



ユーザーガイド

Research and Engineering Studio



Research and Engineering Studio: ユーザーガイド

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

Amazon の商標およびトレードドレスは Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

概要	1
機能とメリット	1
概念と定義	3
アーキテクチャの概要	5
アーキテクチャ図	5
AWS この製品の サービス	6
デモ環境	9
ワンクリックデモスタックを作成する	9
前提条件	9
リソースと入力パラメータを作成する	10
デプロイ後のステップ	11
デプロイを計画する	13
コスト	13
セキュリティ	13
IAM ロール	13
セキュリティグループ	14
データ暗号化	14
クォータ	14
この製品の AWS サービスのクォータ	14
AWS CloudFormation クォータ	15
レジリエンスの計画	15
サポートされる AWS リージョン	15
製品をデプロイする	18
前提条件	18
管理ユーザー AWS アカウント を使用して を作成する	19
Amazon EC2 SSH キーペアを作成する	19
サービスクォータを増やす	19
パブリックドメインを作成する (オプション)	20
ドメインの作成 (GovCloud のみ)	20
外部リソースを提供する	21
環境で LDAPS を設定する (オプション)	22
プライベート VPC を設定する (オプション)	22
外部リソースを作成する	34
ステップ 1: 製品を起動する	40

ステップ 2: 初めてサインインする	48
製品を更新する	49
メジャーバージョンの更新	49
マイナーバージョンの更新	49
製品のアンインストール	51
の使用 AWS Management Console	51
の使用 AWS Command Line Interface	51
shared-storage-security-group の削除	51
Amazon S3 バケットの削除	52
設定ガイド	53
ユーザーとグループの管理	53
IAM アイデンティティセンターでの SSO の設定	53
シングルサインオン (SSO) 用の ID プロバイダーの設定	57
ユーザーのパスワードの設定	64
サブドメインの作成	64
ACM 証明書を作成する	65
Amazon CloudWatch Logs	66
カスタムアクセス許可の境界の設定	67
RES 対応 AMIs を設定する	71
RES 環境にアクセスするための IAM ロールを準備する	72
EC2 Image Builder コンポーネントを作成する	73
EC2 Image Builder レシピを準備する	78
EC2 Image Builder インフラストラクチャを設定する	80
Image Builder イメージパイプラインを設定する	80
Image Builder イメージパイプラインを実行する	81
RES に新しいソフトウェアスタックを登録する	82
管理者ガイド	83
セッション管理	83
ダッシュボード	83
セッション	84
ソフトウェアスタック (AMIs)	85
デバッグ	86
デスクトップ設定	86
環境管理	87
プロジェクト	88
[ユーザー]	91

グループ	92
アクセス許可プロファイル	92
ファイルシステム	96
環境ステータス	97
スナップショットの管理	98
環境設定	103
Amazon S3 バケット	103
シークレットの管理	116
コストのモニタリングと制御	118
製品を使用する	121
仮想デスクトップ	121
サポートされるオペレーティングシステム	121
新しいデスクトップを起動する	121
デスクトップにアクセスする	122
デスクトップの状態を制御する	122
仮想デスクトップを変更する	123
セッション情報を取得する	124
仮想デスクトップをスケジュールする	124
共有デスクトップ	124
デスクトップを共有する	125
共有デスクトップにアクセスする	125
ファイルブラウザ	125
ファイルのアップロード (複数可)	125
ファイルの削除 (複数可)	125
お気に入りを管理する	126
ファイルを編集する	126
ファイルの転送	126
SSH アクセス	127
トラブルシューティング	128
一般的なデバッグとモニタリング	131
便利なログおよびイベント情報ソース	131
一般的な Amazon EC2 コンソールの外観	135
Windows DCV デバッグ	136
NICE DCV バージョン情報の検索	137
RunBooks の問題	137
インストールの問題	139

ID 管理の問題	146
[Storage (ストレージ)]	150
スナップショット	153
インフラストラクチャ	154
仮想デスクトップの起動	155
仮想デスクトップコンポーネント	159
Env 削除	164
デモ環境	169
既知の問題	170
既知の問題 2024.x	171
注意	186
改訂	187
.....	clxxxix

概要

Important

このバージョンのユーザーガイドでは、での Research and Engineering Studio のリリース 2024.08 について説明します AWS。現在のバージョンについては、[「ユーザーガイド」の「Research and Engineering Studio AWS」](#)を参照してください。

Research and Engineering Studio (RES) は、AWS サポートされているオープンソース製品です。IT 管理者は、サイエンティストやエンジニアがテクニカルコンピューティングワークロードを実行できるウェブポータルを提供できます AWS。RES は、ユーザーが安全な仮想デスクトップを起動して、科学研究、製品設計、エンジニアリングシミュレーション、またはデータ分析ワークロードを実行するための単一の画面を提供します。ユーザーは、既存の企業認証情報を使用して RES ポータルに接続し、個々のプロジェクトまたは共同プロジェクトに取り組むことができます。

管理者は、特定のユーザーのセットに対してプロジェクトと呼ばれる仮想コラボレーションスペースを作成し、共有リソースにアクセスしてコラボレーションできます。管理者は、独自のアプリケーションソフトウェアスタック (AMIs) を構築し、RES ユーザーが Windows または Linux 仮想デスクトップを起動できるようにし、共有ファイルシステムを介してプロジェクトデータにアクセスできるようにします。管理者は、ソフトウェアスタックとファイルシステムを割り当て、それらのプロジェクトユーザーのみにアクセスを制限できます。管理者は、組み込みテレメトリを使用して環境の使用状況をモニタリングし、ユーザーの問題をトラブルシューティングできます。また、リソースの過剰消費を防ぐために、個々のプロジェクトの予算を設定することもできます。製品はオープンソースであるため、お客様は自分のニーズに合わせて RES ポータルのユーザーエクスペリエンスをカスタマイズすることもできます。

RES は追加料金なしで利用でき、アプリケーションの実行に必要な AWS リソースに対してのみ料金が発生します。

このガイドでは、での Research and Engineering Studio の概要 AWS、リファレンスアーキテクチャとコンポーネント、デプロイを計画する際の考慮事項、および RES を Amazon Web Services (AWS) クラウドにデプロイするための設定手順について説明します。

機能と利点

の Research and Engineering Studio AWS には、次の機能があります。

ウェブベースのユーザーインターフェイス

RES は、管理者、研究者、エンジニアが研究およびエンジニアリングワークスペースにアクセスして管理するために使用できるウェブベースのポータルを提供します。科学者やエンジニアは、RES を使用するために AWS アカウント やクラウドの専門知識を持つ必要はありません。

プロジェクトベースの設定

プロジェクトを使用して、アクセス許可の定義、リソースの割り当て、一連のタスクまたはアクティビティの予算の管理を行います。整合性とコンプライアンスのために、特定のソフトウェアスタック (オペレーティングシステムと承認済みアプリケーション) とストレージリソースをプロジェクトに割り当てます。プロジェクトごとに支出を監視および管理します。

コラボレーションツール

科学者やエンジニアは、プロジェクトの他のメンバーを招待してコラボレーションし、同僚に求めるアクセス許可レベルを設定できます。これらのユーザーは、RES にサインインしてデスクトップに接続できます。

既存の ID 管理インフラストラクチャとの統合

既存の ID 管理およびディレクトリサービスインフラストラクチャと統合して、ユーザーの既存の企業 ID を使用して RES ポータルに接続し、既存のユーザーおよびグループメンバーシップを使用してプロジェクトにアクセス許可を割り当てます。

永続的ストレージと共有データへのアクセス

仮想デスクトップセッション間で共有データへのアクセスをユーザーに許可するには、既存のファイルシステムに接続するか、RES 内に新しいファイルシステムを作成します。サポートされているストレージサービスには、Linux デスクトップ用の Amazon Elastic File System と、Windows および Linux デスクトップ用の NetApp ONTAP 用の Amazon FSx が含まれます。

モニタリングとレポート

分析ダッシュボードを使用して、インスタンスタイプ、ソフトウェアスタック、オペレーティングシステムタイプのリソース使用状況をモニタリングします。ダッシュボードには、レポート用のプロジェクト別のリソース使用状況の内訳も表示されます。

予算とコストの管理

RES プロジェクト AWS Budgets にリンクして、各プロジェクトのコストをモニタリングします。予算を超えた場合は、VDI セッションの起動を制限できます。

概念と定義

このセクションでは、主要な概念について説明し、この製品に固有の用語を定義します。

ファイルブラウザ

ファイルブラウザは、現在ログインしているユーザーがファイルシステムを表示できる RES ユーザーインターフェイスの一部です。

ファイルシステム

ファイルシステムは、プロジェクトデータ (データセットと呼ばれることが多い) のコンテナとして機能します。プロジェクトの境界内でストレージソリューションを提供し、コラボレーションとデータアクセスコントロールを向上させます。

グローバル管理者

RES 環境間で共有されている RES リソースにアクセスできる管理者代理。スコープとアクセス許可は複数のプロジェクトにまたがります。プロジェクトを作成または変更し、プロジェクト所有者を割り当てることができます。プロジェクト所有者とプロジェクトメンバーにアクセス許可を委任または割り当てることができます。組織のサイズによっては、同じ人が RES 管理者として機能する場合があります。

プロジェクト

プロジェクトは、データとコンピューティングリソースの明確な境界として機能するアプリケーション内の論理パーティションであり、データフローのガバナンスを確保し、プロジェクト間でのデータと VDI ホストの共有を防止します。

プロジェクトベースのアクセス許可

プロジェクトベースのアクセス許可は、複数のプロジェクトが存在するシステム内のデータと VDI ホストの両方の論理パーティションを記述します。プロジェクト内のデータと VDI ホストへのユーザーのアクセスは、関連するロール (複数可) によって決まります。ユーザーには、アクセスが必要なプロジェクトごとにアクセス (またはプロジェクトメンバーシップ) を割り当てする必要があります。それ以外の場合、ユーザーはメンバーシップが付与されていない場合、プロジェクトデータと VDI 到にアクセスできません。

プロジェクトメンバー

RES リソース (VDI、ストレージなど) のエンドユーザー。スコープとアクセス許可は、割り当てられたプロジェクトに制限されます。アクセス許可を委任または割り当ててはできません。

プロジェクトの所有者

特定のプロジェクトへのアクセス権と所有権を持つ管理代理人。スコープとアクセス許可は、所有するプロジェクト (複数可) に制限されます。所有するプロジェクト内のプロジェクトメンバーにアクセス許可を割り当てることができます。

ソフトウェアスタック

ソフトウェアスタックは、ユーザーが VDI ホスト用にプロビジョニングするために選択したオペレーティングシステムに基づく RES 固有のメタデータを持つ [Amazon マシンイメージ \(AMI\)](#) です。

VDI ホスト

仮想デスクトップインスタンス (VDI) ホストを使用すると、プロジェクトメンバーはプロジェクト固有のデータとコンピューティング環境にアクセスし、安全で隔離されたワークスペースを確保できます。

AWS 用語の一般的なリファレンスについては、AWS 全般のリファレンスの [AWS 用語集](#) を参照してください。

アーキテクチャの概要

このセクションでは、この製品でデプロイされたコンポーネントのアーキテクチャ図を示します。

アーキテクチャ図

デフォルトのパラメータを使用してこの製品をデプロイすると、に次のコンポーネントがデプロイされます AWS アカウント。

図 1: AWS アーキテクチャに関する Research and Engineering Studio

Note

AWS CloudFormation リソースは AWS Cloud Development Kit (AWS CDK) コンストラクトから作成されます。

テンプレートで AWS CloudFormation デプロイされた製品コンポーネントの大まかなプロセスフローは次のとおりです。

1. RES は、ウェブポータルコンポーネントと以下をインストールします。
 - a. インタラクティブワークロード用のエンジニアリング仮想デスクトップ (eVDI) コンポーネント
 - b. メトリクスコンポーネント

Amazon CloudWatch は eVDI コンポーネントからメトリクスを受け取ります。

- c. 踏み台ホストコンポーネント

管理者は、SSH を使用して踏み台ホストコンポーネントに接続し、基盤となるインフラストラクチャを管理できます。

2. RES は、NAT ゲートウェイの背後にあるプライベートサブネットにコンポーネントをインストールします。管理者は、Application Load Balancer (ALB) または踏み台ホストコンポーネントを介してプライベートサブネットにアクセスします。
3. Amazon DynamoDB は環境設定を保存します。
4. AWS Certificate Manager (ACM) は、Application Load Balancer (ALB) のパブリック証明書を生成して保存します。

Note

を使用して AWS Certificate Manager、ドメインの信頼された証明書を生成することをお勧めします。

5. Amazon Elastic File System (EFS) は、該当するすべてのインフラストラクチャホストと eVDI Linux セッションにマウントされたデフォルトの /home ファイルシステムをホストします。
6. RES は Amazon Cognito を使用して、内で clusteradmin という名前の初期ブートストラップユーザーを作成し、インストール時に提供された E メールアドレスに一時的な認証情報を送信します。clusteradmin は、初回ログインでパスワードを変更する必要があります。
7. Amazon Cognito は、アクセス許可管理のために組織の Active Directory およびユーザー ID と統合します。
8. セキュリティゾーンを使用すると、管理者はアクセス許可に基づいて製品内の特定のコンポーネントへのアクセスを制限できます。

AWS この製品の サービス

AWS サービス	説明
Amazon Elastic Compute Cloud	コア。選択したオペレーティングシステムとソフトウェアスタックを使用して仮想デスクトップを作成するための基盤となるコンピューティングサービスを提供します。
エラスティックロードバランシング	コア。踏み台、クラスターマネージャー、VDI ホストは、ロードバランサーの背後にある Auto Scaling グループに作成されます。ELB は、RES ホスト間でウェブポータルからのトラフィックのバランスを取ります。
Amazon Virtual Private Cloud	コア。すべてのコア製品コンポーネントは VPC 内に作成されます。
Amazon Cognito	コア。ユーザー ID と認証を管理します。 。Active Directory ユーザーは Amazon Cognito

AWS サービス	説明
	ユーザーとグループにマッピングされ、アクセスレベルを認証します。
Amazon Elastic File System	コア。/home ファイルブラウザと VDI ホスト用のファイルシステム、および共有外部ファイルシステムを提供します。
Amazon DynamoDB	コア。ユーザー、グループ、プロジェクト、ファイルシステム、コンポーネント設定などの設定データを保存します。
AWS Systems Manager	コア。VDI セッション管理のコマンドを実行するためのドキュメントを保存します。
AWS Lambda	コア。DynamoDB テーブル内の設定の更新、Active Directory 同期ワークフローの開始、プレフィックスリストの更新などの製品機能をサポートします。
Amazon CloudWatch	サポート。すべての Amazon EC2 ホストと Lambda 関数のメトリクスとアクティビティログを提供します。
Amazon Simple Storage Service	サポート。ホストブートストラップと設定のアプリケーションバイナリを保存します。
AWS Key Management Service	サポート。Amazon SQS キュー、DynamoDB テーブル、Amazon SNS トピックでの保管時の暗号化に使用されます。
AWS Secrets Manager	サポート。サービスアカウントの認証情報を Active Directory と VDI の自己署名証明書に保存します。
AWS CloudFormation	サポート。製品のデプロイメカニズムを提供します。

AWS サービス	説明
AWS Identity and Access Management	サポート。ホストのアクセスレベルを制限します。
Amazon Route 53	サポート。内部ロードバランサーと踏み台ホストドメイン名を解決するためのプライベートホストゾーンを作成します。
Amazon Simple Queue Service	サポート。非同期実行をサポートするタスクキューを作成します。
Amazon Simple Notification Service	サポート。コントローラーやホストなどの VDI コンポーネント間のパブリケーションサブスクライブモデルをサポートします。
AWS Fargate	サポート。Fargate タスクを使用して環境をインストール、更新、削除します。
Amazon FSx ファイルゲートウェイ	オプション。外部共有ファイルシステムを提供します。
Amazon FSx for NetApp ONTAP	オプション。外部共有ファイルシステムを提供します。
AWS Certificate Manager	オプション。カスタムドメインの信頼された証明書を生成します。
AWS Backup	オプション。Amazon EC2 ホスト、ファイルシステム、DynamoDB のバックアップ機能を提供します。

デモ環境を作成する

Research and Engineering Studio を試すには、このセクションのステップに従います AWS。このデモでは、[AWS デモ環境スタックテンプレートの Research and Engineering Studio を使用して、最小限のパラメータセットで非本番環境をデプロイします](#)。SSO には Keycloak サーバーを使用します。

スタックをデプロイした後、ログインする前に、[デプロイ後のステップ](#)以下の手順に従って 環境でユーザーを設定する必要があります。

ワンクリックデモスタックを作成する

この AWS CloudFormation スタックは、Research and Engineering Studio に必要なすべてのコンポーネントを作成します。

デプロイまでの時間: ~90 分

前提条件

トピック

- [管理ユーザー AWS アカウント を使用して を作成する](#)
- [Amazon EC2 SSH キーペアを作成する](#)
- [サービスクォータを増やす](#)

管理ユーザー AWS アカウント を使用して を作成する

管理ユーザー AWS アカウント を持つ が必要です。

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

サインアップ手順の一環として、電話またはテキストメッセージを受け取り、電話キーパッドで検証コードを入力します。

にサインアップすると AWS アカウント、AWS アカウントのルートユーザー が作成されます。ルートユーザーには、アカウントのすべての AWS のサービス とリソースへのアクセス権があ

ります。セキュリティベストプラクティスとして、ユーザーに管理アクセス権を割り当て、[ルートユーザーアクセスが必要なタスク](#)の実行にはルートユーザーのみを使用するようにしてください。

Amazon EC2 SSH キーペアを作成する

Amazon EC2 SSH キーペアがない場合は、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 を使用したキーペアの作成」を参照してください。

サービスクォータを増やす

[以下のサービスクォータを増やす](#)ことをお勧めします。

- [Amazon VPC](#)
 - NAT ゲートウェイあたりの Elastic IP アドレスクォータを 5 から 8 に増やす
 - アベイラビリティゾーンあたりの NAT ゲートウェイを 5 から 10 に増やす
- [Amazon EC2](#)
 - EC2-VPC Elastic IPs

AWS アカウントには、AWS サービスごとに、以前は制限と呼ばれていたデフォルトのクォータがあります。特に明記されていない限り、クォータは地域固有です。一部のクォータについては引き上げをリクエストできますが、その他のクォータについては引き上げることはできません。詳細については、「[the section called “この製品の AWS サービスのクォータ”](#)」を参照してください。

リソースと入力パラメータを作成する

1. にサインイン AWS Management Console し、<https://console.aws.amazon.com/cloudformation> で AWS CloudFormation コンソールを開きます。

Note

管理者アカウントにいることを確認します。

2. コンソールで[テンプレートを起動](#)します。
3. パラメータ で、この製品テンプレートのパラメータを確認し、必要に応じて変更します。

パラメータ	デフォルト	説明
EnvironmentName	<i>#res-demo#</i>	res- で始まり、11 文字以下の RES 環境に与えられる一意の名前。
AdministratorEmail		製品のセットアップを完了したユーザーの E メールアドレス。Active Directory のシングルサインオン統合に障害が発生した場合、このユーザーはさらにブレイクグラスユーザーとして機能します。
KeyPair		インフラストラクチャホストへの接続に使用されるキーペア。
ClientIPCIDR	<0.0.0.0/0>	システムへの接続を制限する IP アドレスフィルター。デプロイ後に ClientIpCidr を更新できます。
InboundPrefixList		(オプション) 踏み台ホストへのウェブ UI と SSH への直接アクセスが許可されている IPs のマネージドプレフィックスリストを指定します。

デプロイ後のステップ

1. でのユーザーパスワードのリセット AWS Directory Service- デモスタックは、admin1、user1、 の 4 人のユーザーをユーザー名で作成admin2しますuser2。
 - a. Directory Service コンソールに移動します。

- b. 環境のディレクトリ ID を選択します。ディレクトリ ID はスタックの出力から取得できません <StackName>*DirectoryService*。
 - c. 右上のアクションドロップダウンメニューから、ユーザーのパスワードをリセットを選択します。
 - d. 使用するすべてのユーザーについて、ユーザー名を入力し、使用するパスワードを入力し、パスワードのリセットを選択します。
2. ユーザーパスワードをリセットしたら、Research and Engineering Studio が環境内のユーザーを同期するまで待つ必要があります。Research and Engineering Studio は、xx.00 に 1 時間ごとにユーザーを同期します。これが発生するのを待つか、「」に記載されている手順に従ってユーザーをすぐに [Active Directory に追加されたが、RES に欠落しているユーザー](#) 同期できます。

これでデプロイの準備ができました。E メールで受け取った EnvironmentUrl を使用して UI にアクセスするか、デプロイされたスタックの出力から同じ URL を取得することもできます。Active Directory で のパスワードをリセットしたユーザーとパスワードを使用して、Research and Engineering Studio 環境にログインできるようになりました。

デプロイを計画する

コスト

の Research and Engineering Studio AWS は追加料金なしで利用でき、アプリケーションの実行に必要なリソースに対して AWS のみ料金が発生します。詳細については、「[AWS この製品の サービス](#)」を参照してください。

Note

この製品の実行中に使用される AWS サービスのコストは、お客様の負担となります。コスト管理を容易にするために、[AWS Cost Explorer](#) を使用して[予算](#)を作成することを推奨しています。価格は変更されることがあります。詳細については、この製品で使用される各 AWS サービスの料金ウェブページを参照してください。

セキュリティ

AWS インフラストラクチャ上にシステムを構築する場合、セキュリティ責任はお客様と の間で共有されます AWS。この[責任共有モデル](#)は、ホストオペレーティングシステム、仮想化レイヤー、サービスが運用されている施設の物理的なセキュリティなどのコンポーネントを AWS 運用、管理、制御するため、運用上の負担を軽減します。AWS セキュリティの詳細については、[AWS クラウド「セキュリティ」](#)を参照してください。

IAM ロール

AWS Identity and Access Management (IAM) ロールを使用すると、 のサービスおよびユーザーにきめ細かなアクセスポリシーとアクセス許可を割り当てることができます AWS クラウド。この製品は、製品の AWS Lambda 関数と Amazon EC2 インスタンスにリージョンリソースを作成するためのアクセス権を付与する IAM ロールを作成します。

RES は IAM 内のアイデンティティベースのポリシーをサポートしています。デプロイされると、RES は管理者のアクセス許可とアクセスを定義するポリシーを作成します。製品を実装する管理者は、RES と統合された既存のカスタマー Active Directory 内でエンドユーザーとプロジェクトリーダーを作成および管理します。詳細については、AWS 「Identity and Access Management ユーザーガイド」の「[IAM ポリシーの作成](#)」を参照してください。

組織の管理者は、アクティブディレクトリを使用してユーザーアクセスを管理できます。エンドユーザーが RES ユーザーインターフェイスにアクセスすると、RES は [Amazon Cognito](#) で認証します。

セキュリティグループ

この製品で作成されたセキュリティグループは、Lambda 関数、EC2 インスタンス、ファイルシステム CSR インスタンス、リモート VPN エンドポイント間のネットワークトラフィックを制御および分離するように設計されています。セキュリティグループを確認し、製品のデプロイ後に必要に応じてアクセスをさらに制限することをお勧めします。

データ暗号化

デフォルトでは、AWS (RES) の Research and Engineering Studio は、RES 所有のキーを使用して、保管中および転送中の顧客データを暗号化します。RES をデプロイするときに、を指定できます AWS KMS key。RES は、認証情報を使用してキーアクセスを付与します。顧客所有および管理の を指定すると AWS KMS key、保管中の顧客データはそのキーを使用して暗号化されます。

RES は、SSL/TLS を使用して転送中の顧客データを暗号化します。TLS 1.2 が必要ですが、TLS 1.3 をお勧めします。

クォータ

サービスクォータ (制限とも呼ばれます) は、AWS アカウントのサービスリソースまたはオペレーションの最大数です。

この製品の AWS サービスのクォータ

この[製品に実装されている各サービス](#)に十分なクォータがあることを確認してください。詳細については、「[AWS サービスクォータ](#)」を参照してください。

この製品では、次のサービスのクォータを引き上げることをお勧めします。

- Amazon Virtual Private Cloud
- Amazon EC2

クォータの引き上げをリクエストするには、Service Quotas ユーザーガイドの「[クォータ引き上げリクエスト](#)」を参照してください。Service Quotas でクォータがまだ利用できない場合は、[\[上限引き上げ\]](#) フォームを使用してください。

AWS CloudFormation クォータ

AWS アカウント には、この製品で [スタックを起動](#) するときに注意すべき AWS CloudFormation クォータがあります。これらのクォータを理解することで、この製品を正常にデプロイできないような制限エラーを回避できます。詳細については、「ユーザーガイド」の [AWS CloudFormation 「のクォータ」](#) を参照してください。AWS CloudFormation

レジリエンスの計画

製品は、Amazon EC2 インスタンスの最小数とサイズでデフォルトのインフラストラクチャをデプロイして、システムを運用します。大規模な本番環境の耐障害性を向上させるには、インフラストラクチャの Auto Scaling グループ (ASG) 内のデフォルトの最小容量設定を増やすことをお勧めします。値を 1 つのインスタンスから 2 つのインスタンスに増やすと、複数のアベイラビリティゾーン (AZ) の利点が得られ、予期しないデータ損失が発生した場合にシステム機能を復元する時間を短縮できます。

ASG 設定は、<https://console.aws.amazon.com/ec2/> の Amazon EC2 コンソール内でカスタマイズできます。製品はデフォルトで 4 つの ASGs を作成し、各名前は で終わります -asg。最小値と必要な値は、本番環境に適した量に変更できます。変更するグループを選択し、アクションと編集を選択します。ASGs、「Amazon EC2 [Auto Scaling ユーザーガイド](#)」の「[Auto Scaling グループのサイズをスケールする](#)」を参照してください。Amazon EC2 Auto Scaling

サポートされる AWS リージョン

この製品は、現在すべての で利用できないサービスを使用します AWS リージョン。この製品は、すべてのサービス AWS リージョン が利用可能な で起動する必要があります。リージョン別の AWS サービスの最新の可用性については、「[al AWS リージョン Services List](#)」を参照してください。

の Research and Engineering Studio AWS は、以下でサポートされています AWS リージョン。

リージョン名	リージョン	リリース 2024.06 以前	リリース 2024 年 8 月
米国東部 (バージニア北部)	us-east-1	はい	はい
米国東部 (オハイオ)	us-east-2	はい	はい

リージョン名	リージョン	リリース 2024.06 以前	リリース 2024 年 8 月
米国西部 (北カリフォルニア)	us-west-1	はい	はい
米国西部 (オレゴン)	us-west-2	はい	はい
アジアパシフィック (東京)	ap-northeast-1	はい	はい
アジアパシフィック (ソウル)	ap-northeast-2	はい	はい
アジアパシフィック (ムンバイ)	ap-south-1	はい	はい
アジアパシフィック (シンガポール)	ap-southeast-1	はい	はい
アジアパシフィック (シドニー)	ap-southeast-2	はい	はい
カナダ (中部)	ca-central-1	はい	はい
欧州 (フランクフルト)	eu-central-1	はい	はい
欧州 (ミラノ)	eu-south-1	はい	はい
欧州 (アイルランド)	eu-west-1	はい	はい
欧州 (ロンドン)	eu-west-2	はい	はい
欧州 (パリ)	eu-west-3	はい	はい
欧州 (ストックホルム)	eu-north-1	いいえ	はい
イスラエル (テルアビブ)	il-central-1	はい	はい

リージョン名	リージョン	リリース 2024.06 以前	リリース 2024 年 8 月
AWS GovCloud (米国西部)	us-gov-west-1	はい	いいえ

製品をデプロイする

Note

この製品は、[AWS CloudFormation テンプレートとスタック](#)を使用してデプロイを自動化します。CloudFormation テンプレートは、この製品に含まれる AWS リソースとそのプロパティを記述します。CloudFormation スタックは、テンプレートに記述されているリソースをプロビジョニングします。

製品を起動する前に、このガイドで前述した[コスト](#)、[アーキテクチャ](#)、[ネットワークセキュリティ](#)、その他の考慮事項を確認してください。

トピック

- [前提条件](#)
- [外部リソースを作成する](#)
- [ステップ 1: 製品を起動する](#)
- [ステップ 2: 初めてサインインする](#)

前提条件

トピック

- [管理ユーザー AWS アカウント を使用して を作成する](#)
- [Amazon EC2 SSH キーペアを作成する](#)
- [サービスクォータを増やす](#)
- [パブリックドメインを作成する \(オプション\)](#)
- [ドメインの作成 \(GovCloud のみ\)](#)
- [外部リソースを提供する](#)
- [環境で LDAPS を設定する \(オプション\)](#)
- [プライベート VPC を設定する \(オプション\)](#)

管理ユーザー AWS アカウント を使用して を作成する

管理ユーザー AWS アカウント を持つ が必要です。

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

サインアップ手順の一環として、電話またはテキストメッセージを受け取り、電話キーパッドで検証コードを入力します。

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

Amazon EC2 SSH キーペアを作成する

Amazon EC2 SSH キーペアがない場合は、キーペアを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 を使用したキーペアの作成」を参照してください。

サービスクォータを増やす

[以下のサービスクォータを増やす](#)ことをお勧めします。

- [Amazon VPC](#)
 - NAT ゲートウェイあたりの Elastic IP アドレスクォータを 5 から 8 に増やす
 - アベイラビリティゾーンあたりの NAT ゲートウェイを 5 から 10 に増やす
- [Amazon EC2](#)
 - EC2-VPC Elastic IPs

AWS アカウントには、AWS サービスごとに、以前は制限と呼ばれていたデフォルトのクォータがあります。特に明記されていない限り、クォータは地域固有です。一部のクォータについては引き上げをリクエストできますが、その他のクォータについては引き上げることはできません。詳細については、「[the section called “この製品の AWS サービスのクォータ”](#)」を参照してください。

パブリックドメインを作成する (オプション)

ユーザーフレンドリーな URL を持つには、製品のカスタムドメインを使用することをお勧めします。Amazon Route 53 または別のプロバイダーを使用してドメインを登録し、を使用してドメインの証明書をインポートする必要があります AWS Certificate Manager。パブリックドメインと証明書が既にある場合は、このステップをスキップできます。

1. 指示に従って、Route53 [にドメインを登録](#)します。確認メールが届きます。
2. ドメインのホストゾーンを取得します。これは Route53 によって自動的に作成されます。
 - a. Route53 コンソールを開きます。
 - b. 左側のナビゲーションからホストゾーンを選択します。
 - c. ドメイン名用に作成されたホストゾーンを開き、ホストゾーン ID をコピーします。
3. を開き AWS Certificate Manager、以下の手順に従って [ドメイン証明書をリクエスト](#)します。ソリューションをデプロイする予定のリージョンにいることを確認します。
4. ナビゲーションから証明書を一覧表示を選択し、証明書リクエストを見つけます。リクエストは保留中である必要があります。
5. 証明書 ID を選択してリクエストを開きます。
6. ドメインセクションから、Route53 でレコードを作成するを選択します。リクエストの処理には約 10 分かかります。
7. 証明書が発行されたら、証明書のステータスセクションから ARN をコピーします。

