左上にある 6 つの縦のドットが示すドラッグインジケーターを選択し、コンソールホームダッシュボードの新しい場所までドラッグします。

ウィジェットのサイズを変更するには

- ウィジェットの右下にあるサイズ変更アイコンを選択し、ウィジェットをページの新しい場所までドラッグします。

ウィジェットの整理と設定をやり直す場合は、コンソールホームダッシュボードをデフォルトのレイアウトにリセットできます。これにより、コンソールホームダッシュボードレイアウトへの変更が元に戻り、すべてのウィジェットがデフォルトの場所とサイズに復元されます。

ページをデフォルトレイアウトにリセットするには

1. ページの右上にある [デフォルトレイアウトにリセット] ボタンを選択します。
2. 確認するには、[リセット] を選択します。

> **Note**
>
> これにより、コンソールホームダッシュボードのレイアウトに対するすべての変更が元に戻ります。

コンソールホームダッシュボードで新しいウィジェットをリクエストするには

1. コンソールホームダッシュボードの左下にある [別のウィジェットをご希望の場合は、当社までお知らせください。] を選択します。

   コンソールホームダッシュボードへの追加を希望するウィジェットについて説明します。
2. [送信] を選択します。

   > **Note**
   >
   > お客様の提案は定期的に確認されており、今後の AWS Management Consoleのアップデートで新しいウィジェットが追加される可能性があります。

統合設定の指定

統合設定ページから、表示、言語、リージョンなどの AWS Management Console 設定とデフォルトを設定できます。ビジュアルモードとデフォルトの言語は、ナビゲーションバーから直接設定することもできます。これらの変更は、すべてのサービスコンソールに適用されます。

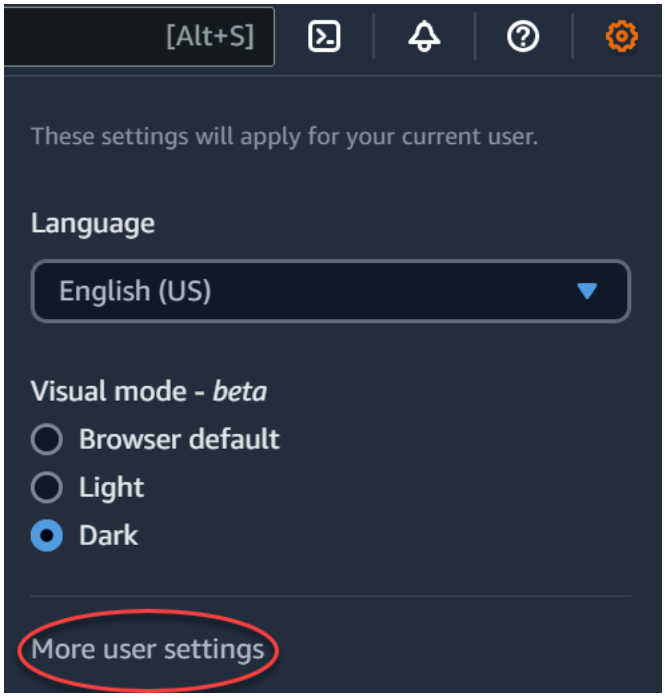
> **Important**
>
> 設定、お気に入りサービス、最近アクセスしたサービスがグローバルに保持されるように、このデータは AWS リージョンデフォルトで無効になっているリージョンを含むすべての に保存されます。対象となるリージョンは、アフリカ (ケープタウン)、アジアパシフィック (香港)、アジアパシフィック (ハイデラバード)、アジアパシフィック (ジャカルタ)、欧州 (ミラノ)、欧州 (スペイン)、欧州 (チューリッヒ)、中東 (バーレーン)、および中東 (UAE) です。アクセスするには、引き続きリージョンを手動で有効にし、そのリージョンでリソースを作成して管理する必要があります。このデータをすべての に保存しない場合は AWS リージョン、すべてリセットを選択して設定をクリアし、最近アクセスした サービスを 設定管理で記憶しないようにします。

統合設定へのアクセス

次の手順では、統合設定にアクセスする方法について説明します。

統合設定にアクセスするには

1. AWS Management Consoleにサインインします。
2. ナビゲーションバーで、歯車アイコンを選択します。
3. [統一された設定] ページを開くには、[その他のユーザー設定] を選択します。



統合設定のリセット

統合設定をリセットすることで、すべての統合設定設定を削除し、デフォルト設定を復元できます。

> **Note**
>
> これは AWS、ナビゲーションやサービスメニューでのお気に入りのサービス、コンソールホームウィジェットや で最近訪問したサービス AWS Console Mobile Application、デフォルト言語、デフォルトリージョン、ビジュアルモードなど、サービス全体に適用されるすべての設定など、 の複数の領域に影響します。

すべての統合設定をリセットするには

1. AWS Management Consoleにサインインします。
2. ナビゲーションバーで、歯車アイコンを選択します。
3. 「その他のユーザー設定」を選択して、統合設定ページを開きます。
4. すべての をリセットを選択します。

統合設定の編集

次の手順では、優先設定を編集する方法について説明します。

統合設定を編集するには

1. AWS Management Consoleにサインインします。
2. ナビゲーションバーで、歯車アイコンを選択します。
3. 「その他のユーザー設定」を選択して、統合設定ページを開きます。
4. 目的の設定の横にある [編集] を選択します。
   - ローカリゼーションとデフォルトのリージョン:
     - [言語] では、コンソールテキストのデフォルト言語を選択できます。
     - [デフォルトのリージョン] では、ログインするたびに適用されるデフォルトのリージョンを選択できます。アカウントで使用可能なリージョンはどれでも選択できます。デフォルトとして最後に使用したリージョンを選択することもできます。

       AWS Management Consoleでのリージョンルーティングの詳細については、「リージョンの選択」を参照してください。
   - [表示:]
     - [Visual mode] (ビジュアルモード) では、コンソールをライトモード、ダークモード、またはブラウザのデフォルトの表示モードに設定できます。

       ダークモードはベータ機能であり、 AWS のすべてのサービスコンソールに適用されるわけではありません。
     - [お気に入りのバーの表示] では、[お気に入り] バーの表示を切り替えて、完全なサービス名とアイコンを表示するか、サービスのアイコンのみを表示します。
     - [お気に入りバーのアイコンサイズ] は、[お気に入り] バーに表示されるサービスアイコンのサイズを、小 (16 x 16 ピクセル) と大 (24 x 24 ピクセル) の間で切り替えます。
   - 設定管理:
     - 最近訪問したサービスでは、 が最近訪問したサービスを AWS Management Console 記憶しているかどうかを選択できます。これをオフにすると、最近アクセスしたサービス履歴も削除されるため、最近アクセスしたサービスはサービスメニュー AWS Console Mobile Applicationやコンソールホームウィジェットに表示されなくなります。
5. [変更を保存] を選択します。

のビジュアルモードを変更する AWS Management Console

ビジュアルモードでは、コンソールがブラウザのライトモード、ダークモード、またはデフォルトの表示モードに設定されます。

ナビゲーションバーからビジュアルモードを変更するには

1. AWS Management Consoleにサインインします。
2. ナビゲーションバーで、歯車アイコンを選択します。
3. [ビジュアルモード] で、ライトモードの場合は [ライト]、ダークモードの場合は [ダーク]、ブラウザのデフォルト表示モードの場合は [ブラウザのデフォルト] を選択します。

統合設定でのデフォルト言語の変更

次の手順では、ナビゲーションバーを使用してデフォルト言語を変更する方法について説明します。

ナビゲーションバーからデフォルトの言語を変更するには

1. AWS Management Consoleにサインインします。
2. ナビゲーションバーで、歯車アイコンを選択します。
3. [言語] で、[ブラウザのデフォルト] を選択するか、ドロップダウンリストから希望する言語を選択します。

リージョンを選択する

多くのサービスでは、リソースの管理場所 AWS リージョン を指定する を選択できます。リージョンは、同じ地理的エリアにある AWS リソースのセットです。AWS Management Console やなどの一部のサービスでは、リージョンを選択する必要はありません AWS Identity and Access Management。 AWS リージョンの詳細については、「AWS 全般のリファレンス」の「AWS リージョンの管理」を参照してください。

リージョンを選択するには

1. AWS Management Consoleにサインインします。
2. サービスを選択して、そのサービスのコンソールにアクセスします。
3. ナビゲーションバーで、現在表示されているリージョン名を選択します。切り替え先となるリージョンを選択します。

既定のリージョンを選択するには

1. ナビゲーションバーで設定アイコンを選択し、[その他のユーザー設定] を選択して [統一された設定] ページに移動します。
2. [ローカリゼーションとデフォルトのリージョン] の横にある [編集] を選択します。
3. デフォルトのリージョンを選択し、設定の保存 を選択します。デフォルトのリージョンを選択しない場合、最後にアクセスしたリージョンがデフォルトになります。
4. (オプション) 新しいデフォルトリージョンに移動を選択して、新しいデフォルトリージョンにすぐに移動します。

> **Note**
>
> AWS リソースを作成したが、それらのリソースがコンソールに表示されない場合、コンソールに別のリージョンのリソースが表示されている可能性があります。一部のリソース (Amazon EC2 インスタンスなど) は、そのリソースが作成されたリージョンに固有です。これらを表示するには、リージョンセレクターを使用してリソースを含むリージョンを選択します。

お気に入りの追加および削除

頻繁に使用するサービスにすばやくアクセスするには、サービスコンソールを [Favorites] (お気に入り) リストに保存できます。

[お気に入り] のリストにサービスを追加するには

1. AWS Management Consoleにサインインします。
2. ページの右上または右下にある [Add widgets] (ウィジェットの追加) ボタンを選択します。
3. [ウィジェットの追加] メニューで、コンソールに追加する [お気に入り] を選択してから [追加] を選択します。

   [お気に入り] は、コンソールホームの最後に追加されます。ウィジェットの上部にあるタイトルバーを選択して [お気に入り] をドラッグアンドドロップし、ウィジェットをページ上の新しい場所にドラッグします。
4. ナビゲーションバーで [サービス] を選択します。

5. [最近アクセスした] リストまたは [すべてのサービス] のリストで、お気に入りとして追加するサービス名にカーソルを合わせます。
6. サービス名の左側にある星印をクリックします。
7. 上記の 2 つのステップを繰り返して、その他のサービスも [お気に入り] のリストに追加できます。

[お気に入り] のリストからサービスを削除するには

1. ナビゲーションバーで [サービス] を選択します。
2. 次のいずれかを行います。
   - [お気に入り] のリストで、サービスの名前の上にマウスを移動します。次に、サービス名の右側に表示された [x] をクリックします。
   - [最近アクセスした] リストまたは [すべてのサービス] リストで、[お気に入り] のリストのサービス名の横の星印の選択を解除します。

パスワードを変更する

アカウント所有者は、 から AWS アカウントのパスワードを変更できますAWS Management Console。

パスワードを変更するには

1. AWS Management Console にサインインします。
2. ナビゲーションバーで、アカウント名をクリックします。
3. [Security credentials] (セキュリティ認証情報) を選択します。
4. 表示されるオプションは、 AWS アカウント タイプによって異なります。コンソールに表示されている手順に従って、パスワードを変更します。
5. 現在のパスワードを 1 回、そして新しいパスワードを 2 回入力します。

   新しいパスワードは 8 文字以上にする必要があります。また、次の文字を含める必要があります。

   - 少なくとも 1 つの記号
   - 少なくとも 1 つの数値
   - 少なくとも 1 つの大文字

- 少なくとも 1 つの小文字

6. [Change Password] (パスワードの変更) または [Save changes] (パスワードの保存) を選択します。

の言語の変更 AWS Management Console

AWS Console Home エクスペリエンスには、 の AWS サービスのデフォルト言語を変更できる統合設定ページが含まれています AWS Management Console。設定メニュー (ナビゲーションバーからアクセスできます) から、デフォルトの言語をすばやく変更することもできます。これは、 コンソールのどこからでも変更できます。

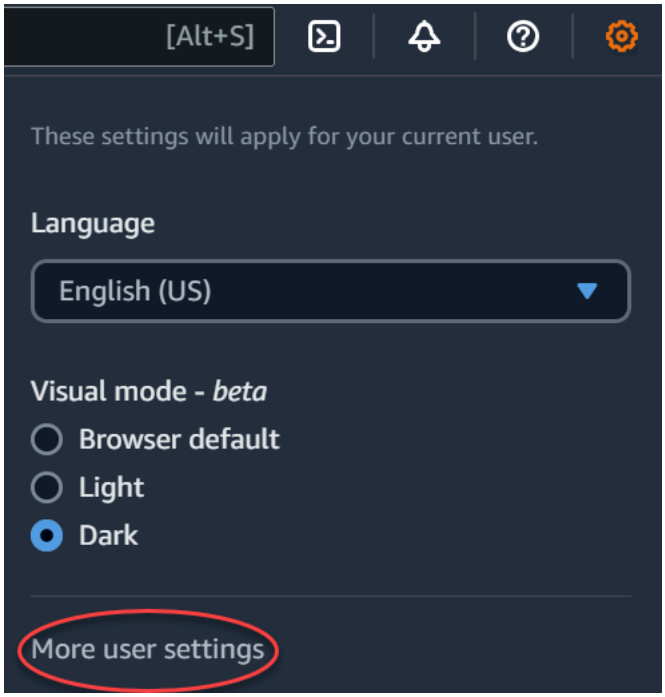
> **Note**
>
> この手順では、すべてのコンソールの言語が変更されますが、 AWS ドキュメントの言語は変更されません。ドキュメントに使用される言語を変更するには、すべてのドキュメントページの右上にある言語メニューを使用します。

AWS Management Console は現在、次の言語をサポートしています。

- 英語 (米国)
- 英語 (英国)
- バハサインドネシア語
- ドイツ語
- フランス語
- 日本語
- スペイン語
- イタリア語
- ポルトガル語
- 韓国語
- 簡体字中国語
- 繁体字中国語

[統一された設定] でデフォルトの言語を変更するには

1. AWS Management Consoleにサインインします。
2. ナビゲーションバーで、設定アイコンを選択します。
3. [統一された設定] ページを開くには、[その他のユーザー設定] を選択します。



4. [統一された設定] で [ローカリゼーションとデフォルトのリージョン] の横にある [編集] を選択します。
5. コンソールで使用する言語を選択し、以下のオプションの 1 つを選択します。

   - ドロップダウンリストから [ブラウザのデフォルト] を選択し、[設定を保存] を選択します。

     すべてのサービスのコンソールテキストは、ブラウザ設定で設定した任意の言語 AWS で表示されます。

     > **Note**
     >
     > ブラウザのデフォルトは、 AWS Management Consoleでサポートされている言語のみをサポートしています。

   - ドロップダウンリストから希望する言語を選択し、[設定を保存] を選択します。

     すべてのサービスのコンソールテキスト AWS が任意の言語で表示されます。

ナビゲーションバーからデフォルトの言語を変更するには

1. AWS Management Consoleにサインインします。
2. ナビゲーションバーで、設定アイコンを選択します。
3. [言語] で、[ブラウザのデフォルト] を選択するか、ドロップダウンリストから希望する言語を選択します。

サービスの使用を開始する

AWS Management Console には、個々のサービスコンソールに移動する複数の方法があります。

サービスのコンソールを開くには

次のいずれかを実行します。

- ナビゲーションバーで、サービスの名前の全部または一部を入力します。[サービス] で、検索結果のリストから必要なサービスを選択します。詳細については、「統合検索を使用した製品、サービス、機能などの検索」を参照してください。
- [最近アクセスしたサービス] ウィジェットで、サービス名を選択します。
- [最近アクセスしたサービス] ウィジェットで、[すべての AWS のサービスを表示] を選択します。次に、[すべての AWS のサービス] ページで、サービス名を選択します。
- ナビゲーションバーで、サービスの詳細なリストを開くには、[サービス] を選択します。次に、[最近アクセスした] または [すべてのサービス] でサービスを選択します。

統合検索を使用した製品、サービス、機能などの検索

ナビゲーションバーの検索ボックスは、 AWS サービス、機能、サービスドキュメント、 AWS Marketplaceを見つけるための統一された検索ツールです。いくつかの文字を入力するだけで、これらすべてのカテゴリの結果が表示されます。入力する文字数が多いほど、検索結果は絞り込まれます。