ドメインの作成 (GovCloud のみ)

AWS GovCloud (米国西部) リージョンにデプロイする場合は、これらの前提条件のステップを完了する必要があります。

1. パブリックホストドメインが作成された商用パーティション AWS アカウントに [Certificate AWS CloudFormation スタック](#)をデプロイします。
2. Certificate CloudFormation 出力から、とを見つけCertificateARNでメモしますPrivateKeySecretARN。
3. GovCloud パーティションアカウントで、CertificateARN出力の値を持つシークレットを作成します。ガシークレット値にアクセスできるように、新しいシークレット ARN をメモしvdc-gateway、シークレットに 2 つのタグを追加します。
 - a. res:ModuleName = virtual-desktop-controller

- b. `res:EnvironmentName = [environment name]` (res-demo の可能性があります)
- 4. GovCloud パーティションアカウントで、`PrivateKeySecretArn`出力の値を持つシークレットを作成します。がシークレット値にアクセスできるように、新しいシークレット ARN をメモし `vdc-gateway`、シークレットに 2 つのタグを追加します。
 - a. `res:ModuleName = virtual-desktop-controller`
 - b. `res:EnvironmentName = [environment name]` (res-demo の可能性があります)

外部リソースを提供する

の Research and Engineering Studio では、デプロイ時に次の外部リソースが存在することを AWS 想定しています。

- ネットワーク (VPC、パブリックサブネット、プライベートサブネット)

ここでは、RES 環境、Active Directory (AD)、共有ストレージのホストに使用される EC2 インスタンスを実行します。

- ストレージ (Amazon EFS)

ストレージボリュームには、仮想デスクトップインフラストラクチャ (VDI) に必要なファイルとデータが含まれています。

- ディレクトリサービス (AWS Directory Service for Microsoft Active Directory)

ディレクトリサービスは、RES 環境に対してユーザーを認証します。

- サービスアカウントのパスワードを含むシークレット

Research and Engineering Studio は、を使用して、サービスアカウントのパスワードなど、指定した[シークレット](#)にアクセスします[AWS Secrets Manager](#)。

Tip

デモ環境をデプロイしていて、これらの外部リソースが利用できない場合は、AWS ハイパフォーマンスコンピューティングレシピを使用して外部リソースを生成できます。アカウントにリソースをデプロイするには[外部リソースを作成する](#)、次のセクション「」を参照してください。

AWS GovCloud (米国西部) リージョンでのデモデプロイでは、「」の前提条件ステップを完了する必要があります [ドメインの作成 \(GovCloud のみ\)](#)。

環境で LDAPS を設定する (オプション)

環境で LDAPS 通信を使用する場合は、以下の手順を実行して、証明書を作成して AWS Managed Microsoft AD (AD) ドメインコントローラーにアタッチし、AD と RES 間の通信を提供する必要があります。

1. 「の [サーバー側の LDAPS を有効にする方法 AWS Managed Microsoft AD](#)」に記載されているステップに従います。LDAPS を既に有効にしている場合は、このステップをスキップできます。
2. AD で LDAPS が設定されていることを確認したら、AD 証明書をエクスポートします。
 - a. Active Directory サーバーに移動します。
 - b. 管理者として PowerShell を開きます。
 - c. certmgr.msc を実行して証明書リストを開きます。
 - d. 最初に信頼されたルート認証機関を開き、次に証明書を開いて、証明書リストを開きます。
 - e. AD サーバーと同じ名前の証明書を長押し (または右クリック) し、すべてのタスクを選択してからエクスポートします。
 - f. Base-64 でエンコードされた X.509 (.CER) を選択し、次へを選択します。
 - g. ディレクトリを選択し、次へを選択します。
3. シークレットの作成先 AWS Secrets Manager :

シークレットマネージャーでシークレットを作成する場合は、[シークレットのタイプ] で [その他のシークレット] を選択し、[プレーンテキスト] フィールドに PEM エンコードの証明書を貼り付けます。

4. 作成された ARN を書き留めて、の DomainTLSCertificateSecretARN パラメータとして入力します [the section called “ステップ 1: 製品を起動する”](#)。

プライベート VPC を設定する (オプション)

Research and Engineering Studio を分離された VPC にデプロイすると、組織のコンプライアンスとガバナンス要件を満たすためのセキュリティが強化されます。ただし、標準の RES デプロイは、依

存関係のインストールにインターネットアクセスに依存しています。プライベート VPC に RES をインストールするには、次の前提条件を満たす必要があります。

トピック

- [Amazon マシンイメージ \(AMIsを準備する](#)
- [VPC エンドポイントの設定](#)
- [VPC エンドポイントのないサービスに接続する](#)
- [プライベート VPC デプロイパラメータを設定する](#)

Amazon マシンイメージ (AMIsを準備する

1. [依存関係](#)をダウンロードします。分離された VPC にデプロイするには、RES インフラストラクチャでパブリックインターネットアクセスなしで依存関係を利用できる必要があります。
2. Amazon S3 読み取り専用アクセスと Amazon EC2 として信頼できる ID を持つ IAM ロールを作成します。
 - a. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
 - b. ロール から、ロールの作成 を選択します。
 - c. [信頼されたエンティティを選択] ページで以下を行います。
 - 信頼されたエンティティタイプで、 を選択します AWS のサービス。
 - 「サービス」または「ユースケース」の EC2」を選択し、「次へ」を選択します。
 - d. アクセス許可の追加で、次のアクセス許可ポリシーを選択し、次へを選択します。
 - AmazonS3ReadOnlyAccess
 - AmazonSSMManagedInstanceCore
 - EC2InstanceProfileForImageBuilder
 - e. ロール名と説明を追加し、ロールの作成を選択します。
3. EC2 Image Builder コンポーネントを作成します。
 - a. で EC2 Image Builder コンソールを開きます <https://console.aws.amazon.com/imagebuilder>。
 - b. 「保存されたリソース」で、「コンポーネント」を選択し、「コンポーネントの作成」を選択します。
 - c. コンポーネントの作成ページで、次の詳細を入力します。

- コンポーネントタイプ で、ビルド を選択します。
- コンポーネントの詳細については、以下を選択します。

パラメータ	ユーザーエントリ
Image operating system (OS)	Linux
Compatible OS Versions	Amazon Linux 2
Component name	Choose a name such as: <i><research-and-engineering-studio-infrastructure></i>
Component version	We recommend starting with 1.0.0.
Description	Optional user entry.

- d. コンポーネントの作成ページで、ドキュメントコンテンツの定義を選択します。
 - i. 定義ドキュメントの内容を入力する前に、tar.gz ファイルのファイル URI が必要です。RES が提供する tar.gz ファイルを Amazon S3 バケットにアップロードし、バケットプロパティからファイルの URI をコピーします。
 - ii. 次のように入力します。

 Note

AddEnvironmentVariables はオプションであり、インフラストラクチャホストにカスタム環境変数が必要ない場合は削除できます。

http_proxy および https_proxy環境変数を設定する場合、インスタンスがプロキシを使用して localhost、インスタンスメタデータ IP アドレス、および VPC エンドポイントをサポートするサービスをクエリしないようにするには、no_proxyパラメータが必要です。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License"). You may
not use this file except in compliance
```

```
# with the License. A copy of the License is located at
#
#      http://www.apache.org/licenses/LICENSE-2.0
#
# or in the 'license' file accompanying this file. This file is
# distributed on an 'AS IS' BASIS, WITHOUT WARRANTIES
# OR CONDITIONS OF ANY KIND, express or implied. See the License for the
# specific language governing permissions
# and limitations under the License.
name: research-and-engineering-studio-infrastructure
description: An RES EC2 Image Builder component to install required RES
  software dependencies for infrastructure hosts.
schemaVersion: 1.0

parameters:
  - AWSAccountID:
      type: string
      description: RES Environment AWS Account ID
  - AWSRegion:
      type: string
      description: RES Environment AWS Region
phases:
  - name: build
    steps:
      - name: DownloadRESInstallScripts
        action: S3Download
        onFailure: Abort
        maxAttempts: 3
        inputs:
          - source: '<s3 tar.gz file uri>'
            destination: '/root/bootstrap/res_dependencies/
res_dependencies.tar.gz'
            expectedBucketOwner: '{{ AWSAccountID }}'
      - name: RunInstallScript
        action: ExecuteBash
        onFailure: Abort
        maxAttempts: 3
        inputs:
          commands:
            - 'cd /root/bootstrap/res_dependencies'
            - 'tar -xf res_dependencies.tar.gz'
            - 'cd all_dependencies'
            - '/bin/bash install.sh'
      - name: AddEnvironmentVariables
```

```

    action: ExecuteBash
    onFailure: Abort
    maxAttempts: 3
    inputs:
      commands:
        - |
          echo -e "
          http_proxy=http://<ip>:<port>
          https_proxy=http://<ip>:<port>

          no_proxy=127.0.0.1,169.254.169.254,169.254.170.2,localhost,
          {{ AWSRegion }}.res,{{ AWSRegion }}.vpce.amazonaws.com,
          {{ AWSRegion }}.elb.amazonaws.com,s3.
          {{ AWSRegion }}.amazonaws.com,s3.dualstack.
          {{ AWSRegion }}.amazonaws.com,ec2.{{ AWSRegion }}.amazonaws.com,ec2.
          {{ AWSRegion }}.api.aws,ec2messages.{{ AWSRegion }}.amazonaws.com,ssm.
          {{ AWSRegion }}.amazonaws.com,ssmmessages.
          {{ AWSRegion }}.amazonaws.com,kms.
          {{ AWSRegion }}.amazonaws.com,secretsmanager.
          {{ AWSRegion }}.amazonaws.com,sqs.
          {{ AWSRegion }}.amazonaws.com,elasticloadbalancing.
          {{ AWSRegion }}.amazonaws.com,sns.{{ AWSRegion }}.amazonaws.com,logs.
          {{ AWSRegion }}.amazonaws.com,logs.
          {{ AWSRegion }}.api.aws,elasticfilesystem.
          {{ AWSRegion }}.amazonaws.com,fsx.{{ AWSRegion }}.amazonaws.com,dynamodb.
          {{ AWSRegion }}.amazonaws.com,api.ecr.
          {{ AWSRegion }}.amazonaws.com,.dkr.ecr.
          {{ AWSRegion }}.amazonaws.com,kinesis.{{ AWSRegion }}.amazonaws.com,.data-
          kinesis.{{ AWSRegion }}.amazonaws.com,.control-
          kinesis.{{ AWSRegion }}.amazonaws.com,events.
          {{ AWSRegion }}.amazonaws.com,cloudformation.
          {{ AWSRegion }}.amazonaws.com,sts.
          {{ AWSRegion }}.amazonaws.com,application-autoscaling.
          {{ AWSRegion }}.amazonaws.com,monitoring.{{ AWSRegion }}.amazonaws.com
          " > /etc/environment

```

- e. [コンポーネントを作成] を選択します。
4. Image Builder イメージレシピを作成します。
 - a. レシピの作成ページで、次のように入力します。

セクション	パラメータ	ユーザーエントリ
レシピ詳細	名前	Enter an appropriate name such as res-recipe-linux-x86.
	バージョン	Enter a version, typically starting with 1.0.0.
	説明	Add an optional description.
基本の イメージ	イメージの選択	Select managed images.
	OS	Amazon Linux
	イメージオリジン	Quick start (Amazon-managed)
	[イメージ名]	Amazon Linux 2 x86
	自動バージョンニングオプション	Use latest available OS version.
インスタンス設定	–	Keep everything in the default settings, and make sure Remove SSM agent after pipeline execution is not selected.
作業ディレクトリパス	作業ディレクトリパス	/root/bootstrap/requirements_dependencies

セクション	パラメータ	ユーザーエントリ
コンポーネント	コンポーネントの構築	以下を検索して選択します。 - Amazon マネージド: aws-cli-version-2-linux - Amazon マネージド: amazon-cloudwatch-agent-linux - 所有: 以前に作成された Amazon EC2 コンポーネント。フィールドに AWS アカウント ID と最新の AWS リージョン を入力します。
	テストコンポーネント	Search for and select: Amazon マネージド: simple-boot-test-linux

b. [レシピを作成する] を選択します。

5. Image Builder インフラストラクチャ設定を作成します。

- 「保存されたリソース」で、「インフラストラクチャ設定」を選択します。
- インフラストラクチャー構成の作成 を選択します。
- インフラストラクチャ設定の作成ページで、次のように入力します。

セクション	パラメータ	ユーザーエントリ
全般	名前	Enter an appropriate name such as res-infra-linux-x86.
	説明	Add an optional description.

セクション	パラメータ	ユーザーエントリ
	IAM ロール	Select the IAM role created previously.
AWS インフラストラクチャ	インスタンスタイプ	Choose t3.medium.
	VPC、サブネット、セキュリティグループ	Select an option that permits internet access and access to the Amazon S3 bucket. If you need to create a security group, you can create one from the Amazon EC2 console with the following inputs: • VPC: インフラストラクチャ設定に使用されているのと同じ VPC を選択します。この VPC にはインターネットアクセスが必要です。 • インバウンドルール : • タイプ: SSH • [Source]: Custom • CIDR ブロック: 0.0.0.0/0

d. インフラストラクチャ構成の作成 を選択します。

6. 新しい EC2 Image Builder パイプラインを作成します。

a. 「イメージパイプライン」に移動し、「イメージパイプラインの作成」を選択します。

b. パイプラインの詳細を指定ページで、次のように入力し、次へを選択します。

- パイプライン名とオプションの説明

- ビルドスケジュールで、スケジュールを設定するか、AMI ベーキングプロセスを手動で開始する場合は手動を選択します。
 - c. レシピの選択ページで、既存のレシピを使用を選択し、前に作成したレシピ名を入力します。[次へ] を選択します。
 - d. 画像プロセスの定義ページで、デフォルトのワークフローを選択し、次へを選択します。
 - e. 「インフラストラクチャ設定の定義」ページで、「既存のインフラストラクチャ設定を使用する」を選択し、以前に作成したインフラストラクチャ設定の名前を入力します。[次へ] を選択します。
 - f. ディストリビューション設定の定義ページで、選択について次の点を考慮してください。
 - RES がそこからインフラストラクチャホストインスタンスを適切に起動できるように、出カイメージはデプロイされた RES 環境と同じリージョンに存在する必要があります。サービスのデフォルトを使用すると、EC2 Image Builder サービスが使用されているリージョンに出カイメージが作成されます。
 - RES を複数のリージョンにデプロイする場合は、新しいディストリビューション設定を作成し、そこにリージョンを追加できます。
 - g. 選択内容を確認し、パイプラインの作成を選択します。
7. EC2 Image Builder パイプラインを実行します。
- a. イメージパイプラインから、作成したパイプラインを見つけて選択します。
 - b. アクション を選択し、パイプラインの実行 を選択します。

パイプラインは、AMI イメージの作成に約 45 分から 1 時間かかる場合があります。

8. 生成された AMI の AMI ID を書き留め、 の InfrastructureHostAMI パラメータの入力として使用します [the section called “ステップ 1: 製品を起動する”](#)。

VPC エンドポイントの設定

RES をデプロイして仮想デスクトップを起動するには、プライベートサブネットへのアクセス AWS のサービス が必要です。必要なアクセスを提供するように VPC エンドポイントを設定する必要があります。また、エンドポイントごとにこれらのステップを繰り返す必要があります。

1. エンドポイントが以前に設定されていない場合は、[「インターフェイス VPC エンドポイント AWS のサービス を使用して にアクセスする」](#)に記載されている手順に従ってください。
2. 2 つのアベイラビリティーゾーンのそれぞれで 1 つのプライベートサブネットを選択します。

AWS のサービス	サービス名
Application Auto Scaling	com.amazonaws.region.application-autoscaling
AWS CloudFormation	com.amazonaws.region.cloudformation
Amazon CloudWatch	com.amazonaws.region.monitoring
Amazon CloudWatch Logs	com.amazonaws.region.logs
Amazon DynamoDB	com.amazonaws. <i>region</i> .dynamodb (ゲートウェイエンドポイントが必要)
Amazon EC2	com.amazonaws.region.ec2
Amazon ECR	com.amazonaws.region.ecr.api
	com.amazonaws.region.ecr.dkr
Amazon Elastic File System	com.amazonaws.region.elasticfilesystem
エラスティックロードバランシング	com.amazonaws.region.elasticloadbalancing
Amazon EventBridge	com.amazonaws.region.events
Amazon FSx	com.amazonaws.region.fsx
AWS Key Management Service	com.amazonaws.region.kms
Amazon Kinesis Data Streams	com.amazonaws.region.kinesis-streams
Amazon S3	com.amazonaws. <i>region</i> .s3 (RES でデフォルトで作成されるゲートウェイエンドポイントが必要です)。
AWS Secrets Manager	com.amazonaws. <i>region</i> .secretsmanager
Amazon SES	com.amazonaws. <i>region</i> .email-smtp (次のアベイラビリティゾーンではサポートされていません: use-1-az2、use1-az3、use1-az5、usw1-az2、usw2-az4、apne2-az4、cac1-az3、cac1-az4)

AWS のサービス	サービス名
AWS Security Token Service	com.amazonaws. <i>region</i> .sts
Amazon SNS	com.amazonaws. <i>region</i> .sns
Amazon SQS	com.amazonaws. <i>region</i> .sqs
AWS Systems Manager	com.amazonaws. <i>region</i> .ec2messages
	com.amazonaws. <i>region</i> .ssm
	com.amazonaws. <i>region</i> .ssmmessages

VPC エンドポイントのないサービスに接続する

VPC エンドポイントをサポートしていないサービスと統合するには、VPC のパブリックサブネットにプロキシサーバーを設定できます。ID プロバイダーとして AWS Identity Center を使用して、Research and Engineering Studio デプロイに必要な最小限のアクセス権を持つプロキシサーバーを作成するには、次の手順に従います。

- RES デプロイに使用する VPC のパブリックサブネットで Linux インスタンスを起動します。
 - Linux ファミリー – Amazon Linux 2 または Amazon Linux 3
 - アーキテクチャ – x86
 - インスタンスタイプ – t2.micro 以上
 - セキュリティグループ – 0.0.0.0/0 からのポート 3128 での TCP
- インスタンスに接続してプロキシサーバーを設定します。
 - http 接続を開きます。
 - 関連するすべてのサブネットから次のドメインへの接続を許可します。
 - .amazonaws.com (汎用 AWS サービスの場合)
 - .amazoncognito.com (Amazon Cognito の場合)
 - .awsapps.com (アイデンティティセンター用)
 - .signin.aws (アイデンティティセンター用)
 - .amazonaws-us-gov.com (Gov Cloud の場合)

- c. 他のすべての接続を拒否します。
 - d. プロキシサーバーをアクティブ化して起動します。
 - e. プロキシサーバーがリッスンする PORT を書き留めます。
3. プロキシサーバーへのアクセスを許可するようにルートテーブルを設定します。
 - a. VPC コンソールに移動し、インフラストラクチャホストと VDI ホストに使用するサブネットのルートテーブルを特定します。
 - b. ルートテーブルを編集して、すべての着信接続が前のステップで作成したプロキシサーバーインスタンスに移動できるようにします。
 - c. これは、インフラストラクチャ/VDIs に使用するすべてのサブネット (インターネットアクセスなし) のルートテーブルに対して行います。
4. プロキシサーバー EC2 インスタンスのセキュリティグループを変更し、プロキシサーバーがリッスンしている PORT でインバウンド TCP 接続が許可されていることを確認します。

プライベート VPC デプロイパラメータを設定する

では[the section called “ステップ 1: 製品を起動する”](#)、AWS CloudFormation テンプレートに特定のパラメータを入力することが期待されます。設定したプライベート VPC に正常にデプロイするには、次のパラメータを必ず設定してください。

パラメータ	Input
InfrastructureHostAMI	Use the infrastructure AMI ID created in the section called “Amazon マシンイメージ (AMIs) を準備する” .
IsLoadBalancerInternetFacing	Set to false.
LoadBalancerSubnets	Choose private subnets without internet access.
InfrastructureHostSubnets	Choose private subnets without internet access.
VdiSubnets	Choose private subnets without internet access.
ClientIP	You can choose your VPC CIDR to allow access for all VPC IP addresses.

外部リソースを作成する

この CloudFormation スタックは、ネットワーク、ストレージ、アクティブディレクトリ、ドメイン証明書 (PortalDomainName が指定されている場合) を作成します。製品をデプロイするには、これらの外部リソースが必要です。

デプロイ前に [recipes テンプレートをダウンロードできます](#)。

デプロイ時間：約 40～90 分

1. にサインイン AWS Management Console し、<https://console.aws.amazon.com/cloudformation> で AWS CloudFormation コンソールを開きます。

Note

管理者アカウントにいることを確認します。

2. コンソールで[テンプレートを起動](#)します。

AWS GovCloud (米国西部) リージョンにデプロイする場合は、GovCloud パーティションアカウントで[テンプレートを起動](#)します。

3. テンプレートパラメータを入力します。

パラメータ	デフォルト	説明
DomainName	corp.res.com	アクティブディレクトリに使用されるドメイン。デフォルト値は、ブートストラップユーザーを設定する LDIF ファイルで提供されます。デフォルトユーザーを使用する場合は、値をデフォルトのままにします。値を変更するには、を更新して別の LDIF ファイルを指定します。これは、アクティブディレクトリに使用されるドメイ

パラメータ	デフォルト	説明
		ンと一致する必要はありません。
SubDomain (GovCloud のみ)		このパラメータは商用リージョンではオプションですが、GovCloud リージョンでは必須です。 SubDomain を指定すると、パラメータには指定された DomainName のプレフィックスが付けられます。指定された Active Directory ドメイン名はサブドメインになります。

パラメータ	デフォルト	説明
AdminPassword		アクティブディレクトリ管理者のパスワード (ユーザー名 Admin)。このユーザーは、初期ブートストラップフェーズのアクティブディレクトリに作成され、その後は使用されません。 重要： このフィールドの形式は、(1) プレーンテキストのパスワード、または (2) キーと値のペアとしてフォーマットされた AWS シークレットの ARN のいずれかです <code>{"password": "somepassword"}</code>。 注： このユーザーのパスワードは、Active Directory のパスワードの複雑さの要件を満たしている必要があります。

パラメータ	デフォルト	説明
ServiceAccountPassword		サービスアカウントの作成に使用されるパスワード (ReadOnlyUser)。このアカウントは同期に使用されます。 重要： このフィールドの形式は、(1) プレーンテキストのパスワード、または (2) キーと値のペアとしてフォーマットされた AWS シークレットの ARN のいずれかです <code>{"password": "somepassword"}</code> 。 注： このユーザーのパスワードは、Active Directory のパスワードの複雑さの要件を満たしている必要があります。
キーペア		SSH クライアントを使用して管理インスタンスを接続します。 注: AWS Systems Manager Session Manager を使用してインスタンスに接続することもできます。

パラメータ	デフォルト	説明
LDIFS3Path	aws-hpc-recipes/main/recipes/res/res_demo_env/assets/res.ldif	Active Directory セットアップのブートストラップフェーズ中にインポートされた LDIF ファイルへの Amazon S3 パス。詳細については、「LDIF サポート」を参照してください。パラメータには、アクティブディレクトリに多数のユーザーを作成するファイルが事前に入力されています。 ファイルを表示するには、GitHub で利用可能な res.ldif ファイルを参照してください。
ClientIpCidr		サイトにアクセスする IP アドレス。例えば、IP アドレスを選択し、[IPADDRESS]/32 を使用してホストからのアクセスのみを許可できます。このデプロイ後に更新できます。
ClientPrefixList		プレフィックスリストを入力して、アクティブディレクトリ管理ノードへのアクセスを提供します。マネージドプレフィックスリストの作成については、「カスタマーマネージドプレフィックスリストの操作」を参照してください。

パラメータ	デフォルト	説明
EnvironmentName	res- <i>[environment name]</i>	PortalDomainName が指定されている場合、このパラメータを使用して生成されたシークレットにタグを追加し、環境内で使用できます。これは、RES スタックの作成時に使用された EnvironmentName パラメータと一致する必要があります。アカウントに複数の環境をデプロイする場合、これは一意である必要があります。
PortalDomainName		GovCloud デプロイの場合は、このパラメータを入力しないでください。証明書とシークレットは、前提条件に従って手動で作成されました。 アカウントの Amazon Route 53 のドメイン名。これを指定すると、パブリック証明書とキーファイルが生成され、 にアップロードされます AWS Secrets Manager。 独自のドメインと証明書がある場合は、このパラメータとを空白のままに EnvironmentName することができます。

- 機能のすべてのチェックボックスを確認し、スタックの作成を選択します。

ステップ 1: 製品を起動する

このセクションのstep-by-stepの手順に従って、製品を設定してアカウントにデプロイします。

デプロイ時間：約 60 分

この製品の [CloudFormation テンプレート](#)は、デプロイする前にダウンロードできます。

AWS GovCloud (米国西部) でデプロイする場合は、この[テンプレート](#)を使用します。

res-stack - このテンプレートを使用して、製品と関連するすべてのコンポーネントを起動します。デフォルト設定では、RES メインスタックと認証、フロントエンド、バックエンドリソースがデプロイされます。

Note

AWS CloudFormation リソースは AWS Cloud Development Kit (AWS CDK) (AWS CDK) コンストラクトから作成されます。

AWS CloudFormation テンプレートは、の AWS に Research and Engineering Studio をデプロイします AWS クラウド。スタックを起動する前に、[前提条件](#)を満たす必要があります。

1. にサインイン AWS Management Console し、<https://console.aws.amazon.com/cloudformation> で AWS CloudFormation コンソールを開きます。
2. [テンプレート](#) を起動します。

AWS GovCloud (米国西部) にデプロイするには、この[テンプレート](#)を起動します。

3. テンプレートはデフォルトで米国東部 (バージニア北部) リージョンで起動します。別の でソリューションを起動するには AWS リージョン、コンソールナビゲーションバーのリージョンセレクターを使用します。

Note

この製品では Amazon Cognito サービスを使用していますが、現在すべての で利用できるわけではありません AWS リージョン。この製品は、Amazon Cognito AWS リージョン が利用可能な で起動する必要があります。リージョン別の最新の可用性については、「[AWS リージョン Services List](#)」を参照してください。

4. パラメータで、この製品テンプレートのパラメータを確認し、必要に応じて変更します。自動外部リソースをデプロイした場合、これらのパラメータは外部リソーススタックの出力タブにあります。

パラメータ	デフォルト	説明
EnvironmentName	<i>#res-demo#</i>	res- で始まり、11 文字以下の RES 環境に与えられる一意の名前。
AdministratorEmail		製品のセットアップを完了したユーザーの E メールアドレス。さらに、このユーザーは、Active Directory シングルサインオン統合に障害が発生した場合に、ブレークグラスユーザーとして機能します。
InfrastructureHostAMI	ami-#####	(オプション) すべてのインフラストラクチャホストに使用するカスタム AMI ID を指定できます。現在サポートされているベース OS は Amazon Linux 2 です。詳細については、「 RES 対応 AMIs を設定する 」を参照してください。
SSHKeyPair		インフラストラクチャホストへの接続に使用されるキーペア。
ClientIP	<i>x.x.x.0/24</i> または <i>x.x.x.0/32</i>	システムへの接続を制限する IP アドレスフィルター。デプロイ後に ClientIpCidr を更新できます。

パラメータ	デフォルト	説明
ClientPrefixList		(オプション) 踏み台ホストへのウェブ UI と SSH への直接アクセスが許可されている IPs のマネージドプレフィックスリストを指定します。
IAMPermissionBoundary		(オプション) RES で作成されたすべてのロールにアクセス許可の境界としてアタッチされる管理ポリシー ARN を指定できます。詳細については、「 カスタムアクセス許可の境界の設定 」を参照してください。
VpcId		インスタンスが起動する VPC の IP。
IsLoadBalancerInternetFacing		インターネット向けロードバランサーをデプロイするには true を選択します (ロードバランサーにはパブリックサブネットが必要です)。制限されたインターネットアクセスを必要とするデプロイの場合は、false を選択します。

パラメータ	デフォルト	説明
LoadBalancerSubnets		ロードバランサーが起動する異なるアベイラビリティーゾーンで、少なくとも2つのサブネットを選択します。制限されたインターネットアクセスを必要とするデプロイの場合は、プライベートサブネットを選択します。インターネットアクセスが必要なデプロイの場合は、パブリックサブネットを選択します。外部ネットワークスタックによって3つ以上作成された場合は、作成されたすべてのを選択します。
InfrastructureHostSubnets		インフラストラクチャホストが起動する異なるアベイラビリティーゾーンで、少なくとも2つのプライベートサブネットを選択します。外部ネットワークスタックによって3つ以上作成された場合は、作成されたすべてのを選択します。
VdiSubnets		VDI インスタンスが起動する異なるアベイラビリティーゾーンで、少なくとも2つのプライベートサブネットを選択します。外部ネットワークスタックによって3つ以上作成された場合は、作成されたすべてのを選択します。

パラメータ	デフォルト	説明
ActiveDirectoryName	<i>corp.res.com</i>	アクティブディレクトリのドメイン。ポータルドメイン名と一致する必要はありません。
ADShortName	<i>corp</i>	アクティブディレクトリの短縮名。これは NetBIOS 名とも呼ばれます。
LDAP ベース	<i>DC=corp,DC=res,DC=com</i>	LDAP 階層内のベースへの LDAP パス。
LDAPConnectionURI		アクティブディレクトリのホストサーバーが到達できる単一の ldap:// パス。デフォルトの AD ドメインで自動外部リソースをデプロイした場合は、ldap://corp.res.com を使用できます。
ServiceAccountUserName	ServiceAccount	AD への接続に使用されるサービスアカウントのユーザー名。このアカウントには、ComputersOU 内にコンピュータを作成するためのアクセス権が必要です。
ServiceAccountPasswordSecretArn		ServiceAccount のプレーンテキストパスワードを含むシークレット ARN を指定します。
UsersOU		同期するユーザーの AD 内の組織単位。

パラメータ	デフォルト	説明
GroupsOU		同期するグループの AD 内の組織単位。
SudoersOU		グローバル sudoers の AD 内の組織単位。
SudoersGroupName	RESAdministrators	インストール時にインスタンスへの sudoer アクセスと RES への管理者アクセスを持つすべてのユーザーを含むグループ名。
ComputersOU		インスタンスが参加する AD 内の組織単位。
DomainTLSCertificateSecretARN		(オプション) AD への TLS 通信を有効にするドメイン TLS 証明書シークレット ARN を指定します。
EnableLdapIDMapping		UID 番号と GID 番号が SSSD によって生成されるか、AD によって提供される番号を使用するかを決定します。SSSD が生成した UID と GID を使用するには true、AD が提供する UID と GID を使用するには False に設定します。ほとんどの場合、このパラメータは True に設定する必要があります。

パラメータ	デフォルト	説明
DisableADJoin	False	Linux ホストがディレクトリドメインに参加しないようにするには、を True に変更します。それ以外の場合は、デフォルト設定の False のままにします。
ServiceAccountUserDN		Directory でサービスアカウントユーザーの識別名 (DN) を指定します。
SharedHomeFilesystemID		Linux VDI ホストの共有ホームファイルシステムに使用する EFS ID。
CustomDomainNameforWebApp		(オプション) システムのウェブ部分へのリンクを提供するためにウェブポータルで使われるサブドメイン。
CustomDomainNameforVDI		(オプション) システムの VDI 部分へのリンクを提供するためにウェブポータルで使われるサブドメイン。

パラメータ	デフォルト	説明
ACMCertificateARNforWebApp		(オプション) デフォルト設定を使用する場合、製品はドメイン amazonaws.com でウェブアプリケーションをホストします。ドメインで製品サービスをホストできます。自動外部リソースをデプロイした場合、これは自動的に生成され、情報は res-bi スタックの出力にあります。ウェブアプリケーションの証明書を生成する必要がある場合は、「」を参照してください 設定ガイド 。
CertificateSecretARNforVDI		(オプション) この ARN シークレットは、ウェブポータルパブリック証明書のパブリック証明書を保存します。自動外部リソースにポータルドメイン名を設定すると、res-bi スタックの出力タブにこの値が表示されます。
PrivateKeySecretARNforVDI		(オプション) この ARN シークレットは、ウェブポータル証明書のプライベートキーを保存します。自動外部リソースにポータルドメイン名を設定すると、res-bi スタックの出力タブにこの値が表示されます。

5. [スタックの作成] を選択してスタックをデプロイします。

スタックのステータスは、AWS CloudFormation コンソールの Status 列で表示できます。約 60 分で CREATE_COMPLETE ステータスが表示されます。

ステップ 2: 初めてサインインする

製品スタックがアカウントにデプロイされると、認証情報が記載された E メールが送信されます。URL を使用してアカウントにサインインし、他のユーザーのワークスペースを設定します。

初めてサインインしたら、ウェブポータルで SSO プロバイダーに接続するように設定することができます。デプロイ後の設定情報については、「」を参照してください[設定ガイド](#)。clusteradmin はブレイクグラスアカウントです。これを使用してプロジェクトを作成し、それらのプロジェクトにユーザーまたはグループのメンバーシップを割り当てることができます。ソフトウェアスタックを割り当てたり、それ自体にデスクトップをデプロイしたりすることはできません。

製品を更新する

Research and Engineering Studio (RES) には、バージョン更新がメジャーかマイナーかに応じて、製品を更新する 2 つの方法があります。

RES は日付ベースのバージョニングスキームを使用します。メジャーリリースでは年と月が使用され、マイナーリリースでは必要に応じてシーケンス番号が追加されます。たとえば、バージョン 2024.01 はメジャーリリースとして 2024 年 1 月にリリースされました。バージョン 2024.01.01 はそのバージョンのマイナーリリース更新でした。

トピック

- [メジャーバージョンの更新](#)
- [マイナーバージョンの更新](#)

メジャーバージョンの更新

Research and Engineering Studio は、スナップショットを使用して、環境設定を失うことなく、以前の RES 環境から最新の環境への移行をサポートします。このプロセスを使用して、ユーザーをオンボーディングする前に環境の更新をテストおよび検証することもできます。

環境を最新バージョンの RES で更新するには：

1. 現在の環境のスナップショットを作成します。「[the section called “スナップショットを作成する”](#)」を参照してください。
2. 新しいバージョンで RES を再デプロイします。「[the section called “ステップ 1: 製品を起動する”](#)」を参照してください。
3. 更新された環境にスナップショットを適用します。「[the section called “スナップショットを適用する”](#)」を参照してください。
4. 新しい環境に正常に移行されたすべてのデータを検証します。

マイナーバージョンの更新

RES のマイナーバージョン更新の場合、新しいインストールは必要ありません。テンプレートを更新することで、既存の RES スタックを更新できます AWS CloudFormation。更新をデプロイ AWS CloudFormation する前に、で現在の RES 環境のバージョンを確認してください。テンプレートの先頭にバージョン番号があります。

例: "Description": "RES_2024.1"

マイナーバージョンを更新するには：

1. 最新の AWS CloudFormation テンプレートを にダウンロードします [the section called “ステップ 1: 製品を起動する”](#)。
2. <https://console.aws.amazon.com/cloudformation> で AWS CloudFormation コンソールを開きます。
3. スタックから、プライマリスタックを検索して選択します。として表示されます *<stack-name>*。
4. [更新] を選択します。
5. 現在のテンプレートを置き換える を選択します。
6. [テンプレートソース] で、[テンプレートファイルのアップロード] を選択します。
7. ファイルの選択を選択し、ダウンロードしたテンプレートをアップロードします。
8. スタックの詳細を指定する で、次へ を選択します。パラメータを更新する必要はありません。
9. スタックオプションを設定する で、次へ を選択します。
10. レビュー *<stack-name>* で、送信を選択します。

製品のアンインストール

AWS 製品上の Research and Engineering Studio は、 から、 AWS Management Console または を使用してアンインストールできます AWS Command Line Interface。この製品によって作成された Amazon Simple Storage Service (Amazon S3) バケットを手動で削除する必要があります。この製品は、保持するデータを保存している場合、<EnvironmentName>-shared-storage-security-group を自動的に削除しません。

の使用 AWS Management Console

1. [AWS CloudFormation コンソール](#) にサインインします。
2. スタックページで、この製品のインストールスタックを選択します。
3. [削除] を選択します。

の使用 AWS Command Line Interface

AWS Command Line Interface (AWS CLI) が環境で使用可能かどうかを判断します。インストール手順については、「AWS CLI ユーザーガイド」の「[AWS Command Line Interfaceとは](#)」を参照してください。AWS CLI が使用可能で、製品がデプロイされたリージョンの管理者アカウントに設定されていることを確認したら、次のコマンドを実行します。

```
$ aws cloudformation delete-stack --stack-name  
<RES-stack-name>
```

shared-storage-security-group の削除

Warning

製品は、意図しないデータ損失を防ぐために、このファイルシステムをデフォルトで保持します。セキュリティグループと関連するファイルシステムを削除すると、それらのシステム内に保持されているデータはすべて完全に削除されます。データをバックアップするか、新しいセキュリティグループにデータを再割り当てすることをお勧めします。

1. にサインイン AWS Management Console し、<https://console.aws.amazon.com/efs/> で Amazon EFS コンソールを開きます。
2. `<RES-stack-name>-shared-storage-security-group` に関連付けられているすべてのファイルシステムを削除します。または、これらのファイルシステムを別のセキュリティグループに再割り当てして、データを維持することもできます。
3. にサインイン AWS Management Console し、<https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
4. `<RES-stack-name>-shared-storage-security-group` を削除します。

Amazon S3 バケットの削除

この製品は、誤ってデータが失われないように AWS CloudFormation スタックを削除する場合に、製品によって作成された Amazon S3 バケット (オプトインリージョンにデプロイする場合) を保持するように設定されています。製品をアンインストールした後、データを保持する必要がない場合は、この S3 バケットを手動で削除できます。Amazon S3 バケットを削除するには、次の手順に従います。

1. にサインイン AWS Management Console し、<https://console.aws.amazon.com/s3/> で Amazon S3 コンソールを開きます。
2. ナビゲーションペインで [バケット] を選択します。
3. `stack-name` S3 バケットを見つけます。
4. 各 Amazon S3 バケットを選択し、空を選択します。各バケットを空にする必要があります。
5. S3 バケットを選択し、続いて [削除] を選択します。

を使用して S3 バケットを削除するには AWS CLI、次のコマンドを実行します。

```
$ aws s3 rb s3://<bucket-name> --force
```

Note

--force コマンドは、その内容のバケットを空にします。

設定ガイド

この設定ガイドでは、AWS 製品の Research and Engineering Studio をさらにカスタマイズして統合する方法に関するデプロイ後の手順について説明します。

トピック

- [ユーザーとグループの管理](#)
- [サブドメインの作成](#)
- [ACM 証明書を作成する](#)
- [Amazon CloudWatch Logs](#)
- [カスタムアクセス許可の境界の設定](#)
- [RES 対応 AMIs を設定する](#)

ユーザーとグループの管理

Research and Engineering Studio は、SAML 2.0 準拠の任意の ID プロバイダーを使用できます。外部リソースを使用して RES をデプロイした場合、または IAM Identity Center を使用する予定の場合は、「」を参照してください[IAM Identity Center でのシングルサインオン \(SSO\) の設定](#)。独自の SAML 2.0 準拠の ID プロバイダーがある場合は、「」を参照してください[シングルサインオン \(SSO\) 用の ID プロバイダーの設定](#)。

トピック

- [IAM Identity Center でのシングルサインオン \(SSO\) の設定](#)
- [シングルサインオン \(SSO\) 用の ID プロバイダーの設定](#)
- [ユーザーのパスワードの設定](#)

IAM Identity Center でのシングルサインオン (SSO) の設定

マネージド Active Directory に接続しているアイデンティティセンターがまだない場合は、から始めます[ステップ 1: アイデンティティセンターを設定する](#)。マネージド Active Directory に接続されたアイデンティティセンターが既にある場合は、から始めます[ステップ 2: アイデンティティセンターに接続する](#)。

 Note

AWS GovCloud (米国西部) リージョンにデプロイする場合は、Research and Engineering Studio を AWS GovCloud (US) デプロイしたパーティションアカウントに SSO を設定します。

ステップ 1: アイデンティティセンターを設定する

IAM アイデンティティセンターを有効にする

1. [AWS Identity and Access Management コンソール](#) にサインインします。
2. アイデンティティセンターを開きます。
3. [Enable] (有効化) を選択します。
4. Enable with AWS Organizationsを選択します。
5. [Continue] (続行) をクリックします。

 Note

マネージド Active Directory があるリージョンと同じリージョンにいることを確認します。

IAM Identity Center をマネージド Active Directory に接続する

IAM Identity Center を有効にしたら、以下の推奨セットアップステップを完了します。

1. ナビゲーションペインで、設定を選択します。
2. ID ソースで、アクションを選択し、ID ソースの変更を選択します。
3. 既存のディレクトリで、ディレクトリを選択します。
4. [次へ] を選択してください。
5. 変更を確認し、確認ボックスに **ACCEPT** と入力します。
6. [アイデンティティソースを変更] を選択します。

ユーザーとグループの ID センターへの同期

で行われた変更 [IAM Identity Center をマネージド Active Directory に接続する](#) が完了すると、緑色の確認バナーが表示されます。

1. 確認バナーで、ガイド付きセットアップの開始を選択します。
2. 属性マッピングの設定 から、次へ を選択します。
3. ユーザー セクションで、同期するユーザーを入力します。
4. [追加] を選択します。
5. [次へ] を選択してください。
6. 変更を確認し、設定の保存を選択します。
7. 同期プロセスには数分かかる場合があります。同期していないユーザーに関する警告メッセージが表示された場合は、同期を再開を選択します。

ユーザーの有効化

1. メニューから、ユーザーを選択します。
2. アクセスを有効にするユーザー (複数可) を選択します。
3. ユーザーアクセスを有効にするを選択します。

ステップ 2: アイデンティティセンターに接続する

IAM Identity Center でのアプリケーションのセットアップ

1. [IAM Identity Center コンソール](#)を開きます。
2. アプリケーションを選択します。
3. アプリケーションの追加 を選択します。
4. セットアップ設定で、セットアップするアプリケーションがあるを選択します。
5. アプリケーションタイプで、SAML 2.0 を選択します。
6. [次へ] を選択してください。
7. 使用する表示名と説明を入力します。
8. IAM Identity Center メタデータで、IAM Identity Center SAML メタデータファイルのリンクをコピーします。これは、RES ポータルで IAM Identity Center を設定するときに必要になります。

9. アプリケーションプロパティで、アプリケーション開始 URL を入力します。例えば、<your-portal-domain>/sso。
10. Application ACS URL で、RES ポータルからリダイレクト URL を入力します。これを見つけるには：
 - a. 環境管理で、全般設定を選択します。
 - b. [Identity provider] タブを選択します。
 - c. Single Sign-On の下に、SAML リダイレクト URL が表示されます。
11. Application SAML audience で、Amazon Cognito URN を入力します。

URL を作成するには：

- a. RES ポータルから、全般設定を開きます。
- b. ID プロバイダータブで、ユーザープール ID を見つけます。
- c. ユーザープール ID をこの文字列に追加します。

```
urn:amazon:cognito:sp:<user_pool_id>
```

12. Amazon Cognito URN を入力したら、送信を選択します。

アプリケーションの属性マッピングの設定

1. アイデンティティセンターから、作成したアプリケーションの詳細を開きます。
2. アクションを選択し、属性マッピングの編集を選択します。
3. [件名] に `${user:email}` と入力します。
4. フォーマット で、emailAddress を選択します。
5. 新しい属性マッピングの追加を選択します。
6. アプリケーションの User 属性に「email」と入力します。
7. 「IAM Identity Center のこの文字列値またはユーザー属性にマップする」に「」と入力します `${user:email}`。
8. Format に「unspecified」と入力します。
9. [変更を保存] を選択します。

IAM Identity Center でのアプリケーションへのユーザーの追加

1. アイデンティティセンターから、作成したアプリケーションの割り当て済みユーザーを開き、ユーザーの割り当てを選択します。
2. アプリケーションアクセスを割り当てるユーザーを選択します。
3. ユーザーの割り当てを選択します。

RES 環境内での IAM Identity Center のセットアップ

1. Research and Engineering Studio 環境から、環境管理で全般設定を開きます。
2. ID プロバイダータブを開きます。
3. シングルサインオンで、編集 (ステータスの横) を選択します。
4. フォームに以下の情報を入力します。
 - a. SAML を選択します。
 - b. プロバイダー名に、わかりやすい名前を入力します。
 - c. Enter metadata document endpoint URL を選択します。
 - d. 中にコピーした URL を入力します [IAM Identity Center でのアプリケーションのセットアップ](#)。
 - e. Provider email 属性に「email」と入力します。
 - f. [Submit] を選択してください。
5. ページを更新し、ステータスが有効と表示されることを確認します。

シングルサインオン (SSO) 用の ID プロバイダーの設定

Research and Engineering Studio は、任意の SAML 2.0 ID プロバイダーと統合して、RES ポータルへのユーザーアクセスを認証します。これらのステップでは、選択した SAML 2.0 ID プロバイダーと統合する手順を示します。IAM Identity Center を使用する場合は、「」を参照してください [the section called “IAM アイデンティティセンターでの SSO の設定”](#)。

Note

ユーザーの E メールは、IDP SAML アサーションと Active Directory で一致する必要があります。ID プロバイダーを Active Directory に接続し、定期的にユーザーを同期する必要があります。

トピック

- [ID プロバイダーを設定する](#)
- [ID プロバイダーを使用するように RES を設定する](#)
- [非本番環境での ID プロバイダーの設定](#)
- [SAML IdP の問題のデバッグ](#)

ID プロバイダーを設定する

このセクションでは、RES Amazon Cognito ユーザープールからの情報を使用して ID プロバイダーを設定する手順について説明します。

1. RES は、RES ポータルとプロジェクトへのアクセスが許可されているユーザー ID を持つ AD (AWS マネージド AD またはセルフプロビジョニング AD) があることを前提としています。AD を ID サービスプロバイダーに接続し、ユーザー ID を同期します。AD を接続し、ユーザー ID を同期する方法については、ID プロバイダーのドキュメントを参照してください。例えば、「AWS IAM Identity Center ユーザーガイド」の「[ID ソースとしての Active Directory の使用](#)」を参照してください。
2. ID プロバイダー (IdP) で RES の SAML 2.0 アプリケーションを設定します。この設定には、次のパラメータが必要です。
 - SAML リダイレクト URL — IdP が SAML 2.0 レスポンスをサービスプロバイダーに送信するために使用する URL。

Note

IdP によっては、SAML リダイレクト URL の名前が異なる場合があります。

- アプリケーション URL
- アサーションコンシューマーサービス (ACS) URL
- ACS POST バインディング URL

URL を取得するには

1. 管理者または clusteradmin として RES にサインインします。
2. 環境管理 ⇒ 一般設定 ⇒ ID プロバイダーに移動します。
3. SAML リダイレクト URL を選択します。

- SAML オーディエンス URI — サービスプロバイダー側の SAML オーディエンスエンティティの一意の ID。

 Note

IdP によっては、SAML オーディエンス URI の名前が異なる場合があります。

- ClientID
- アプリケーション SAML 対象者
- SP エンティティ ID

入力を次の形式で指定します。

```
urn:amazon:cognito:sp:user-pool-id
```

SAML オーディエンス URI を検索するには

1. 管理者または clusteradmin として RES にサインインします。
 2. 環境管理 ⇒ 一般設定 ⇒ ID プロバイダーに移動します。
 3. ユーザープール ID を選択します。
3. RES に投稿される SAML アサーションには、次のフィールド/クレームがユーザーの E メールアドレスに設定されている必要があります。
- SAML Subject または NameID
 - SAML E メール
4. IdP は、設定に基づいて SAML アサーションにフィールド/クレームを追加します。RES にはこれらのフィールドが必要です。ほとんどのプロバイダーは、デフォルトでこれらのフィールドを自動的に入力します。設定する必要がある場合は、次のフィールド入力と値を参照してください。
- AudienceRestriction — を に設定します `urn:amazon:cognito:sp:user-pool-id`。 *user-pool-id* を Amazon Cognito ユーザープールの ID に置き換えます。

```
<saml:AudienceRestriction>  
  <saml:Audience> urn:amazon:cognito:sp:user-pool-id
```

```
</saml:AudienceRestriction>
```

- レスポンス — InResponseToを に設定しますhttps://*user-pool-domain*/saml2/idpresponse。 *user-pool-domain* を Amazon Cognito ユーザープールのドメイン名に置き換えます。

```
<saml2p:Response
  Destination="http://user-pool-domain/saml2/idpresponse"
  ID="id123"
  InResponseTo="_dd0a3436-bc64-4679-a0c2-cb4454f04184"
  IssueInstant="Date-time stamp"
  Version="2.0"
  xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

- SubjectConfirmationData — Recipient ユーザープールsaml2/idpresponseエンドポイントと元の SAML リクエスト ID InResponseToに設定します。

```
<saml2:SubjectConfirmationData
  InResponseTo="_dd0a3436-bc64-4679-a0c2-cb4454f04184"
  NotOnOrAfter="Date-time stamp"
  Recipient="https://user-pool-domain/saml2/idpresponse"/>
```

- AuthnStatement — 次のように を設定します。

```
<saml2:AuthnStatement AuthnInstant="2016-10-30T13:13:28.152TZ"
  SessionIndex="32413b2e54db89c764fb96ya2k"
  SessionNotOnOrAfter="2016-10-30T13:13:28">
  <saml2:SubjectLocality />
  <saml2:AuthnContext>

  <saml2:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:Password</saml2:AuthnContextClassRef>
  </saml2:AuthnContext>
</saml2:AuthnStatement>
```

5. SAML アプリケーションにログアウト URL フィールドがある場合は、 に設定します<*domain-url*>/saml2/logout。

ドメイン URL を取得するには

1. 管理者または clusteradmin として RES にサインインします。
 2. 環境管理 ⇒ 一般設定 ⇒ ID プロバイダーに移動します。
 3. ドメイン URL を選択します。
6. IdP が Amazon Cognito との信頼を確立するために署名証明書を受け入れる場合は、Amazon Cognito 署名証明書をダウンロードし、IdP にアップロードします。

署名証明書を取得するには

1. の開始方法のAmazon Cognito コンソールを開く[AWS Management Console](#)」
2. ユーザープールを選択します。ユーザープールは である必要があります `res-environment name-user-pool`。
3. [Sign-in experience] (サインインエクスペリエンス) タブを選択します。
4. フェデレーテッド ID プロバイダーのサインインセクションで、署名証明書の表示を選択します。

この証明書を使用して、Active Directory IDP をセットアップし、 を追加しrelying party trust、この証明書利用者に対して SAML サポートを有効にできます。

 Note

これは Keycloak と IDC には適用されません。

5. アプリケーションのセットアップが完了したら、SAML 2.0 アプリケーションメタデータ XML または URL をダウンロードします。次のセクションで使用します。

ID プロバイダーを使用するように RES を設定する

RES のシングルサインオン設定を完了するには

1. 管理者または clusteradmin として RES にサインインします。
2. 環境管理 ⇒ 一般設定 ⇒ ID プロバイダーに移動します。

3. Single Sign-On で、ステータスインジケータの横にある編集アイコンを選択して Single Sign-On Configuration ページを開きます。
 - a. ID プロバイダーで、SAML を選択します。
 - b. プロバイダー名には、ID プロバイダーの一意の名前を入力します。
-  **Note**

次の名前は使用できません。

 - Cognito
 - IdentityCenter
- c. メタデータドキュメントソースで、適切なオプションを選択し、メタデータ XML ドキュメントをアップロードするか、ID プロバイダーから URL を指定します。
 - d. プロバイダー E メール属性 に、テキスト値 を入力しますemail。
 - e. [Submit] を選択してください。
 4. 環境設定ページを再ロードします。設定が正しい場合、シングルサインオンが有効になります。

非本番環境での ID プロバイダーの設定

提供された[外部リソース](#)を使用して非本番環境の RES 環境を作成し、IAM Identity Center を ID プロバイダーとして設定した場合は、Okta などの別の ID プロバイダーを設定することをお勧めします。RES SSO 有効化フォームは、3 つの設定パラメータを要求します。

1. プロバイダー名 — 変更できません
2. メタデータドキュメントまたは URL — 変更可能
3. プロバイダー E メール属性 — 変更可能

メタデータドキュメントとプロバイダー E メール属性を変更するには、次の手順を実行します。

1. [Amazon Cognito コンソール](#)に移動します。
2. ナビゲーションから、ユーザープールを選択します。
3. ユーザープールを選択すると、ユーザープールの概要が表示されます。

4. サインインエクスペリエンスタブから、フェデレーテッド ID プロバイダーのサインインに移動し、設定された ID プロバイダーを開きます。
5. 通常、メタデータを変更し、属性マッピングを変更しないだけで済みます。属性マッピングを更新するには、編集 を選択します。メタデータドキュメントを更新するには、メタデータの置き換えを選択します。
6. 属性マッピングを編集した場合は、DynamoDB で<environment name>.cluster-settingsテーブルを更新する必要があります。
 - a. DynamoDB コンソールを開き、ナビゲーションからテーブルを選択します。
 - b. <environment name>.cluster-settings テーブルを検索して選択し、Actions メニューから Explore items を選択します。
 - c. スキャンまたはクエリ項目で、フィルターに移動し、次のパラメータを入力します。
 - 属性名 — key
 - 値 — identity-provider.cognito.sso_idp_provider_email_attribute
 - d. [Run] (実行) を選択します。
7. 返された項目 で文字列を検索identity-provider.cognito.sso_idp_provider_email_attributeし、編集 を選択して、Amazon Cognito の変更と一致するように文字列を変更します。

SAML IdP の問題のデバッグ

SAML トレーサー — Chrome ブラウザでこの拡張機能を使用して SAML リクエストを追跡し、SAML アサーション値を確認できます。詳細については、Chrome ウェブストアの「[SAML トレーサー](#)」を参照してください。

SAML 開発者ツール — OneLogin には、SAML エンコードされた値をデコードし、SAML アサーションの必須フィールドをチェックするために使用できるツールが用意されています。詳細については、OneLogin ウェブサイトの「[Base 64 Decode + Inflate](#)」を参照してください。

Amazon CloudWatch Logs — CloudWatch Logs で RES ログのエラーや警告を確認できます。ログは、 という名前のロググループにあります *res-environment-name*/cluster-manager。

Amazon Cognito ドキュメント — Amazon Cognito との SAML 統合の詳細については、「Amazon Amazon Cognito デベロッパーガイド」の「[ユーザープールへの SAML ID プロバイダーの追加](#)」を参照してください。

ユーザーのパスワードの設定

1. [AWS Directory Service コンソール](#)から、作成したスタックのディレクトリを選択します。
2. アクションメニューで、ユーザーパスワードのリセットを選択します。
3. ユーザーを選択し、新しいパスワードを入力します。
4. パスワードのリセットを選択します。

サブドメインの作成

カスタムドメインを使用している場合は、ポータルのウェブ部分と VDI 部分をサポートするようにサブドメインを設定する必要があります。

Note

AWS GovCloud (米国西部) リージョンにデプロイする場合は、ドメインパブリックホストゾーンをホストする商用パーティションアカウントでウェブアプリケーションと VDI サブドメインを設定します。

1. [Route 53 コンソール](#)を開きます。
2. 作成したドメインを検索し、レコードの作成を選択します。
3. レコード名として「web」と入力します。
4. レコードタイプとして CNAME を選択します。
5. Value には、最初の E メールで受け取ったリンクを入力します。
6. [レコードを作成] を選択します。
7. " のレコードを作成するには、NLB アドレスを取得します。
 - a. [AWS CloudFormation コンソール](#)を開きます。
 - b. <environment-name>-vdc を選択してください。
 - c. リソースを選択し、を開きます<environmentname>-vdc-external-nlb。
 - d. NLB から DNS 名をコピーします。

8. [Route 53 コンソール](#)を開きます。
9. ドメインを検索し、レコードの作成を選択します。
10. レコード名に「」と入力しますvdc。
11. [レコードタイプ] で、[CNAME] を選択します。
12. NLB の場合は、DNS を入力します。
13. [Create record] (レコードを作成) を選択します。

ACM 証明書を作成する

デフォルトでは、RES はドメイン amazonaws.com を使用してアプリケーションロードバランサーでウェブポータルをホストします。独自のドメインを使用するには、ユーザーが提供する、または AWS Certificate Manager (ACM) からリクエストされたパブリック SSL/TLS 証明書を設定する必要があります。ACM を使用する場合、クライアントとウェブサービスホスト間の SSL/TLS チャンネルを暗号化するためのパラメータとして指定する必要がある AWS リソース名を受け取ります。

Tip

外部リソースデモパッケージをデプロイする場合は、に外部リソーススタックをデプロイPortalDomainNameするときに、選択したドメインを に入力する必要があります[外部リソースを作成する](#)。

カスタムドメインの証明書を作成するには：

1. コンソールから [AWS Certificate Manager](#) を開き、パブリック証明書をリクエストします。AWS GovCloud (米国西部) にデプロイする場合は、GovCloud パーティションアカウントに証明書を作成します。
2. 「パブリック証明書をリクエストする」を選択し、「次へ」を選択します。
3. ドメイン名で、 *.PortalDomainNameと の両方の証明書をリクエストしますPortalDomainName。
4. 検証メソッドで、DNS 検証を選択します。
5. [リクエスト] を選択します。
6. Certificates リストから、リクエストされた証明書を開きます。各証明書のステータスは、検証保留中になります。

Note

証明書が表示されない場合は、リストを更新します。

7. 次のいずれかを行います：

- 商用デプロイ：

リクエストされた各証明書の証明書の詳細から、Route 53 でレコードを作成するを選択します。証明書のステータスは「発行済み」に変わります。

- GovCloud デプロイ：

AWS GovCloud (米国西部) にデプロイする場合は、CNAME キーと値をコピーします。商用パーティションアカウントから、値を使用してパブリックホストゾーンに新しいレコードを作成します。証明書のステータスは「発行済み」に変わります。

8. 新しい証明書 ARN をコピーして、のパラメータとして入力しますACMCertificateARNforWebApp。

Amazon CloudWatch Logs

Research and Engineering Studio は、インストール中に CloudWatch に次のロググループを作成します。デフォルトの保持については、次の表を参照してください。

CloudWatch Log グループ	Retention
/aws/lambda/<installation-stack-name>-cluster-endpoints	有効期限なし
/aws/lambda/<installation-stack-name>-cluster-manager-scheduled-ad-sync	有効期限なし
/aws/lambda/<installation-stack-name>-cluster-settings	有効期限なし
/aws/lambda/<installation-stack-name>-oauth-credentials	有効期限なし

CloudWatch Log グループ	Retention
/aws/lambda/<installation-stack-name>-self-signed-certificate	有効期限なし
/aws/lambda/<installation-stack-name>-update-cluster-prefix-list	有効期限なし
/aws/lambda/<installation-stack-name>-vdc-scheduled-event-transformer	有効期限なし
/aws/lambda/<installation-stack-name>-vdc-update-cluster-manager-client-scope	有効期限なし
/<installation-stack-name>/cluster-manager	3 か月間
/<installation-stack-name>/vdc/controller	3 か月間
/<installation-stack-name>/vdc/dcv-broker	3 か月間
/<installation-stack-name>/vdc/dcv-connection-gateway	3 か月間

ロググループのデフォルトの保持を変更する場合は、[CloudWatch コンソール](#)に移動し、[CloudWatch Logs のログデータ保持を変更する](#) 指示に従ってください。

カスタムアクセス許可の境界の設定

2024 年 4 月現在、カスタムアクセス許可の境界をアタッチすることで、オプションで RES によって作成されたロールを変更できます。カスタムアクセス許可の境界は、IAMPermissionBoundary パラメータの一部としてアクセス許可の境界の ARN を指定することで、RES AWS CloudFormation のインストールの一部として定義できます。このパラメータを空のままにすると、どの RES ロールにもアクセス許可の境界は設定されません。以下は、RES ロールが動作するために必要なアクションのリストです。使用する予定のアクセス許可の境界で、次のアクションが明示的に許可されていることを確認します。

```
[
  {
    "Effect": "Allow",
```

```
"Resource": "*",
"Sid": "ResRequiredActions",
"Action": [
  "access-analyzer:*",
  "account:GetAccountInformation",
  "account:ListRegions",
  "acm:*",
  "airflow:*",
  "amplify:*",
  "amplifybackend:*",
  "amplifyuibuilder:*",
  "aoss:*",
  "apigateway:*",
  "appflow:*",
  "application-autoscaling:*",
  "appmesh:*",
  "apprunner:*",
  "aps:*",
  "athena:*",
  "auditmanager:*",
  "autoscaling-plans:*",
  "autoscaling:*",
  "backup-gateway:*",
  "backup-storage:*",
  "backup:*",
  "batch:*",
  "bedrock:*",
  "budgets:*",
  "ce:*",
  "cloud9:*",
  "cloudformation:*",
  "cloudfront:*",
  "cloudtrail-data:*",
  "cloudtrail:*",
  "cloudwatch:*",
  "codeartifact:*",
  "codebuild:*",
  "codeguru-profiler:*",
  "codeguru-reviewer:*",
  "codepipeline:*",
  "codestar-connections:*",
  "codestar-notifications:*",
  "codestar:*",
  "cognito-identity:*
```

```
"cognito-idp:*",
"cognito-sync:*",
"comprehend:*",
"compute-optimizer:*",
"cur:*",
"databrew:*",
"datapipeline:*",
"datasync:*",
"dax:*",
"detective:*",
"devops-guru:*",
"dlm:*",
"dms:*",
"drs:*",
"dynamodb:*",
"ebs:*",
"ec2-instance-connect:*",
"ec2:*",
"ec2messages:*",
"ecr:*",
"ecs:*",
"eks:*",
"elastic-inference:*",
"elasticache:*",
"elasticbeanstalk:*",
"elasticfilesystem:*",
"elasticloadbalancing:*",
"elasticmapreduce:*",
"elastictranscoder:*",
"es:*",
"events:*",
"firehose:*",
"fis:*",
"fms:*",
"forecast:*",
"fsx:*",
"geo:*",
"glacier:*",
"glue:*",
"grafana:*",
"guardduty:*",
"health:*",
"iam:*",
"identitystore:*
```

```
"imagebuilder:*",
"inspector2:*",
"inspector:*",
"internetmonitor:*",
"iot:*",
"iotanalytics:*",
"kafka:*",
"kafkaconnect:*",
"kinesis:*",
"kinesisanalytics:*",
"kms:*",
"lambda:*",
"lightsail:*",
"logs:*",
"memorydb:*",
"mgh:*",
"mobiletargeting:*",
"mq:*",
"neptune-db:*",
"organizations:DescribeOrganization",
"osis:*",
"personalize:*",
"pi:*",
"pipes:*",
"polly:*",
"qldb:*",
"quicksight:*",
"rds-data:*",
"rds:*",
"redshift-data:*",
"redshift-serverless:*",
"redshift:*",
"rekognition:*",
"resiliencehub:*",
"resource-groups:*",
"route53:*",
"route53domains:*",
"route53resolver:*",
"rum:*",
"s3:*",
"sagemaker:*",
"scheduler:*",
"schemas:*",
"sdb:*
```

```
    "secretsmanager:*",
    "securityhub:*",
    "serverlessrepo:*",
    "servicecatalog:*",
    "servicequotas:*",
    "ses:*",
    "signer:*",
    "sns:*",
    "sqs:*",
    "ssm:*",
    "ssmmessages:*",
    "states:*",
    "storagegateway:*",
    "sts:*",
    "support:*",
    "tag:GetResources",
    "tag:GetTagKeys",
    "tag:GetTagValues",
    "textract:*",
    "timestream:*",
    "transcribe:*",
    "transfer:*",
    "translate:*",
    "vpc-lattice:*",
    "waf-regional:*",
    "waf:*",
    "wafv2:*",
    "wellarchitected:*",
    "wisdom:*",
    "xray:*"
  ]
}
```

RES 対応 AMIs を設定する

RES 対応 AMIs を使用すると、仮想デスクトップインスタンス (VDIs) の RES 依存関係をカスタム AMIs にプリインストールできます。RES 対応 AMIs を使用すると、事前にバイクされたイメージを使用した VDI インスタンスの起動時間が短縮されます。EC2 Image Builder を使用すると、AMIs を構築して新しいソフトウェアスタックとして登録できます。Image Builder の詳細については、[「Image Builder ユーザーガイド」](#)を参照してください。

開始する前に、[最新バージョンの RES をデプロイ](#)する必要があります。

トピック

- [RES 環境にアクセスするための IAM ロールを準備する](#)
- [EC2 Image Builder コンポーネントを作成する](#)
- [EC2 Image Builder レシピを準備する](#)
- [EC2 Image Builder インフラストラクチャを設定する](#)
- [Image Builder イメージパイプラインを設定する](#)
- [Image Builder イメージパイプラインを実行する](#)
- [RES に新しいソフトウェアスタックを登録する](#)

RES 環境にアクセスするための IAM ロールを準備する

EC2 Image Builder から RES 環境サービスにアクセスするには、RES-EC2InstanceProfileForImageBuilder という IAM ロールを作成または変更する必要があります。Image Builder で使用する IAM ロールの設定については、Image Builder ユーザーガイドの [AWS Identity and Access Management \(IAM\)](#) を参照してください。

ロールには以下が必要です。

- 信頼された関係には Amazon EC2 サービスが含まれます
- AmazonSSMManagedInstanceCore および EC2InstanceProfileForImageBuilder ポリシー
- デプロイされた RES 環境への DynamoDB および Amazon S3 アクセスが制限されたカスタム RES ポリシー

(このポリシーは、カスタマー管理ポリシードキュメントまたはカスタマーインラインポリシードキュメントのいずれかになります。ここに示すポリシーを使用するには、*us-east-1* を目的の AWS リージョンに、*123456789012* を your AWS AccountID に、*{RES_ENVIRONMENT_NAME}* を RES EnvironmentName.)