サービス、機能、ドキュメント、または AWS Marketplace 製品を検索するには

1. のナビゲーションバーの検索ボックスに AWS Management Console、検索語の全部または一部を入力します。
2. 検索を絞り込み、詳細を表示するには、次のいずれかを実行します。
   - 目的のコンテンツの種類で検索結果を絞り込むには、左側のカテゴリの 1 つを選択します。
   - 特定のカテゴリの結果を表示するには、各カテゴリ見出しの横の [*n* 件すべての結果を表示する] を選択します。メインの結果リストに戻るには、左上隅の [戻る] をクリックします。
   - サービスの一般的な機能にすばやく移動するには、検索結果のサービス名の上で一時停止し、リンクをクリックします。
   - ドキュメントまたは AWS Marketplace 結果の詳細については、結果のタイトルを一時停止します。
3. リンクをクリックすると、目的のサービス、トピック、または AWS Marketplace ページに移動します。

> **Tip**
>
> キーボードを使用して、上位の検索結果にすばやく移動することもできます。まず、[Alt+s] (Windows) または[Option+s] (macOS) キーを押して検索バーにアクセスします。次に、検索する語句を入力します。意図した結果がリストの最上部に表示されたら、[Enter] キーを押します。例えば、Amazon EC2 コンソールにすばやく移動するには、「ec2」と入力し、[Enter] キーを押します。

Amazon Q デベロッパーとチャットする

Amazon Q Developer は、生成人工知能 (AI) を活用した会話型アシスタントで、 AWS アプリケーションの理解、構築、拡張、運用に役立ちます。 AWS アーキテクチャ、リソース AWS、 AWS ベストプラクティス、ドキュメントなど、 に関する質問は Amazon Q にお問い合わせいただけます。サポートケースを作成し、ライブエージェントからサポートを受けることもできます。詳細については、「Amazon Q デベロッパーユーザーガイド」の「Amazon Q とは」を参照してください。

Amazon Q の使用を開始する

、 AWS ドキュメントウェブサイト AWS Management Console、 AWS ウェブサイト、または AWS コンソールモバイルアプリケーションで Amazon Q とのチャットを開始するには、六角形の Amazon Q アイコンを選択します。詳細については、「Amazon Q デベロッパーユーザーガイド」の「Amazon Q デベロッパーの開始方法」を参照してください。

質問例

Amazon Q に質問できる質問の例を次に示します。

- `How do I get billing support?`
- `How do I create an EC2 instance?`
- `How do I troubleshoot a "Failed to load" error?`
- `How do I close an AWS account?`
- `Can you connect me with a person?`

での myApplications とは AWS

myApplications は、コンソールホームの拡張機能であり、 AWSでのアプリケーションのコスト、ヘルス、セキュリティ体制、パフォーマンスの管理とモニタリングに役立ちます。アカウント内のすべてのアプリケーション、すべてのアプリケーションの主要なメトリクス、コスト、セキュリティ、オペレーションのメトリクスとインサイトの概要には、 の 1 つのビューからアクセスできます AWS Management Console。myApplications には、以下が含まれます。

- コンソールホームページのアプリケーションウィジェット
- アプリケーションリソースのコストとセキュリティ検出結果を表示するために使用できる myApplications
- コスト、パフォーマンス、セキュリティ検出結果などの主要なアプリケーションメトリクスを表示する myApplications ダッシュボード

myApplications の機能

- アプリケーションの作成 — 新しいアプリケーションを作成し、そのリソースを整理します。アプリケーションは myApplications に自動的に表示されるため、、API AWS Management Console、CLI、および SDKs でアクションを実行できます。 APIs アプリケーションの作成時に Infrastructure as code (IaC) が生成され、myApplication ダッシュボードからアクセスできます。IaC は、 AWS CloudFormation や Terraform などの IaC ツールで使用できます。
- アプリケーションへのアクセス — どのアプリケーションでも、myApplications ウィジェットから選択してすばやくアクセスできます。
- アプリケーションのメトリクスの比較 — myApplications を使用すると、複数のアプリケーションにわたってアプリケーションリソースのコストや重要なセキュリティ検出結果の数など、アプリケーションの主要なメトリクスを比較できます。
- アプリケーションのモニタリングと管理 — アラーム、Canary、 のサービスレベルの目標、 の検出結果 Amazon CloudWatch、 のコスト傾向を使用して AWS Security Hub、アプリケーションのヘルスとパフォーマンスを評価します AWS Cost Explorer Service。コンピューティングメトリクスの概要と最適化を確認したり、 からリソースのコンプライアンスと設定ステータスを管理したりすることもできます AWS Systems Manager。

関連サービス

myApplications は、以下のサービスを利用します。

- AppRegistry
- AppManager
- Amazon CloudWatch
- Amazon EC2
- AWS Lambda
- AWS Resource Explorer
- AWS Security Hub
- Systems Manager
- AWS Service Catalog
- タグ付け

myApplications へのアクセス

myApplications にアクセスするには、AWS Management Consoleの左側のサイドバーで [myApplications] を選択します。

料金

の myApplications AWS は追加料金なしで提供されます。セットアップ料金や前払いの義務はありません。myApplication ダッシュボードに集約されている基盤となるリソースやサービスの使用料金は、該当するリソースの公表価格で引き続き適用されます。

サポートされるリージョン

myApplications は、次の で使用できます AWS リージョン。

- 米国東部 (オハイオ)
- 米国東部 (バージニア北部)
- 米国西部 (北カリフォルニア)

- 米国西部 (オレゴン)
- アジアパシフィック (ムンバイ)
- アジアパシフィック (大阪)
- アジアパシフィック (ソウル)
- アジアパシフィック (シンガポール)
- アジアパシフィック (シドニー)
- アジアパシフィック (東京)
- カナダ (中部)
- 欧州 (フランクフルト)
- 欧州 (アイルランド)
- 欧州 (ロンドン)
- 欧州 (パリ)
- 欧州 (ストックホルム)
- 南米 (サンパウロ)

オプトインリージョン

デフォルトでは、オプトインリージョンは有効ではありません。これらのリージョンを myApplications で使用するには、手動で各リージョンを有効にする必要があります。の詳細については AWS リージョン、「 の管理 AWS リージョン」を参照してください。次のオプトインリージョンがサポートされています。

- アフリカ (ケープタウン)
- アジアパシフィック (香港)
- アジアパシフィック (ハイデラバード)
- アジアパシフィック (ジャカルタ)
- アジアパシフィック (メルボルン)
- 欧州 (ミラノ)
- 欧州 (スペイン)
- 欧州 (チューリッヒ)
- 中東 (バーレーン)

- 中東 (アラブ首長国連邦)
- イスラエル (テルアビブ)

myApplications の開始方法

myApplications を使用してアプリケーションの作成、モニタリング、管理を開始するには、次の手順に従います。

ステップ 1: アプリケーションの作成

myApplications の使用を開始するには、2023 年 11 月 8 日より前に作成された新しい AppRegistry アプリケーションを作成するか、既存のアプリケーションをオンボードします。

Create an application

アプリケーションを作成するには

1. AWS Management Consoleにサインインします。
2. 左側のサイドバーで [myApplications] を選択します。
3. [アプリケーションを作成] を選択します。
4. アプリケーション名を入力します。
5. (オプション) アプリケーションの説明を入力します。
6. (オプション) タグを追加します。タグはリソースに適用されるキーと値のペアで、リソースに関するメタデータを保持します。

   > **Note**
   >
   > AWS アプリケーションタグは、新しく作成されたアプリケーションに自動的に適用され、アプリケーションに関連付けられたリソースを識別するために使用できます。詳細については、「 AWS Service Catalog AppRegistry 管理者ガイド」の AWS 「アプリケーションタグ」を参照してください。

7. (オプション) 属性グループを追加します。属性グループを使用してアプリケーションのメタデータを保存できます。
8. [次へ] をクリックします。
9. (オプション) 既存のリソースを追加します。

> **Note**
>
> リソースを検索して追加するには、 AWS Resource Explorerをオンにする必要があります。詳細については、「 の開始方法 AWS Resource Explorer」を参照してください。
> 追加されたすべてのリソースには、 AWS アプリケーションタグが付けられます。

a. [リソースを選択] を選択します。

b. (オプション) ビューを選択します。

c. リソースを検索します。キーワード、名前、またはタイプで検索するか、リソースタイプを選択することができます。

> **Note**
>
> 探しているリソースが見つからない場合は、 でトラブルシューティングします AWS Resource Explorer。詳細については、[Resource Explorer ユーザーガイド」の「Resource Explorer での検索に関する問題のトラブルシューティング」を参照してください。

d. 追加するユーザーの横のチェックボックスをオンにします。

e. [追加] を選択します。

f. [Next (次へ)] を選択します。

10. 選択内容を確認します。

11. AWS CloudFormation スタックを関連付ける場合は、ページの下部にあるチェックボックスをオンにします。

> **Note**
>
> アプリケーションに AWS CloudFormation スタックを追加すると、アプリケーションに追加されたすべてのリソースにアプリケーション AWS タグが付けられるため、スタックの更新が必要です。スタックの最終更新後に実行した手動設定は、この更新後に反映されない場合があります。これにより、ダウンタイムなどのアプリケーションの問題が発生する可能性があります。詳細については、「AWS CloudFormation ユーザーガイド」の「スタックのリソースの更新動作」を参照してください。

12. [アプリケーションを作成] を選択します。

Onboard existing application

既存の AppRegistry アプリケーションをオンボードするには

1. AWS Management Consoleにサインインします。
2. 左側のサイドバーで [myApplications] を選択します。
3. 検索バーを使用してアプリケーションを見つけます。
4. アプリケーションを選択します。
5. [#########をオンボード」を選択します。
6. CloudFormation スタックを関連付ける場合は、アラートボックスのチェックボックスをオンにします。
7. [アプリケーションをオンボード] を選択します。

ステップ 2: アプリケーションの表示

すべてのリージョンまたは特定のリージョンのアプリケーションおよび関連情報をカードビューまたはテーブルビューで表示できます。

アプリケーションを表示するには

1. 左側のサイドバーで [myApplications] を選択します。
2. [リージョン] で、[現在のリージョン] または [サポートされているリージョン] を選択します。
3. 特定のアプリケーションを検索するには、その名前、キーワード、または説明を検索バーに入力します。
4. (オプション) デフォルトのビューはカードビューです。アプリケーションページをカスタマイズするには、次の手順に従います。

   a. 歯車アイコンを選択します。
   b. (オプション) ページサイズを選択します。
   c. (オプション) カードビューまたはテーブルビューを選択します。
   d. (オプション) ページサイズを選択します。
   e. (オプション) テーブルビューを使用する場合は、テーブルビューのプロパティを選択します。

f. (オプション) 表示するアプリケーションのプロパティと表示順序を切り替えます。

g. [確認] を選択します。

アプリケーションの管理

このトピックでは、アプリケーションの管理方法について説明します。

アプリケーションの編集

アプリケーションを編集 AppRegistry して、説明を更新できます。を使用して AppRegistry 、アプリケーションのタグと属性グループを編集することもできます。

アプリケーションを編集するには

1. AWS Management Consoleを開きます。
2. コンソールの左側のサイドバーで、[myApplications] を選択します。
3. 編集するアプリケーションを選択します。
4. myApplication ダッシュボードで、[アクション]、[アプリケーションを編集] の順に選択します。
5. [アプリケーションの説明を編集] で説明を更新し、[変更を保存] を選択します。

タグを編集するには

- 「 AWS Service Catalog AppRegistry 管理者ガイド」の「タグの管理」のステップに従います。

属性グループを編集するには

- 「 AWS Service Catalog AppRegistry 管理者ガイド」の「属性グループの編集」のステップに従います。

アプリケーションの削除

アプリケーションが不要になった場合は、削除できます。

アプリケーションを削除するには

1. AWS Management Consoleを開きます。

2. コンソールの左側のサイドバーで、[myApplications] を選択します。
3. 削除するアプリケーションを選択します。
4. myApplication ダッシュボードで、[アクション] を選択します。
5. [アプリケーションを削除] を選択します。
6. [削除] を選択します。
7. 削除することを確認し、[アプリケーションを削除] を選択します。

コードスニペットの作成

myApplications は、すべてのアプリケーションのコードスニペットを作成します。コードスニペットを使用すると、Infrastructure as Code (IaC) ツールを使用して、新しく作成したリソースをアプリケーションに自動的に追加できます。追加されたすべてのリソースには、 AWS アプリケーションに関連付けるアプリケーションタグが付けられます。

アプリケーションのコードスニペットを作成するには

1. AWS Management Consoleを開きます。
2. コンソールの左側のサイドバーで、[myApplications] を選択します。
3. アプリケーションを検索して選択します。
4. [アクション] を選択します。
5. [コードスニペットを取得] を選択します。
6. コードスニペットタイプを選択します。
7. [コピー] を選択して、コードをクリップボードにコピーします。
8. コードを IaC ツールに貼り付けます。

リソースの管理

このトピックでは、リソースを管理する方法について説明します。

リソースの追加

アプリケーションにリソースを追加すると、リソースをグループ化して、セキュリティ、パフォーマンス、コンプライアンスを管理できます。

リソースを追加するには

1. AWS Management Consoleを開きます。
2. コンソールの左側のサイドバーで、[myApplications] を選択します。
3. アプリケーションを検索して選択します。
4. [リソースを管理] を選択します。
5. [Add resources] (リソースを追加) を選択します。
6. (オプション) ビューを選択します。
7. リソースを検索します。キーワード、名前、またはタイプで検索するか、リソースタイプを選択することができます。

> **Note**
>
> 探しているリソースが見つからない場合は、 でトラブルシューティングします AWS Resource Explorer。詳細については、[Resource Explorer ユーザーガイド」の「Resource Explorer での検索に関する問題のトラブルシューティング」を参照してください。

8. 追加するユーザーの横のチェックボックスをオンにします。
9. [追加] を選択します。

リソースの削除

リソースを削除して、アプリケーションとの関連付けを解除できます。

リソースを削除するには

1. AWS Management Consoleを開きます。
2. コンソールの左側のサイドバーで、[myApplications] を選択します。
3. アプリケーションを検索して選択します。
4. [リソースを管理] を選択します。
5. (オプション) ビューを選択します。
6. リソースを検索します。キーワード、名前、またはタイプで検索するか、リソースタイプを選択することができます。

> **Note**
>
> 探しているリソースが見つからない場合は、 でトラブルシューティングします AWS Resource Explorer。詳細については、[Resource Explorer ユーザーガイド」の「Resource Explorer での検索に関する問題のトラブルシューティング」を参照してください。

7. [削除] を選択します。
8. [リソースを削除] を選択して、リソースを削除することを確認します。

myApplications ダッシュボード

作成またはオンボードするアプリケーションごとに、独自の myApplications ダッシュボードがあります。myApplications ダッシュボードには、複数の AWS サービスからのインサイトを示すコスト、セキュリティ、運用ウィジェットが含まれています。各ウィジェットのお気に入り登録、並べ替え、削除、またはサイズ変更も可能です。詳細については、「ウィジェットの操作」を参照してください。

アプリケーションダッシュボード設定ウィジェット

このウィジェットには、アプリケーションリソース AWS のサービス を管理するための の設定に役立つ、推奨される開始アクティビティのリストが含まれています。

アプリケーション概要ウィジェット

このウィジェットには、アプリケーションの名前、説明、AWS アプリケーションタグが表示されます。Infrastructure as Code (IAC) のアプリケーションタグにアクセスしてコピーし、リソースに手動でタグを付けることができます。

コンピューティングウィジェット

このウィジェットには、アプリケーションに追加するコンピューティングリソースの情報とメトリクスが表示されます。これには、アラームの合計数とコンピューティングリソースタイプの合計数が含まれます。このウィジェットには、Amazon EC2 インスタンスの CPU 使用率と Lambda 呼び出し Amazon CloudWatch に関する からのリソースパフォーマンスメトリクスの傾向グラフも表示されます。

コンピューティングウィジェットの設定

コンピューティングウィジェットにデータを入力するには、アプリケーションに少なくとも 1 つの Amazon EC2 インスタンスまたは Lambda 関数を設定します。詳細については、Amazon Elastic Compute Cloud ドキュメントと「AWS Lambda デベロッパーガイド」の「Lambda の開始方法」を参照してください。