信頼された関係エンティティ：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

        "Effect": "Allow",
        "Principal": {
            "Service": "ec2.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }
]
}

```

RES ポリシー :

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RESDynamoDBAccess",
      "Effect": "Allow",
      "Action": "dynamodb:GetItem",
      "Resource": "arn:aws:dynamodb:us-east-1:123456789012:table/{RES_ENVIRONMENT_NAME}.cluster-settings",
      "Condition": {
        "ForAllValues:StringLike": {
          "dynamodb:LeadingKeys": [
            "global-settings.gpu_settings.*",
            "global-settings.package_config.*"
          ]
        }
      }
    },
    {
      "Sid": "RESS3Access",
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::{RES_ENVIRONMENT_NAME}-cluster-us-east-1-123456789012/idea/vdc/res-ready-install-script-packages/*"
    }
  ]
}

```

EC2 Image Builder コンポーネントを作成する

Image [Builder ユーザーガイドの「Image Builder コンソールを使用してコンポーネントを作成する」](#)の指示に従います。

コンポーネントの詳細を入力します。

1. Type で、Build を選択します。
2. イメージオペレーティングシステム (OS) の場合は、Linux または Windows を選択します。
3. コンポーネント名には、などのわかりやすい名前を入力します **research-and-engineering-studio-vdi-<operating-system>**。
4. コンポーネントのバージョン番号を入力し、オプションで説明を追加します。
5. 定義ドキュメントには、次の定義ファイルを入力します。エラーが発生した場合、YAML ファイルはスペースに敏感であり、最も可能性の高い原因です。

Linux

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License"). You may not
# use this file except in compliance
# with the License. A copy of the License is located at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# or in the 'license' file accompanying this file. This file is distributed on
# an 'AS IS' BASIS, WITHOUT WARRANTIES
# OR CONDITIONS OF ANY KIND, express or implied. See the License for the
# specific language governing permissions
# and limitations under the License.
name: research-and-engineering-studio-vdi-linux
description: An RES EC2 Image Builder component to install required RES software
dependencies for Linux VDI.
schemaVersion: 1.0
parameters:
  - AWSAccountID:
    type: string
    description: RES Environment AWS Account ID
  - RESEnvName:
    type: string
    description: RES Environment Name
  - RESEnvRegion:
    type: string
    description: RES Environment Region
  - RESEnvReleaseVersion:
    type: string
```

```

description: RES Release Version

phases:
  - name: build
    steps:
      - name: PrepareRESBootstrap
        action: ExecuteBash
        onFailure: Abort
        maxAttempts: 3
        inputs:
          commands:
            - 'mkdir -p /root/bootstrap/logs'
            - 'mkdir -p /root/bootstrap/latest'
      - name: DownloadRESLinuxInstallPackage
        action: S3Download
        onFailure: Abort
        maxAttempts: 3
        inputs:
          - source: 's3://{{ RESEnvName }}-cluster-{{ RESEnvRegion }}-{{ AWSAccountID }}/idea/vdc/res-ready-install-script-packages/linux/res_linux_install_{{ RESEnvReleaseVersion }}.tar.gz'
            destination: '/root/bootstrap/res_linux_install_{{ RESEnvReleaseVersion }}.tar.gz'
            expectedBucketOwner: '{{ AWSAccountID }}'
      - name: RunInstallScript
        action: ExecuteBash
        onFailure: Abort
        maxAttempts: 3
        inputs:
          commands:
            - 'tar -xvf
            {{ build.DownloadRESLinuxInstallPackage.inputs[0].destination }} -C /root/bootstrap/latest'
            - '/bin/bash /root/bootstrap/latest/virtual-desktop-host-linux/install.sh -r {{ RESEnvRegion }} -n {{ RESEnvName }} -g NONE'
      - name: FirstReboot
        action: Reboot
        onFailure: Abort
        maxAttempts: 3
        inputs:
          delaySeconds: 0
      - name: RunInstallPostRebootScript
        action: ExecuteBash
        onFailure: Abort

```

```

        maxAttempts: 3
        inputs:
          commands:
            - '/bin/bash /root/bootstrap/latest/virtual-desktop-host-linux/
install_post_reboot.sh'
        - name: SecondReboot
          action: Reboot
          onFailure: Abort
          maxAttempts: 3
          inputs:
            delaySeconds: 0

```

Windows

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License"). You may not
# use this file except in compliance
# with the License. A copy of the License is located at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# or in the 'license' file accompanying this file. This file is distributed on
# an 'AS IS' BASIS, WITHOUT WARRANTIES
# OR CONDITIONS OF ANY KIND, express or implied. See the License for the
# specific language governing permissions
# and limitations under the License.
name: research-and-engineering-studio-vdi-windows
description: An RES EC2 Image Builder component to install required RES software
dependencies for Windows VDI.
schemaVersion: 1.0
parameters:
  - AWSAccountID:
    type: string
    description: RES Environment AWS Account ID
  - RESEnvName:
    type: string
    description: RES Environment Name
  - RESEnvRegion:
    type: string
    description: RES Environment Region
  - RESEnvReleaseVersion:
    type: string

```

```

description: RES Release Version

phases:
  - name: build
    steps:
      - name: CreateRESBootstrapFolder
        action: CreateFolder
        onFailure: Abort
        maxAttempts: 3
        inputs:
          - path: 'C:\Users\Administrator\RES\Bootstrap'
            overwrite: true
      - name: DownloadRESWindowsInstallPackage
        action: S3Download
        onFailure: Abort
        maxAttempts: 3
        inputs:
          - source: 's3://{{ RESEnvName }}-cluster-{{ RESEnvRegion }}-{{ AWSAccountID }}/idea/vdc/res-ready-install-script-packages/windows/res_windows_install_{{ RESEnvReleaseVersion }}.tar.gz'
            destination:
              '{{ build.CreateRESBootstrapFolder.inputs[0].path }}\res_windows_install_{{ RESEnvReleaseVersion }}.tar.gz'
            expectedBucketOwner: '{{ AWSAccountID }}'
      - name: RunInstallScript
        action: ExecutePowerShell
        onFailure: Abort
        maxAttempts: 3
        inputs:
          commands:
            - 'cd {{ build.CreateRESBootstrapFolder.inputs[0].path }}'
            - 'Tar -xf
              res_windows_install_{{ RESEnvReleaseVersion }}.tar.gz'
            - 'Import-Module .\virtual-desktop-host-windows\Install.ps1'
            - 'Install-WindowsEC2Instance'
      - name: Reboot
        action: Reboot
        onFailure: Abort
        maxAttempts: 3
        inputs:
          delaySeconds: 0

```

6. オプションのタグを作成し、コンポーネントの作成を選択します。

EC2 Image Builder レシピを準備する

EC2 Image Builder レシピでは、新しいイメージを作成するための開始点として使用するベースイメージと、イメージをカスタマイズしてすべてが期待どおりに動作することを確認するために追加する一連のコンポーネントを定義します。レシピを作成または変更して、必要な RES ソフトウェアの依存関係を持つターゲット AMI を構築する必要があります。レシピの詳細については、[「レシピの管理」](#)を参照してください。

RES は、次のイメージオペレーティングシステムをサポートしています。

- Amazon Linux 2 (x86 および ARM64)
- Ubuntu 22.04.3 (x86)
- Windows 2019、2022 (x86)

Create a new recipe

1. で EC2 Image Builder コンソールを開きます<https://console.aws.amazon.com/imagebuilder>。
2. 保存済みリソースで、イメージレシピを選択します。
3. [イメージレシピの作成] を選択します。
4. 一意の名前とバージョン番号を入力します。
5. RES でサポートされているベースイメージを選択します。
6. インスタンス設定で、SSM エージェントがプリインストールされていない場合はインストールします。ユーザーデータおよびその他の必要なユーザーデータに情報を入力します。

Note

SSM エージェントをインストールする方法については、以下を参照してください。

- [Linux 用 EC2 インスタンスに SSM エージェントを手動でインストールする](#)
- [Windows Server 用の EC2 インスタンスに SSM Agent を手動でインストールおよびアンインストールする](#)

7. Linux ベースのレシピの場合は、Amazon が管理するaws-cli-version-2-linuxビルドコンポーネントをレシピに追加します。RES インストールスクリプトは を使用して AWS CLI、DynamoDB クラスター設定の設定値への VDI アクセスを提供します。Windows では、このコンポーネントは必要ありません。

- Linux または Windows 環境用に作成された EC2 Image Builder コンポーネントを追加し、必要なパラメータ値を入力します。次のパラメータは必須の入力です: AWSAccountID、RESEnvName、RESEnvRegion、RESEnvReleaseVersion。

 Important

Linux 環境では、aws-cli-version-2-linuxビルドコンポーネントを最初に追加した状態で、これらのコンポーネントを追加する必要があります。

- (推奨) Amazon が管理するsimple-boot-test-<linux-or-windows>テストコンポーネントを追加して、AMI を起動できることを確認します。これは最小限の推奨事項です。要件を満たす他のテストコンポーネントを選択できます。
- 必要に応じてオプションのセクションを完了し、他の必要なコンポーネントを追加して、レシピの作成を選択します。

Modify a recipe

既存の EC2 Image Builder レシピがある場合は、次のコンポーネントを追加して使用できます。

- Linux ベースのレシピの場合は、Amazon が管理するaws-cli-version-2-linuxビルドコンポーネントをレシピに追加します。RES インストールスクリプトはを使用して AWS CLI、DynamoDB クラスター設定の設定値への VDI アクセスを提供します。Windows では、このコンポーネントは必要ありません。
- Linux または Windows 環境用に作成された EC2 Image Builder コンポーネントを追加し、必要なパラメータ値を入力します。次のパラメータは必須の入力です: AWSAccountID、RESEnvName、RESEnvRegion、RESEnvReleaseVersion。

 Important

Linux 環境では、aws-cli-version-2-linuxビルドコンポーネントを最初に追加した状態で、これらのコンポーネントを追加する必要があります。

- 必要に応じてオプションのセクションを完了し、他の必要なコンポーネントを追加して、レシピの作成を選択します。

EC2 Image Builder インフラストラクチャを設定する

インフラストラクチャ設定を使用して、Image Builder が Image Builder イメージの構築とテストに使用する Amazon EC2 インフラストラクチャを指定できます。RES で使用するには、新しいインフラストラクチャ設定を作成するか、既存の設定を使用するかを選択できます。

- 新しいインフラストラクチャ設定を作成するには、[「インフラストラクチャ設定の作成」](#)を参照してください。
- 既存のインフラストラクチャ設定を使用するには、[インフラストラクチャ設定を更新します](#)。

Image Builder インフラストラクチャを設定するには：

1. IAM ロールには、で以前に設定したロールを入力します[the section called “RES 環境にアクセスするための IAM ロールを準備する”](#)。
2. インスタンスタイプでは、少なくとも 4 GB のメモリを持つタイプを選択し、選択したベース AMI アーキテクチャをサポートします。[Amazon EC2 インスタンスタイプ](#)」を参照してください。
3. VPC、サブネット、およびセキュリティグループの場合、ソフトウェアパッケージをダウンロードするためにインターネットアクセスを許可する必要があります。RES 環境の cluster-settings DynamoDB テーブルと Amazon S3 クラスターバケットへのアクセスも許可する必要があります。

Image Builder イメージパイプラインを設定する

Image Builder イメージパイプラインは、ベースイメージ、構築とテスト用のコンポーネント、インフラストラクチャ設定、およびディストリビューション設定をアセンブルします。RES 対応 AMIs のイメージパイプラインを設定するには、新しいパイプラインを作成するか、既存のパイプラインを使用するかを選択できます。詳細については、Image Builder ユーザーガイドの[「AMI イメージパイプラインの作成と更新」](#)を参照してください。

Create a new Image Builder pipeline

1. で Image Builder コンソールを開きます<https://console.aws.amazon.com/imagebuilder>。
2. ナビゲーションから、イメージパイプラインを選択します。
3. 「イメージパイプラインの作成」を選択します。

4. 一意の名前、オプションの説明、スケジュール、頻度を入力して、パイプラインの詳細を指定します。
5. 「レシピの選択」で、「既存のレシピを使用」を選択し、「」で作成したレシピを選択します [the section called “EC2 Image Builder レシピを準備する”](#)。レシピの詳細が正しいことを確認します。
6. イメージ作成プロセスを定義する では、ユースケースに応じてデフォルトワークフローまたはカスタムワークフローを選択します。ほとんどの場合、デフォルトのワークフローで十分です。詳細については、[EC2 Image Builder パイプラインのイメージワークフローを設定する](#)」を参照してください。
7. 「インフラストラクチャ設定の定義」で、「既存のインフラストラクチャ設定の選択」を選択し、「」で作成したインフラストラクチャ設定を選択します [the section called “EC2 Image Builder インフラストラクチャを設定する”](#)。インフラストラクチャの詳細が正しいことを確認します。
8. ディストリビューション設定を定義する で、サービスのデフォルトを使用してディストリビューション設定を作成する を選択します。出力イメージは、RES 環境 AWS リージョンと同じに存在する必要があります。サービスのデフォルトを使用すると、Image Builder が使用されているリージョンにイメージが作成されます。
9. パイプラインの詳細を確認し、パイプラインの作成を選択します。

Modify an existing Image Builder pipeline

1. 既存のパイプラインを使用するには、 で作成されたレシピを使用するように詳細を変更します [the section called “EC2 Image Builder レシピを準備する”](#)。
2. [Save changes] (変更の保存) をクリックします。

Image Builder イメージパイプラインを実行する

設定された出力イメージを生成するには、イメージパイプラインを開始する必要があります。イメージレシピのコンポーネント数によっては、構築プロセスに最大 1 時間かかる場合があります。

イメージパイプラインを実行するには：

1. イメージパイプラインから、 で作成されたパイプラインを選択します [the section called “Image Builder イメージパイプラインを設定する”](#)。
2. アクションから、パイプラインの実行を選択します。

RES に新しいソフトウェアスタックを登録する

1. 「」の指示に従って[the section called “ソフトウェアスタック \(AMIs\)”](#)、ソフトウェアスタックを登録します。
2. AMI ID には、に構築された出カイメージの AMI ID を入力します[the section called “Image Builder イメージパイプラインを実行する”](#)。

管理者ガイド

この管理者ガイドでは、AWS 製品の Research and Engineering Studio をさらにカスタマイズして統合する方法について、技術的な対象者向けの追加の手順を提供します。

トピック

- [セッション管理](#)
- [環境管理](#)
- [シークレットの管理](#)
- [コストのモニタリングと制御](#)

セッション管理

セッション管理は、セッションを開発およびテストするための柔軟でインタラクティブな環境を提供します。管理ユーザーとして、プロジェクト環境内でインタラクティブセッションを作成および管理することをユーザーに許可できます。

トピック

- [ダッシュボード](#)
- [セッション](#)
- [ソフトウェアスタック \(AMIs\)](#)
- [デバッグ](#)
- [デスクトップ設定](#)

ダッシュボード

セッション管理ダッシュボードは、管理者に以下に関するクイックビューを提供します。

1. インスタンスのタイプ
2. セッションの状態
3. ベース OS
4. プロジェクト
5. アベイラビリティゾーン

6. ソフトウェアスタック

さらに、管理者は次のことができます。

7. ダッシュボードを更新して情報を更新します。
8. セッションの表示を選択してセッションに移動します。

セッション

セッションには、Research and Engineering Studio 内で作成されたすべての仮想デスクトップが表示されます。セッションページから、セッション情報をフィルタリングして表示したり、新しいセッションを作成したりできます。

1. メニューを使用して、指定した期間内に作成または更新されたセッションで結果をフィルタリングします。
2. セッションを選択し、アクションメニューを使用して以下を行います。
 - a. セッションを再開する (複数可)
 - b. 停止/休止セッション (複数可)
 - c. 強制停止/休止セッション (複数可)
 - d. セッションの終了 (複数可)
 - e. セッションの強制終了 (複数可)
 - f. セッションの状態 (複数可)
 - g. ソフトウェアスタックの作成
3. セッションの作成を選択して新しいセッションを作成します。
4. 名前でセッションを検索し、状態とオペレーティングシステムでフィルタリングします。
5. セッション名を選択すると、詳細が表示されます。

セッションを作成する

1. セッションの作成 を選択します。新しい仮想デスクトップの起動モーダルが開きます。
2. 新しいセッションの詳細を入力します。
3. (オプション) Show Advanced Options をオンにして、サブネット ID や DCV セッションタイプなどの追加の詳細を指定します。

4. [Submit] を選択してください。

セッションの詳細

セッションリストから、セッション名を選択してセッションの詳細を表示します。

ソフトウェアスタック (AMIs)

Note

で提供されている CentSO7 ソフトウェアスタックを実行するには AWS GovCloud (US)、[リンクされた標準アカウント](#) AWS Marketplace を使用して 内の AMI にサブスクライブする必要があります。

ソフトウェアスタックページから、Amazon マシンイメージ (AMIs) を設定し、既存の AMIs を管理できます。

1. 既存のソフトウェアスタックを検索するには、オペレーティングシステムのドロップダウンを使用して OS でフィルタリングします。
2. ソフトウェアスタックの名前を選択して、スタックの詳細を表示します。
3. ソフトウェアスタックを選択したら、アクションメニューを使用してスタックを編集し、スタックをプロジェクトに割り当てます。
4. ソフトウェアスタックの登録ボタンを使用すると、新しいスタックを作成できます。
 1. 「ソフトウェアスタックの登録」を選択します。
 2. 新しいソフトウェアスタックの詳細を入力します。
 3. [Submit] を選択してください。

プロジェクトにソフトウェアスタックを割り当てる

新しいソフトウェアスタックを作成するときは、スタックをプロジェクトに割り当てることができます。初回作成後にスタックをプロジェクトに追加する必要がある場合は、次の操作を行います。

Note

ソフトウェアスタックは、自分がメンバーであるプロジェクトにのみ割り当てることができません。

1. ソフトウェアスタックページからプロジェクトに追加する必要があるソフトウェアスタックを選択します。
2. [アクション] を選択します。
3. [編集] を選択します。
4. プロジェクトドロップダウンを使用してプロジェクトを選択します。
5. [Submit] を選択してください。

スタックの詳細ページからソフトウェアスタックを編集することもできます。

ソフトウェアスタックの詳細を表示する

ソフトウェアスタックリストから、ソフトウェアスタック名を選択して詳細を表示します。詳細ページから、編集を選択してソフトウェアスタックを編集することもできます。

デバッグ

デバッグパネルには、仮想デスクトップに関連付けられたメッセージトラフィックが表示されます。このパネルを使用して、ホスト間のアクティビティを監視できます。VD ホストタブにはインスタンス固有のアクティビティが表示され、VD セッションタブには進行中のセッションアクティビティが表示されます。

デスクトップ設定

デスクトップ設定ページを使用して、仮想デスクトップに関連付けられたリソースを設定できます。サーバータブでは、次のような設定にアクセスできます。

DCV セッションアイドルタイムアウト

DCV セッションが自動的に切断されるまでの時間。これにより、デスクトップセッションの状態は変更されず、DCV クライアントまたはウェブブラウザからのみセッションが閉じられます。

アイドルタイムアウトの警告

アイドル警告がクライアントに提供されるまでの時間。

CPU 使用率のしきい値

アイドルと見なされる CPU 使用率。

ユーザーあたりの許可されたセッション

個々のユーザーが一度に持つことができる VDI セッションの数。ユーザーがこの値以上になると、My Virtual Desktops ページから新しいセッションを起動できなくなります。セッションページからセッションを起動する機能は、この値の影響を受けません。

最大ルートボリュームサイズ

仮想デスクトップセッションのルートボリュームのデフォルトサイズ。

許可されたインスタンスタイプ

この RES 環境で起動できるインスタンスファミリーとサイズのリスト。インスタンスファミリーとインスタンスサイズの組み合わせの両方が受け入れられます。たとえば、「m7a」を指定すると、m7a ファミリーのすべてのサイズが VDI セッションとして起動できるようになります。'm7a.24xlarge' を指定した場合、VDI セッションとして起動できるのは m7a.24xlarge のみです。このリストは、環境内のすべてのプロジェクトに影響します。

環境管理

RES の環境管理セクションから、管理ユーザーは研究およびエンジニアリングプロジェクト用に分離された環境を作成および管理できます。これらの環境には、コンピューティングリソース、ストレージ、その他の必要なコンポーネントを含めることができ、すべて安全な環境内にあります。ユーザーは、プロジェクトの特定の要件を満たすようにこれらの環境を設定およびカスタマイズできるため、他のプロジェクトや環境に影響を与えることなく、ソリューションの実験、テスト、反復が容易になります。

トピック

- [プロジェクト](#)
- [ユーザー](#)

- [グループ](#)
- [アクセス許可プロファイル](#)
- [ファイルシステム](#)
- [環境ステータス](#)
- [スナップショットの管理](#)
- [環境設定](#)
- [Amazon S3 バケット](#)

プロジェクト

プロジェクトは、仮想デスクトップ、チーム、予算の境界を形成します。プロジェクトを作成するときは、名前、説明、環境設定などの設定を定義します。プロジェクトには通常、コンピューティングリソースのタイプとサイズ、ソフトウェアスタック、ネットワーク設定など、プロジェクトの特定の要件を満たすようにカスタマイズできる 1 つ以上の環境が含まれます。

トピック

- [プロジェクトを表示する](#)
- [プロジェクトを作成する](#)
- [プロジェクトを編集する](#)
- [プロジェクトへのタグの追加または削除](#)
- [プロジェクトに関連付けられたファイルシステムを表示する](#)
- [起動テンプレートを追加する](#)

プロジェクトを表示する

プロジェクトダッシュボードには、利用可能なプロジェクトのリストが表示されます。プロジェクトダッシュボードから、次のことができます。

1. 検索フィールドを使用してプロジェクトを検索できます。
2. プロジェクトを選択すると、アクションメニューを使用して次のことができます。
 - a. プロジェクトを編集する
 - b. プロジェクトの無効化または有効化

c. プロジェクトタグを更新する

3. プロジェクトの作成を選択して、新しいプロジェクトを作成できます。

プロジェクトを作成する

1. [プロジェクトを作成] を選択します。
2. プロジェクトの詳細を入力します。

プロジェクト ID は、コスト配分を追跡するために使用できるリソースタグです AWS Cost Explorer Service。詳細については、[「ユーザー定義のコスト配分タグのアクティブ化」](#)を参照してください。

Important

作成後にプロジェクト ID を変更することはできません。

詳細オプションの詳細については、「」を参照してください[起動テンプレートを追加する](#)。

3. (オプション) プロジェクトの予算を有効にします。予算の詳細については、「」を参照してください[コストのモニタリングと制御](#)。
4. ユーザーまたはグループに適切なロール (「プロジェクトメンバー」または「プロジェクト所有者」) を割り当てます。各ロールが実行できるアクション[デフォルトのアクセス許可プロファイル](#)については、「」を参照してください。
5. [Submit] を選択してください。

プロジェクトを編集する

1. プロジェクトリストでプロジェクトを選択します。
2. Actions メニューから、Edit Project を選択します。
3. 更新を入力します。予算を有効にする場合は、[コストのモニタリングと制御](#)「」を参照してください。詳細オプションの詳細については、「」を参照してください[起動テンプレートを追加する](#)。
4. [Submit] を選択してください。

プロジェクトへのタグの追加または削除

プロジェクトタグは、そのプロジェクトで作成されたすべてのインスタンスにタグを割り当てます。

1. プロジェクトリストでプロジェクトを選択します。
2. Actions メニューから、Update Tags を選択します。
3. タグの追加を選択し、キーの値を入力します。
4. タグを削除するには、削除するタグの横にある「削除」を選択します。

プロジェクトに関連付けられたファイルシステムを表示する

プロジェクトを選択すると、画面の下部にあるファイルシステムペインを展開して、プロジェクトに関連付けられたファイルシステムを表示できます。

起動テンプレートを追加する

プロジェクトを作成または編集するときは、プロジェクト設定内のアドバンスドオプションを使用して起動テンプレートを追加できます。起動テンプレートは、セキュリティグループ、IAM ポリシー、起動スクリプトなどの追加の設定をプロジェクト内のすべての VDI インスタンスに提供します。

ポリシーの追加

IAM ポリシーを追加して、プロジェクトの下にデプロイされたすべてのインスタンスの VDI アクセスを制御できます。ポリシーをオンボードするには、ポリシーに次のキーと値のペアをタグ付けします。

```
res:Resource/vdi-host-policy
```

IAM ロールの詳細については、[「IAM のポリシーとアクセス許可」](#)を参照してください。

セキュリティグループの追加

セキュリティグループを追加して、プロジェクト内のすべての VDI インスタンスの出力データと進入データを制御できます。セキュリティグループをオンボードするには、セキュリティグループに次のキーと値のペアをタグ付けします。

```
res:Resource/vdi-security-group
```

セキュリティグループの詳細については、「Amazon VPC ユーザーガイド」の「[セキュリティグループを使用して AWS リソースへのトラフィックを制御する](#)」を参照してください。

起動スクリプトを追加する

プロジェクト内のすべての VDI セッションで開始する起動スクリプトを追加できます。RES は Linux および Windows のスクリプト開始をサポートしています。スクリプトを開始するには、次のいずれかを選択できます。

VDI の開始時にスクリプトを実行する

このオプションは、RES 設定またはインストールを実行する前に、VDI インスタンスの先頭でスクリプトを開始します。

VDI が設定されている場合にスクリプトを実行する

このオプションは、RES 設定の完了後にスクリプトを開始します。

スクリプトは、次のオプションをサポートしています。

スクリプト設定	例
S3 URI	s3://bucketname/script.sh
HTTPS URL	https://sample.samplecontent.com/sample
ローカルファイル	file:///user/scripts/example.sh

引数には、カンマで区切られた引数を指定します。

プロジェクト設定の例

[ユーザー]

アクティブディレクトリから同期されたすべてのユーザーがユーザーページに表示されます。ユーザーは、製品の設定中に cluster-admin ユーザーによって同期されます。初期ユーザー設定の詳細については、「[設定ガイド](#)」を参照してください。

Note

管理者は、アクティブなユーザーのセッションのみを作成できます。デフォルトでは、すべてのユーザーは製品環境にサインインするまで非アクティブ状態になります。ユーザーが非アクティブの場合は、セッションを作成する前にサインインするように依頼します。

ユーザーページから、次のことができます。

1. ユーザーを検索します。
2. ユーザー名を選択したら、アクションメニューを使用して次の操作を行います。
 - a. 管理者ユーザーとして設定する
 - b. ユーザーを無効にする

グループ

アクティブディレクトリから同期されたすべてのグループは、グループページに表示されます。グループの設定と管理の詳細については、「」を参照してください[設定ガイド](#)。

グループページから、次のことができます。

1. ユーザーグループを検索します。
2. ユーザーグループを選択したら、アクションメニューを使用してグループを無効または有効にします。
3. ユーザーグループを選択すると、画面の下部にあるユーザーペインを展開して、グループのユーザーを表示できます。

アクセス許可プロファイル

概要

Research and Engineering Studio (RES) を使用すると、管理ユーザーは、選択したユーザーに、自分が所属するプロジェクトを管理するための追加のアクセス許可を付与するカスタムアクセス許可プロファイルを作成できます。各プロジェクトには、デプロイ後にカスタマイズできる「プロジェクト

トメンバー」と「プロジェクト所有者」の 2 つの[デフォルトのアクセス許可プロファイル](#)があります。

現在、管理者はアクセス許可プロファイルを使用して 2 つのアクセス許可のコレクションを付与できます。

1. 指定されたユーザーがプロジェクトに他のユーザーやグループを追加または削除できるようにする「プロジェクトメンバーシップの更新」と、指定されたユーザーがプロジェクトを有効または無効にできるようにする「プロジェクトステータスの更新」で構成されるプロジェクト管理アクセス許可。
2. 指定されたユーザーがプロジェクト内に VDI セッションを作成できるようにする「セッションの作成」と、指定されたユーザーがプロジェクト内の他のユーザーのセッションを作成または終了できるようにする「別のユーザーのセッションの作成/終了」で構成される VDI セッション管理アクセス許可。

これにより、管理者は 環境内の管理者以外のユーザーにプロジェクトベースのアクセス許可を委任できます。

プロジェクト管理のアクセス許可

プロジェクトメンバーシップを更新する

このアクセス許可により、付与された管理者以外のユーザーは、プロジェクトからユーザーまたはグループを追加および削除できます。また、アクセス許可プロファイルを設定し、そのプロジェクトの他のすべてのユーザーとグループのアクセスレベルを決定することもできます。

プロジェクトのステータスを更新する

このアクセス許可により、付与された管理者以外のユーザーは、プロジェクトページのアクションボタンを使用してプロジェクトを有効または無効にできます。

VDI セッション管理のアクセス許可

セッションを作成する

ユーザーが My Virtual Desktops ページから独自の VDI セッションを起動できるかどうかを制御します。これを無効にして、管理者以外のユーザーが独自の VDI セッションを起動できないようにします。ユーザーはいつでも独自の VDI セッションを停止および終了できます。

管理者以外のユーザーにセッションを作成するアクセス許可がない場合、新しい仮想デスクトップを起動するボタンは、次に示すように無効になります。

他のユーザーのセッションを作成または終了する

管理者以外のユーザーが左側のナビゲーションペインからセッションページにアクセスできるようにします。これらのユーザーは、このアクセス許可が付与されているプロジェクトで他のユーザーの VDI セッションを起動できます。

管理者以外のユーザーが他のユーザーのセッションを起動するアクセス許可を持っている場合、左側のナビゲーションペインには、次に示すようにセッション管理の下でセッションリンクが表示されます。

管理者以外のユーザーに他のユーザーのセッションを作成するアクセス許可がない場合、左側のナビゲーションペインには、次に示すようにセッション管理は表示されません。

アクセス許可プロファイルの管理

RES 管理者は、次のアクションを実行してアクセス許可プロファイルを管理できます。

アクセス許可プロファイルを一覧表示する

- Research and Engineering Studio コンソールページから、左側のナビゲーションペインでアクセス許可プロファイルを選択します。このページから、アクセス許可プロファイルを作成、更新、一覧表示、表示、削除できます。

アクセス許可プロファイルを表示する

- メインのアクセス許可プロファイルページで、表示するアクセス許可プロファイルの名前を選択します。このページから、選択したアクセス許可プロファイルを編集または削除できます。
- アクセス許可プロファイルを現在使用しているプロジェクトを表示するには、影響を受けるプロジェクトタブを選択します。

アクセス許可プロファイルを作成する

1. メインのアクセス許可プロファイルページで、プロファイルの作成を選択してアクセス許可プロファイルを作成します。
2. アクセス許可プロファイルの名前と説明を入力し、このプロファイルに割り当てるユーザーまたはグループに付与するアクセス許可を選択します。

アクセス許可プロファイルを編集する

- メインのアクセス許可プロファイルページで、プロファイルの横にある円をクリックしてプロファイルを選択し、アクションを選択し、プロファイルの編集を選択してそのアクセス許可プロファイルを更新します。

アクセス許可プロファイルを削除する

- メインのアクセス許可プロファイルページで、プロファイルの横にある円をクリックしてプロファイルを選択し、アクションを選択し、プロファイルの削除を選択します。既存のプロジェクトで使用されているアクセス許可プロファイルは削除できません。

デフォルトのアクセス許可プロファイル

すべての RES プロジェクトには、グローバル管理者が設定できる 2 つのデフォルトのアクセス許可プロファイルが付属しています。(さらに、グローバル管理者はプロジェクトの新しいアクセス許可プロファイルを作成および変更できます)。次の表は、「プロジェクトメンバー」と「プロジェクト所有者」のデフォルトのアクセス許可プロファイルで許可されるアクセス許可を示しています。アクセス許可プロファイル、およびプロジェクトの特定のユーザーに付与するアクセス許可は、自分が属するプロジェクトにのみ適用されます。グローバル管理者は、すべてのプロジェクトで以下のすべてのアクセス許可を持つスーパーユーザーです。

アクセス許可	説明	プロジェクトメンバー	プロジェクト所有者
セッションの作成	独自のセッションを作成しま	X	X

アクセス許可	説明	プロジェクトメンバー	プロジェクト所有者	
	す。ユーザーは、このアクセス許可の有無にかかわらず、いつでも自分のセッションを停止および終了できます。			
他のユーザーのセッションを作成/終了する	プロジェクト内で別のユーザーのセッションを作成または終了します。		X	
プロジェクトメンバーシップの更新	プロジェクトに関連付けられたユーザーとグループを更新します。		X	
プロジェクトステータスの更新	プロジェクトを有効または無効にします。		X	

ファイルシステム

ファイルシステムページから、次のことができます。

1. ファイルシステムを検索します。
2. ファイルシステムを選択したら、アクションメニューを使用して次の操作を行います。
 - a. ファイルシステムをプロジェクトに追加する

- b. プロジェクトからファイルシステムを削除する
3. 新しいファイルシステムをオンボードします。
4. ファイルシステムを作成します。
5. ファイルシステムを選択すると、画面の下部にあるペインを展開してファイルシステムの詳細を表示できます。

ファイルシステムを作成する

1. [ファイルシステムの作成] を選択します。
2. 新しいファイルシステムの詳細を入力します。
3. VPC からサブネット IDs。IDs は、環境管理 > 設定 > ネットワークタブにあります。
4. [Submit] を選択してください。

ファイルシステムのオンボード

1. ファイルシステムのオンボードを選択します。
2. ドロップダウンからファイルシステムを選択します。モーダルは、追加の詳細エントリで展開されます。
3. ファイルシステムの詳細を入力します。
4. [Submit] を選択してください。

環境ステータス

環境ステータスページには、製品内にデプロイされたソフトウェアとホストが表示されます。これには、ソフトウェアバージョン、モジュール名、その他のシステム情報などの情報が含まれます。

スナップショットの管理

スナップショット管理は、環境間でデータを保存および移行するプロセスを簡素化し、一貫性と正確性を確保します。スナップショットを使用すると、環境の状態を保存し、同じ状態の新しい環境にデータを移行できます。

スナップショット管理ページから、次のことができます。

1. 作成されたすべてのスナップショットとそのステータスを表示します。
2. スナップショットを作成します。スナップショットを作成する前に、適切なアクセス許可を持つバケットを作成する必要があります。
3. 適用されたすべてのスナップショットとそのステータスを表示します。
4. スナップショットを適用します。

スナップショットを作成する

スナップショットを作成する前に、必要なアクセス許可を Amazon S3 バケットに提供する必要があります。バケットの作成については、「[バケットを作成する](#)」を参照してください。バケットのバージョニングとサーバーアクセスのログ記録を有効にすることをお勧めします。これらの設定は、プロビジョニング後にバケットのプロパティタブから有効にできます。

Note

この Amazon S3 バケットのライフサイクルは、製品内で管理されません。コンソールからバケットのライフサイクルを管理する必要があります。

バケットにアクセス許可を追加するには：

1. バケットリストから作成したバケットを選択します。
2. [アクセス許可] タブを選択します。
3. [バケットポリシー] で [編集] を選択します。
4. バケットポリシーに次のステートメントを追加します。以下の値を自分の値に置き換えてください。
 - **111122223333** -> AWS アカウント ID

- `{RES_ENVIRONMENT_NAME}` -> RES 環境名
- `us-east-1` -> AWS リージョン
- `amzn-s3-demo-bucket` -> S3 バケット名

 Important

でサポートされている限定バージョンの文字列があります AWS。詳細については、https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_version.htmlを参照してください。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Export-Snapshot-Policy",
      "Effect": "Allow",
      "Principal": {
        "AWS":
          "arn:aws:iam::111122223333:role/{RES_ENVIRONMENT_NAME}-cluster-manager-
          role-us-east-1"}
      },
      "Action": [
        "s3:GetObject",
        "s3:ListBucket",
        "s3:AbortMultipartUpload",
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3::amzn-s3-demo-bucket",
        "arn:aws:s3::amzn-s3-demo-bucket/*"
      ]
    },
    {
      "Sid": "AllowSSLRequestsOnly",
      "Action": "s3:*",
```

```
        "Effect": "Deny",
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket",
            "arn:aws:s3:::amzn-s3-demo-bucket/*"
        ],
        "Condition": {
            "Bool": {
                "aws:SecureTransport": "false"
            }
        },
        "Principal": "*"
    }
}
```

スナップショットを作成するには：

1. [スナップショットの作成] を選択します。
2. 作成した Amazon S3 バケットの名前を入力します。
3. バケット内にスナップショットを保存するパスを入力します。例えば、**october2023/23**。
4. [Submit] を選択してください。
5. 5～10 分後、スナップショットページで更新を選択してステータスを確認します。スナップショットは、ステータスが IN_PROGRESS から COMPLETED に変わるまで有効ではありません。

スナップショットを適用する

環境のスナップショットを作成したら、そのスナップショットを新しい環境に適用してデータを移行できます。環境がスナップショットを読み取れるように、バケットに新しいポリシーを追加する必要があります。

スナップショットを適用すると、ユーザーアクセス許可、プロジェクト、ソフトウェアスタック、アクセス許可プロファイル、ファイルシステムなどのデータが新しい環境に関連付けられてコピーされます。ユーザーセッションはレプリケートされません。スナップショットが適用されると、各リソースレコードの基本情報をチェックして、既に存在するかどうかを判断します。レコードが重複している場合、スナップショットは新しい環境でのリソースの作成をスキップします。名前やキーを共有するなど、似たようなレコードの場合、他の基本的

なリソース情報は異なるため、次の規則を使用して名前とキーを変更して新しいレコードを作成します: RecordName_SnapshotRESVersion_ApplySnapshotID。はタイムスタンプのApplySnapshotIDように見えるため、スナップショットを適用しようとするたびに識別されません。

スナップショットアプリケーション中に、スナップショットはリソースの可用性をチェックします。新しい環境で使用できないリソースは作成されません。依存リソースを持つリソースの場合、スナップショットは依存リソースの可用性をチェックします。依存リソースが使用できない場合、依存リソースなしでメインリソースが作成されます。

新しい環境が想定どおりにない場合、または失敗する場合は、ロググループにある CloudWatch ログで /res-<env-name>/cluster-manager 詳細を確認できます。各ログには [apply snapshot] タグがあります。スナップショットを適用したら、[the section called “スナップショットの管理”](#) ページからそのステータスを確認できます。

バケットにアクセス許可を追加するには：

1. バケットリストから作成したバケットを選択します。
2. [アクセス許可] タブを選択します。
3. [バケットポリシー] で [編集] を選択します。
4. バケットポリシーに次のステートメントを追加します。以下の値を自分の値に置き換えてください。
 - *111122223333* -> AWS アカウント ID
 - *{RES_ENVIRONMENT_NAME}* -> RES 環境名
 - *us-east-1* -> AWS リージョン
 - *amzn-s3-demo-bucket* -> S3 バケット名

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Export-Snapshot-Policy",
      "Effect": "Allow",
      "Principal": {
```

```

        "AWS":
        "arn:aws:iam::111122223333:role/{RES_ENVIRONMENT_NAME}-cluster-manager-
        role-us-east-1}"
    },
    "Action": [
        "s3:GetObject",
        "s3:ListBucket"
    ],
    "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ]
},
{
    "Sid": "AllowSSLRequestsOnly",
    "Action": "s3:*",
    "Effect": "Deny",
    "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ],
    "Condition": {
        "Bool": {
            "aws:SecureTransport": "false"
        }
    },
    "Principal": "*"
}
]
}

```

スナップショットを適用するには：

1. スナップショットの適用 を選択します。
2. スナップショットを含む Amazon S3 バケットの名前を入力します。
3. バケット内のスナップショットへのファイルパスを入力します。
4. [Submit] を選択してください。
5. 5～10 分後、スナップショット管理ページで更新を選択してステータスを確認します。

環境設定

環境設定には、次のような製品設定の詳細が表示されます。

- 全般

製品をプロビジョニングしたユーザーの管理者ユーザー名や E メールなどの情報を表示します。ウェブポータルのタイトルと著作権テキストを編集できます。

- ID プロバイダー

シングルサインオンステータスなどの情報を表示します。

- ネットワーク

アクセス用の VPC ID、プレフィックスリスト IDs を表示します。

- Directory Service

ユーザー名とパスワードのアクティブディレクトリ設定とサービスアカウントのシークレットマネージャー ARN を表示します。

Amazon S3 バケット

トピック

- [Amazon S3 バケットのマウント](#)
- [Amazon S3 バケットを追加する](#)
- [Amazon S3 バケットを編集する](#)
- [Amazon S3 バケットを削除する](#)
- [データ分離](#)
- [クロスアカウントバケットアクセス](#)
- [プライベート VPC でのデータ流出の防止](#)
- [トラブルシューティング](#)
- [CloudTrail の有効化](#)

Amazon S3 バケットのマウント

Research and Engineering Studio (RES) は、Linux Virtual Desktop Infrastructure (VDI) インスタンスへの Amazon S3 バケットのマウントをサポートしています。RES 管理者は、環境管理の S3 バケッ

トタブで、S3 バケットを RES にオンボードしたり、プロジェクトにアタッチしたり、設定を編集したり、バケットを削除したりできます。

S3 バケットダッシュボードには、利用可能なオンボード S3 バケットのリストが表示されます。S3 バケットダッシュボードから、次のことができます。

1. バケットの追加を使用して、S3 バケットを RES にオンボードします。
2. S3 バケットを選択し、アクションメニューを使用して次の操作を行います。
 - バケットを編集する
 - バケットを削除する
3. 検索フィールドを使用してバケット名で検索し、オンボードされた S3 バケットを検索します。

Amazon S3 バケットを追加する

RES 環境に S3 バケットを追加するには：

1. [Add bucket (バケットの追加)] を選択します。
2. バケット名、ARN、マウントポイントなどのバケットの詳細を入力します。

Important

- 指定されたバケット ARN、マウントポイント、モードは、作成後に変更することはできません。
- バケット ARN には、オンボードされた S3 バケットをそのプレフィックスに分離するプレフィックスを含めることができます。

3. バケットをオンボードするモードを選択します。

Important

- 特定のモードによるデータ分離の詳細については、[データ分離](#)「」を参照してください。

4. 詳細オプションでは、クロスアカウントアクセス用にバケットをマウントするための IAM ロール ARN を指定できます。の手順に従って[クロスアカウントバケットアクセス](#)、クロスアカウントアクセスに必要な IAM ロールを作成します。
5. (オプション) バケットをプロジェクトに関連付けます。プロジェクトは後で変更できます。ただし、S3 バケットをプロジェクトの既存の VDI セッションにマウントすることはできません。プロジェクトがバケットに関連付けられた後に起動されたセッションのみがバケットをマウントします。
6. [Submit] を選択してください。

Amazon S3 バケットを編集する

1. S3 バケットリストで S3 バケットを選択します。
2. [Actions] (アクション) メニューで、[Edit] (編集) を選択します。
3. 更新を入力します。

Important

- プロジェクトを S3 バケットに関連付けると、そのプロジェクトの既存の仮想デスクトップインフラストラクチャ (VDI) インスタンスにバケットがマウントされません。バケットは、バケットがそのプロジェクトに関連付けられた後に、プロジェクトで起動された VDI セッションにのみマウントされます。
- S3 バケットからプロジェクトの関連付けを解除しても、S3 バケット内のデータには影響しませんが、デスクトップユーザーはそのデータにアクセスできなくなります。

4. バケット設定の保存 を選択します。

Amazon S3 バケットを削除する

1. S3 バケットリストで S3 バケットを選択します。
2. Actions メニューから、Remove を選択します。

Important

- まず、バケットからすべてのプロジェクトの関連付けを削除する必要があります。
- 削除オペレーションは、S3 バケット内のデータには影響しません。S3 バケットと RES の関連付けのみが削除されます。
- バケットを削除すると、そのセッションの認証情報の有効期限 (約 1 時間) に、既存の VDI セッションがそのバケットの内容にアクセスできなくなります。

データ分離

RES に S3 バケットを追加する場合、バケット内のデータを特定のプロジェクトとユーザーに分離するオプションがあります。バケットの追加ページで、読み取り専用 (R) または読み取り/書き込み (R/W) のモードを選択できます。

読み取り専用

Read Only (R) を選択した場合、バケット ARN (Amazon リソースネーム) のプレフィックスに基づいてデータ分離が適用されます。たとえば、管理者が ARN を使用して RES にバケットを追加 `arn:aws:s3:::bucket-name/example-data`/し、このバケットをプロジェクト A とプロジェクト B に関連付けると、プロジェクト A とプロジェクト B 内から VDI を起動するユーザーは、`/example-data` パスの `bucket-name` にあるデータのみを読み取ることができます。そのパス外のデータにはアクセスできません。バケット ARN にプレフィックスが付加されていない場合、バケット全体がそれに関連付けられたプロジェクトで利用可能になります。

読み取りと書き込み

Read and Write (R/W) を選択した場合でも、上記のように、バケット ARN のプレフィックスに基づいてデータ分離が適用されます。このモードには、管理者が S3 バケットに変数ベースのプレフィックスを提供できるようにする追加オプションがあります。Read and Write (R/W) を選択すると、カスタムプレフィックスセクションが利用可能になり、以下のオプションを含むドロップダウンメニューが表示されます。

- カスタムプレフィックスなし
- `/%p`
- `/%p/%u`

カスタムデータ分離なし

カスタムプレフィックスに No custom prefix を選択すると、バケットはカスタムデータ分離なしで追加されます。これにより、バケットに関連付けられたすべてのプロジェクトに読み取りおよび書き込みアクセスが許可されます。例えば、管理者が `arn:aws:s3:::bucket-name` No custom prefix 選択した ARN を使用して RES にバケットを追加し、このバケットをプロジェクト A とプロジェクト B に関連付けると、プロジェクト A とプロジェクト B 内から VDI を起動するユーザーは、バケットへの無制限の読み取りおよび書き込みアクセスが可能になります。

プロジェクトレベルごとのデータ分離

カスタムプレフィックスに `/%p` を選択すると、バケット内のデータは、それに関連付けられた特定のプロジェクトごとに分離されます。`%p` 変数はプロジェクトコードを表します。例えば、管理者が `arn:aws:s3:::bucket-name/%p` 選択したと `/bucket` のマウントポイントを使用して RES にバケットを追加し、このバケットをプロジェクト A とプロジェクト B に関連付けると、プロジェクト A のユーザー A は `/bucket` にファイルを書き込むことができます。プロジェクト A のユーザー B は、ユーザー A が `/bucket` で書き込んだファイルを表示することもできます。ただし、ユーザー B がプロジェクト B で VDI を起動し、`/bucket` を検索すると、データがプロジェクトによって分離されるため、ユーザー A が作成したファイルが表示されません。ユーザー A が書き込んだファイルは、プレフィックスの下に S3 バケットにあります `/ProjectA` が、ユーザー B はプロジェクト B から VDI を使用する `/ProjectB` 場合にのみアクセスできます。

プロジェクトごと、ユーザーごとのデータ分離

カスタムプレフィックスに `/%p/%u` を選択すると、バケット内のデータは、そのプロジェクトに関連付けられた特定のプロジェクトとユーザーに分離されます。`%p` 変数はプロジェクトコードを表し、ユーザー名 `%u` を表します。たとえば、管理者は `/%p/%u`、選択したと `/bucket` のマウントポイント `arn:aws:s3:::bucket-name` を持つ ARN を使用して RES にバケットを追加します。このバケットはプロジェクト A とプロジェクト B に関連付けられています。プロジェクト A のユーザー A は `/bucket` にファイルを書き込むことができます。`%p` 分離のみの以前のシナリオとは異なり、この場合のユーザー B には、`/bucket` のプロジェクト A で書き込まれたファイルが表示されません。これは、データがプロジェクトとユーザーの両方によって分離されるためです。ユーザー A が書き込んだファイルはプレフィックスの S3 バケットにあります `/ProjectA/UserA` が、ユーザー B はプロジェクト A で VDI を使用する `/ProjectA/UserB` 場合にのみアクセスできます。

クロスアカウントバケットアクセス

RES は、これらのバケットに適切なアクセス許可がある場合、他の AWS アカウントからバケットをマウントできます。次のシナリオでは、アカウント A の RES 環境がアカウント B に S3 バケットをマウントしたいと考えています。

ステップ 1: RES がデプロイされているアカウントに IAM ロールを作成します（これはアカウント A と呼ばれます）。

1. S3 バケット (アカウント A) へのアクセスを必要とする RES アカウントの AWS マネジメントコンソールにサインインします。
2. IAM コンソールを開きます。
 - a. IAM ダッシュボードに移動します。
 - b. ナビゲーションペインで [ポリシー] を選択します。
3. ポリシーを作成する：
 - a. [Create policy] (ポリシーを作成) を選択します。
 - b. [JSON] タブを選択します。
 - c. 次の JSON ポリシーを貼り付けます (*amzn-s3-demo-bucket* をアカウント B にある S3 バケットの名前に置き換えます)。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:ListBucket",
        "s3:DeleteObject",
        "s3:AbortMultipartUpload"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ]
    }
  ]
}
```

```

    ]
  }
}

```

- d. [次へ] を選択してください。
4. ポリシーを確認して作成します。
 - a. ポリシーの名前を指定します (例: "S3AccessPolicy")。
 - b. ポリシーの目的を説明するオプションの説明を追加します。
 - c. ポリシーを確認し、ポリシーの作成を選択します。
5. IAM コンソールを開きます。
 - a. IAM ダッシュボードに移動します。
 - b. ナビゲーションペインで、[ロール] を選択します。
6. ロールを作成する :
 - a. [Create role] (ロールの作成) を選択します。
 - b. 信頼されたエンティティのタイプとしてカスタム信頼ポリシーを選択します。
 - c. 次の JSON ポリシーを貼り付けます (**111122223333** をアカウント A の実際のアカウント ID、**<ENVIRONMENT_NAME>** を RES デプロイの環境名、**us-east-1** を RES がデプロイされる AWS リージョンに置き換えます)。

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS":
          "arn:aws:iam::111122223333:role/<ENVIRONMENT_NAME>-custom-credential-broker-lambda-role-us-east-1"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

```
}
```

- d. 「次へ」を選択します。
7. アクセス許可ポリシーをアタッチする：
 - a. 前に作成したポリシーを検索して選択します。
 - b. 「次へ」を選択します。
8. ロールのタグ付け、確認、作成：
 - a. ロール名 (S3AccessRole など) を入力します。
 - b. ステップ 3 で、タグの追加を選択し、次のキーと値を入力します。
 - キー: res:Resource
 - 値: s3-bucket-iam-role
 - c. ロールを確認し、ロールの作成を選択します。
9. RES で IAM ロールを使用します。
 - a. 作成した IAM ロール ARN をコピーします。
 - b. RES コンソールにログインします。
 - c. 左側のナビゲーションペインで、S3 バケットを選択します。
 - d. バケットの追加を選択し、フォームにクロスアカウントの S3 バケット ARN を入力します。
 - e. 詳細設定 - オプションのドロップダウンを選択します。
 - f. IAM ロール ARN フィールドにロール ARN を入力します。
 - g. バケットの追加 を選択します。

ステップ 2: アカウント B でバケットポリシーを変更する

1. アカウント B の AWS マネジメントコンソールにサインインします。
2. S3 コンソールを開きます。
 - a. S3 ダッシュボードに移動します。
 - b. アクセスを許可するバケットを選択します。
3. バケットポリシーを編集します。
 - a. アクセス許可タブを選択し、バケットポリシーを選択します。

- b. 次のポリシーを追加して、アカウント A の IAM ロールにバケットへのアクセスを許可します (**111122223333** をアカウント A の実際のアカウント ID に置き換え、**amzn-s3-demo-bucket** を S3 バケットの名前に置き換えます)。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/S3AccessRole"
      },
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:ListBucket",
        "s3:DeleteObject",
        "s3:AbortMultipartUpload"
      ],
      "Resource": [
        "arn:aws:s3::amzn-s3-demo-bucket",
        "arn:aws:s3::amzn-s3-demo-bucket/*"
      ]
    }
  ]
}
```

- c. [保存] を選択します。

プライベート VPC でのデータ流出の防止

ユーザーが安全な S3 バケットからアカウント内の独自の S3 バケットにデータを流出しないようにするには、VPC エンドポイントをアタッチしてプライベート VPC を保護します。次の手順は、アカウント内の S3 バケットへのアクセスをサポートする S3 サービスの VPC エンドポイントと、クロスアカウントバケットを持つ追加のアカウントを作成する方法を示しています。

1. Amazon VPC コンソールを開きます。
 - a. AWS マネジメントコンソールにサインインします。

- b. <https://console.aws.amazon.com/vpc/> で Amazon VPC コンソールを開きます。
2. S3 の VPC エンドポイントを作成する :
 - a. 左側のナビゲーションペインで、エンドポイントを選択します。
 - b. [エンドポイントの作成] を選択します。
 - c. [Service category] (サービスカテゴリ) で、[AWS services] (AWS のサービス) が選択されていることを確認します。
 - d. サービス名フィールドに「」と入力するか `com.amazonaws.<region>.s3` (<region> を自分の AWS リージョンに置き換えます)、「S3」を検索します。
 - e. リストから S3 サービスを選択します。
3. エンドポイント設定の構成 :
 - a. VPC の場合は、エンドポイントを作成する VPC を選択します。
 - b. サブネットの場合は、デプロイ中に VDI サブネットに使用されるプライベートサブネットの両方を選択します。
 - c. DNS 名を有効にする で、オプションがオンになっていることを確認します。これにより、プライベート DNS ホスト名をエンドポイントネットワークインターフェイスに解決できます。
4. アクセスを制限するように ポリシーを設定します。
 - a. ポリシー で、カスタム を選択します。
 - b. ポリシーエディタで、アカウントまたは特定のアカウント内のリソースへのアクセスを制限するポリシーを入力します。ポリシーの例を次に示します (`amzn-s3-demo-bucket` を S3 バケット名に置き換え、`111122223333` と `444455556666` をアクセスする適切な AWS アカウント IDs に置き換えます)。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
```

```
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ],
    "Condition": {
        "StringEquals": {
            "aws:PrincipalAccount": [
                "111122223333",
                "444455556666"
            ]
        }
    }
}
```

5. エンドポイントを作成します。

- a. 設定を確認します。
- b. エンドポイントの作成を選択します。

6. エンドポイントを検証する：

- a. エンドポイントが作成されたら、VPC コンソールのエンドポイントセクションに移動します。
- b. 新しく作成したエンドポイントを選択します。
- c. 状態が使用可能であることを確認します。

これらのステップに従って、アカウントまたは指定されたアカウント ID 内のリソースに制限された S3 アクセスを許可する VPC エンドポイントを作成します。

トラブルシューティング

バケットが VDI へのマウントに失敗したかどうかを確認する方法

バケットが VDI へのマウントに失敗した場合、エラーをチェックできる場所がいくつかあります。以下のステップに従います。

1. VDI ログを確認します。
 - a. AWS マネジメントコンソールにログインします。
 - b. EC2 コンソールを開き、インスタンスに移動します。

- c. 起動した VDI インスタンスを選択します。
- d. Session Manager を介して VDI に接続します。
- e. 以下の コマンドを実行します。