コストと使用状況ウィジェット

このウィジェットには、アプリケーションリソースの AWS コストと使用状況のデータが表示されます。このデータを使用して、 AWS のサービスごとの毎月のコストを比較し、コストの内訳を表示できます。このウィジェットは、 AWS アプリケーションタグでタグ付けされたリソースのコストのみを要約します。ただし、税金、手数料、およびリソースに直接関連付けられていないその他の共有コストは除きます。コストは、非ブレンドとして表示され、24 時間ごとに最低 1 回更新されます。詳細については、「AWS Cost Management ユーザーガイド」の「AWS Resource Explorerを用いてコストを分析する」を参照してください。

コストと使用状況ウィジェットの設定

コストと使用状況ウィジェットを設定するには、アプリケーションとアカウント AWS Cost Explorer Service で を有効にします。このサービスは追加料金なしで提供され、セットアップ料金や前払いの義務もありません。詳細については、「AWS Cost Management ユーザーガイド」の「Cost Explorer を有効にする」を参照してください。

AWS セキュリティウィジェット

このウィジェットには、アプリケーションの AWS Security のセキュリティ検出結果が表示されます。 AWS Security は、 のアプリケーションのセキュリティ検出結果を包括的に表示します AWS。最近の優先度の高い検出結果に対する重大度別のアクセス、セキュリティ体制のモニタリング、最近の重要度/重大度の高い検出結果へのアクセス、次のステップに向けたインサイトの取得を行うことができます。詳細については、「AWS Security Hub」を参照してください。

AWS セキュリティウィジェットの設定

AWS セキュリティウィジェットを設定するには、アプリケーションとアカウント AWS Security Hub 用に を設定します。詳細については、「 ユーザーガイド」の「 とは AWS Security HubAWS Security Hub 」を参照してください。料金情報については、「AWS Security Hub ユーザーガイド」の「AWS Security Hub の無料トライアル、使用状況、料金」を参照してください。。

AWS Security Hub では、 AWS Config Recording を設定する必要があります。このサービスは、AWS アカウントに関連付けられたリソースの詳細ビューを提供します。詳細については、AWS Systems Manager ユーザーガイドの AWS Systems Managerを参照してください。

DevOps ウィジェット

このウィジェットには運用上のインサイトが表示されるため、コンプライアンスを評価して、アプリケーションに対してアクションを実行できます。これらのインサイトには以下が含まれます。

- フリートの管理
- 状態の管理
- パッチ管理
- 設定と OpsItems 管理

DevOps ウィジェットの設定

DevOps ウィジェットを設定するには、アプリケーションとアカウント AWS Systems Manager OpsCenter で を有効にします。詳細については、「 ユーザーガイド」の「Systems Manager Explorer の開始方法」および OpsCenterAWS Systems Manager 「」を参照してください。OpsCenter を有効にすると AWS Systems Manager Explorer 、 は AWS Config と を設定 Amazon CloudWatch して、一般的に使用されるルールとイベント OpsItems に基づいてイベントが自動的に作成されるようにします。詳細については、「 ユーザーガイド」の「セットアップ OpsCenterAWS Systems Manager 」を参照してください。

Systems Manager エージェントを実行するようにインスタンスを設定し、パッチスキャンを有効にするアクセス許可を適用できます。詳細については、「AWS Systems Manager ユーザーガイド」の「AWS Systems Manager Quick Setup」を参照してください。

Patch Manager を設定することで、アプリケーションの Amazon EC2 AWS Systems Manager インスタンスの自動パッチ適用を設定することもできます。詳細については、「AWS Systems Manager ユーザーガイド」の「Quick Setup パッチポリシーの使用」を参照してください。

料金情報については、「AWS Systems Manager の料金」を参照してください。

モニタリングと運用ウィジェット

このウィジェットには以下が表示されます。

- アプリケーションに関連するリソースのアラームとアラート

- アプリケーションのサービスレベル目標 (SLO) とメトリクス
- 使用可能な AWS Application Signals メトリクス

モニタリングと運用ウィジェットの設定

モニタリングとオペレーションウィジェットを設定するには、 AWS アカウントに CloudWatch アラームと Canary を作成します。詳細については、「Amazon ユーザーガイド」の「Amazon CloudWatch アラームの使用」および「Canary の作成」を参照してください。 CloudWatch CloudWatch アラームと合成 Canary の料金については、「Amazon の CloudWatch 料金」とAWS「クラウドオペレーションと移行ブログ」をそれぞれ参照してください。

CloudWatch Application Signals の詳細については、「Amazon ユーザーガイド」の「Amazon CloudWatch アプリケーションインサイトを有効にする」を参照してください。 CloudWatch

タグウィジェット

このウィジェットには、アプリケーションに関連するすべてのタグが表示されます。このウィジェットを使用して、アプリケーションのメタデータ (重要度、環境、コストセンター) を追跡および管理できます。詳細については、「 リソースのタグ付けのベストプラクティス」ホワイトペーパーの「タグとは」を参照してください。 AWS AWS

AWS Management Console プライベートアクセス

AWS Management Console プライベートアクセスは、 へのアクセスを制御するための高度なセキュリティ機能です AWS Management Console。 AWS Management Console プライベートアクセスは、ユーザーがネットワーク内 AWS アカウント から予期しない にサインインできないようにする場合に役立ちます。この機能を使用すると、トラフィックがネットワーク内から発信された AWS アカウント ときに、 へのアクセスを指定された既知のセット AWS Management Console のみに制限できます。

トピック

サポートされている AWS リージョン、サービスコンソール、および機能

AWS Management Console プライベートアクセスは、リージョンと AWS サービスのサブセットのみをサポートします。サポートされていないサービスコンソールは、 AWS Management Consoleで非アクティブになります。さらに、統合設定のデフォルトリージョンの選択など、 AWS Management Console プライベートアクセスの使用時に特定の AWS Management Console 機能が無効になる場合があります。

以下のリージョンとサービスコンソールがサポートされています。

サポートされるリージョン

- 米国東部 (オハイオ)
- 米国東部 (バージニア北部)
- 米国西部 (北カリフォルニア)

- 米国西部 (オレゴン)
- アジアパシフィック (ハイデラバード)
- アジアパシフィック (ムンバイ)
- アジアパシフィック (ソウル)
- アジアパシフィック (大阪)
- アジアパシフィック (シンガポール)
- アジアパシフィック (シドニー)
- アジアパシフィック (東京)
- カナダ (中部)
- 欧州 (フランクフルト)
- 欧州 (アイルランド)
- 欧州 (ロンドン)
- 欧州 (パリ)
- 欧州 (ストックホルム)
- 南米 (サンパウロ)
- アフリカ (ケープタウン)
- アジアパシフィック (香港)
- アジアパシフィック (ジャカルタ)
- アジアパシフィック (メルボルン)
- カナダ西部 (カルガリー)
- 欧州 (ミラノ)
- 欧州 (スペイン)
- 欧州 (チューリッヒ)
- 中東 (バーレーン)
- 中東 (アラブ首長国連邦)
- イスラエル (テルアビブ)

サポートされているサービスコンソール

- Amazon API Gateway

- AWS App Mesh
- AWS Application Migration Service
- Amazon Athena
- AWS Auto Scaling
- AWS Billing Conductor
- AWS Certificate Manager
- AWS Cloud Map
- Amazon CloudFront
- Amazon CloudWatch
- AWS CodeArtifact
- AWS CodeBuild
- Amazon CodeGuru
- Amazon Comprehend
- Amazon Comprehend Medical
- AWS Compute Optimizer
- AWS Console Home
- AWS Database Migration Service
- AWS DeepRacer
- Amazon DocumentDB
- Amazon DynamoDB
- Amazon EC2
- Amazon EC2 Global View
- EC2 Image Builder
- Amazon EC2 Instance Connect
- Amazon Elastic Container Registry
- Amazon Elastic Container Service
- AWS Elastic Disaster Recovery
- Amazon Elastic File System
- Amazon Elastic Kubernetes Service
- Amazon ElastiCache

- Amazon EMR
- Amazon EventBridge
- Amazon GameLift
- AWS Global Accelerator
- AWS Glue DataBrew
- AWS Ground Station
- Amazon GuardDuty
- AWS Identity and Access Management
- AWS Identity and Access Management Access Analyzer
- Amazon Inspector
- Amazon Kendra
- AWS Key Management Service
- Amazon Kinesis
- Amazon Managed Service for Apache Flink
- Amazon Data Firehose
- Amazon Kinesis Video Streams
- AWS Lambda
- Amazon Lex
- AWS License Manager
- Amazon Managed Grafana
- Amazon Managed Streaming for Apache Kafka
- Amazon Managed Workflows for Apache Airflow (MWAA)
- AWS Migration Hub Strategy Recommendations
- Amazon MQ
- Network Access Analyzer
- AWS Network Manager
- Amazon OpenSearch サービス
- AWS Organizations
- Amazon S3 on Outposts
- Amazon SageMaker ランタイム

- Amazon SageMaker Synthetic Data
- AWS Secrets Manager
- Service Quotas
- AWS Signer
- Amazon Simple Email Service
- Amazon Simple Queue Service
- Amazon Simple Storage Service (Amazon S3)
- AWS SQL Workbench
- AWS Step Functions
- AWS Support
- AWS Systems Manager
- AWS Transfer Family
- 統一された設定
- Amazon VPC IP Address Manager

AWS Management Console プライベートアクセスのセキュリティコントロールの概要

ネットワークからの AWS Management Console アカウント制限

AWS Management Console プライベートアクセスは、ネットワーク AWS Management Console からの へのアクセスを組織 AWS アカウント 内の指定された既知のセットのみに制限する場合に役立ちます。そうすることにより、ユーザーがネットワーク内から予期しない AWS アカウント にログインするのを防ぐことができます。これらのコントロールは、 AWS Management Console VPC エンドポイントポリシーを使用して実装できます。詳細については、「サービスコントロールポリシーと VPC エンドポイントポリシーの実装」を参照してください。

ネットワークからインターネットへの接続

静的コンテンツ (、CSS AWS Management Console、イメージ) など、 が使用するアセットにアクセスするためにJavaScript、ネットワークからのインターネット接続が引き続き必要です。また、 では有効 AWS のサービス になっていませんAWS PrivateLink。で使用される最上位ドメインのリストについては AWS Management Console、「」を参照してくださいトラブルシューティング。

> **Note**
>
> 現在、 AWS Management Console プライベートアクセスは `status.aws.amazon.com`、、 などのエンドポイントをサポートしていません`health.aws.amazon.comdocs.aws.amazon.com`。これらのドメインはパブリックインターネットにルーティングする必要があります。

必要な VPC エンドポイントと DNS 設定

AWS Management Console プライベートアクセスには、リージョンごとに次の 2 つの VPC エンドポイントが必要です。*#####*を、自身のリージョン情報に置き換えます。

1. の com.amazonaws.*region.*console AWS Management Console
2. の com.amazonaws.*region* .signin AWS サインイン

> **Note**
>
> インフラストラクチャとネットワーク接続は、 AWS Management Consoleで使用する他のリージョンに関係なく、常に米国東部 (バージニア北部) (us-east-1) リージョンにプロビジョニングします。 AWS Transit Gateway を使用して、米国東部 (バージニア北部) と他のすべてのリージョンとの接続を設定できます。詳細については、「Amazon VPC Transit Gateway ガイド」の「トランジットゲートウェイの開始方法」を参照してください。Amazon VPC ピアリング接続を使用することもできます。詳細については、「Amazon VPC ピアリング接続ガイド」の「VPC ピア機能とは」を参照してください。これらのオプションを比較するには、「Amazon Virtual Private Cloud 接続オプションホワイトペーパー」の「Amazon VPC 間の接続オプション」を参照してください。

DNSAWS Management Console および の設定 AWS サインイン

ネットワークトラフィックをそれぞれの VPC エンドポイントにルーティングするには、 AWS Management Consoleにユーザーがアクセスする元のネットワーク内の DNS レコードを設定します。これらの DNS レコードにより、ユーザーのブラウザトラフィックは、作成した VPC エンドポイントに誘導されます。

1 つのホストゾーンを作成できます。ただし、VPC エンドポイントがないため、`health.aws.amazon.com` や `docs.aws.amazon.com` などのエンドポイントにはアクセスできません。これらのドメインはパブリックインターネットにルーティングする必要があります。リージョンごとに 2 つのプライベートホストゾーンを作成することをお勧めします。1 つは `signin.aws.amazon.com` 用、別の 1 つは `console.aws.amazon.com` 用で、以下の CNAME レコードを使用します。

- リージョンの CNAME レコード (すべてのリージョン)
- サインインDNSゾーンの AWS サインイン VPC エンドポイントを指す region.signin.aws.amazon.com
- コンソールDNSゾーンの AWS Management Console VPC エンドポイントを指す region.console.aws.amazon.com
- 米国東部 (バージニア北部) リージョン専用のリージョンレス CNAME レコード。常に米国東部 (バージニア北部) リージョンを設定する必要があります。
  - 米国東部 (バージニア北部) (us-east-1) の AWS サインイン VPC エンドポイントを指す signin.aws.amazon.com
  - 米国東部 (バージニア北部) (us-east-1) の AWS Management Console VPC エンドポイントを指す console.aws.amazon.com

CNAME レコードを作成する手順については、「Amazon Route 53 デベロッパーガイド」の「レコードを使用する」を参照してください。

Amazon S3 を含む一部の AWS コンソールでは、DNS名前に異なるパターンが使用されます。以下に 2 つの例を示します。

- support.console.aws.amazon.com
- s3.console.aws.amazon.com

このトラフィックを AWS Management Console VPC エンドポイントに送信できるようにするには、これらの名前を個別に追加する必要があります。完全にプライベートなエクスペリエンスを実現するために、すべてのエンドポイントにルーティングを設定することをお勧めします。ただし、AWS Management Console これはプライベートアクセスを使用するためには必要ありません。

次の`jsonファイル`には、リージョンごとに設定する AWS のサービスとコンソールエンドポイントの完全なリストが含まれています。DNS の名前には、`com.amazonaws.`*`region`*`.console` エンドポイントの下の `PrivateIpv4DnsNames` フィールドを使用します。

- https://configuration.private-access.console.amazonaws.com/us-east-1.config.json
- https://configuration.private-access.console.amazonaws.com/us-east-2.config.json
- https://configuration.private-access.console.amazonaws.com/us-west-2.config.json
- https://configuration.private-access.console.amazonaws.com/ap-northeast-1.config.json
- https://configuration.private-access.console.amazonaws.com/ap-northeast-2.config.json
- https://configuration.private-access.console.amazonaws.com/ap-southeast-1.config.json
- https://configuration.private-access.console.amazonaws.com/ap-southeast-2.config.json
- https://configuration.private-access.console.amazonaws.com/ap-south-1.config.json
- https://configuration.private-access.console.amazonaws.com/ap-south-2.config.json
- https://configuration.private-access.console.amazonaws.com/ca-central-1.config.json
- https://configuration.private-access.console.amazonaws.com/eu-west-1.config.json
- https://configuration.private-access.console.amazonaws.com/eu-west-2.config.json
- https://configuration.private-access.console.amazonaws.com/il-central-1.config.json

> **Note**
>
> このリストは、 AWS Management Console プライベートアクセスの範囲にエンドポイントが追加されるたびに毎月更新されます。プライベートホストゾーンを最新の状態に保つには、前述のファイルリストを定期的に取得してください。

Route 53 を使用して DNS を設定する場合は、https://console.aws.amazon.com/route53/v2/hostedzones# にアクセスして DNS のセットアップを確認してください。Route 53 のプライベートホストゾーンごとに、次のレコードセットが存在することを確認します。

- console.aws.amazon.com
- signin.aws.amazon.com
- region.console.aws.amazon.com
- region.signin.aws.amazon.com
- support.console.aws.amazon.com
- global.console.aws.amazon.com
- 前述の JSON ファイルにある追加レコード

AWS サービスの VPC エンドポイントとDNS設定

は、直接ブラウザリクエストとウェブサーバーによってプロキシされるリクエストの組み合わせ AWS のサービス を介して AWS Management Console 呼び出します。このトラフィックを AWS Management Console VPC エンドポイントに転送するには、DNSVPC エンドポイントを追加し、依存 AWS サービスごとに を設定する必要があります。

次のjsonファイルには、 AWS PrivateLink サポートされている AWS のサービス が一覧表示されています。サービスと が統合されていない場合 AWS PrivateLink、サービスはこれらのファイルに含まれません。

- https://configuration.private-access.console.amazonaws.com/us-east-1.config.json
- https://configuration.private-access.console.amazonaws.com/us-east-2.config.json
- https://configuration.private-access.console.amazonaws.com/us-west-2.config.json
- https://configuration.private-access.console.amazonaws.com/ap-northeast-1.config.json
- https://configuration.private-access.console.amazonaws.com/ap-northeast-2.config.json
- https://configuration.private-access.console.amazonaws.com/ap-southeast-1.config.json
- https://configuration.private-access.console.amazonaws.com/ap-southeast-2.config.json
- https://configuration.private-access.console.amazonaws.com/ap-south-1.config.json
- https://configuration.private-access.console.amazonaws.com/ap-south-2.config.json
- https://configuration.private-access.console.amazonaws.com/ca-central-1.config.json
- https://configuration.private-access.console.amazonaws.com/eu-west-1.config.json
- https://configuration.private-access.console.amazonaws.com/eu-west-2.config.json
- https://configuration.private-access.console.amazonaws.com/il-central-1.config.json

対応するサービスの VPC エンドポイントの `[ServiceName]` フィールドを使用して VPC に追加します。

> **Note**
>
> このリストは毎月更新され、 AWS Management Console プライベートアクセスのサポートがサービスコンソールに追加されます。常に最新の状態に保つには、前述のファイルリストを定期的に取得し、VPC エンドポイントを更新してください。

サービスコントロールポリシーと VPC エンドポイントポリシーの実装

プライベートアクセスのサービスコントロールポリシー (SCPsと VPC エンドポイントポリシーを使用して、VPC AWS Management Console 内および接続されているオンプレミスネットワーク AWS Management Console から の使用が許可されているアカウントのセットを制限できます。

サービスコントロールポリシーでの AWS Management ConsoleAWS Organizations プライベートアクセスの使用

AWS 組織が特定のサービスを許可するサービスコントロールポリシー (SCP) を使用している場合は、許可されたアクション`signin:*`に を追加する必要があります。このアクセス許可は、プライベートアクセス VPC エンドポイント AWS Management Console 経由で にサインインすると、SCP が アクセス許可なしでブロックする IAM 認証が実行されるために必要です。例えば、次のサービスコントロールポリシーでは、プライベートアクセスエンドポイントを使用してアクセスされるときを含め、組織内で Amazon EC2 AWS Management Console および CloudWatch のサービスを使用することを許可します。

```
{
  "Effect": "Allow",
  "Action": [
    "signin:*",
    "ec2:*",
    "cloudwatch:*",
    ... Other services allowed
  },
  "Resource": "*"
}
```

SCP の詳細については、AWS Organizations ユーザーガイド の「サービスコントロールポリシー (SCP)」を参照してください。

予想されるアカウントと組織にのみ AWS Management Console 使用を許可する (信頼できる ID)

AWS Management Console と は、サインインアカウントの ID を具体的に制御する VPC エンドポイントポリシー AWS サインイン をサポートします。

他の VPC エンドポイントポリシーとは異なり、このポリシーは認証前に評価されます。その結果、認証されたセッションのサインインと使用のみを具体的に制御し、セッションが実行する AWS サービス固有のアクションは制御しません。例えば、セッションが Amazon EC2 コンソールなどの AWS サービスコンソールにアクセスする場合、これらの VPC エンドポイントポリシーは、そのページを表示するために実行される Amazon EC2 アクションに対して評価されません。代わりに、サインインした IAM プリンシパルに関連付けられた IAM ポリシーを使用して、 AWS サービスアクションに対するアクセス許可を制御できます。

> **Note**
>
> AWS Management Console および VPC エンドポイントの SignIn VPC エンドポイントポリシーは、ポリシー策定の限定されたサブセットのみをサポートします。各 `Principal` と `Resource` は `*` に設定する必要があります。また、`Action` は `*` または `signin:*` のいずれかにする必要があります。VPC エンドポイントへのアクセスを制御するには、`aws:PrincipalOrgId` および `aws:PrincipalAccount` 条件キーを使用します。

コンソールエンドポイントと SignIn VPC エンドポイントの両方には、次のポリシーが推奨されます。

この VPC エンドポイントポリシーは、指定された AWS 組織の AWS アカウント へのサインインを許可し、他のアカウントへのサインインをブロックします。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgId": "o-xxxxxxxxxxx"
        }
      }
    }
  ]
}
```

この VPC エンドポイントポリシーは、特定の のリストへのサインインを制限 AWS アカウント し、他のアカウントへのサインインをブロックします。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalAccount": [ "111122223333", "222233334444" ]
        }
      }
    }
  ]
}
```

AWS Management Console およびサインイン VPC エンドポイントで AWS アカウント または 組織を制限するポリシーは、サインイン時に評価され、既存のセッションに対して定期的に再評価されます。

アイデンティティベースのポリシーとその他のポリシータイプの実装

でアクセスを管理する AWS には、ポリシーを作成し、IAM ID (ユーザー、ユーザーのグループ、またはロール) または AWS リソースにアタッチします。このページでは、 ポリシーを AWS Management Console プライベートアクセスと一緒に使用した場合の仕組みについて説明します。

サポートされている AWS グローバル条件コンテキストキー

AWS Management Console プライベートアクセスは、 `aws:SourceVpce`および `aws:VpcSourceIp` AWS グローバル条件コンテキストキーをサポートしていません。 AWS Management Console のプライベートアクセスを使用する場合は、代わりに `aws:SourceVpc` IAM 条件をポリシーで使用できます。

AWS Management Console プライベートアクセスと aws の連携方法：SourceVpc

このセクションでは、 によって生成されたリクエストが に対して AWS Management Console 実行できるさまざまなネットワークパスについて説明します AWS のサービス。一般に、 AWS サービスコンソールは、ブラウザの直接リクエストと、 AWS Management Console ウェブサーバーから にプロキシされるリクエストを組み合わせて実装されます AWS のサービス。これらの実装は、予告なしに変更される可能性があります。セキュリティ要件に VPC エンドポイント AWS のサービス を使用した へのアクセスが含まれている場合は、VPC から直接使用するか、プライベートアクセスを介して使用するかにかかわらず、VPC エンドポイントを VPC AWS Management Console から使用するすべてのサービスに設定することをお勧めします。さらに、プライベートアクセス機能では、特定の`aws:SourceVpce`値ではなく、ポリシーで `aws:SourceVpc`IAM AWS Management Console 条件を使用する必要があります。このセクションでは、さまざまなネットワークパスの仕組みについて詳しく説明します。

ユーザーが にサインインすると AWS Management Console、ブラウザへの直接リクエストと、AWS Management Console ウェブサーバーから AWS サーバーにプロキシされるリクエスト AWS のサービス を組み合わせて、 にリクエストを行います。例えば、 CloudWatch グラフデータのリクエストはブラウザから直接行われます。一方、Amazon S3 などの一部の AWS サービスコンソールリクエストは、ウェブサーバーによって Amazon S3 にプロキシされます。

直接ブラウザリクエストの場合、 AWS Management Console プライベートアクセスを使用しても何も変更されません。以前と同様、リクエストは VPC が monitoring.region.amazonaws.com に到達するように設定したネットワークパスを通じてサービスに到達します。VPC が の VPC エンドポイントで設定されている場合com.amazonaws.region.monitoring、リクエストはその CloudWatch VPC エンドポイントを介して に到達 CloudWatchします。用の VPC エンドポイントがない場合 CloudWatch、リクエストは VPC 上のインターネットゲートウェイを介してパブリック CloudWatch エンドポイントに到達します。 CloudWatch VPC エンドポイント CloudWatch を介して に到着したリクエストには IAM 条件があり`aws:SourceVpc`、それぞれの値に`aws:SourceVpce`設定されます。パブリックエンドポイント CloudWatch 経由で に到達するユーザーは、 をリクエストの送信元 IP アドレス`aws:SourceIp`に設定します。これらの IAM 条件キーの詳細については、「IAM ユーザーガイド」の「 グローバル条件コンテキストキー」を参照してください。

Amazon S3 コンソールにアクセスしたときに Amazon S3 コンソールがバケットを一覧表示するリクエストなど、 AWS Management Console ウェブサーバーによってプロキシされるリクエストの場合Amazon S3、ネットワークパスは異なります。これらのリクエストは VPC から開始されないため、そのサービス用に VPC に設定した VPC エンドポイントを使用しません。この場合、Amazon S3 の VPC エンドポイントがあっても、バケットを一覧表示する Amazon S3 へのセッションのリ

クエストは Amazon S3 VPC エンドポイントを使用しません。ただし、サポートされているサービスで AWS Management Console プライベートアクセスを使用する場合、これらのリクエスト (Amazon S3 へのリクエストなど) には、リクエストコンテキストに `aws:SourceVpc`条件キーが含まれます。`aws:SourceVpc` 条件キーは、サインインとコンソールの AWS Management Console プライベートアクセスエンドポイントがデプロイされる VPC ID に設定されます。そのため、アイデンティティベースのポリシーで `aws:SourceVpc` 制限を使用している場合、 AWS Management Console プライベートアクセスサインインとコンソールエンドポイントをホストしているこの VPC の VPC ID を追加する必要があります。`aws:SourceVpce` 条件は、それぞれのサインインまたはコンソール VPC エンドポイント ID に設定されます。

> **Note**
>
> ユーザーが AWS Management Console のプライベートアクセスでサポートされていないサービスコンソールへのアクセスを必要とする場合は、ユーザーのアイデンティティベースのポリシーで `aws:SourceIP` 条件キーを使用し、必要なパブリックネットワークアドレス (オンプレミスのネットワーク範囲など) のリストを含める必要があります。

さまざまなネットワークパスがどのように反映されるか CloudTrail

によって生成されたリクエストで使用されるさまざまなネットワークパス AWS Management Console は、 CloudTrail イベント履歴に反映されます。

直接ブラウザリクエストの場合、 AWS Management Console プライベートアクセスを使用しても何も変更されません。 CloudTrail イベントには、サービス API コールの実行に使用された VPC エンドポイント ID など、接続に関する詳細が含まれます。

AWS Management Console ウェブサーバーによってプロキシされるリクエストの場合、 CloudTrail イベントには VPC 関連の詳細は含まれません。ただし、`AwsConsoleSignIn`イベントタイプなど、ブラウザセッションを確立 AWS サインイン するために必要な への初期リクエストには、イベントの詳細に AWS サインイン VPC エンドポイント ID が含まれます。

AWS Management Console プライベートアクセスを試す

このセクションでは、新しいアカウントで AWS Management Console プライベートアクセスをセットアップしてテストする方法について説明します。

AWS Management Console プライベートアクセスは高度なセキュリティ機能であり、ネットワークと VPCsの設定に関する事前の知識が必要です。このトピックでは、本格的なインフラストラクチャなしで AWS Management Console プライベートアクセスを試行する方法について説明します。

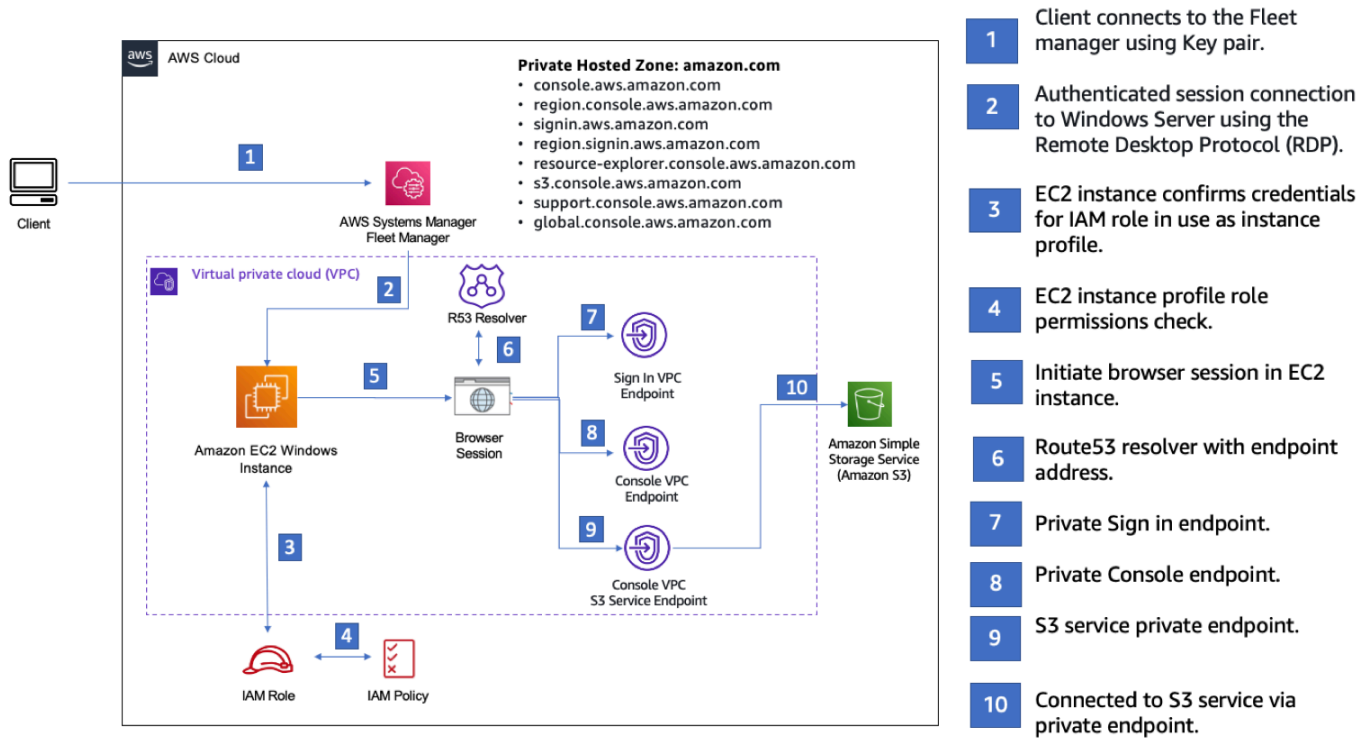
トピック

Amazon EC2 でのテスト設定

Amazon Elastic Compute Cloud (Amazon EC2) は、Amazon Web Service クラウドでスケーラブルなコンピューティングキャパシティーを提供します。Amazon EC2 を使用すると、必要な数 (またはそれ以下) の仮想サーバーの起動、セキュリティおよびネットワーキングの構成、ストレージの管理ができます。このセットアップでは、 AWS Systems Managerの一機能である Fleet Manager を使用して、リモートデスクトッププロトコル (RDP) を使って Amazon EC2 Windows インスタンスに接続できます。

このガイドでは、Amazon EC2 AWS Management Console インスタンスから Amazon Simple Storage Service へのプライベートアクセス接続をセットアップして体験するためのテスト環境を示します。このチュートリアルでは AWS CloudFormation 、 を使用して、この機能を視覚化するために Amazon EC2 が使用するネットワーク設定を作成および設定します。

次の図は、Amazon EC2 を使用して AWS Management Console のプライベートアクセス設定にアクセスするためのワークフローを示しています。これは、ユーザーがプライベートエンドポイントを使用して Amazon S3 に接続する方法を示しています。



次の AWS CloudFormation テンプレートをコピーし、「ネットワークをセットアップするには」の手順 3 で使用するファイルに保存します。

> **Note**
>
> この AWS CloudFormation テンプレートは、イスラエル (テルアビブ) リージョンで現在サポートされていない設定を使用します。