```
sudo su
cd ~/bootstrap/logs
```

ここでは、ブートストラップログを確認できます。障害の詳細は `configure.log`. `{time}` ファイルにあります。

さらに、詳細については `/etc/message` ログを確認してください。

2. カスタム認証情報ブローカーの Lambda CloudWatch Logs を確認する :
 - a. AWS マネジメントコンソールにログインします。
 - b. CloudWatch コンソールを開き、ロググループに移動します。
 - c. ロググループ を検索します `/aws/lambda/<stack-name>-vdc-custom-credential-broker-lambda`。
 - d. 最初に使用可能なロググループを調べ、ログ内のエラーを見つけます。これらのログには、S3 バケットをマウントするための一時的なカスタム認証情報を提供する潜在的な問題に関する詳細が含まれます。
3. カスタム認証情報ブローカー API Gateway CloudWatch Logs を確認します。
 - a. AWS マネジメントコンソールにログインします。
 - b. CloudWatch コンソールを開き、ロググループに移動します。
 - c. ロググループ を検索します `<stack-name>-vdc-custom-credential-broker-lambda vdc custom credential broker api gateway access logs <nonce>`。
 - d. 最初に使用可能なロググループを調べ、ログ内のエラーを見つけます。これらのログには、S3 バケットのマウントに必要なカスタム認証情報の API Gateway へのリクエストとレスポンスに関する詳細が含まれます。

オンボーディング後にバケットの IAM ロール設定を編集する方法

1. [AWS DynamoDB コンソール](#)にサインインします。
2. テーブルを選択します。
 - a. 左側のナビゲーションペインで、テーブルを選択します。

- b. を検索して選択します `<stack-name>.cluster-settings`。
3. テーブルをスキャンします。
 - a. Explore table items を選択します。
 - b. スキャンが選択されていることを確認します。
4. フィルターを追加する：
 - a. フィルターを選択してフィルターエントリセクションを開きます。
 - b. キーと一致するようにフィルターを設定します。
 - 属性: キーを入力します。
 - 条件: Begins with を選択します。
 - 値: `shared-storage.<filesystem_id>.s3_bucket.iam_role_arn`
`<filesystem_id>` を変更する必要があるファイルシステムの値に置き換えます。
5. スキャンを実行します。

Run を選択して、フィルターを使用してスキャンを実行します。

6. 値を確認します。

エントリが存在する場合は、適切な IAM ロール ARN で値が正しく設定されていることを確認します。

エントリが存在しない場合：

- a. 項目の作成を選択します。
 - b. 項目の詳細を入力します。
 - key 属性には、 と入力します `shared-storage.<filesystem_id>.s3_bucket.iam_role_arn`。
 - 正しい IAM ロール ARN を追加します。
 - c. 保存を選択して項目を追加します。
7. VDI インスタンスを再起動します。

インスタンスを再起動して、誤った IAM ロール ARN の影響を受ける VDI が再度マウントされるようにします。

CloudTrail の有効化

CloudTrail コンソールを使用してアカウントで CloudTrail を有効にするには、「CloudTrail AWS ユーザーガイド CloudTrail」の[CloudTrail コンソールを使用した証跡の作成](#)に記載されている手順に従ってください。CloudTrail は、S3 バケットにアクセスした IAM ロールを記録することで、S3 バケットへのアクセスを記録します。これは、プロジェクトまたはユーザーにリンクされたインスタンス ID にリンクできます。

シークレットの管理

Research and Engineering Studio は、を使用して次のシークレットを維持します AWS Secrets Manager。RES は、環境の作成中にシークレットを自動的に作成します。環境の作成中に管理者が入力したシークレットはパラメータとして入力されます。

シークレット名	説明	生成された RES	入力された管理者
<envname>-sso-client-secret	環境用のシングルサインオン OAuth2 クライアントシークレット	✓	
<envname>-vdc-client-secret	vdc ClientSecret	✓	
<envname>-vdc-client-id	vdc ClientId	✓	
<envname>-vdc-gateway-certificate-private-key	ドメインの自己署名証明書プライベートキー	✓	
<envname>-vdc-gateway-certificate-certificate	ドメインの自己署名証明書	✓	
<envname>-cluster-manager-client-secret	クラスターマネージャー ClientSecret	✓	

シークレット名	説明	生成された RES	入力された管理者
<envname>-cluster-manager-client-id	クラスターマネージャー ClientId	✓	
<envname>-external-private-key	ドメインの自己署名証明書プライベートキー	✓	
<envname>-external-certificate	ドメインの自己署名証明書	✓	
<envname>-internal-private-key	ドメインの自己署名証明書プライベートキー	✓	
<envname>-internal-certificate	ドメインの自己署名証明書	✓	
<envname>-director-yservice-ServiceAccountUsername			✓
<envname>-director-yservice-ServiceAccountPassword			✓

次のシークレット ARN 値は、DynamoDB の <envname>-cluster-settings テーブルに含まれていません。

キー	ソース
identity-provider.cognito.sso_client_secret	
vdc.dcv_connection_gateway.certificate.certificate_secret_arn	スタック
vdc.dcv_connection_gateway.certificate.private_key_secret_arn	スタック

キー	ソース
cluster.load_balancers.internal_alb.certificates.private_key_secret_arn	スタック
directoryservice.root_username_secret_arn	
vdc.client_secret	スタック
cluster.load_balancers.external_alb.certificates.certificate_secret_arn	スタック
cluster.load_balancers.internal_alb.certificates.certificate_secret_arn	スタック
directoryservice.root_password_secret_arn	
cluster.secretsmanager.kms_key_id	
cluster.load_balancers.external_alb.certificates.private_key_secret_arn	スタック
cluster-manager.client_secret	

コストのモニタリングと制御

Note

Research and Engineering Studio プロジェクトをに関連付ける AWS Budgets ことは、ではサポートされていません AWS GovCloud (US)。

Cost [AWS Cost Explorer](#)を使用して**予算**を作成し、コストを管理することをお勧めします。価格は変更されることがあります。詳細については、各の料金ウェブページを参照してください [the section called “AWS この製品の サービス”](#)。

コスト追跡を支援するために、RES プロジェクトを内部で作成された予算に関連付けることができます AWS Budgets。まず、請求コスト配分タグ内で環境タグをアクティブ化する必要があります。

1. にサインイン AWS Management Console し、<https://console.aws.amazon.com/costmanagement/> で AWS Billing and Cost Management コンソールを開きます。
2. コスト配分タグを選択します。
3. タグと `res:EnvironmentName` タグを検索 `res:Project` して選択します。
4. [アクティブ化] を選択します。

 Note

RES タグがデプロイ後に表示されるまでに最大 1 日かかる場合があります。

RES リソースの予算を作成するには：

1. 請求コンソールから、予算を選択します。
2. 予算の作成を選択します。
3. [Budget setup] (予算の設定) で、[Customize (advanced)] (カスタマイズ (高度)) を選択します。
4. Budget types で、Cost budget - Recommended を選択します。
5. [次へ] を選択します。
6. 詳細 に、予算のわかりやすい Budget 名を入力して、アカウントの他の予算と区別します。たとえば、[EnvironmentName]-[ProjectName]-[BudgetName] です。
7. 予算額を設定する に、プロジェクトの予算額を入力します。
8. Budget scope で、Filter specific AWS cost dimensions を選択します。
9. [Add filter] (フィルターを追加) を選択します。
10. ディメンション で、タグ を選択します。
11. タグで、`res:Project` を選択します。

 Note

タグと値が使用可能になるまでに最大 2 日かかる場合があります。プロジェクト名が使用可能になったら、予算を作成できます。

12. 値 で、プロジェクト名を選択します。

13. フィルターを適用 を選択して、プロジェクトフィルターを予算にアタッチします。
14. [次へ] を選択します。
15. (オプション) アラートしきい値を追加します。
16. [次へ] を選択します。
17. (オプション) アラートが設定されている場合は、アタッチアクションを使用して、アラートで目的のアクションを設定します。
18. [次へ] を選択します。
19. 予算設定を確認し、追加の予算パラメータで正しいタグが設定されていることを確認します。
20. [予算を作成] をクリックします。

予算が作成されたら、プロジェクトの予算を有効にできます。プロジェクトの予算を有効にするには、「」を参照してください[the section called “プロジェクトを編集する”](#)。予算を超えると、仮想デスクトップの起動がブロックされます。デスクトップの起動中に予算を超えた場合、デスクトップは引き続き動作します。

予算を変更する必要がある場合は、コンソールに戻って予算額を編集します。RES 内で変更が有効になるまでに最大 15 分かかる場合があります。または、プロジェクトを編集して予算を無効にすることもできます。

製品を使用する

このセクションでは、仮想デスクトップを使用して他のユーザーとコラボレーションするためのガイドをユーザーに提供します。

トピック

- [仮想デスクトップ](#)
- [共有デスクトップ](#)
- [ファイルブラウザ](#)
- [SSH アクセス](#)

仮想デスクトップ

仮想デスクトップインターフェイス (VDI) モジュールを使用すると、ユーザーは Windows または Linux 仮想デスクトップを作成および管理できます AWS。ユーザーは、お気に入りのツールとアプリケーションがプリインストールおよび設定された状態で Amazon EC2 インスタンスを起動できます。

サポートされるオペレーティングシステム

RES は現在、次のオペレーティングシステムを使用した仮想デスクトップの起動をサポートしています。

- Amazon Linux 2 (x86 および ARM64)
- Ubuntu 22.04.03 (x86)
- Windows 2019、2022 (x86)

新しいデスクトップを起動する

1. メニューから、仮想デスクトップを選択します。
2. 新しい仮想デスクトップを起動を選択します。
3. 新しいデスクトップの詳細を入力します。

4. [Submit] を選択してください。

デスクトップ情報を含む新しいカードがすぐに表示され、デスクトップは 10～15 分以内に使用できるようになります。起動時間は、選択したイメージによって異なります。RES は GPU インスタンスを検出し、関連するドライバーをインストールします。

デスクトップにアクセスする

仮想デスクトップにアクセスするには、デスクトップのカードを選択し、ウェブクライアントまたは DCV クライアントを使用して接続します。

Web connection

ウェブブラウザからデスクトップにアクセスするのが最も簡単な接続方法です。

- Connect を選択するか、サムネイルを選択してブラウザから直接デスクトップにアクセスします。

DCV connection

DCV クライアントを介してデスクトップにアクセスすると、最高のパフォーマンスが得られます。DCV 経由で にアクセスするには：

1. DCV セッションファイルを選択して、.dcv ファイルをダウンロードします。DCV クライアントがシステムにインストールされている必要があります。
2. インストール手順については、 ? アイコンを選択します。

デスクトップの状態を制御する

デスクトップの状態を制御するには：

1. [アクション] を選択します。
2. Virtual Desktop State を選択します。次の 4 つの状態から選択できます。
 - 停止

停止したセッションではデータが失われることはなく、停止したセッションはいつでも再開できます。

- 再起動

現在のセッションを再起動します。

- 終了

セッションを完全に終了します。エフェメラルストレージを使用している場合、セッションを終了するとデータが失われる可能性があります。終了する前に、データを RES ファイルシステムにバックアップする必要があります。

- 休止

デスクトップの状態はメモリに保存されます。デスクトップを再起動すると、アプリケーションは再開されますが、リモート接続が失われる可能性があります。すべてのインスタンスが休止をサポートしているわけではなく、オプションはインスタンスの作成時に有効になっている場合にのみ使用できます。インスタンスがこの状態をサポートしているかどうかを確認するには、[「休止の前提条件」](#)を参照してください。

仮想デスクトップを変更する

仮想デスクトップのハードウェアを更新するか、セッション名を変更できます。

1. インスタンスサイズを変更する前に、セッションを停止する必要があります。
 - a. [アクション] を選択します。
 - b. Virtual Desktop State を選択します。
 - c. [停止] を選択します。

 Note

休止したセッションのデスクトップサイズは更新できません。

2. デスクトップが停止したことを確認したら、アクションを選択し、セッションの更新を選択します。
3. セッション名を変更するか、必要なデスクトップサイズを選択します。
4. [Submit] を選択してください。

5. インスタンスが更新されたら、デスクトップを再起動します。

- a. [アクション] を選択します。
- b. Virtual Desktop State を選択します。
- c. [開始] を選択します。

セッション情報を取得する

1. [アクション] を選択します。
2. 情報の表示を選択します。

仮想デスクトップをスケジュールする

デフォルトでは、仮想デスクトップにはスケジュールがなく、セッションを停止または終了するまでアクティブのままになります。デスクトップは、誤って停止しないようにアイドル状態でも停止します。アイドル状態は、アクティブな接続がなく、CPU 使用率が少なくとも 15 分間 15% 未満であることによって決まります。デスクトップを自動的に起動および停止するようにスケジュールを設定できます。

1. [アクション] を選択します。
2. [スケジュール] を選択します。
3. 各日のスケジュールを設定します。
4. [保存] を選択します。

共有デスクトップ

共有デスクトップでは、共有されているデスクトップを確認できます。デスクトップに接続するには、管理者または所有者でない限り、セッション所有者も接続されている必要があります。

セッションの共有中に、共同作業者のアクセス許可を設定できます。たとえば、コラボレーションしているチームメイトに読み取り専用アクセス権を付与できます。

デスクトップを共有する

1. デスクトップセッションから、アクションを選択します。
2. セッションのアクセス許可を選択します。
3. ユーザーとアクセス許可レベルを選択します。有効期限を設定することもできます。
4. [保存] を選択します。

アクセス許可の詳細については、「」を参照してください[the section called “アクセス許可プロファイル”](#)。

共有デスクトップにアクセスする

共有デスクトップから、共有されているデスクトップを表示し、インスタンスに接続できます。ウェブブラウザまたは DCV で参加できます。接続するには、「」の指示に従います[the section called “デスクトップにアクセスする”](#)。

ファイルブラウザ

ファイルブラウザを使用すると、ウェブポータルからファイルシステムにアクセスできます。基盤となるファイルシステムへのアクセス許可を持つ使用可能なすべてのファイルを管理できます。バックエンドストレージ (Amazon EFS) は、すべての Linux ノードで使用できます。Linux ノードと Windows ノードでは、FSx for ONTAP を使用できます。仮想デスクトップ上のファイルの更新は、ターミナルまたはウェブベースのファイルブラウザを介したファイルの更新と同じです。

ファイルのアップロード (複数可)

1. [ファイルをアップロード] を選択します。
2. ファイルを削除するか、アップロードするファイルを参照します。
3. アップロード (n) ファイルを選択します。

ファイルの削除 (複数可)

1. 削除するファイル (複数可) を選択します。

2. [アクション] を選択します。
3. [ファイルを削除] を選択します。

または、任意のファイルまたはフォルダを右クリックし、ファイルの削除を選択することもできます。

お気に入りを管理する

重要なファイルやフォルダを固定するには、お気に入りに追加します。

1. ファイルまたはフォルダを選択します。
2. お気に入り を選択します。

または、任意のファイルまたはフォルダを右クリックして、お気に入り を選択することもできます。

Note

お気に入りはローカルブラウザに保存されます。ブラウザを変更したり、キャッシュをクリアしたりする場合は、お気に入りを再ピン留めする必要があります。

ファイルを編集する

ウェブポータル内のテキストベースのファイルのコンテンツを編集できます。

1. 更新するファイルを選択します。モーダルが開き、ファイルの内容が表示されます。
2. 更新を行い、保存を選択します。

ファイルの転送

ファイル転送を使用して、外部ファイル転送アプリケーションを使用してファイルを転送します。次のアプリケーションから選択し、画面の指示に従ってファイルを転送できます。

- FileZilla (Windows、MacOS、Linux)
- WinSCP (Windows)
- AWS Transfer for FTP (Amazon EFS)

SSH アクセス

SSH を使用して踏み台ホストにアクセスするには：

1. RES メニューから、SSH アクセスを選択します。
2. アクセスに SSH または PuTTY を使用するには、画面の指示に従います。

トラブルシューティング

このセクションでは、システムをモニタリングする方法と、発生する可能性のある特定の問題のトラブルシューティング方法について説明します。

トピック

- [一般的なデバッグとモニタリング](#)
- [RunBooks の問題](#)
- [既知の問題](#)

詳細な内容：

- [一般的なデバッグとモニタリング](#)
 - [便利なログおよびイベント情報ソース](#)
 - [環境 Amazon EC2 インスタンスのログファイル](#)
 - [CloudFormation スタック](#)
 - [問題によるシステム障害と Amazon EC2 Auto Scaling グループアクティビティによる反映](#)
 - [一般的な Amazon EC2 コンソールの外観](#)
 - [インフラストラクチャホスト](#)
 - [インフラストラクチャホストと仮想デスクトップ](#)
 - [終了状態のホスト](#)
 - [参照に便利な Active Directory \(AD\) 関連のコマンド](#)
 - [Windows DCV デバッグ](#)
 - [NICE DCV バージョン情報の検索](#)
- [RunBooks の問題](#)
 - [インストールの問題](#)
 - [AWS CloudFormation スタックはWaitCondition received failed message」というメッセージで作成に失敗します。エラー:States.TaskFailed"](#)
 - [スタックが正常に作成された後に AWS CloudFormation E メール通知が受信されない](#)
 - [インスタンスサイクルまたは vdc-controller が失敗状態](#)
 - [環境 CloudFormation スタックが依存オブジェクトエラーにより削除に失敗する](#)
 - [環境の作成中に CIDR ブロックパラメータでエラーが発生しました](#)

- [環境作成中の CloudFormation スタック作成の失敗](#)
- [AdDomainAdminNode CREATE_FAILED で外部リソース \(デモ\) スタックの作成が失敗する](#)
- [ID 管理の問題](#)
 - [iam:PassRole を実行する権限がありません](#)
 - [自分の AWS アカウント以外のユーザーに リソースの AWS Research and Engineering Studio へのアクセスを許可したい](#)
 - [環境にログインすると、すぐにログインページに戻ります。](#)
 - [ログイン試行時の「ユーザーが見つかりません」エラー](#)
 - [Active Directory に追加されたが、RES に欠落しているユーザー](#)
 - [セッションの作成時にユーザーが使用できない](#)
 - [CloudWatch クラスターマネージャーログのサイズ制限超過エラー](#)
- [\[Storage \(ストレージ\)\]](#)
 - [RES を使用してファイルシステムを作成しましたが、VDI ホストにマウントされません](#)
 - [RES を介してファイルシステムをオンボードしたが、VDI ホストにマウントされない](#)
 - [VDI ホストから読み書きできない](#)
 - [アクセス許可処理のユースケースの例](#)
 - [RES から Amazon FSx for NetApp ONTAP を作成しましたが、ドメインに参加していません](#)
- [スナップショット](#)
 - [スナップショットのステータスが Failed である](#)
 - [スナップショットは、テーブルをインポートできなかったことを示すログとともに適用されません。](#)
- [インフラストラクチャ](#)
 - [正常なインスタスがないロードバランサーターゲットグループ](#)
- [仮想デスクトップの起動](#)
 - [以前に動作していた仮想デスクトップが正常に接続できなくなりました](#)
 - [5 つの仮想デスクトップしか起動できない](#)
 - [デスクトップ Windows の接続試行は「接続が閉じられました。トランスポートエラー」](#)
 - [VDIsプロビジョニング状態でスタックする](#)
 - [起動後に VDIがエラー状態になる](#)
- [仮想デスクトップコンポーネント](#)
 - [Amazon EC2 インスタンスがコンソールで終了を繰り返し表示](#)

- [AD への参加に失敗したために vdc-controller インスタンスがサイクルしています / eVDI モジュールが失敗した API ヘルスチェックを表示](#)
- [プロジェクトは、ソフトウェアスタックを編集して追加するときにプルダウンに表示されません](#)
- [cluster-manager Amazon CloudWatch ログには、「<user-home-init> アカウントはまだ利用できません。ユーザーの同期を待っています」と表示されます \(アカウントはユーザー名です\)。](#)
- [ログイン試行時の Windows デスクトップに「アカウントが無効になっています。管理者にお問い合わせください」](#)
- [外部/顧客の AD 設定に関する DHCP オプションの問題](#)
- [Firefox エラー MOZILLA_PKIX_ERROR_REQUIRED_TLS_FEATURE_MISSING](#)
- [Env 削除](#)
 - [res-xxx-cluster スタックが「DELETE_FAILED」状態で、「Role is invalid or cannot be assumed」エラーのため手動で削除できない](#)
- [ログの収集](#)
- [VDI ログのダウンロード](#)
- [Linux EC2 インスタンスからのログのダウンロード](#)
- [Windows EC2 インスタンスからのログのダウンロード](#)
- [WaitCondition エラーの ECS ログの収集](#)
- [デモ環境](#)
 - [ID プロバイダーへの認証リクエストを処理する際のデモ環境ログインエラー](#)
- [既知の問題 2024.x](#)
 - [既知の問題 2024.x](#)
 - [\(2024.06\) AD グループ名にスペースが含まれているとスナップショットの適用が失敗する](#)
 - [\(2024.04-2024.04.02\) VDI インスタンスのロールにアタッチされていない IAM アクセス許可境界が提供されました](#)
 - [\(2024.04.02 以前\) ap-southeast-2 \(シドニー\) の Windows NVIDIA インスタンスが起動に失敗する](#)
 - [\(2024.04 および 2024.04.01\) GovCloud での RES 削除の失敗](#)
 - [\(2024.04 - 2024.04.02\) Linux 仮想デスクトップは再起動時に「RESUMING」ステータスのままになる可能性があります](#)

- [\(2024.04.02 以前\) SAMAccountName 属性に大文字または特殊文字が含まれている AD ユーザーの同期に失敗しました](#)
- [\(2024.04.02 以前\) 踏み台ホストにアクセスするためのプライベートキーが無効です](#)
- [\(2024.06 以前\) AD 同期中に RES に同期されていないグループメンバー](#)
- [\(2024.06 以前\) CVE-2024-6387、RegreSSHion、RHEL9 および Ubuntu VDI のセキュリティ脆弱性](#)

一般的なデバッグとモニタリング

このセクションでは、RES 内の情報がどこにあるかについて説明します。

- [便利なログおよびイベント情報ソース](#)
 - [環境 Amazon EC2 インスタンスのログファイル](#)
 - [CloudFormation スタック](#)
 - [問題によるシステム障害と Amazon EC2 Auto Scaling グループアクティビティによる反映](#)
- [一般的な Amazon EC2 コンソールの外観](#)
 - [インフラストラクチャホスト](#)
 - [インフラストラクチャホストと仮想デスクトップ](#)
 - [終了状態のホスト](#)
 - [参照に便利な Active Directory \(AD\) 関連のコマンド](#)
- [Windows DCV デバッグ](#)
- [NICE DCV バージョン情報の検索](#)

便利なログおよびイベント情報ソース

保持される情報のさまざまなソースは、トラブルシューティングやモニタリングの用途で参照できます。

環境 Amazon EC2 インスタンスのログファイル

ログファイルは、RES で使用されている Amazon EC2 インスタンスに存在します。SSM セッションマネージャーを使用して、これらのファイルを調べるためにインスタンスへのセッションを開くことができます。

cluster-manager や vdc-controller などのインフラストラクチャインスタンスでは、アプリケーションやその他のログは次の場所にあります。

- /opt/idea/app/logs/application.log
- /root/bootstrap/logs/
- /var/log/
- /var/log/sssdl/
- /var/log/messages
- /var/log/user-data.log
- /var/log/cloud-init.log
- /var/log/cloud-init-output.log

Linux 仮想デスクトップでは、以下には便利なログファイルが含まれています。

- /var/log/dcv/
- /root/bootstrap/logs/userdata.log
- /var/log/messages

Windows 仮想デスクトップインスタンスのログについては、「」を参照してください。

- PS C:\ProgramData\nice\dcv\log
- PS C:\ProgramData\nice\DCVSessionManagerAgent\log

Windows では、一部のアプリケーションのログ記録は次の場所にあります。

- PS C:\Program Files\NICE\DCV\Server\bin

Windows では、NICE DCV 証明書ファイルは以下にあります。

- C:\Windows\System32\config\systemprofile\AppData\Local\NICE\dcv\

Amazon CloudWatch ロググループ

Amazon EC2 と AWS Lambda コンピューティングリソースは、Amazon CloudWatch Log Groups に情報をログに記録します。ログエントリ内のログエントリは、潜在的な問題のトラブルシューティングや一般的な情報に役立つ情報を提供します。

これらのグループの名前は次のとおりです。

- `/aws/lambda/<envname>-/` - lambda related
- `/<envname>/`
 - `cluster-manager/` - main infrastructure host
 - `vdc/` - virtual desktop related
 - `dcv-broker/` - desktop related
 - `dcv-connection-gateway/` - desktop related
 - `controller/` - main desktop controller host
 - `dcv-session/` - desktop session related

ロググループを調べるときは、次のような大文字と小文字の文字列を使用してフィルタリングすると便利です。これにより、メモされた文字列を含むメッセージのみが出力されます。

```
? "ERROR" ? "error"
```

問題をモニタリングするもう 1 つの方法は、対象のデータを表示するウィジェットを含む Amazon CloudWatch Dashboards を作成することです。

たとえば、文字列エラーと ERROR の発生をカウントするウィジェットを作成し、線としてグラフ化します。この方法により、パターン変更が発生したことを示す潜在的な問題や傾向の出現を簡単に検出できます。

インフラストラクチャホストの の例を次に示します。これを使用するには、クエリ行を連結し、`<envname>` および `<region>` 属性を適切な値に置き換えます。

```
{
  "widgets": [
    {
      "type": "log",
      "x": 0,
      "y": 0,
      "width": 24,
      "height": 6,
```

```

    "properties": {
      "query": "SOURCE '/<envname>/vdc/controller' |
        SOURCE '/<envname>/cluster-manager' |
        SOURCE '/<envname>/vdc/dcv-broker' |
        SOURCE '/<envname>/vdc/dcv-connection-gateway' |
        fields @timestamp, @message, @logStream, @log\n|
        filter @message like /(?(i)(error|ERROR))/\n|
        sort @timestamp desc|
        stats count() by bin(30s)",
      "region": "<region>",
      "title": "infrastructure hosts",
      "view": "timeSeries",
      "stacked": false
    }
  }
}

```

ダッシュボードの例を次に示します。

CloudFormation スタック

環境の作成時に作成された CloudFormation スタックには、環境の設定に関連するリソース、イベント、出力情報が含まれます。

スタックごとに、イベント、リソース、出力タブを参照してスタックに関する情報を確認できます。

RES スタック：

- <envname>-bootstrap
- <envname>-cluster
- <envname>-metrics
- <envname>-directoryservice
- <envname>-identity-provider
- <envname>-shared-storage
- <envname>-cluster-manager
- <envname>-vdc
- <envname>-踏み台ホスト

デモ環境スタック (デモ環境をデプロイしていて、これらの外部リソースを利用できない場合は、AWS ハイパフォーマンスコンピューティングレシピを使用してデモ環境のリソースを生成できます)。

- <envname>
- <envname>-Networking
- <envname>-DirectoryService
- <envname>-Storage
- <envname>-WindowsManagementHost

問題によるシステム障害と Amazon EC2 Auto Scaling グループアクティビティによる反映

RES UIs がサーバーエラーを示している場合、原因はアプリケーションソフトウェアやその他の問題である可能性があります。

各インフラストラクチャの Amazon EC2 インスタンスの自動スケーリンググループ (ASGs) には、インスタンスのスケーリングアクティビティを検出するのに役立つアクティビティタブが含まれています。UI ページにエラーがある場合、またはアクセスできない場合は、Amazon EC2 コンソールで複数の終了したインスタンスをチェックし、関連する ASG の Auto Scaling グループアクティビティタブをチェックして、Amazon EC2 インスタンスが循環しているかどうかを確認します。

その場合は、インスタンスの関連する Amazon CloudWatch ロググループを使用して、問題の原因を示す可能性のあるエラーがログに記録されているかどうかを確認します。SSM セッションコンソールを使用して、そのタイプの実行中のインスタンスへのセッションを開き、インスタンスが異常とマークされて ASG によって終了される前に、インスタンスのログファイルを調べて原因を特定することもできます。

この問題が発生した場合、ASG コンソールには次のようなアクティビティが表示されることがあります。

一般的な Amazon EC2 コンソールの外観

このセクションには、さまざまな状態で動作しているシステムのスクリーンショットが含まれています。

インフラストラクチャホスト

Amazon EC2 コンソールでは、デスクトップが実行されていない場合、通常、次のようになります。表示されるインスタンスは、RES インフラストラクチャの Amazon EC2 ホストです。インスタンス名のプレフィックスは RES 環境名です。

インフラストラクチャホストと仮想デスクトップ

Amazon EC2 コンソールでは、仮想デスクトップが実行されていると、次のように表示されます。この場合、仮想デスクトップは赤で表示されます。インスタンス名のサフィックスは、デスクトップを作成したユーザーです。中央の名前は起動時に設定されたセッション名であり、デフォルトの「MyDesktop」またはユーザーが設定した名前です。

終了状態のホスト

Amazon EC2 コンソールに終了したインスタンスが表示されると、通常は終了したデスクトップホストになります。コンソールに終了した状態のインフラストラクチャホストが含まれている場合、特に同じタイプの が複数ある場合は、進行中のシステムの問題を示している可能性があります。

次の図は、終了したデスクトップインスタンスを示しています。

参照に便利な Active Directory (AD) 関連のコマンド

以下は、AD 設定関連情報を表示するためにインフラストラクチャホストに入力できる ldap 関連のコマンドの例です。使用するドメインやその他のパラメータには、環境の作成時に入力されたパラメータを反映する必要があります。

```
ldapsearch "(cn=AWS Delegated Add Workstations To Domain Users)" -x -h corp.res.com
  -b "DC=corp,DC=res,DC=com" -D "CN=Admin,OU=Users,OU=CORP,DC=corp,DC=res,DC=com"
  -w <password>

ldapsearch "(&(objectClass=group))" -x -h corp.res.com
  -b "DC=corp,DC=res,DC=com" -D "CN=Admin,OU=Users,OU=CORP,DC=corp,DC=res,DC=com"
  -w <password>
```

Windows DCV デバッグ

Windows デスクトップでは、以下を使用して関連するセッションを一覧表示できます。

```
PS C:\Windows\System32\config\systemprofile\AppData\Local\NICE\dcv> & 'C:\Program Files\NICE\DCV\Server\bin\dcv.exe' list-sessions
Session: 'a7953489-9dbf-492b-8135-7709dccc4cab' (owner:admin2 type:console
name:windows1)
```

NICE DCV バージョン情報の検索

NICE DCV は仮想デスクトップセッションに使用されます。[AWS NICE DCV](#)。次の例は、インストールされている DCV ソフトウェアのバージョンを確認する方法を示しています。

リナックス

```
[root@ip-10-3-157-194 ~]# /usr/bin/dcv version

NICE DCV 2023.0 (r14852)
Copyright (C) 2010-2023 NICE s.r.l.
All rights reserved.

This product is protected by copyright and
licenses restricting use, copying, distribution, and decompilation.
```

Windows

```
PS C:\Windows\System32\config\systemprofile\AppData\Local\NICE\dcv> & 'C:\Program Files\NICE\DCV\Server\bin\dcv.exe' version

NICE DCV 2023.0 (r15065)
Copyright (C) 2010-2023 NICE s.r.l.
All rights reserved.