AWS Management Console プライベートアクセス環境 Amazon EC2 AWS CloudFormation テンプレート

```
Description: |
  AWS Management Console Private Access.
Parameters:
  VpcCIDR:
    Type: String
    Default: 172.16.0.0/16
    Description: CIDR range for VPC
```

```
  Ec2KeyPair:
    Type: AWS::EC2::KeyPair::KeyName
    Description: The EC2 KeyPair to use to connect to the Windows instance

  PublicSubnet1CIDR:
    Type: String
    Default: 172.16.1.0/24
    Description: CIDR range for Public Subnet A

  PublicSubnet2CIDR:
    Type: String
    Default: 172.16.0.0/24
    Description: CIDR range for Public Subnet B

  PublicSubnet3CIDR:
    Type: String
    Default: 172.16.2.0/24
    Description: CIDR range for Public Subnet C

  PrivateSubnet1CIDR:
    Type: String
    Default: 172.16.4.0/24
    Description: CIDR range for Private Subnet A

  PrivateSubnet2CIDR:
    Type: String
    Default: 172.16.5.0/24
    Description: CIDR range for Private Subnet B

  PrivateSubnet3CIDR:
    Type: String
    Default: 172.16.3.0/24
    Description: CIDR range for Private Subnet C

  LatestWindowsAmiId:
    Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
    Default: '/aws/service/ami-windows-latest/Windows_Server-2022-English-Full-Base'

  InstanceTypeParameter:
    Type: String
    Default: 't2.medium'


Resources:
```

```
#########################
# VPC AND SUBNETS
#########################

  AppVPC:
    Type: 'AWS::EC2::VPC'
    Properties:
      CidrBlock: !Ref VpcCIDR
      InstanceTenancy: default
      EnableDnsSupport: true
      EnableDnsHostnames: true

  PublicSubnetA:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PublicSubnet1CIDR
      MapPublicIpOnLaunch: true
      AvailabilityZone:
        Fn::Select:
          - 0
          - Fn::GetAZs: ""

  PublicSubnetB:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PublicSubnet2CIDR
      MapPublicIpOnLaunch: true
      AvailabilityZone:
        Fn::Select:
          - 1
          - Fn::GetAZs: ""

  PublicSubnetC:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PublicSubnet3CIDR
      MapPublicIpOnLaunch: true
      AvailabilityZone:
        Fn::Select:
          - 2
```

```
        - Fn::GetAZs: ""

  PrivateSubnetA:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PrivateSubnet1CIDR
      AvailabilityZone:
        Fn::Select:
          - 0
          - Fn::GetAZs: ""

  PrivateSubnetB:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PrivateSubnet2CIDR
      AvailabilityZone:
        Fn::Select:
          - 1
          - Fn::GetAZs: ""

  PrivateSubnetC:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PrivateSubnet3CIDR
      AvailabilityZone:
        Fn::Select:
          - 2
          - Fn::GetAZs: ""

  InternetGateway:
    Type: AWS::EC2::InternetGateway

  InternetGatewayAttachment:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      InternetGatewayId: !Ref InternetGateway
      VpcId: !Ref AppVPC

  NatGatewayEIP:
    Type: AWS::EC2::EIP
    DependsOn: InternetGatewayAttachment
```

```
  NatGateway:
    Type: AWS::EC2::NatGateway
    Properties:
      AllocationId: !GetAtt NatGatewayEIP.AllocationId
      SubnetId: !Ref PublicSubnetA

#########################
# Route Tables
#########################

  PrivateRouteTable:
    Type: 'AWS::EC2::RouteTable'
    Properties:
      VpcId: !Ref AppVPC

  DefaultPrivateRoute:
    Type: AWS::EC2::Route
    Properties:
      RouteTableId: !Ref PrivateRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      NatGatewayId: !Ref NatGateway

  PrivateSubnetRouteTableAssociation1:
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      RouteTableId: !Ref PrivateRouteTable
      SubnetId: !Ref PrivateSubnetA

  PrivateSubnetRouteTableAssociation2:
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      RouteTableId: !Ref PrivateRouteTable
      SubnetId: !Ref PrivateSubnetB

  PrivateSubnetRouteTableAssociation3:
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      RouteTableId: !Ref PrivateRouteTable
      SubnetId: !Ref PrivateSubnetC

  PublicRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
```

```
      VpcId: !Ref AppVPC

  DefaultPublicRoute:
    Type: AWS::EC2::Route
    DependsOn: InternetGatewayAttachment
    Properties:
      RouteTableId: !Ref PublicRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref InternetGateway

  PublicSubnetARouteTableAssociation1:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PublicRouteTable
      SubnetId: !Ref PublicSubnetA

  PublicSubnetBRouteTableAssociation2:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PublicRouteTable
      SubnetId: !Ref PublicSubnetB

  PublicSubnetBRouteTableAssociation3:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PublicRouteTable
      SubnetId: !Ref PublicSubnetC


#########################
# SECURITY GROUPS
#########################

  VPCEndpointSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: Allow TLS for VPC Endpoint
      VpcId: !Ref AppVPC
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 443
          ToPort: 443
          CidrIp: !GetAtt AppVPC.CidrBlock
```

```
  EC2SecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: Default EC2 Instance SG
      VpcId: !Ref AppVPC

#########################
# VPC ENDPOINTS
#########################

  VPCEndpointGatewayS3:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.s3'
      VpcEndpointType: Gateway
      VpcId: !Ref AppVPC
      RouteTableIds:
        - !Ref PrivateRouteTable

  VPCEndpointInterfaceSSM:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      VpcEndpointType: Interface
      PrivateDnsEnabled: false
      SubnetIds:
        - !Ref PrivateSubnetA
        - !Ref PrivateSubnetB
      SecurityGroupIds:
        - !Ref VPCEndpointSecurityGroup
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.ssm'
      VpcId: !Ref AppVPC

  VPCEndpointInterfaceEc2messages:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      VpcEndpointType: Interface
      PrivateDnsEnabled: false
      SubnetIds:
        - !Ref PrivateSubnetA
        - !Ref PrivateSubnetB
        - !Ref PrivateSubnetC
      SecurityGroupIds:
        - !Ref VPCEndpointSecurityGroup
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.ec2messages'
```

```
      VpcId: !Ref AppVPC

  VPCEndpointInterfaceSsmmessages:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      VpcEndpointType: Interface
      PrivateDnsEnabled: false
      SubnetIds:
        - !Ref PrivateSubnetA
        - !Ref PrivateSubnetB
        - !Ref PrivateSubnetC
      SecurityGroupIds:
        - !Ref VPCEndpointSecurityGroup
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.ssmmessages'
      VpcId: !Ref AppVPC

  VPCEndpointInterfaceSignin:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      VpcEndpointType: Interface
      PrivateDnsEnabled: false
      SubnetIds:
        - !Ref PrivateSubnetA
        - !Ref PrivateSubnetB
        - !Ref PrivateSubnetC
      SecurityGroupIds:
        - !Ref VPCEndpointSecurityGroup
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.signin'
      VpcId: !Ref AppVPC

  VPCEndpointInterfaceConsole:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      VpcEndpointType: Interface
      PrivateDnsEnabled: false
      SubnetIds:
        - !Ref PrivateSubnetA
        - !Ref PrivateSubnetB
        - !Ref PrivateSubnetC
      SecurityGroupIds:
        - !Ref VPCEndpointSecurityGroup
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.console'
      VpcId: !Ref AppVPC
```

```
#########################
# ROUTE53 RESOURCES
#########################

  ConsoleHostedZone:
    Type: "AWS::Route53::HostedZone"
    Properties:
      HostedZoneConfig:
        Comment: 'Console VPC Endpoint Hosted Zone'
      Name: 'console.aws.amazon.com'
      VPCs:
        -
          VPCId: !Ref AppVPC
          VPCRegion: !Ref "AWS::Region"

  ConsoleRecordGlobal:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: 'console.aws.amazon.com'
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  GlobalConsoleRecord:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: 'global.console.aws.amazon.com'
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  ConsoleS3ProxyRecordGlobal:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: 's3.console.aws.amazon.com'
```

```
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  ConsoleSupportProxyRecordGlobal:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: "support.console.aws.amazon.com"
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  ExplorerProxyRecordGlobal:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: "resource-explorer.console.aws.amazon.com"
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  ConsoleRecordRegional:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: !Sub "${AWS::Region}.console.aws.amazon.com"
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  SigninHostedZone:
```

```
    Type: "AWS::Route53::HostedZone"
    Properties:
      HostedZoneConfig:
        Comment: 'Signin VPC Endpoint Hosted Zone'
      Name: 'signin.aws.amazon.com'
      VPCs:
        -
          VPCId: !Ref AppVPC
          VPCRegion: !Ref "AWS::Region"

  SigninRecordGlobal:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'SigninHostedZone'
      Name: 'signin.aws.amazon.com'
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceSignin.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceSignin.DnsEntries]]]
      Type: A

  SigninRecordRegional:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'SigninHostedZone'
      Name: !Sub "${AWS::Region}.signin.aws.amazon.com"
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceSignin.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceSignin.DnsEntries]]]
      Type: A

#########################
# EC2 INSTANCE
#########################

  Ec2InstanceRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
```

```
          -
            Effect: Allow
            Principal:
              Service:
                - ec2.amazonaws.com
            Action:
              - sts:AssumeRole
      Path: /
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore

  Ec2InstanceProfile:
    Type: AWS::IAM::InstanceProfile
    Properties:
      Path: /
      Roles:
       - !Ref Ec2InstanceRole

  EC2WinInstance:
    Type: 'AWS::EC2::Instance'
    Properties:
      ImageId: !Ref LatestWindowsAmiId
      IamInstanceProfile: !Ref Ec2InstanceProfile
      KeyName: !Ref Ec2KeyPair
      InstanceType:
        Ref: InstanceTypeParameter
      SubnetId: !Ref PrivateSubnetA
      SecurityGroupIds:
        - Ref: EC2SecurityGroup
      BlockDeviceMappings:
        - DeviceName: /dev/sda1
          Ebs:
            VolumeSize: 50
      Tags:
      - Key: "Name"
        Value: "Console VPCE test instance"
```

ネットワークを設定するには

1. 組織の管理アカウントにサインインして、AWS CloudFormation コンソールを開きます。
2. [スタックの作成] を選択します。

3. [With new resources (standard)] (新しいリソースの使用 (標準)) を選択します。以前に作成した AWS CloudFormation テンプレートファイルをアップロードし、次へ を選択します。
4. **`PrivateConsoleNetworkForS3`** などスタックの名前を入力し、[次へ] を選択します。
5. VPC とサブネットの場合、希望する IP CIDR 範囲を入力するか、指定されたデフォルト値を使用してください。デフォルト値を使用する場合は、 内の既存の VPC リソースと重複していないことを確認します AWS アカウント。
6. Ec2KeyPair パラメータでは、アカウント内の既存の Amazon EC2 キーペアから 1 つを選択します。既存の Amazon EC2 キーペアがない場合は、次のステップに進む前に作成する必要があります。詳細については、Amazon EC2 ユーザーガイド」の「Amazon EC2 を使用してキーペアを作成するAmazon EC2」を参照してください。
7. [スタックの作成] を選択します。
8. スタックが作成されたら、[リソース] タブを選択して、作成されたリソースを表示します。

Amazon EC2 インスタンスに接続するには

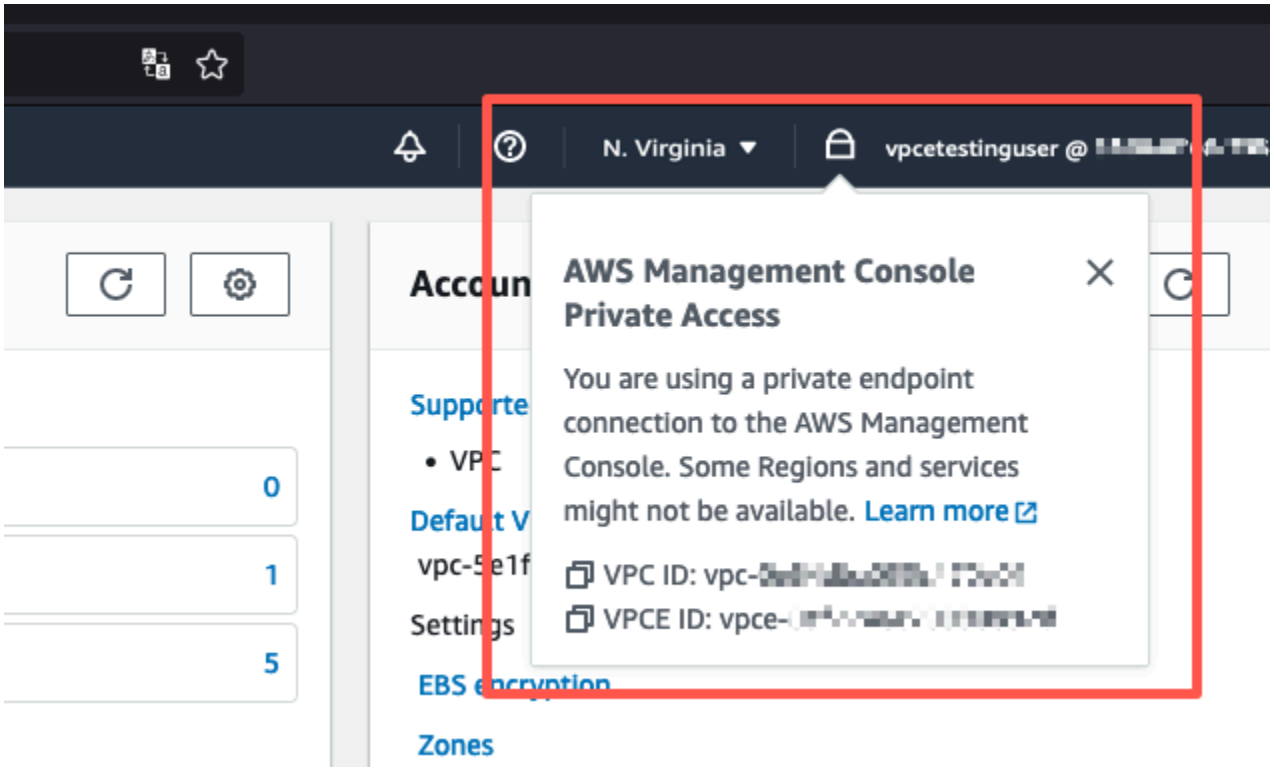
1. 組織の管理アカウントにサインインして、[Amazon EC2 コンソール] を開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスページで、 AWS CloudFormation テンプレートによって作成されたコンソール VPCE テストインスタンスを選択します。次に、[接続] を選択します。

> **Note**
>
> この例では、 の一機能である Fleet Manager AWS Systems Manager Explorerを使用して Windows Server に接続します。接続を開始するまでに数分かかることがあります。

4. [インスタンスに接続] ページで、[RDP クライアント]、[Fleet Manager を使用して接続] の順に選択します。
5. [Fleet Manager リモートデスクトップ] を選択します。
6. Amazon EC2 インスタンスの管理パスワードを取得し、ウェブインターフェイスを使用して Windows Desktop にアクセスするには、 AWS CloudFormation テンプレートの作成時に使用した Amazon EC2 キーペアに関連付けられたプライベートキーを使用します。
7. Amazon EC2 Windows インスタンスから、ブラウザ AWS Management Console で を開きます。
8. AWS 認証情報を使用してサインインしたら、Amazon S3 コンソールを開き、プライベートアクセスを使用して AWS Management Console 接続されていることを確認します。

AWS Management Console プライベートアクセスの設定をテストするには

1. 組織の管理アカウントにサインインして、[Amazon S3 コンソール] を開きます。
2. ナビゲーションバーのロックプライベートアイコンを選択すると、使用中の VPC エンドポイントが表示されます。次のスクリーンショットは、ロックプライベートアイコンの場所と VPC 情報を示しています。



Amazon でセットアップをテストする WorkSpaces

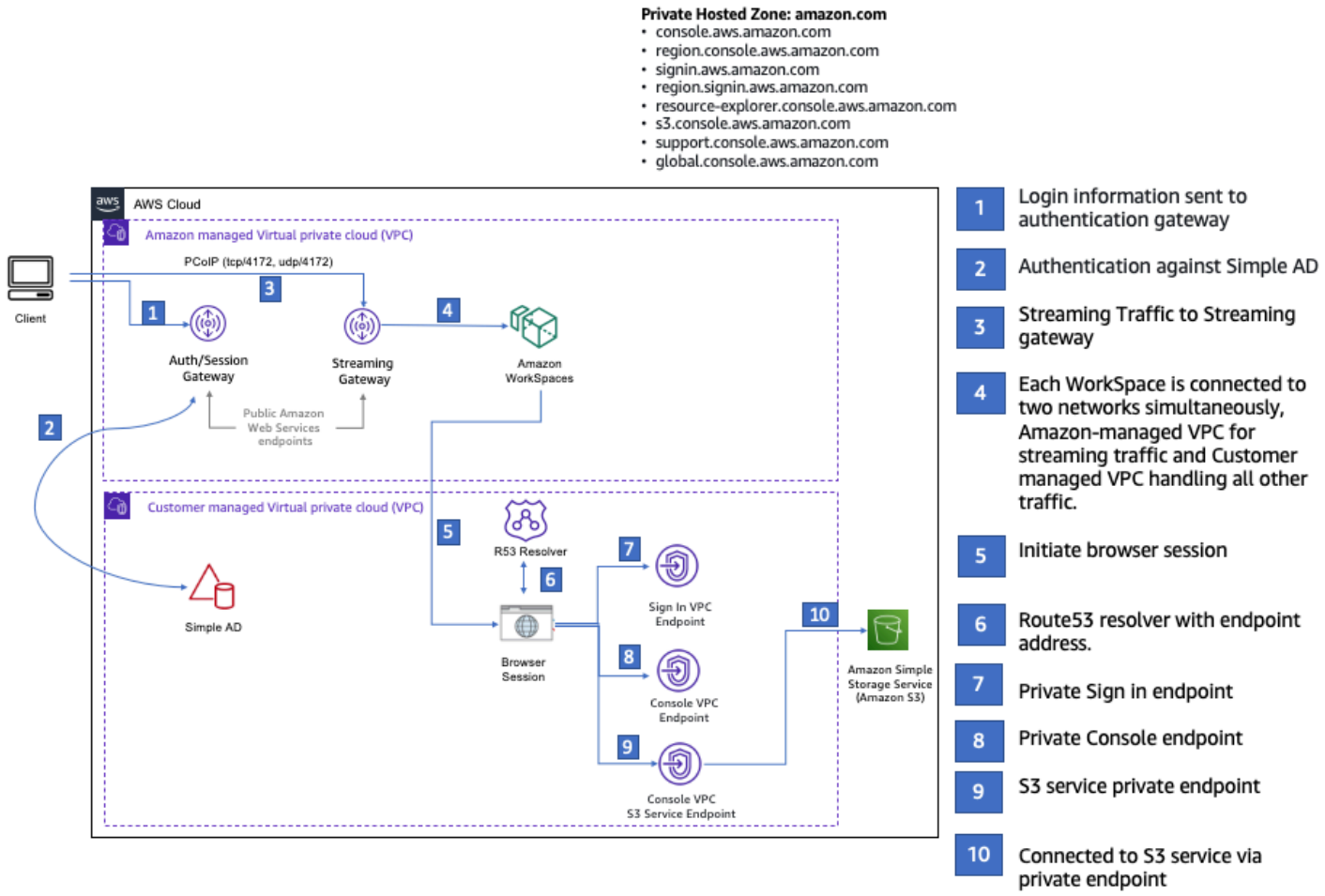
Amazon WorkSpaces では、 と呼ばれる、ユーザー向けの仮想クラウドベースの Windows、Amazon Linux、または Ubuntu Linux デスクトップをプロビジョニングできます WorkSpaces。必要に応じてユーザーをすばやく追加または削除できます。ユーザーは、複数のデバイスまたはウェブブラウザから仮想デスクトップにアクセスできます。の詳細については WorkSpaces、「Amazon WorkSpaces 管理ガイド」を参照してください。

このセクションの例では、ユーザー環境が で実行されているウェブブラウザを使用して AWS Management Console プライベートアクセスにサインイン WorkSpace するテスト環境について説明します。次に、ユーザーは Amazon Simple Storage Service コンソールにアクセスします。WorkSpace これは、VPC に接続されたネットワーク上のラップトップを使用して、ブラウザ AWS

Management Console から にアクセスする企業ユーザーのエクスペリエンスをシミュレートするためのものです。

このチュートリアルでは AWS CloudFormation 、 を使用してネットワーク設定と で使用する Simple Active Directory を作成し、設定します。 WorkSpaces また、 WorkSpace を使用して をセットアップする手順も順を追って説明します AWS Management Console。

次の図は、 を使用してプライベートアクセス設定を WorkSpace AWS Management Console テストするワークフローを示しています。クライアント 、Amazon マネージド VPC WorkSpace、カスタマーマネージド VPC の関係を示します。



次の AWS CloudFormation テンプレートをコピーし、ネットワークをセットアップする手順のステップ 3 で使用するファイルに保存します。

AWS Management Console プライベートアクセス環境 AWS CloudFormation テンプレート

```
Description: |
```

```
  AWS Management Console Private Access.
Parameters:

  VpcCIDR:
    Type: String
    Default: 172.16.0.0/16
    Description: CIDR range for VPC

  PublicSubnet1CIDR:
    Type: String
    Default: 172.16.1.0/24
    Description: CIDR range for Public Subnet A

  PublicSubnet2CIDR:
    Type: String
    Default: 172.16.0.0/24
    Description: CIDR range for Public Subnet B

  PrivateSubnet1CIDR:
    Type: String
    Default: 172.16.4.0/24
    Description: CIDR range for Private Subnet A

  PrivateSubnet2CIDR:
    Type: String
    Default: 172.16.5.0/24
    Description: CIDR range for Private Subnet B

# Amazon WorkSpaces is available in a subset of the Availability Zones for each
 supported Region.
# https://docs.aws.amazon.com/workspaces/latest/adminguide/azs-workspaces.html
Mappings:
  RegionMap:
    us-east-1:
      az1: use1-az2
      az2: use1-az4
      az3: use1-az6
    us-west-2:
      az1: usw2-az1
      az2: usw2-az2
      az3: usw2-az3
    ap-south-1:
      az1: aps1-az1
      az2: aps1-az2
```

```
      az3: aps1-az3
    ap-northeast-2:
      az1: apne2-az1
      az2: apne2-az3
    ap-southeast-1:
      az1: apse1-az1
      az2: apse1-az2
    ap-southeast-2:
      az1: apse2-az1
      az2: apse2-az3
    ap-northeast-1:
      az1: apne1-az1
      az2: apne1-az4
    ca-central-1:
      az1: cac1-az1
      az2: cac1-az2
    eu-central-1:
      az1: euc1-az2
      az2: euc1-az3
    eu-west-1:
      az1: euw1-az1
      az2: euw1-az2
    eu-west-2:
      az1: euw2-az2
      az2: euw2-az3
    sa-east-1:
      az1: sae1-az1
      az2: sae1-az3

Resources:

  iamLambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service:
              - lambda.amazonaws.com
            Action:
              - 'sts:AssumeRole'
      ManagedPolicyArns:
```

```
        - arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole
      Policies:
        - PolicyName: describe-ec2-az
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              - Effect: Allow
                Action:
                  - 'ec2:DescribeAvailabilityZones'
                Resource: '*'
      MaxSessionDuration: 3600
      Path: /service-role/

  fnZoneIdtoZoneName:
    Type: AWS::Lambda::Function
    Properties:
      Runtime: python3.8
      Handler: index.lambda_handler
      Code:
        ZipFile: |
          import boto3
          import cfnresponse

          def zoneId_to_zoneName(event, context):
              responseData = {}
              ec2 = boto3.client('ec2')
              describe_az = ec2.describe_availability_zones()
              for az in describe_az['AvailabilityZones']:
                  if event['ResourceProperties']['ZoneId'] == az['ZoneId']:
                      responseData['ZoneName'] = az['ZoneName']
                      cfnresponse.send(event, context, cfnresponse.SUCCESS,
responseData, str(az['ZoneId']))

          def no_op(event, context):
              print(event)
              responseData = {}
              cfnresponse.send(event, context, cfnresponse.SUCCESS, responseData,
str(event['RequestId']))

          def lambda_handler(event, context):
              if event['RequestType'] == ('Create' or 'Update'):
                  zoneId_to_zoneName(event, context)
              else:
                  no_op(event,context)
```

```
      Role: !GetAtt iamLambdaExecutionRole.Arn

  getAZ1:
    Type: "Custom::zone-id-zone-name"
    Properties:
      ServiceToken: !GetAtt fnZoneIdtoZoneName.Arn
      ZoneId: !FindInMap [ RegionMap, !Ref 'AWS::Region', az1 ]
  getAZ2:
    Type: "Custom::zone-id-zone-name"
    Properties:
      ServiceToken: !GetAtt fnZoneIdtoZoneName.Arn
      ZoneId: !FindInMap [ RegionMap, !Ref 'AWS::Region', az2 ]

#########################
# VPC AND SUBNETS
#########################

  AppVPC:
    Type: 'AWS::EC2::VPC'
    Properties:
      CidrBlock: !Ref VpcCIDR
      InstanceTenancy: default
      EnableDnsSupport: true
      EnableDnsHostnames: true

  PublicSubnetA:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PublicSubnet1CIDR
      MapPublicIpOnLaunch: true
      AvailabilityZone: !GetAtt getAZ1.ZoneName

  PublicSubnetB:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PublicSubnet2CIDR
      MapPublicIpOnLaunch: true
      AvailabilityZone: !GetAtt getAZ2.ZoneName

  PrivateSubnetA:
    Type: 'AWS::EC2::Subnet'
    Properties:
```

```
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PrivateSubnet1CIDR
      AvailabilityZone: !GetAtt getAZ1.ZoneName

  PrivateSubnetB:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref AppVPC
      CidrBlock: !Ref PrivateSubnet2CIDR
      AvailabilityZone: !GetAtt getAZ2.ZoneName

  InternetGateway:
    Type: AWS::EC2::InternetGateway

  InternetGatewayAttachment:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      InternetGatewayId: !Ref InternetGateway
      VpcId: !Ref AppVPC

  NatGatewayEIP:
    Type: AWS::EC2::EIP
    DependsOn: InternetGatewayAttachment

  NatGateway:
    Type: AWS::EC2::NatGateway
    Properties:
      AllocationId: !GetAtt NatGatewayEIP.AllocationId
      SubnetId: !Ref PublicSubnetA

#########################
# Route Tables
#########################

  PrivateRouteTable:
    Type: 'AWS::EC2::RouteTable'
    Properties:
      VpcId: !Ref AppVPC

  DefaultPrivateRoute:
    Type: AWS::EC2::Route
    Properties:
      RouteTableId: !Ref PrivateRouteTable
      DestinationCidrBlock: 0.0.0.0/0
```

```
      NatGatewayId: !Ref NatGateway

  PrivateSubnetRouteTableAssociation1:
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      RouteTableId: !Ref PrivateRouteTable
      SubnetId: !Ref PrivateSubnetA

  PrivateSubnetRouteTableAssociation2:
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      RouteTableId: !Ref PrivateRouteTable
      SubnetId: !Ref PrivateSubnetB

  PublicRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref AppVPC

  DefaultPublicRoute:
    Type: AWS::EC2::Route
    DependsOn: InternetGatewayAttachment
    Properties:
      RouteTableId: !Ref PublicRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref InternetGateway

  PublicSubnetARouteTableAssociation1:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PublicRouteTable
      SubnetId: !Ref PublicSubnetA

  PublicSubnetBRouteTableAssociation2:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref PublicRouteTable
      SubnetId: !Ref PublicSubnetB


##########################
# SECURITY GROUPS
##########################
```

```
  VPCEndpointSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: Allow TLS for VPC Endpoint
      VpcId: !Ref AppVPC
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 443
          ToPort: 443
          CidrIp: !GetAtt AppVPC.CidrBlock

#########################
# VPC ENDPOINTS
#########################

  VPCEndpointGatewayS3:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.s3'
      VpcEndpointType: Gateway
      VpcId: !Ref AppVPC
      RouteTableIds:
        - !Ref PrivateRouteTable

  VPCEndpointInterfaceSignin:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      VpcEndpointType: Interface
      PrivateDnsEnabled: false
      SubnetIds:
        - !Ref PrivateSubnetA
        - !Ref PrivateSubnetB
      SecurityGroupIds:
        - !Ref VPCEndpointSecurityGroup
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.signin'
      VpcId: !Ref AppVPC

  VPCEndpointInterfaceConsole:
    Type: 'AWS::EC2::VPCEndpoint'
    Properties:
      VpcEndpointType: Interface
      PrivateDnsEnabled: false
      SubnetIds:
        - !Ref PrivateSubnetA
```

```
        - !Ref PrivateSubnetB
      SecurityGroupIds:
        - !Ref VPCEndpointSecurityGroup
      ServiceName: !Sub 'com.amazonaws.${AWS::Region}.console'
      VpcId: !Ref AppVPC

#########################
# ROUTE53 RESOURCES
#########################

  ConsoleHostedZone:
    Type: "AWS::Route53::HostedZone"
    Properties:
      HostedZoneConfig:
        Comment: 'Console VPC Endpoint Hosted Zone'
      Name: 'console.aws.amazon.com'
      VPCs:
        -
          VPCId: !Ref AppVPC
          VPCRegion: !Ref "AWS::Region"

  ConsoleRecordGlobal:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: 'console.aws.amazon.com'
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  GlobalConsoleRecord:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: 'global.console.aws.amazon.com'
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A
```

```
  ConsoleS3ProxyRecordGlobal:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: 's3.console.aws.amazon.com'
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  ConsoleSupportProxyRecordGlobal:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: "support.console.aws.amazon.com"
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  ExplorerProxyRecordGlobal:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: "resource-explorer.console.aws.amazon.com"
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  ConsoleRecordRegional:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'ConsoleHostedZone'
      Name: !Sub "${AWS::Region}.console.aws.amazon.com"
      AliasTarget:
```

```
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceConsole.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceConsole.DnsEntries]]]
      Type: A

  SigninHostedZone:
    Type: "AWS::Route53::HostedZone"
    Properties:
      HostedZoneConfig:
        Comment: 'Signin VPC Endpoint Hosted Zone'
      Name: 'signin.aws.amazon.com'
      VPCs:
        -
          VPCId: !Ref AppVPC
          VPCRegion: !Ref "AWS::Region"

  SigninRecordGlobal:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'SigninHostedZone'
      Name: 'signin.aws.amazon.com'
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceSignin.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceSignin.DnsEntries]]]
      Type: A

  SigninRecordRegional:
    Type: AWS::Route53::RecordSet
    Properties:
      HostedZoneId: !Ref 'SigninHostedZone'
      Name: !Sub "${AWS::Region}.signin.aws.amazon.com"
      AliasTarget:
        DNSName: !Select ['1', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceSignin.DnsEntries]]]
        HostedZoneId: !Select ['0', !Split [':', !Select ['0', !GetAtt
 VPCEndpointInterfaceSignin.DnsEntries]]]
      Type: A

##########################
# WORKSPACE RESOURCES
##########################
```

```
  ADAdminSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      Name: "ADAdminSecret"
      Description: "Password for directory services admin"
      GenerateSecretString:
        SecretStringTemplate: '{"username": "Admin"}'
        GenerateStringKey: password
        PasswordLength: 30
        ExcludeCharacters: '"@/\'

  WorkspaceSimpleDirectory:
    Type: AWS::DirectoryService::SimpleAD
    DependsOn: AppVPC
    DependsOn: PrivateSubnetA
    DependsOn: PrivateSubnetB
    Properties:
      Name: "corp.awsconsole.com"
      Password: '{{resolve:secretsmanager:ADAdminSecret:SecretString:password}}'
      Size: "Small"
      VpcSettings:
        SubnetIds:
          - Ref: PrivateSubnetA
          - Ref: PrivateSubnetB

        VpcId:
          Ref: AppVPC


Outputs:
  PrivateSubnetA:
    Description: Private Subnet A
    Value: !Ref PrivateSubnetA

  PrivateSubnetB:
    Description: Private Subnet B
    Value: !Ref PrivateSubnetB

  WorkspaceSimpleDirectory:
    Description: Directory to be used for Workspaces
    Value: !Ref WorkspaceSimpleDirectory

  WorkspacesAdminPassword:
```