This product is protected by copyright and
licenses restricting use, copying, distribution, and decompilation.
```

RunBooks の問題

次のセクションには、発生する可能性のある問題、検出方法、問題の解決方法に関する提案が含まれています。

- [インストールの問題](#)

- AWS CloudFormation スタックはWaitCondition received failed message」というメッセージで作成に失敗します。エラー:States.TaskFailed"
- スタックが正常に作成された後に AWS CloudFormation E メール通知が受信されない
- インスタンスサイクルまたは vdc-controller が失敗状態
- 環境 CloudFormation スタックが依存オブジェクトエラーにより削除に失敗する
- 環境の作成中に CIDR ブロックパラメータでエラーが発生しました
- 環境作成中の CloudFormation スタック作成の失敗
- AdDomainAdminNode CREATE_FAILED で外部リソース (デモ) スタックの作成が失敗する
- ID 管理の問題
 - iam:PassRole を実行する権限がありません
 - 自分の AWS アカウント以外のユーザーに リソースの AWS Research and Engineering Studio へのアクセスを許可したい
 - 環境にログインすると、すぐにログインページに戻ります。
 - ログイン試行時の「ユーザーが見つかりません」エラー
 - Active Directory に追加されたが、RES に欠落しているユーザー
 - セッションの作成時にユーザーが使用できない
 - CloudWatch クラスタマネージャーログのサイズ制限超過エラー
- [Storage (ストレージ)]
 - RES を使用してファイルシステムを作成しましたが、VDI ホストにマウントされません
 - RES を介してファイルシステムをオンボードしたが、VDI ホストにマウントされない
 - VDI ホストから読み書きできない
 - アクセス許可処理のユースケースの例
 - RES から Amazon FSx for NetApp ONTAP を作成しましたが、ドメインに参加していません
- スナップショット
 - スナップショットのステータスが Failed である
 - スナップショットは、テーブルをインポートできなかったことを示すログとともに適用されません。
- インフラストラクチャ
 - 正常なインスタンスがないロードバランサーターゲットグループ
- 仮想デスクトップの起動
 - 以前に動作していた仮想デスクトップが正常に接続できなくなりました

- [5 つの仮想デスクトップしか起動できない](#)
- [デスクトップ Windows の接続試行は「接続が閉じられました。トランスポートエラー」](#)
- [VDIsプロビジョニング状態でスタックする](#)
- [起動後に VDIがエラー状態になる](#)
- [仮想デスクトップコンポーネント](#)
 - [Amazon EC2 インスタンスがコンソールで終了を繰り返し表示](#)
 - [AD への参加に失敗したために vdc-controller インスタンスがサイクルしています / eVDI モジュールが失敗した API ヘルスチェックを表示](#)
 - [プロジェクトは、ソフトウェアスタックを編集して追加するときにプルダウンに表示されません](#)
 - [cluster-manager Amazon CloudWatch ログには、「<user-home-init> アカウントはまだ利用できません。ユーザーの同期を待っています」と表示されます \(アカウントはユーザー名です\)。](#)
 - [ログイン試行時の Windows デスクトップに「アカウントが無効になっています。管理者にお問い合わせください」](#)
 - [外部/顧客の AD 設定に関する DHCP オプションの問題](#)
 - [Firefox エラー MOZILLA_PKIX_ERROR_REQUIRED_TLS_FEATURE_MISSING](#)
- [Env 削除](#)
 - [res-xxx-cluster スタックが「DELETE_FAILED」状態で、「Role is invalid or cannot be assumed」エラーのため手動で削除できない](#)
 - [ログの収集](#)
 - [VDI ログのダウンロード](#)
 - [Linux EC2 インスタンスからのログのダウンロード](#)
 - [Windows EC2 インスタンスからのログのダウンロード](#)
 - [WaitCondition エラーの ECS ログの収集](#)
- [デモ環境](#)
 - [ID プロバイダーへの認証リクエストを処理する際のデモ環境ログインエラー](#)

インストールの問題

トピック

- [AWS CloudFormation スタックはWaitCondition received failed message」というメッセージで作成に失敗します。エラー:States.TaskFailed"](#)
- [スタックが正常に作成された後に AWS CloudFormation E メール通知が受信されない](#)

- [インスタンスサイクルまたは vdc-controller が失敗状態](#)
- [環境 CloudFormation スタックが依存オブジェクトエラーにより削除に失敗する](#)
- [環境の作成中に CIDR ブロックパラメータでエラーが発生しました](#)
- [環境作成中の CloudFormation スタック作成の失敗](#)
- [AdDomainAdminNode CREATE_FAILED で外部リソース \(デモ\) スタックの作成が失敗する](#)

.....

AWS CloudFormation スタックは「WaitCondition received failed message」というメッセージで作成に失敗します。エラー:States.TaskFailed"

問題を特定するには、という名前の Amazon CloudWatch ロググループを調べます<stack-name>-InstallerTasksCreateTaskDefCreateContainerLogGroup<nonce>-<nonce>。同じ名前のロググループが複数ある場合は、最初に使用可能なロググループを調べます。ログ内のエラーメッセージには、問題に関する詳細情報が表示されます。

 Note

パラメータ値にスペースがないことを確認します。

.....

スタックが正常に作成された後に AWS CloudFormation E メール通知が受信されない

AWS CloudFormation スタックが正常に作成された後に Eメールの招待を受信しなかった場合は、以下を確認します。

1. E メールアドレスパラメータが正しく入力されたことを確認します。

E メールアドレスが正しくないか、アクセスできない場合は、Research and Engineering Studio 環境を削除して再デプロイします。

2. インスタンスのサイクルの証拠については、Amazon EC2 コンソールを確認してください。

<envname> プレフィックスが の Amazon EC2 インスタンスが終了済みとして表示され、新しいインスタンスに置き換えられる場合、ネットワークまたは Active Directory の設定に問題がある可能性があります。

3. High AWS Performance Compute レシピをデプロイして外部リソースを作成した場合は、VPC、プライベートサブネットとパブリックサブネット、およびその他の選択したパラメータがスタックによって作成されたことを確認します。

パラメータのいずれかが正しくない場合は、RES 環境を削除して再デプロイする必要がある場合があります。詳細については、「[製品のアインストール](#)」を参照してください。

4. 独自の外部リソースを使用して製品をデプロイした場合は、ネットワークと Active Directory が予想される設定と一致していることを確認します。

インフラストラクチャインスタンスが Active Directory に正常に参加したことを確認することが重要です。のステップを試 [the section called “インスタンスサイクルまたは vdc-controller が失敗状態”](#)して問題を解決します。

.....

インスタンスサイクルまたは vdc-controller が失敗状態

この問題の最も可能性の高い原因は、リソース (複数可) が Active Directory に接続または参加できないことです。

問題を検証するには：

1. コマンドラインから、vdc-controller の実行中のインスタンスで SSM とのセッションを開始します。
2. `sudo su -` を実行します。
3. `systemctl status sssd` を実行します。

ステータスが非アクティブ、失敗、またはログにエラーが表示される場合、インスタンスは Active Directory に参加できませんでした。

SSM エラーログ

問題を解決するには：

- 同じコマンドラインインスタンスから、`cat /root/bootstrap/logs/userdata.log` を実行してログを調査します。

この問題には、3 つの根本原因のいずれかが考えられます。

根本原因 1: 入力された ldap 接続の詳細が正しくない

ログを見直します。以下が複数回繰り返される場合、インスタンスは Active Directory に参加できませんでした。

```
+ local AD_AUTHORIZATION_ENTRY=
+ [[ -z '' ]]
+ [[ 0 -le 180 ]]
+ local SLEEP_TIME=34
+ log_info '(0 of 180) waiting for AD authorization, retrying in 34 seconds ...'
++ date '+%Y-%m-%d %H:%M:%S,%3N'
+ echo '[2024-01-16 22:02:19,802] [INFO] (0 of 180) waiting for AD authorization,
retrying in 34 seconds ...'
[2024-01-16 22:02:19,802] [INFO] (0 of 180) waiting for AD authorization, retrying in
34 seconds ...
+ sleep 34
+ (( ATTEMPT_COUNT++ ))
```

1. RES スタックの作成中に以下のパラメータ値が正しく入力されたことを確認します。
 - directoryservice.ldap_connection_uri
 - directoryservice.ldap_base
 - directoryservice.users.ou
 - directoryservice.groups.ou
 - directoryservice.sudoers.ou
 - directoryservice.computers.ou
 - directoryservice.name
2. DynamoDB テーブルの誤った値を更新します。テーブルは、テーブルの下で DynamoDB コンソールにあります。テーブル名は `<stack name>.cluster-settings`。
3. テーブルを更新したら、現在環境インスタンスを実行している cluster-manager と vdc-controller を削除します。Auto Scaling は、DynamoDB テーブルの最新の値を使用して新しいインスタンスを起動します。

根本原因 2: ServiceAccount ユーザー名が正しく入力されていない

ログが を返す場合 `Insufficient permissions to modify computer account`、スタックの作成時に入力した ServiceAccount 名が正しくない可能性があります。

1. AWS コンソールから Secrets Manager を開きます。
2. `directoryserviceServiceAccountUsername` を検索します。シークレットは である必要があります `<stack name>-directoryservice-ServiceAccountUsername`。
3. シークレットを開いて詳細ページを表示します。シークレット値 で、シークレット値の取得 を選択し、プレーンテキスト を選択します。
4. 値が更新された場合は、環境の現在実行中の `cluster-manager` インスタンスと `vdc-controller` インスタンスを削除します。自動スケーリングは、Secrets Manager の最新の値を使用して新しいインスタンスを開始します。

根本原因 3: 入力された ServiceAccount パスワードが正しくない

ログに と表示される場合 `Invalid credentials`、スタックの作成時に入力した ServiceAccount パスワードが正しくない可能性があります。

1. AWS コンソールから Secrets Manager を開きます。
2. `directoryserviceServiceAccountPassword` を検索します。シークレットは である必要があります `<stack name>-directoryservice-ServiceAccountPassword`。
3. シークレットを開いて詳細ページを表示します。シークレット値 で、シークレット値の取得 を選択し、プレーンテキスト を選択します。
4. パスワードを忘れた場合、または入力したパスワードが正しいかどうか分からない場合は、Active Directory と Secrets Manager でパスワードをリセットできます。
 - a. でパスワードをリセットするには AWS Managed Microsoft AD :
 - i. AWS コンソールを開き、 に移動します AWS Directory Service。
 - ii. RES ディレクトリのディレクトリ ID を選択し、アクションを選択します。
 - iii. [Reset user password] (ユーザーパスワードをリセットする) を選択します。
 - iv. ServiceAccount ユーザー名を入力します。
 - v. 新しいパスワードを入力し、パスワードのリセットを選択します。
 - b. Secrets Manager でパスワードをリセットするには :

- i. AWS コンソールを開き、Secrets Manager に移動します。
 - ii. `directoryserviceServiceAccountPassword` を検索します。シークレットは `<stack name>-directoryservice-ServiceAccountPassword`。
 - iii. シークレットを開いて詳細ページを表示します。シークレット値 で、シークレット値の取得 を選択し、プレーンテキスト を選択します。
 - iv. [Edit] (編集) を選択します。
 - v. ServiceAccount ユーザーの新しいパスワードを設定し、保存を選択します。
5. 値を更新した場合は、環境の現在実行中の `cluster-manager` インスタンスと `vdc-controller` インスタンスを削除します。Auto Scaling は、最新の値を使用して新しいインスタンスを起動します。

環境 CloudFormation スタックが依存オブジェクトエラーにより削除に失敗する

などの依存オブジェクトエラーが原因で `<env-name>-vdc` CloudFormation スタックの削除が失敗した場合 `vdcvhostsecuritygroup`、コンソールを使用して AWS RES が作成したサブネットまたはセキュリティグループに起動された Amazon EC2 インスタンスが原因である可能性があります。

この問題を解決するには、この方法で起動されたすべての Amazon EC2 インスタンスを検索して終了します。その後、環境の削除を再開できます。

環境の作成中に CIDR ブロックパラメータでエラーが発生しました

環境を作成すると、レスポンスステータスが [FAILED] の CIDR ブロックパラメータにエラーが表示されます。

エラーの例：

```
Failed to update cluster prefix list:
  An error occurred (InvalidParameterValue) when calling the
  ModifyManagedPrefixList operation:
    The specified CIDR (52.94.133.132/24) is not valid. For example, specify a CIDR
    in the following form: 10.0.0.0/16.
```

この問題を解決するために想定される形式は x.x.x.0/24 または x.x.x.0/32 です。

環境作成中の CloudFormation スタック作成の失敗

環境の作成には、一連のリソース作成オペレーションが含まれます。一部のリージョンでは、容量の問題が発生し、CloudFormation スタックの作成が失敗する可能性があります。

この場合、環境を削除し、作成を再試行します。または、別のリージョンで作成を再試行することもできます。

AdDomainAdminNode CREATE_FAILED で外部リソース (デモ) スタックの作成が失敗する

デモ環境スタックの作成が次のエラーで失敗した場合、インスタンスの起動後のプロビジョニング中に Amazon EC2 パッチ適用が予期せず発生した可能性があります。

```
AdDomainAdminNode CREATE_FAILED Failed to receive 1 resource signal(s) within the specified duration
```

失敗の原因を特定するには：

1. SSM ステートマネージャーで、パッチ適用が設定されているかどうか、およびすべてのインスタンスに対して設定されているかどうかを確認します。
2. SSM RunCommand/Automation の実行履歴で、パッチ適用関連の SSM ドキュメントの実行がインスタンスの起動と一致するかどうかを確認します。
3. 環境の Amazon EC2 インスタンスのログファイルで、ローカルインスタンスのログ記録を確認して、プロビジョニング中にインスタンスが再起動したかどうかを確認します。

パッチ適用が原因で問題が発生した場合は、起動から少なくとも 15 分後に RES インスタンスのパッチ適用を遅らせます。

ID 管理の問題

シングルサインオン (SSO) と ID 管理のほとんどの問題は、設定ミスが原因で発生します。SSO 設定の設定については、以下を参照してください。

- [the section called “IAM アイデンティティセンターでの SSO の設定”](#)
- [the section called “シングルサインオン \(SSO\) 用の ID プロバイダーの設定”](#)

ID 管理に関連するその他の問題をトラブルシューティングするには、以下のトラブルシューティングトピックを参照してください。

トピック

- [iam:PassRole を実行する権限がありません](#)
- [自分の AWS アカウント以外のユーザーに リソースの AWS Research and Engineering Studio へのアクセスを許可したい](#)
- [環境にログインすると、すぐにログインページに戻ります。](#)
- [ログイン試行時の「ユーザーが見つかりません」エラー](#)
- [Active Directory に追加されたが、RES に欠落しているユーザー](#)
- [セッションの作成時にユーザーが使用できない](#)
- [CloudWatch クラスターマネージャーログのサイズ制限超過エラー](#)

.....

iam:PassRole を実行する権限がありません

iam:PassRole アクションを実行する権限がないというエラーが表示された場合は、ポリシーを更新して RES にロールを渡すことができるようにする必要があります。

一部の AWS サービスでは、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、そのサービスに既存のロールを渡すことができます。そのためには、サービスにロールを渡す権限が必要です。

次の例のエラーは、marymajor という名前の IAM ユーザーがコンソールを使用して RES でアクションを実行しようとするとき発生します。ただし、このアクションをサービスが実行するには、サービスロールから付与された権限が必要です。メアリーには、ロールをサービスに渡す許可がありません。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

この場合、Mary のポリシーを更新してメアリーに iam:PassRole アクションの実行を許可する必要があります。サポートが必要な場合は、AWS 管理者にお問い合わせください。サインイン資格情報を提供した担当者が管理者です。

自分の AWS アカウント以外のユーザーに リソースの AWS Research and Engineering Studio へのアクセスを許可したい

他のアカウントのユーザーや組織外の人、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。リソースベースのポリシーまたはアクセスコントロールリスト (ACL) をサポートするサービスの場合、それらのポリシーを使用して、リソースへのアクセスを付与できます。

詳細については、以下を参照してください:

- 所有している AWS アカウント間でリソースへのアクセスを提供する方法については、IAM ユーザーガイドの [「所有している別の AWS アカウントの IAM ユーザーへのアクセスを提供する」](#) を参照してください。
- サードパーティー AWS アカウントにリソースへのアクセスを提供する方法については、IAM ユーザーガイドの [「サードパーティーが所有する AWS アカウントへのアクセスを提供する」](#) を参照してください。
- ID フェデレーションを通じてアクセスを提供する方法については、IAM ユーザーガイドの [「外部で認証されたユーザー \(ID フェデレーション\) へのアクセス」](#) を参照してください。
- クロスアカウントアクセスにロールとリソースベースのポリシーを使用する方法の違いについては、[IAM ユーザーガイドの「IAM ロールとリソースベースのポリシーの違い」](#) を参照してください。

環境にログインすると、すぐにログインページに戻ります。

この問題は、SSO 統合の設定が間違っている場合に発生します。問題を特定するには、コントローラーインスタンスログをチェックし、エラーがないか設定を確認します。

ログを確認するには：

1. [CloudWatch コンソール](#)を開きます。
2. ロググループから、 という名前のグループを見つけます/`<environment-name>/cluster-manager`。
3. ロググループを開いて、ログストリームのエラーを検索します。

設定を確認するには：

1. [DynamoDB コンソール](#)を開く
2. テーブルから、 という名前のテーブルを見つけます`<environment-name>.cluster-settings`。
3. テーブルを開き、Explore テーブル項目を選択します。
4. フィルターセクションを展開し、次の変数を入力します。
 - 属性名 – キー
 - 条件 – を含む
 - 値 – sso
5. [実行] を選択します。
6. 返された文字列で、SSO 設定値が正しいことを確認します。正しくない場合は、sso_enabled キーの値を False に変更します。
7. RES ユーザーインターフェイスに戻り、SSO を再設定します。

.....

ログイン試行時の「ユーザーが見つかりません」エラー

ユーザーが RES インターフェイスにログインしようとしたときに「ユーザーが見つかりません」というエラーが表示され、そのユーザーが Active Directory に存在する場合：

- ユーザーが RES に存在せず、最近 AD にユーザーを追加した場合
 - ユーザーがまだ RES に同期されていない可能性があります。RES は 1 時間ごとに同期するため、次の同期後にユーザーが追加されたことを待機して確認する必要がある場合があります。

す。すぐに同期するには、「」の手順に従います[Active Directory に追加されたが、RES に欠落しているユーザー](#)。

• ユーザーが RES に存在する場合：

1. 属性マッピングが正しく設定されていることを確認します。詳細については、「[シングルサインオン \(SSO\) 用の ID プロバイダーの設定](#)」を参照してください。
2. SAML 件名と SAML E メール の両方がユーザーの E メールアドレスにマッピングされていることを確認します。

.....

Active Directory に追加されたが、RES に欠落しているユーザー

ユーザーを Active Directory に追加しても RES にない場合は、AD 同期をトリガーする必要があります。AD 同期は、AD エントリを RES 環境にインポートする Lambda 関数によって 1 時間ごとに実行されます。場合によっては、新しいユーザーまたはグループを追加した後、次の同期プロセスが実行されるまで遅延することがあります。Amazon Simple Queue Service から手動で同期を開始できます。

同期プロセスを手動で開始します。

1. [Amazon SQS コンソール](#) を開きます。
2. キューから、 を選択します <environment-name>-cluster-manager-tasks.fifo。
3. メッセージの送受信を選択します。
4. メッセージ本文には、次のように入力します。

```
{ "name": "adsync.sync-from-ad", "payload": {} }
```

5. メッセージグループ ID には、次のように入力します。 **adsync.sync-from-ad**
6. メッセージ重複排除 ID には、ランダムな英数字の文字列を入力します。このエントリは、過去 5 分以内に行われたすべての呼び出しとは異なる必要があります。そうしないと、リクエストは無視されます。

.....

セッションの作成時にユーザーが使用できない

セッションを作成する管理者が、セッションの作成時に Active Directory に属しているユーザーが利用できない場合、ユーザーは初めてログインする必要がある場合があります。セッションはアクティブなユーザーに対してのみ作成できます。アクティブなユーザーは、少なくとも 1 回環境にログインする必要があります。

.....

CloudWatch クラスターマネージャーログのサイズ制限超過エラー

```
2023-10-31T18:03:12.942-07:00 ldap.SIZELIMIT_EXCEEDED: {'msgtype': 100, 'msgid': 11, 'result': 4, 'desc': 'Size limit exceeded', 'ctrls': []}
```

CloudWatch クラスターマネージャーログにこのエラーが表示された場合、ldap 検索が返したユーザーレコードが多すぎる可能性があります。この問題を修正するには、IDP の ldap 検索結果の制限を増やします。

.....

[Storage (ストレージ)]

トピック

- [RES を使用してファイルシステムを作成しましたが、VDI ホストにマウントされません](#)
- [RES を介してファイルシステムをオンボードしたが、VDI ホストにマウントされない](#)
- [VDI ホストから読み書きできない](#)
- [RES から Amazon FSx for NetApp ONTAP を作成しましたが、ドメインに参加していません](#)

.....

RES を使用してファイルシステムを作成しましたが、VDI ホストにマウントされません

ファイルシステムは、VDI ホストでマウントする前に「使用可能」状態である必要があります。以下の手順に従って、ファイルシステムが必須状態であることを確認します。

Amazon EFS

1. [Amazon EFS コンソール](#)に移動します。

2. ファイルシステムの状態が使用可能であることを確認します。
3. ファイルシステムの状態が使用可能でない場合は、VDI ホストを起動するまで待ちます。

1. [Amazon FSx コンソール](#)に移動します。
2. ステータスが使用可能であることを確認します。
3. Status が使用可能でない場合は、VDI ホストを起動する前に待ちます。

.....

RES を介してファイルシステムをオンボードしたが、VDI ホストにマウントされない

RES にオンボードされるファイルシステムには、VDI ホストがファイルシステムをマウントできるように、必要なセキュリティグループルールが設定されている必要があります。これらのファイルシステムは RES の外部で作成されるため、RES は関連するセキュリティグループルールを管理しません。

オンボードされたファイルシステムに関連付けられたセキュリティグループは、次のインバウンドトラフィックを許可する必要があります。

- Linux " ホストからの NFS トラフィック (ポート: 2049)
- Windows " ホストからの SMB トラフィック (ポート: 445)

.....

VDI ホストから読み書きできない

ONTAP は、ボリュームの UNIX、NTFS、MIXED セキュリティスタイルをサポートしています。セキュリティスタイルは、ONTAP がデータアクセスを制御するために使用するアクセス許可のタイプと、これらのアクセス許可を変更できるクライアントタイプを決定します。

たとえば、ボリュームが UNIX セキュリティスタイルを使用している場合でも、ONTAP のマルチプロトコル特性により、SMB クライアントは引き続きデータにアクセスできます (ただし、適切に認証および認可される場合に限ります)。ただし、ONTAP は UNIX クライアントのみがネイティブツールを使用して変更できる UNIX アクセス許可を使用します。

アクセス許可処理のユースケースの例

Linux ワークロードでの UNIX スタイルのボリュームの使用

アクセス許可は、他のユーザーの sudoer で設定できます。たとえば、次の例では、/`<project-name>` ディレクトリに対する `<group-ID>` 完全な読み取り/書き込みアクセス許可のすべてのメンバーに付与します。

```
sudo chown root:<group-ID> /<project-name>
sudo chmod 770 /<project-name>
```

Linux および Windows ワークロードでの NTFS スタイルのボリュームの使用

共有アクセス許可は、特定のフォルダの共有プロパティを使用して設定できます。たとえば、ユーザー `user_01` とフォルダ `がある場合 myfolder`、Full Control、Change または Read のアクセス許可を Allow または Deny に設定できます。

Linux クライアントと Windows クライアントの両方でボリュームを使用する場合は、Linux ユーザー名を同じユーザー名に `domain\username` の NetBIOS ドメイン名形式に関連付ける名前マッピングを SVM に設定する必要があります。これは、Linux ユーザーと Windows ユーザーの間で変換するために必要です。リファレンスについては、[「Amazon FSx for NetApp ONTAP によるマルチプロトコルワークロードの有効化」](#) を参照してください。

.....

RES から Amazon FSx for NetApp ONTAP を作成しましたが、ドメインに参加していません

現在、RES コンソールから Amazon FSx for NetApp ONTAP を作成すると、ファイルシステムはプロビジョニングされますが、ドメインには参加しません。作成した ONTAP ファイルシステム SVM をドメインに結合するには、[「Microsoft Active Directory SVMs の結合」](#) を参照して、[Amazon FSx コンソール](#) の手順に従ってください。必要な [アクセス許可が AD の Amazon FSx サービスアカウントに委任](#) されていることを確認します。SVM がドメインに正常に参加したら、SVM 概要 > エンドポイント > SMB DNS 名に移動し、後で必要になるため DNS 名をコピーします。

ドメインに結合したら、クラスター設定 DynamoDB テーブルで SMB DNS 設定キーを編集します。

1. [Amazon DynamoDB コンソール](#) に移動します。
2. テーブルを選択し、`<stack-name>-cluster-settings` を選択します。
3. 「Explore table items」で、フィルターを展開し、次のフィルターを入力します。
 - 属性名 - キー
 - 条件 - に等しい

- 値 - shared-storage.<file-system-name>.fsx_netapp_ontap.svm.smb_dns
4. 返された項目を選択し、次にアクション、編集項目を選択します。
 5. 以前にコピーした SMB DNS 名で値を更新します。
 6. 保存して閉じるを選択します。

さらに、ファイルシステムに関連付けられたセキュリティグループが、[Amazon VPC によるファイルシステムアクセスコントロール](#)で推奨されているトラフィックを許可していることを確認します。ファイルシステムを使用する新しい VDI ホストは、ドメインに参加している SVM とファイルシステムをマウントできるようになりました。

または、RES Onboard File System capability- from Environment Management select File Systems, Onboard File System を使用して、ドメインに既に参加している既存のファイルシステムをオンボードすることもできます。

.....

スナップショット

トピック

- [スナップショットのステータスが Failed である](#)
- [スナップショットは、テーブルをインポートできなかったことを示すログとともに適用されません。](#)

.....

スナップショットのステータスが Failed である

RES スナップショットページで、スナップショットのステータスが Failed の場合、エラーが発生した時間、クラスターマネージャーの Amazon CloudWatch ロググループに移動することで原因を特定できます。

```
[2023-11-19 03:39:20,208] [INFO] [snapshots-service] creating snapshot in S3 Bucket:
asdf at path s31
[2023-11-19 03:39:20,381] [ERROR] [snapshots-service] An error occurred while
creating the snapshot: An error occurred (TableNotFoundException)
when calling the UpdateContinuousBackups operation:
Table not found: res-demo.accounts.sequence-config
```

.....

スナップショットは、テーブルをインポートできなかったことを示すログとともに適用されません。

以前の env から取得したスナップショットが新しい env に適用されない場合は、クラスターマネージャーの CloudWatch ログを調べて問題を特定します。必要なテーブルクラウドがインポートされないことが問題で言及されている場合は、スナップショットが有効な状態であることを確認します。

1. metadata.json ファイルをダウンロードし、さまざまなテーブルの ExportStatus のステータスが COMPLETED であることを確認します。さまざまなテーブルに ExportManifest フィールドが設定されていることを確認します。上記のフィールドが設定されていない場合、スナップショットは無効な状態であり、スナップショットの適用機能では使用できません。
2. スナップショットの作成を開始したら、RES でスナップショットのステータスが COMPLETED になっていることを確認します。スナップショットの作成プロセスには最大 5～10 分かかります。スナップショット管理ページを再ロードまたは再アクセスして、スナップショットが正常に作成されたことを確認します。これにより、作成されたスナップショットが有効な状態になります。

.....

インフラストラクチャ

トピック

- [正常なインスタスがないロードバランサーターゲットグループ](#)

.....

正常なインスタスがないロードバランサーターゲットグループ

サーバーエラーメッセージなどの問題が UI に表示される場合、またはデスクトップセッションが接続できない場合、インフラストラクチャの Amazon EC2 インスタンスに問題がある可能性があります。

問題の原因を特定する方法は、まず Amazon EC2 コンソールで、繰り返し終了し、新しいインスタンスに置き換えられていると思われる Amazon EC2 インスタスがないかを確認することです。その場合は、Amazon CloudWatch logs をチェックして原因を特定できます。

もう 1 つの方法は、システム内のロードバランサーを確認することです。システムに問題がある可能性があることを示すのは、Amazon EC2 コンソールで見つかったロードバランサーに、登録された正常なインスタンスが表示されない場合です。

通常の外観の例を以下に示します。

Healthy エントリが 0 の場合、リクエストを処理できる Amazon EC2 インスタンスがないことを示します。

Unhealthy エントリが 0 以外の場合は、Amazon EC2 インスタンスが循環している可能性があります。これは、インストールされているアプリケーションソフトウェアがヘルスチェックに合格していないことが原因である可能性があります。

Healthy エントリと Unhealthy エントリの両方が 0 の場合、ネットワークの設定ミスの可能性を示します。たとえば、パブリックサブネットとプライベートサブネットには、対応する AZs がない場合があります。この条件が発生した場合、ネットワーク状態が存在することを示す追加のテキストがコンソールに表示されることがあります。

.....

仮想デスクトップの起動

トピック

- [以前に動作していた仮想デスクトップが正常に接続できなくなりました](#)
- [5 つの仮想デスクトップしか起動できない](#)
- [デスクトップ Windows の接続試行は「接続が閉じられました。トランスポートエラー」](#)
- [VDIs プロビジョニング状態でスタックする](#)
- [起動後に VDI がエラー状態になる](#)

.....

以前に動作していた仮想デスクトップが正常に接続できなくなりました

デスクトップ接続が閉じられたり、接続できなくなったりすると、基盤となる Amazon EC2 インスタンスが失敗するか、Amazon EC2 インスタンスが RES 環境外で終了または停止されたことが原因である可能性があります。管理者 UI のステータスは、準備完了状態を引き続き表示する場合がありますが、接続の試行は失敗します。

Amazon EC2 コンソールを使用して、インスタンスが終了または停止されたかどうかを判断する必要があります。停止した場合は、もう一度開始してみてください。状態が終了した場合は、別のデスクトップを作成する必要があります。ユーザーのホームディレクトリに保存されたデータは、新しいインスタンスの起動時に引き続き使用できます。

以前に失敗したインスタンスが管理者 UI にまだ表示されている場合は、管理者 UI を使用して終了する必要がある場合があります。

.....

5 つの仮想デスクトップしか起動できない

ユーザーが起動できる仮想デスクトップの数のデフォルトの制限は 5 です。これは、次のように管理者 UI を使用して管理者が変更できます。

- デスクトップ設定に移動します。
- サーバータブを選択します。
- DCV セッションパネルで、右側の編集アイコンをクリックします。
- ユーザーあたりの許可されたセッションの値を、必要な新しい値に変更します。
- [Submit] (送信) を選択します。
- ページを更新して、新しい設定が設定されていることを確認します。

.....

デスクトップ Windows の接続試行は「接続が閉じられました。トランスポートエラー」

Windows デスクトップ接続が UI エラー「接続が閉じられました。トランスポートエラー」。Windows インスタンスでの証明書の作成に関連する DCV サーバーソフトウェアの問題が原因である可能性があります。

Amazon CloudWatch ロググループは、次のようなメッセージで接続試行エラーをログに記録する <envname>/vdc/dcv-connection-gateway 場合があります。

```
Nov 24 20:24:27.631 DEBUG HTTP:Splicer Connection{id=9}:  
Websocket{session_id="1291e75f-7816-48d9-bbb2-7371b3b911cd":  
Resolver lookup{client_ip=Some(52.94.36.19)}
```

```
session_id="1291e75f-7816-48d9-bbb2-7371b3b911cd"
protocol_type=WebSocket extension_data=None}:NoStrictCertVerification:
Additional stack certificate (0): [s/n: 0E9E9C4DE7194B37687DC4D2C0F5E94AF0DD57E]
```

```
Nov 24 20:25:15.384 INFO HTTP:Splicer Connection{id=21}:Websocket{
session_id="d1d35954-f29d-4b3f-8c23-6a53303ebc3f"}:
Connection initiated error: unreachable, server io error Custom {
kind: InvalidData, error:
General("Invalid certificate: certificate has expired (code: 10)") }
```

```
Nov 24 20:25:15.384 WARN HTTP:Splicer Connection{id=21}:
Websocket{session_id="d1d35954-f29d-4b3f-8c23-6a53303ebc3f"}:
Error in websocket connection: Server unreachable: Server error: IO error:
unexpected error: Invalid certificate: certificate has expired (code: 10)
```

この場合、SSM セッションマネージャーを使用して Windows インスタンスへの接続を開き、次の 2 つの証明書関連ファイルを削除することが解決される可能性があります。