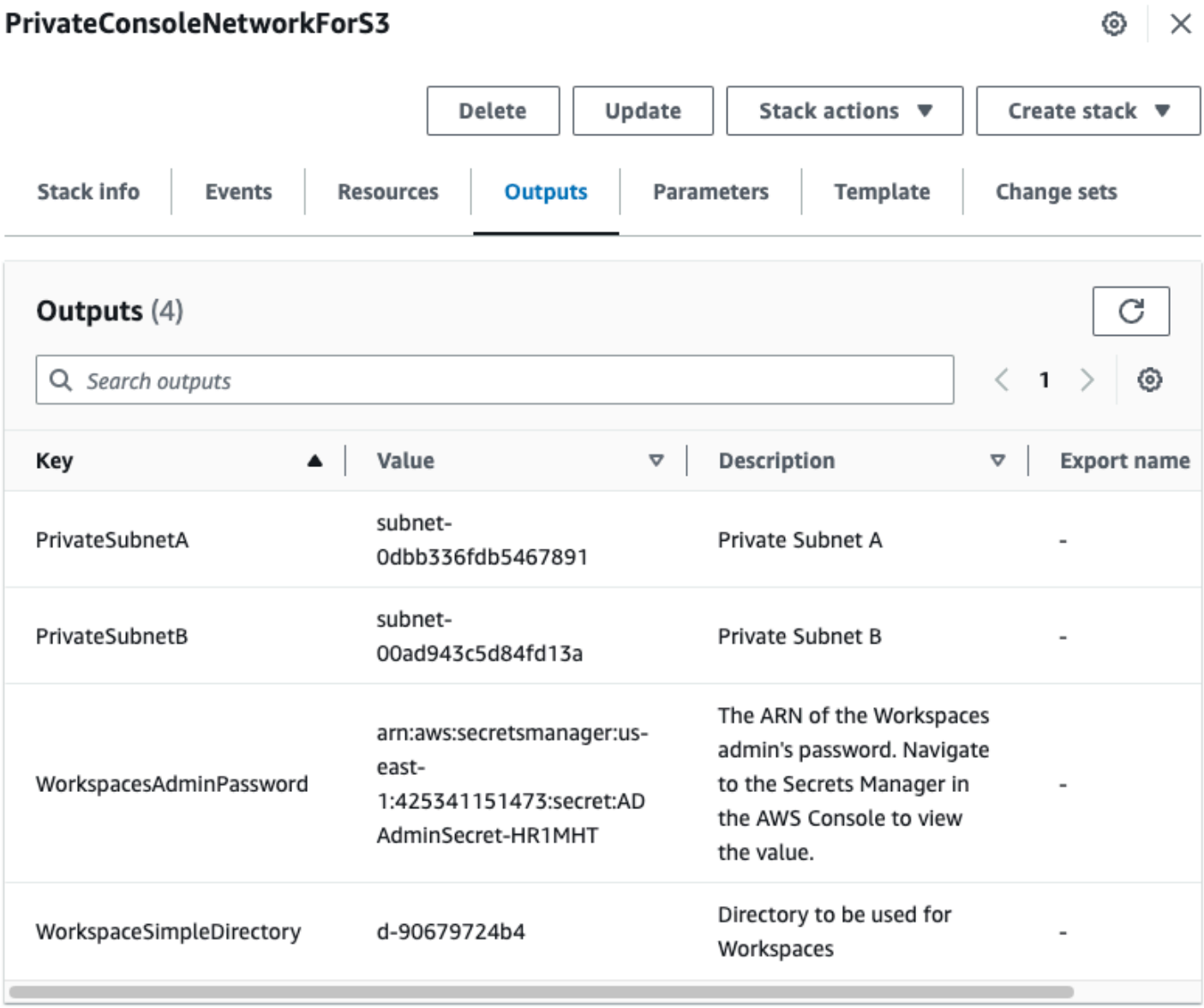
```
    Description : "The ARN of the Workspaces admin's password.  Navigate to the Secrets
Manager in the AWS Console to view the value."
    Value: !Ref ADAdminSecret
```

> **Note**
>
> このテスト設定は、米国東部 (バージニア北部) (us-east-1) リージョンで実行するように設計されています。

ネットワークを設定するには

1. 組織の管理アカウントにサインインして、AWS CloudFormation コンソールを開きます。
2. [スタックの作成] を選択します。
3. [With new resources (standard)] (新しいリソースの使用 (標準)) を選択します。以前に作成した AWS CloudFormation テンプレートファイルをアップロードし、次へ を選択します。
4. **PrivateConsoleNetworkForS3** などスタックの名前を入力し、[次へ] を選択します。
5. VPC とサブネットの場合、希望する IP CIDR 範囲を入力するか、指定されたデフォルト値を使用してください。デフォルト値を使用する場合は、 内の既存の VPC リソースと重複していないことを確認します AWS アカウント。
6. [スタックの作成] を選択します。
7. スタックが作成されたら、[リソース] タブを選択して、作成されたリソースを表示します。
8. [出力] タブを選択すると、プライベートサブネットと Workspace Simple Directory の値が表示されます。これらの値は、 を作成および設定するための次の手順のステップ 4 で使用するため、書き留めておきます WorkSpace。

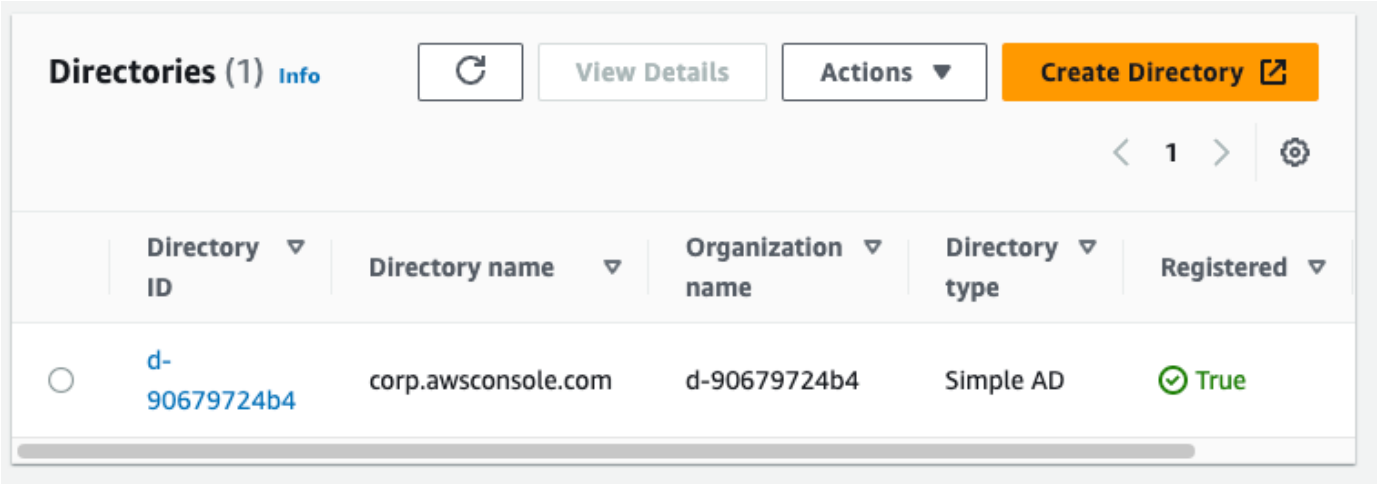
次のスクリーンショットは、プライベートサブネットと Workspace Simple Directory の値が表示された [出力] タブのビューを示しています。



ネットワークを作成したので、次の手順を使用して を作成してアクセスします WorkSpace。

を作成するには WorkSpace

1. WorkSpaces コンソールを開きます。
2. ナビゲーションペインで [ディレクトリ] を選択します。
3. [ディレクトリ] ページで、ディレクトリのステータスが [アクティブ] であることを確認します。次のスクリーンショットは、アクティブディレクトリを含む [ディレクトリ] ページを示しています。



4. でディレクトリを使用するには WorkSpaces、ディレクトリを登録する必要があります。ナビゲーションペインで、 を選択しWorkSpaces、 の作成 WorkSpacesを選択します。
5. [ディレクトリを選択] で、前の手順で AWS CloudFormation が作成したディレクトリを選択します。[アクション] メニューで、[登録] を選択します。
6. サブネット選択については、前の手順のステップ 9 で説明した 2 つのプライベートサブネットを選択します。
7. [セルフサービス許可を有効化] を選択し、[登録] を選択します。
8. ディレクトリが登録されたら、 の作成を続行します WorkSpace。登録したディレクトリを選択し、[次へ] を選択します。
9. [ユーザーの作成] ページで、[追加ユーザーの作成] を選択します。の名前と E メールを入力して、 を使用できるようにします WorkSpace。 WorkSpace ログイン情報がこの E メールアドレスに送信されるため、E メールアドレスが有効であることを確認します。
10. [次へ] をクリックします。
11. [ユーザーの識別] ページで、手順 9 で作成したユーザーを選択し、[次へ] を選択します。
12. [バンドルの選択] ページで、[Amazon Linux 2 のスタンダード]、[次へ] の順に選択します。
13. 実行モードとユーザーカスタマイズにデフォルト設定を使用し、次に [ワークスペースを作成] を選択します。 WorkSpace は `Pending`ステータスで開始し、`Available`約 20 分以内に に移行します。
14. WorkSpace が利用可能になると、ステップ 9 で指定した E メールアドレスでアクセスする手順が記載された E メールが届きます。

にサインインしたら WorkSpace、 AWS Management Console プライベートアクセスを使用してアクセスしていることをテストできます。

にアクセスするには WorkSpace

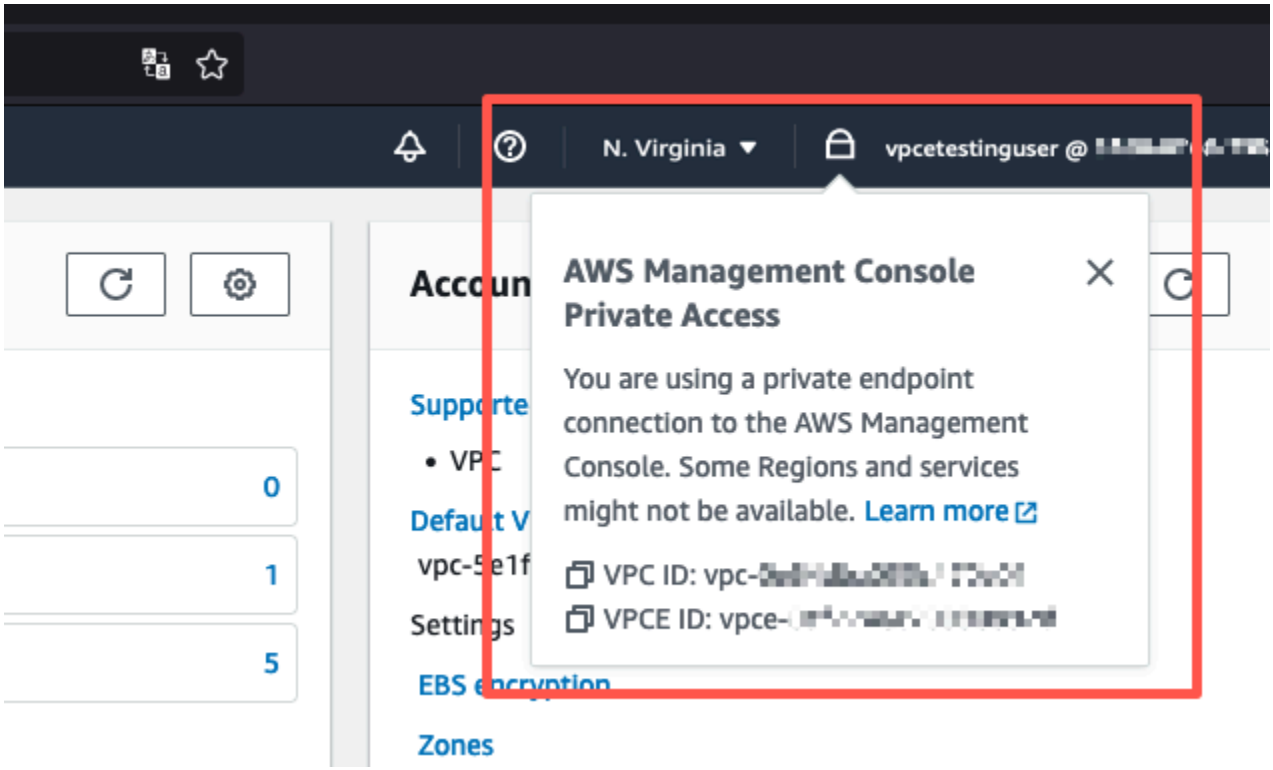
1. 前の手順のステップ 14 で受信した E メールを開きます。
2. E メールで、プロファイルを設定して WorkSpaces クライアントをダウンロードするために提供されている一意のリンクを選択します。
3. パスワードを設定します。
4. 任意のクライアントをダウンロードします。
5. クライアントをインストールして起動します。E メールに記載されている登録コードを入力して、[登録] を選択します。
6. ステップ 3 で作成した認証情報 WorkSpaces を使用して Amazon にサインインします。

AWS Management Console プライベートアクセスの設定をテストするには

1. から WorkSpaceブラウザを開きます。次に、AWS Management Consoleに移動し、認証情報を使用してサインインします。

   > **Note**
   >
   > Firefox をブラウザとして使用している場合は、ブラウザの設定で [DNS over HTTPS を有効にする] オプションがオフになっていることを確認してください。

2. Amazon Amazon S3コンソールを開き、 AWS Management Console プライベートアクセスを使用して接続されていることを確認します。
3. ナビゲーションバーのロックプライベートアイコンを選択すると、使用中の VPC と VPC エンドポイントが表示されます。次のスクリーンショットは、ロックプライベートアイコンの場所と VPC 情報を示しています。



IAM ポリシーを使った VPC 設定のテスト

Amazon EC2 で設定した VPC をさらにテストしたり、アクセスを制限する IAM ポリシーをデプロイ WorkSpaces したりできます。

指定された VPC を使用していない限り、次のポリシーは Amazon S3 へのアクセスを拒否します。

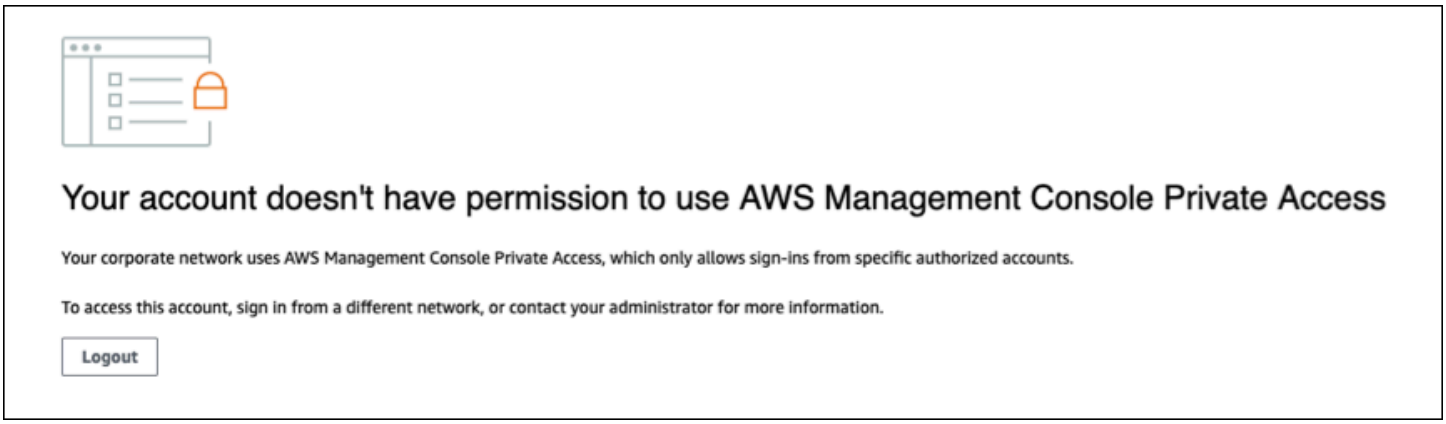
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "S3:*",
            "Resource": "*",
            "Condition": {
                "StringNotEqualsIfExists": {
                    "aws:SourceVpc": "sourceVPC"
                },
                "Bool": {
                    "aws:ViaAwsService": "false"
                }
            }
```

```
        }
    ]
}
```

次のポリシーは、サインインエンドポイントのプライベートアクセスポリシーを使用して、 AWS Management Console 選択した AWS アカウント IDs へのサインインを制限します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalAccount": [
                        "AWSAccountID"
                    ]
                }
            }
        }
    ]
}
```

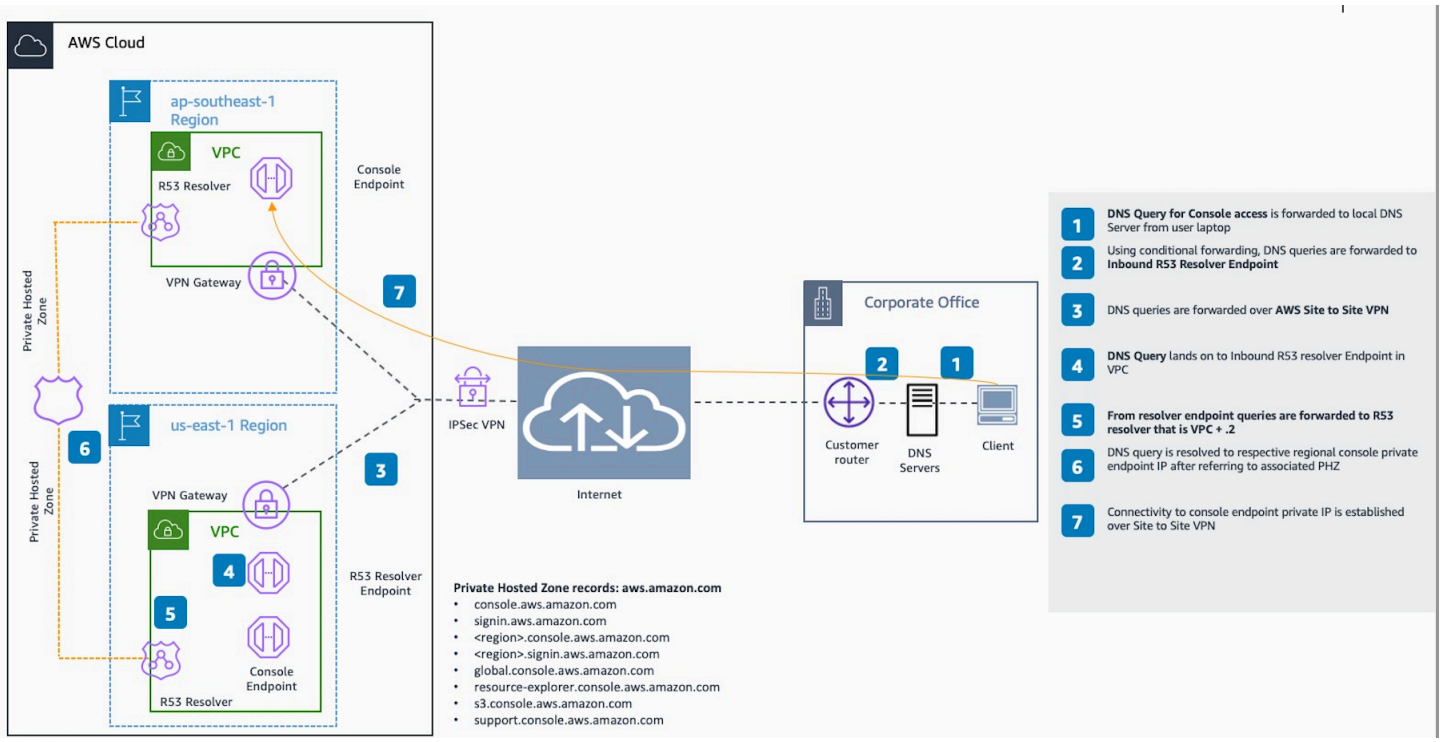
自分のアカウント以外の ID で接続すると、次のエラーページが表示されます。



リファレンスアーキテクチャ

オンプレミスネットワークから AWS Management Console プライベートアクセスにプライベートに接続するには、 AWS Site-to-Site VPN から AWS Virtual Private Gateway (VGW) への接続オプションを利用できます。 AWS Site-to-Site VPN は、接続を作成し、接続を介してトラフィックを渡すようにルーティングを設定することで、VPC からリモートネットワークへのアクセスを有効にします。詳細については、AWS 「Site-to-Site VPN ユーザーガイドAWS 」の「Site-to-Site VPN とは」を参照してください。 AWS 仮想プライベートゲートウェイ (VGW) は、VPC とオンプレミスネットワーク間のゲートウェイとして機能する高可用性のリージョンサービスです。

AWS Site-to-Site VPN AWS Virtual Private Gateway (VGW) への送信



このリファレンスアーキテクチャ設計の重要なコンポーネントは、 Amazon Route 53 Resolver、特にインバウンドリゾルバーです。プライベートアクセスエンドポイントが作成される VPC AWS Management Console で設定すると、リゾルバーエンドポイント (ネットワークインターフェイス) が指定されたサブネットに作成されます。その後、その IP アドレスをオンプレミスの DNS サーバー上の条件付きフォワーダーで参照して、プライベートホストゾーンのレコードをクエリできます。オンプレミスクライアントが に接続すると AWS Management Console、 AWS Management Console プライベートアクセスエンドポイントのプライベート IPsにルーティングされます。

AWS Management Console プライベートアクセスエンドポイントへの接続を設定する前に、 にアクセスするすべてのリージョン AWS Management Consoleと米国東部 (バージニア北部) リージョン

で AWS Management Console プライベートアクセスエンドポイントを設定し、プライベートホストゾーンを設定する前提条件のステップを完了します。

Console Toolbar での AWS CloudShell の起動

AWS CloudShell はブラウザベースの事前に認証されたシェルで、Console Toolbar の AWS Management Consoleから直接起動できます。ご希望のシェル (Bash、PowerShell、または Z シェル) を使用して、 サービスに対して AWS CLI コマンドを実行できます。

次の 2 つの方法のうちの 1 つを使用して、Console Toolbar から CloudShell を起動することができます。

- コンソールの左下にある CloudShell アイコンを選択します。
- コンソールナビゲーションバーで、CloudShell アイコンを選択します。

このサービスの詳細については、AWS CloudShell ユーザーガイドを参照してください。

AWS CloudShell が使用可能な AWS リージョンについては、「AWS リージョン別のサービス表」を参照してください。コンソールリージョンの選択は CloudShell リージョンと同期しています。選択したリージョンで CloudShell が利用できない場合、CloudShell は最も近いリージョンで実行されます。

請求情報を取得する

必要なアクセス権限を持っている場合、コンソールから AWS の料金に関する情報を取得できます。

請求情報を取得するには

1. ナビゲーションバーで、アカウント名を選択します。
2. [Billing Dashboard] (請求ダッシュボード) をクリックします。
3. AWS Billing and Cost Management ダッシュボードを使用して、毎月の使用量の概要と内訳を表示します。詳細については、「AWS Billing ユーザーガイド」を参照してください。

コンソールでの Markdown の使用

Amazon などの の一部のサービスは AWS Management Console、特定のフィールドでの Markdown の使用 CloudWatchをサポートしています。このトピックでは、コンソールでサポートされている Markdown のフォーマットのタイプについて説明します。

内容

段落、線の間隔、および水平線

段落は空白行で区切ります。HTML に変換されたときに段落間の空白行が確実にレンダリングされるようにするには、改行しないスペース ( ) を含む新しい行を追加し、それに続けて空白行を追加します。次の例のように、複数の空白行を 1 つずつ挿入するには、この行のペアを繰り返します。

```
 

 
```

段落を区切る水平の罫線を作成するには、3 つの連続したハイフン (---) を含む新しい行を追加します。

```
Previous paragraph.
---
Next paragraph.
```

等幅タイプのテキストブロックを作成するには、3 つのバックティック (`) を含む行を追加します。等幅タイプで表示するテキストを入力します。次に、3 つのバックティックを含む別の新しい行を追加します。次の例は、表示時に等幅に変換されるテキストを示しています。

```
```
This appears in a text box with a background shading.
The text is in monospace.
```
```

ヘッダー

見出しを作成するには、シャープ記号 (#) を使用します。1 つのシャープ記号とスペースは、トップレベルの見出しを示します。2 つのシャープ記号を使用すると第 2 レベルのヘッダーが作成され、3 つのシャープ記号を使用すると第 3 レベルのヘッダーが作成されます。次の例は、最上位レベル、第 2 レベル、第 3 レベルの見出しを示しています。

```
# Top-level heading
```

```
## Second-level heading
```

```
### Third-level heading
```

テキストのフォーマット

テキストを斜体でフォーマットするには、両端を 1 つのアンダースコア (_) またはアスタリスク (*) で囲みます。

```
*This text appears in italics.*
```

テキストを太字でフォーマットするには、両端を 2 つのアンダースコアまたは 2 つのアスタリスクで囲みます。

```
**This text appears in bold.**
```

テキストを取り消し線でフォーマットするには、両端を 2 つのチルダ (~) で囲みます。

```
~~This text appears in strikethrough.~~
```

リンク

テキストのハイパーリンクを追加するには、角かっこ ([ ]) で囲まれたリンクテキストを入力します。その後に、かっこで囲んだ完全な URL (( )) を入力します。次に例を示します。

```
Choose [link_text](http://my.example.com).
```

リスト

行を箇条書きの一部としてフォーマットするには、1 つのアスタリスク (*) に続いてスペースで始まる別々の行に追加します。次に例を示します。

```
Here is a bulleted list:
* Ant
* Bug
* Caterpillar
```

行を番号付きリストの一部としてフォーマットするには、別々の行に、数値、ピリオド (.)、およびスペースで始まる行に追加します。次に例を示します。

```
Here is a numbered list:
1. Do the first step
2. Do the next step
3. Do the final step
```

テーブルとボタン (CloudWatch ダッシュボード)

CloudWatch ダッシュボードのテキストウィジェットは Markdown テーブルとボタンをサポートしています。

表を作成するには、縦棒 (|) を使用して列を区切り、新しい行を使用して行を区切ります。最初の行をヘッダー行にするには、ヘッダー行と、値の最初の行の間に行を挿入します。次に、表の各列に少なくとも 3 つのハイフン (-) を追加します。縦棒を使用して列を区切ります。次の例は、2 つの列、ヘッダー行、および 2 行のデータを含む表の Markdown を示しています。

```
Table | Header
----|-----
Amazon Web Services | AWS
```

```
1 | 2
```

前の例の Markdown テキストでは、以下の表が作成されます。

テーブル	[Header] (ヘッダー)
Amazon Web Services	AWS
1	2

CloudWatch ダッシュボードのテキストウィジェットでは、ハイパーリンクをボタンとしてフォーマットすることもできます。ボタンを作成するには、[button:*Button text*] を使用し、その後に、かっこで囲んだ完全な URL (( )) を入力します。次に例を示します。

```
[button:Go to AWS](http://my.example.com)
[button:primary:This button stands out even more](http://my.example.com)
```

トラブルシューティング

に関する一般的な問題の解決策については、このセクションを参照してください AWS Management Console。

Amazon Q Developer を使用して、一部の AWS サービスの一般的なエラーを診断およびトラブルシューティングすることもできます。詳細については、「Amazon Q デベロッパーユーザーガイド」の「Amazon Q デベロッパーによるコンソールでの一般的なエラーの診断」を参照してください。

トピック

ページが正しく読み込まれない

- この問題がたまにしか発生しない場合は、インターネット接続を確認してください。別のネットワーク経由で接続するか、VPN の有無にかかわらず接続を試みるか、別のウェブブラウザを使用してみてください。
- 影響を受けるすべてのユーザーが同じチームに属する場合は、プライバシーブラウザの拡張機能またはセキュリティファイアウォールの問題である可能性があります。プライバシーブラウザの拡張機能とセキュリティファイアウォールは、 が使用するドメインへのアクセスをブロックできます AWS Management Console。これらの拡張機能をオフにするか、ファイアウォールの設定の調整をお勧めします。接続の問題を確認するには、お使いのブラウザのデベロッパーツール (Chrome、Firefoxの) をクリックし、[コンソール] タブに表示されたエラーを確認します。は、次のリストを含むドメインのサフィックス AWS Management Console を使用します。これはすべてを網羅したリストではありません。これらのドメインのサフィックスは、 AWSのみが排他的に使用するわけではありません。
  - .a2z.com
  - .amazon.com
  - .amazonaws.com
  - .aws

- .aws.com
- .aws.dev
- .awscloud.com
- .awsplayer.com
- .awsstatic.com
- .cloudfront.net
- .live-video.net

> **Warning**
>
> 2022 年 7 月 31 日以降、 は Internet Explorer 11 をサポートし AWS なくなりました。サポートされている他のブラウザ AWS Management Console で を使用することをお勧めします。詳細については、AWS ニュースブログを参照してください。

に接続すると、ブラウザに「アクセス拒否」エラーが表示される AWS Management Console

コンソールに最近加えられた変更は、以下のすべてを使用している場合、アクセスに影響する可能性があります。

- VPC 内からのブラウザ。
- VPC エンドポイント。
- `aws:SourceIp` グローバル条件キーを含む IAM ポリシー。

コンソールで、IAM ポリシーページ に移動します。`aws:SourceIp` グローバル条件キーを含む IAM ポリシーを確認し、`aws:SourceVpc`キーを追加することをお勧めします。

または、 AWS Management Console プライベートアクセス機能へのオンボーディングを検討して、VPC エンドポイント AWS Management Console を介して にアクセスし、ポリシー`aws:SourceVpc`の条件を使用することを検討することもできます。詳細については、「AWS Management Console プライベートアクセス」を参照してください。

への接続時にブラウザにタイムアウトエラーが表示される AWS Management Console

デフォルトの でサービスが停止した場合 AWS リージョン、 に接続しようとすると、ブラウザに 504 Gateway タイムアウトエラーが表示されることがあります AWS Management Console。別のリージョン AWS Management Console から にログインするには、URL で代替リージョンエンドポイントを指定します。例えば、`us-west-1` (北カリフォルニア) リージョンでサービス停止があった場合に、`us-west-2` (オレゴン) リージョンにアクセスするには、次のテンプレートを使用します。

```
https://region-code.console.aws.amazon.com
```

詳細については、AWS 全般のリファレンス の「AWS Management Console サービスエンドポイント」を参照してください。

を含むすべての のステータスを表示するには AWS のサービス、 AWS Management Console「」を参照してくださいAWS Health Dashboard。

AWS Management Console の言語を変更したいが、ページ下部の言語選択メニューが見つからない

言語選択メニューは新しい [Unified Settings] (統合設定) ページに移動しました。の言語を変更するには AWS Management Console、統合設定ページ に移動し、コンソールの言語を選択します。

詳細については、AWS Management Consoleの言語の変更を参照してください。

ドキュメント履歴

以下の表は、AWS Management Console 入門ガイドの 2021 年 3 月以降の重要な変更点をまとめたものです。

変更	説明	日付
Amazon Q とチャットする	ユーザーが Amazon Q Developer に AWS 質問する方法を詳述した新しい設定ページ。詳細については、「Amazon Q デベロッパーとチャットする」を参照してください。	2024 年 5 月 29 日
myApplications	myApplications を紹介する新しいページ。詳細については、「 での myApplications とは AWS」を参照してください。	2023 年 11 月 29 日
統合設定の指定	言語や地域など、現在のユーザーに適用される設定とデフォルト値を設定するための新しい設定ページ。詳細については、「統合設定の指定」を参照してください。	2022 年 4 月 6 日
新しい AWS Console Home UI	新しい AWS Console Home UI には、重要な使用状況情報を表示するためのウィジェットと AWS サービスへのショートカットが含まれています。詳細については、「ウィジェットの操作」を参照してください。	2022 年 2 月 25 日

変更	説明	日付
コンソールの言語の変更	AWS Management Console の別の言語を選択します。詳細については、「AWS Management Consoleの言語の変更」を参照してください。	2021 年 4 月 1 日
の起動 CloudShell	AWS CloudShell から AWS Management Console を開き、 AWS CLI コマンドを実行します。詳細については、「 の起動 AWS CloudShell」を参照してください。	2021 年 3 月 22 日

AWS 用語集

AWS の最新の用語については、「AWS の用語集リファレンス」の「AWS 用語集」を参照してください。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。