```
PS C:\Windows\system32\config\systemprofile\AppData\Local\NICE\dcv> dir

Directory: C:\Windows\system32\config\systemprofile\AppData\Local\NICE\dcv
```

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a----	8/4/2022 12:59 PM	1704	dcv.key
-a----	8/4/2022 12:59 PM	1265	dcv.pem

ファイルは自動的に再作成され、それ以降の接続試行が成功する可能性があります。

この方法で問題を解決し、Windows デスクトップの新しい起動で同じエラーが発生した場合は、ソフトウェアスタックの作成 関数を使用して、再生成された証明書ファイルを含む固定インスタンスの新しい Windows ソフトウェアスタックを作成します。これにより、正常な起動と接続に使用できる Windows ソフトウェアスタックが生成されます。

.....

VDIsプロビジョニング状態でスタックする

デスクトップ起動が管理者 UI のプロビジョニング状態のままである場合は、いくつかの理由が考えられます。

原因を特定するには、デスクトップインスタンスのログファイルを調べ、問題の原因となっている可能性のあるエラーを探します。このドキュメントには、ログファイルと Amazon CloudWatch ロググループのリストが含まれており、有用なログおよびイベント情報ソースというラベルが付いたセクションに関連情報が含まれています。

この問題の考えられる原因は次のとおりです。

- 使用されている AMI ID は software-stack として登録されていますが、RES ではサポートされていません。

AMI に必要な設定またはツールが想定されていないため、ブートストラッププロビジョニングスクリプトを完了できませんでした。Linux インスタンスなど、インスタンス/root/bootstrap/logs/のログファイルには、これに関する有用な情報が含まれている場合があります。AWS Marketplace から取得した AMIs ID は、RES デスクトップインスタンスでは機能しない場合があります。サポートされているかどうかを確認するには、テストが必要です。

- ユーザーデータスクリプトは、Windows 仮想デスクトップインスタンスがカスタム AMI から起動されたときに実行されません。

デフォルトでは、ユーザーデータスクリプトは Amazon EC2 インスタンスの起動時に 1 回実行されます。既存の仮想デスクトップインスタンスから AMI を作成し、その AMI にソフトウェアスタックを登録して、このソフトウェアスタックで別の仮想デスクトップを起動しようとする、ユーザーデータスクリプトは新しい仮想デスクトップインスタンスでは実行されません。

この問題を解決するには、AMI の作成に使用した元の仮想デスクトップインスタンスで管理者として PowerShell コマンドウィンドウを開き、次のコマンドを実行します。

```
C:\ProgramData\Amazon\EC2-Windows\Launch\Scripts\InitializeInstance.ps1 -Schedule
```

次に、インスタンスから新しい AMI を作成します。新しい AMI を使用してソフトウェアスタックを登録し、後で新しい仮想デスクトップを起動できます。プロビジョニング状態のままのインスタンスで同じコマンドを実行し、インスタンスを再起動して仮想デスクトップセッションを修正することもできますが、設定ミスのある AMI から別の仮想デスクトップを起動すると、同じ問題が再度発生することに注意してください。

.....

起動後に VDI がエラー状態になる

考えられる問題 1: ホームファイルシステムに、異なる POSIX アクセス許可を持つユーザーのディレクトリがあります。

これは、次のシナリオが当てはまる場合に直面する問題である可能性があります。

1. デプロイされた RES バージョンは 2024.01 以降です。
2. RES スタックのデプロイ中に、 の 属性が に設定EnableLdapIDMappingされましたTrue。
3. RES スタックのデプロイ時に指定されたホームファイルシステムは、RES 2024.01 より前のバージョンで使用されたか、 を EnableLdapIDMappingに設定して以前の環境で使用されましたFalse。

解決手順: ファイルシステム内のユーザーディレクトリを削除します。

1. クラスタマネージャーホストへの SSM。
2. `cd /home.`
3. `ls -` は、`admin1`、`admin2..` などのユーザー名に一致するディレクトリ名を持つディレクトリを一覧表示する必要があります。
4. ディレクトリ を削除します`sudo rm -r 'dir_name'`。ssm-user ディレクトリと ec2-user ディレクトリを削除しないでください。
5. ユーザーが新しい env に既に同期されている場合は、ユーザーの DDB テーブルからユーザーの を削除します (clusteradmin を除く)。
6. AD 同期の開始 - クラスタマネージャー Amazon EC2 `sudo /opt/idea/python/3.9.16/bin/resctl ldap sync-from-ad`で実行します。
7. RES ウェブページから Error状態の VDI インスタンスを再起動します。VDI が約 20 分で Ready状態に移行することを確認します。

.....

仮想デスクトップコンポーネント

トピック

- [Amazon EC2 インスタンスがコンソールで終了を繰り返し表示](#)
- [AD への参加に失敗したために vdc-controller インスタンスがサイクルしています / eVDI モジュールが失敗した API ヘルスチェックを表示](#)

- [プロジェクトは、ソフトウェアスタックを編集して追加するときにプルダウンに表示されません](#)
- [cluster-manager Amazon CloudWatch ログには、「<user-home-init> アカウントはまだ利用できません。ユーザーの同期を待っています」と表示されます \(アカウントはユーザー名です\)。](#)
- [ログイン試行時の Windows デスクトップに「アカウントが無効になっています。管理者にお問い合わせください」](#)
- [外部/顧客の AD 設定に関する DHCP オプションの問題](#)
- [Firefox エラー MOZILLA_PKIX_ERROR_REQUIRED_TLS_FEATURE_MISSING](#)

.....

Amazon EC2 インスタンスがコンソールで終了を繰り返し表示

インフラストラクチャインスタンスが Amazon EC2 コンソールで終了と繰り返し表示される場合、原因はその設定に関連している可能性があり、インフラストラクチャインスタンスタイプによって異なります。以下に、原因を特定する方法を示します。

Amazon EC2 コンソールで vdc-controller インスタンスが終了状態を繰り返す場合は、シークレットタグが正しくない可能性があります。RES によって維持されるシークレットには、インフラストラクチャの Amazon EC2 インスタンスにアタッチされた IAM アクセスコントロールポリシーの一部として使用されるタグがあります。vdc-controller がサイクルしていて、CloudWatch ロググループに次のエラーが表示された場合、シークレットが正しくタグ付けされていない可能性があります。シークレットには、次のタグを付ける必要があることに注意してください。

```
{
  "res:EnvironmentName": "<envname>" # e.g. "res-demo"
  "res:ModuleName": "virtual-desktop-controller"
}
```

このエラーの Amazon CloudWatch ログメッセージは、次のように表示されます。

```
An error occurred (AccessDeniedException) when calling the GetSecretValue
operation: User: arn:aws:sts::160215750999:assumed-role/<envname>-vdc-gateway-role-us-
east-1/i-043f76a2677f373d0
is not authorized to perform: secretsmanager:GetSecretValue on resource:
arn:aws:secretsmanager:us-east-1:160215750999:secret:Certificate-res-bi-
Certs-5W9SPUXF08IB-F1sNRv
because no identity-based policy allows the secretsmanager:GetSecretValue action
```

Amazon EC2 インスタンスのタグをチェックし、それらが上記のリストと一致することを確認します。

AD への参加に失敗したために vdc-controller インスタンスがサイクルしています / eVDI モジュールが失敗した API ヘルスチェックを表示

eVDI モジュールがヘルスチェックに失敗した場合、環境ステータスセクションに以下が表示されます。

この場合、デバッグの一般的なパスは、クラスターマネージャーの [CloudWatch](#) ログを調べることです。(という名前のロググループを探します <env-name>/cluster-manager。)

考えられる問題：

- ログにテキストが含まれている場合は Insufficient permissions、res スタックの作成時に指定された ServiceAccount ユーザー名のスペルが正しいことを確認してください。

ログ行の例：

```
Insufficient permissions to modify computer account:
CN=IDEA-586BD25043,OU=Computers,OU=RES,OU=CORP,DC=corp,DC=res,DC=com:
000020E7: AttrErr: DSID-03153943, #1: 0: 000020E7: DSID-03153943, problem 1005
(CONSTRAINT_ATT_TYPE), data 0, Att 90008 (userAccountControl):len 4 >> 432 ms -
request will be retried in 30 seconds
```

- [SecretsManager コンソール](#) から、RES デプロイ中に提供される ServiceAccount ユーザー名にアクセスできます。Secrets Manager で対応するシークレットを検索し、プレーンテキストの取得を選択します。ユーザー名が正しくない場合は、編集を選択してシークレット値を更新します。現在の cluster-manager インスタンスと vdc-controller インスタンスを終了します。新しいインスタンスは安定した状態になります。
- 提供された [外部リソーススタック](#) によって作成されたリソースを使用している場合、ユーザー名は「ServiceAccount」である必要があります。RES のデプロイ中に DisableADJoin/パラメータが False に設定されている場合は、「ServiceAccount」ユーザーに AD でコンピュータオブジェクトを作成するアクセス許可があることを確認します。
- 使用したユーザー名が正しいが、ログにテキストが含まれている場合 Invalid credentials、入力したパスワードが間違っているか、有効期限が切れている可能性があります。

ログ行の例：

```
{'msgtype': 97, 'msgid': 1, 'result': 49, 'desc': 'Invalid credentials', 'ctrls': [],  
'info': '80090308: LdapErr: DSID-0C090569, comment: AcceptSecurityContext error,  
data 532, v4563'}
```

- 環境の作成時に入力したパスワードは、[Secrets Manager コンソールにパスワードを保存するシークレット](#)にアクセスして読み取ることができます。シークレット (など <env_name>directoryserviceServiceAccountPassword) を選択し、プレーンテキストの取得を選択します。
- シークレットのパスワードが正しくない場合は、編集を選択してシークレットの値を更新します。現在の cluster-manager インスタンスと vdc-controller インスタンスを終了します。新しいインスタンスは更新されたパスワードを使用し、安定した状態になります。
- パスワードが正しい場合は、接続された Active Directory でパスワードの有効期限が切れている可能性があります。最初に Active Directory でパスワードをリセットしてから、シークレットを更新する必要があります。[Directory Service コンソール](#)から Active Directory でユーザーのパスワードをリセットできます。
 1. 適切なディレクトリ ID を選択する
 2. アクションを選択し、ユーザーパスワードをリセットしてから、ユーザー名 (ServiceAccount など) と新しいパスワードをフォームに入力します。
 3. 新しく設定したパスワードが以前のパスワードと異なる場合は、対応する Secret Manager シークレットのパスワードを更新します (例: <env_name>directoryserviceServiceAccountPassword。
 4. 現在の cluster-manager インスタンスと vdc-controller インスタンスを終了します。新しいインスタンスは安定した状態になります。

.....

プロジェクトは、ソフトウェアスタックを編集して追加するときにプルダウンに表示されません

この問題は、ユーザーアカウントと AD の同期に関連する次の問題に関連している可能性があります。この問題が表示された場合は、クラスターマネージャーの Amazon CloudWatch ロググループでエラー <user-home-init> account not available yet. waiting for user to be synced 「」 をチェックして、原因が同じか関連しているかを判断します。

.....

cluster-manager Amazon CloudWatch ログには、「<user-home-init> アカウントはまだ利用できません。ユーザーの同期を待っています」と表示されます (アカウントはユーザー名です)。

SQS サブスクライバーは、ユーザーアカウントにアクセスできないため、ビジー状態で無限ループに陥っています。このコードは、ユーザーの同期中にユーザーのホームファイルシステムを作成しようとしたときにトリガーされます。

ユーザーアカウントにアクセスできないのは、使用中の AD に対して RES が正しく設定されていないためです。たとえば、BI/RES 環境の作成時に使用された ServiceAccountUsername パラメータが、「Admin」ではなく ServiceAccount」を使用するなど、正しい値ではなかったとします。

.....

ログイン試行時の Windows デスクトップに「アカウントが無効になっています。管理者にお問い合わせください」

ユーザーがロックされた画面にログインできない場合、SSO 経由で正常にサインオンした後、RES 用に設定された AD でユーザーが無効化されている可能性があります。

AD でユーザーアカウントが無効になっている場合、SSO ログインは失敗します。

外部/顧客の AD 設定に関する DHCP オプションの問題

独自の Active Directory "The connection has been closed. Transport error"で RES を使用するとき Windows 仮想デスクトップで というエラーが発生した場合は、dcv-connection-gateway Amazon CloudWatch ログに次のようなものがないか確認してください。

```
Oct 28 00:12:30.626 INFO HTTP:Splicer Connection{id=263}:
Websocket{session_id="96cffa6e-cf2e-410f-9eea-6ae8478dc08a"}: Connection initiated
error: unreachable, server io error Custom { kind: Uncategorized, error: "failed to
lookup address information: Name or service not known" }

Oct 28 00:12:30.626 WARN HTTP:Splicer Connection{id=263}:
Websocket{session_id="96cffa6e-cf2e-410f-9eea-6ae8478dc08a"}: Error in websocket
connection: Server unreachable: Server error: IO error: failed to lookup address
information: Name or service not known
```

```
Oct 28 00:12:30.627 DEBUG HTTP:Splicer Connection{id=263}: ConnectionGuard dropped
```

独自の VPC の DHCP オプションに AD ドメインコントローラーを使用している場合は、以下を行う必要があります。

1. AmazonProvidedDNS を 2 つのドメインコントローラー IPs。
2. ドメイン名を ec2.internal に設定します。

以下に例を示します。この設定がないと、RES/DCV は ip-10-0-x-xx.ec2.internal ホスト名を検索するため、Windows デスクトップでトランスポートエラー が表示されます。

.....

Firefox エラー MOZILLA_PKIX_ERROR_REQUIRED_TLS_FEATURE_MISSING

Firefox ウェブブラウザを使用すると、仮想デスクトップに接続しようとする
と、MOZILLA_PKIX_ERROR_REQUIRED_TLS_FEATURE_MISSING というエラーメッセージが表示されることがあります。

原因は、RES ウェブサーバーが TLS + Stapling On でセットアップされているが、Stapling Validation で応答していないことです (<https://support.mozilla.org/en-US/questions/1372483>)。

これは、https://really-simple-ssl.com/mozilla_pkix_error_required_tls_feature_missing の指示に従って修正できます。

.....

Env 削除

トピック

- [res-xxx-cluster スタックが「DELETE_FAILED」状態で、「Role is invalid or cannot be assumed」エラーのため手動で削除できない](#)
- [ログの収集](#)
- [VDI ログのダウンロード](#)
- [Linux EC2 インスタンスからのログのダウンロード](#)
- [Windows EC2 インスタンスからのログのダウンロード](#)

- [WaitCondition エラーの ECS ログの収集](#)

.....

res-xxx-cluster スタックが「DELETE_FAILED」状態で、「Role is invalid or cannot be assumed」エラーのため手動で削除できない

「res-xxx-cluster」スタックが「DELETE_FAILED」状態で、手動で削除できない場合は、次の手順を実行して削除できます。

スタックが「DELETE_FAILED」状態になっている場合は、まず手動で削除してみてください。スタックの削除を確認するダイアログが表示される場合があります。[削除] を選択します。

必要なスタックリソースをすべて削除しても、保持するリソースを選択するメッセージが表示されることがあります。その場合は、「保持するリソース」としてすべてのリソースを選択し、削除を選択します。

次のようなエラーが表示される場合があります。Role: arn:aws:iam::... is Invalid or cannot be assumed

これは、スタックの削除に必要なロールが、スタックの前に最初に削除されたことを意味します。これを回避するには、ロールの名前をコピーします。IAM コンソールに移動し、次に示すパラメータを使用して、その名前のロールを作成します。

- Trusted entity type では、AWS サービスを選択します。
- ユースケースで、Use cases for other AWS servicesを選択しますCloudFormation。

[次へ] を選択してください。ロールにAWSCloudFormationFullAccess「」およびAdministratorAccess「」のアクセス許可を付与してください。レビューページは次のようになります。

次に、CloudFormation コンソールに戻り、スタックを削除します。これで、ロールを作成した後で削除できるようになります。最後に、IAM コンソールに移動し、作成したロールを削除します。

.....

ログの収集

EC2 コンソールから EC2 インスタンスにログインする

- Linux EC2 インスタンスにログインするには、[次の手順に従います](#)。
- Windows EC2 インスタンスにログインするには、[次の手順に従います](#)。次に、Windows PowerShell を開いてコマンドを実行します。

インフラストラクチャホストログの収集

1. Cluster-manager: 次の場所からクラスターマネージャーのログを取得し、チケットにアタッチします。
 - a. CloudWatch ロググループ からのすべてのログ<env-name>/cluster-manager。
 - b. <env-name>-cluster-manager EC2 インスタンスの /root/bootstrap/logs ディレクトリにあるすべてのログ。このセクションの冒頭にあるEC2 コンソールから EC2 インスタンスにログインする」から にリンクされている手順に従って、インスタンスにログインします。
2. Vdc-controller: 次の場所から vdc-controller のログを取得し、チケットにアタッチします。
 - a. CloudWatch ロググループ からのすべてのログ<env-name>/vdc-controller。
 - b. <env-name>-vdc-controller EC2 インスタンスの /root/bootstrap/logs ディレクトリにあるすべてのログ。このセクションの冒頭にあるEC2 コンソールから EC2 インスタンスにログインする」から にリンクされている手順に従って、インスタンスにログインします。

ログを簡単に取得する方法の 1 つは、[Linux EC2 インスタンスからのログのダウンロード](#)セクションの指示に従うことです。モジュール名はインスタンス名になります。

VDI ログの収集

対応する Amazon EC2 インスタンスを特定する

ユーザーがセッション名 で VDI を起動した場合VDI1、Amazon EC2 コンソールのインスタンスの対応する名前は になります<env-name>-VDI1-<user name>。

Linux VDI ログの収集

このセクションの冒頭にあるAmazon EC2 コンソールから EC2 インスタンスにログインする」の「」にリンクされた手順に従って、Amazon EC2 コンソールから対応する Amazon EC2 インスタンスにログインします。VDI Amazon EC2 インスタンスの /root/bootstrap/logsおよび /var/log/dcv/ ディレクトリにあるすべてのログを取得します。

ログを取得する方法の 1 つは、ログを s3 にアップロードし、そこからダウンロードすることです。そのためには、以下の手順に従って 1 つのディレクトリからすべてのログを取得し、アップロードします。

1. /root/bootstrap/logs ディレクトリの下に dcv ログをコピーするには、次の手順に従います。

```
sudo su -  
cd /root/bootstrap  
mkdir -p logs/dcv_logs  
cp -r /var/log/dcv/* logs/dcv_logs/
```

2. 次に、次のセクション「」に記載されている手順に従ってログ[VDI ログのダウンロード](#)をダウンロードします。

Windows VDI ログの収集

このセクションの冒頭にある Amazon EC2 コンソールから EC2 インスタンスにログインする」の「」にリンクされた手順に従って、Amazon EC2 コンソールから対応する Amazon EC2 インスタンスにログインします。VDI EC2 インスタンスの \$env:SystemDrive\Users\Administrator\RES\Bootstrap\Log\ ディレクトリですべてのログを取得します。

ログを取得する方法の 1 つは、ログを S3 にアップロードし、そこからダウンロードすることです。これを行うには、次のセクション「」に記載されているステップに従います[VDI ログのダウンロード](#)。

.....

VDI ログのダウンロード

1. VDI EC2 インスタンスの IAM ロールを更新して、S3 アクセスを許可します。
2. EC2 コンソールに移動し、VDI インスタンスを選択します。
3. 使用している IAM ロールを選択します。
4. アクセス許可の追加ドロップダウンメニューのアクセス許可ポリシーセクションで、ポリシーのアタッチを選択し、AmazonS3FullAccess ポリシーを選択します。
5. アクセス許可を追加を選択して、そのポリシーをアタッチします。
6. その後、VDI タイプに基づいて以下の手順に従ってログをダウンロードします。モジュール名はインスタンス名になります。
 - a. [Linux EC2 インスタンスからのログのダウンロード](#) Linux 用。

b. [Windows EC2 インスタンスからのログのダウンロード](#) for Windows。

7. 最後に、ロールを編集してAmazonS3FullAccessポリシーを削除します。

Note

すべての VDI、と同じ IAM ロールを使用します。 <env-name>-vdc-host-role-<region>

Linux EC2 インスタンスからのログのダウンロード

ログをダウンロードする EC2 インスタンスにログインし、次のコマンドを実行してすべてのログを s3 バケットにアップロードします。

```
sudo su -
ENV_NAME=<environment_name>
REGION=<region>
ACCOUNT=<aws_account_number>
MODULE=<module_name>

cd /root/bootstrap
tar -czvf ${MODULE}_logs.tar.gz logs/ --overwrite
aws s3 cp ${MODULE}_logs.tar.gz s3://${ENV_NAME}-cluster-${REGION}-${ACCOUNT}/${MODULE}_logs.tar.gz
```

その後、S3 コンソールに移動<environment_name>-cluster-<region>-<aws_account_number>し、名前が のバケットを選択し、以前にアップロードした<module_name>_logs.tar.gzファイルをダウンロードします。

Windows EC2 インスタンスからのログのダウンロード

ログをダウンロードする EC2 インスタンスにログインし、次のコマンドを実行してすべてのログを S3 バケットにアップロードします。

```
$ENV_NAME="<environment_name>"
```

```

$REGION="<region>"
$ACCOUNT="<aws_account_number>"
$MODULE="<module_name>"

$logDirPath = Join-Path -Path $env:SystemDrive -ChildPath "Users\Administrator\RES
\Bootstrap\Log"
$zipFilePath = Join-Path -Path $env:TEMP -ChildPath "logs.zip"
Remove-Item $zipFilePath
Compress-Archive -Path $logDirPath -DestinationPath $zipFilePath
$bucketName = "${ENV_NAME}-cluster-${REGION}-${ACCOUNT}"
$keyName = "${MODULE}_logs.zip"
Write-S3Object -BucketName $bucketName -Key $keyName -File $zipFilePath

```

その後、S3 コンソールに移動**<environment_name>-cluster-<region>-<aws_account_number>**し、名前が のバケットを選択し、以前にアップロードした**<module_name>_logs.zip**ファイルをダウンロードします。

WaitCondition エラーの ECS ログの収集

1. デプロイされたスタックに移動し、リソースタブを選択します。
2. Deploy → ResearchAndEngineeringStudio → Installer → Tasks → CreateTaskDef → CreateContainer → LogGroup を展開し、ロググループを選択して CloudWatch ログを開きます。
3. このロググループから最新のログを取得します。

デモ環境

トピック

- [ID プロバイダーへの認証リクエストを処理する際のデモ環境ログインエラー](#)

ID プロバイダーへの認証リクエストを処理する際のデモ環境ログインエラー

問題

ログインしようとして、ID プロバイダーへの認証リクエストを処理するときに「予期しないエラー」が発生した場合、パスワードの有効期限が切れている可能性があります。これは、ログインしようとしているユーザーのパスワードまたは Active Directory サービスアカウントのいずれかです。

緩和策

1. [Directory サービスコンソール](#)でユーザーとサービスアカウントのパスワードをリセットします。
2. [Secrets Manager](#) のサービスアカウントのパスワードを、上記で入力した新しいパスワードと一致するように更新します。
 - Keycloak スタックの : PasswordSecret-...-RESExternal-...-DirectoryService-... with Description: Password for Microsoft Active Directory
 - for RES: res-ServiceAccountPassword-... with 説明: Active Directory サービスアカウントのパスワード
3. [EC2 コンソール](#)に移動し、クラスターマネージャーインスタンスを終了します。Auto Scaling ルールは、新しいインスタンスのデプロイを自動的にトリガーします。

既知の問題

- [既知の問題 2024.x](#)
 - [\(2024.06\) AD グループ名にスペースが含まれているとスナップショットの適用が失敗する](#)
 - [\(2024.04-2024.04.02\) VDI インスタンスのロールにアタッチされていない IAM アクセス許可境界が提供されました](#)
 - [\(2024.04.02 以前\) ap-southeast-2 \(シドニー\) の Windows NVIDIA インスタンスが起動に失敗する](#)
 - [\(2024.04 および 2024.04.01\) GovCloud での RES 削除の失敗](#)
 - [\(2024.04 - 2024.04.02\) Linux 仮想デスクトップは再起動時に「RESUMING」ステータスのままになる可能性があります](#)
 - [\(2024.04.02 以前\) SAMAccountName 属性に大文字または特殊文字が含まれている AD ユーザーの同期に失敗しました](#)
 - [\(2024.04.02 以前\) 踏み台ホストにアクセスするためのプライベートキーが無効です](#)
 - [\(2024.06 以前\) AD 同期中に RES に同期されていないグループメンバー](#)

- [\(2024.06 以前\) CVE-2024-6387、RegreSSHion、RHEL9 および Ubuntu VDI のセキュリティ脆弱性](#)

既知の問題 2024.x

.....

(2024.06) AD グループ名にスペースが含まれているとスナップショットの適用が失敗する

問題

AD グループに名前にスペースが含まれている場合、RES 2024.06 は以前のバージョンのスナップショットを適用できません。

クラスターマネージャーの CloudWatch ログ (<environment-name>/cluster-manager ロググループの下) には、AD 同期中に次のエラーが含まれます。

```
[apply-snapshot] authz.role-assignments/<Group name with spaces>:group#<projectID>:project FAILED_APPLY because: [INVALID_PARAMS] Actor key doesn't match the regex pattern ^[a-zA-Z0-9_.-]{1,20}:(user|group)$
```

エラーは、以下の要件を満たすグループ名のみを RES が受け入れることが原因です。

- 小文字と大文字の ASCII 文字、数字、ダッシュ (-)、ピリオド (.)、アンダースコア (_) のみを含めることができます。
- ダッシュ (-) は最初の文字として使用できません
- スペースを含めることはできません。

影響を受けるバージョン

2024 年 6 月

緩和策

1. パッチスクリプトとパッチファイル ([patch.py](#) および [groupname_regex.patch](#)) をダウンロードするには、次のコマンドを実行し、をファイルを配置するディレクトリ<output-directory>に、を RES 環境の名前<environment-name>に置き換えます。

- a. パッチは RES 2024.06 にのみ適用されます
- b. パッチスクリプトには、[AWS CLI v2](#)、Python 3.9.16 以降、および [Boto3](#) が必要です。
- c. RES AWS がデプロイされているアカウントとリージョンの CLI を設定し、RES によって作成されたバケットに書き込む S3 アクセス許可があることを確認します。

```
OUTPUT_DIRECTORY=<output-directory>
ENVIRONMENT_NAME=<environment-name>

mkdir -p ${OUTPUT_DIRECTORY}
curl https://research-engineering-studio-us-east-1.s3.amazonaws.com/
releases/2024.06/patch_scripts/patch.py --output ${OUTPUT_DIRECTORY}/patch.py
curl https://research-engineering-studio-us-east-1.s3.amazonaws.com/
releases/2024.06/patch_scripts/patches/groupname_regex.patch --output
${OUTPUT_DIRECTORY}/groupname_regex.patch
```

2. パッチスクリプトとパッチファイルがダウンロードされるディレクトリに移動します。次のパッチコマンドを実行します。

```
python3 patch.py --environment-name ${ENVIRONMENT_NAME} --res-version 2024.06 --
module cluster-manager --patch ${OUTPUT_DIRECTORY}/groupname_regex.patch
```

3. 環境の Cluster Manager インスタンスを再起動するには、次のコマンドを実行します。Amazon EC2 マネジメントコンソールからインスタンスを終了することもできます。

```
INSTANCE_ID=$(aws ec2 describe-instances \
  --filters \
    Name=tag:Name,Values=${ENVIRONMENT_NAME}-cluster-manager \
    Name=tag:res:EnvironmentName,Values=${ENVIRONMENT_NAME} \
  --query "Reservations[0].Instances[0].InstanceId" \
  --output text)

aws ec2 terminate-instances --instance-ids ${INSTANCE_ID}
```

Note

このパッチでは、AD グループ名に小文字と大文字の ASCII 文字、数字、ダッシュ (-)、ピリオド (.)、アンダースコア (_)、および合計長が 1~30 のスペースを含めることができます。

(2024.04-2024.04.02) VDI インスタンスのロールにアタッチされていない IAM アクセス許可境界が提供されました

問題

仮想デスクトップセッションがプロジェクトのアクセス許可の境界設定を適切に継承していない。これは、IAMPermissionBoundary パラメータで定義されたアクセス許可の境界が、そのプロジェクトの作成中にプロジェクトに適切に割り当てられていないためです。

影響を受けるバージョン

2024 年 4 月 - 2024.04.02

緩和策

VDIs がプロジェクトに割り当てられたアクセス許可の境界を適切に継承できるようにするには、次の手順に従います。

1. パッチスクリプトとパッチファイル ([patch.py](#) および [vdi_host_role_permission_boundary.patch](#)) をダウンロードするには、次のコマンドを実行し、をファイルを配置するローカルディレクトリ<output-directory>に置き換えます。
 - a. パッチは RES 2024.04.02 にのみ適用されます。バージョン 2024.04 または 2024.04.01 を使用している場合は、[マイナーバージョンの更新についてパブリックドキュメントに記載されている手順に従って](#)、環境を 2024.04.02 に更新できます。
 - b. パッチスクリプトには、[AWS CLI v2](#)、Python 3.9.16 以降、および [Boto3](#) が必要です。
 - c. RES AWS がデプロイされているアカウントとリージョンの CLI を設定し、RES によって作成されたバケットに書き込む S3 アクセス許可があることを確認します。

```
OUTPUT_DIRECTORY=<output-directory>
```

```
curl https://research-engineering-studio-us-east-1.s3.amazonaws.com/
releases/2024.04.02/patch_scripts/patch.py --output ${OUTPUT_DIRECTORY}/patch.py
```

```
curl https://research-engineering-studio-us-east-1.s3.amazonaws.com/
releases/2024.04.02/patch_scripts/patches/vdi_host_role_permission_boundary.patch
--output ${OUTPUT_DIRECTORY}/vdi_host_role_permission_boundary.patch
```

2. パッチスクリプトとパッチファイルがダウンロードされるディレクトリに移動します。次のパッチコマンドを実行し、 を RES 環境の名前<environment-name>に置き換えます。

```
python3 patch.py --environment-name <environment-name> --res-version 2024.04.02 --  
module cluster-manager --patch vdi_host_role_permission_boundary.patch
```

3. を RES 環境の名前<environment-name>に置き換えて、このコマンドを実行して環境内の cluster-manager インスタンスを再起動します。Amazon EC2 マネジメントコンソールからインスタンスを終了することもできます。

```
ENVIRONMENT_NAME=<environment-name>  
  
INSTANCE_ID=$(aws ec2 describe-instances \  
  --filters \  
    Name=tag:Name,Values=${ENVIRONMENT_NAME}-cluster-manager \  
    Name=tag:res:EnvironmentName,Values=${ENVIRONMENT_NAME}\  
  --query "Reservations[0].Instances[0].InstanceId" \  
  --output text)  
  
aws ec2 terminate-instances --instance-ids ${INSTANCE_ID}
```

.....

(2024.04.02 以前) ap-southeast-2 (シドニー) の Windows NVIDIA インスタンスが起動に失敗する

問題

Amazon マシンイメージ (AMIs) は、特定の設定で RES で仮想デスクトップ (VDIs) をスピンアップするために使用されます。各 AMI には、リージョンごとに異なる ID が関連付けられています。RES で ap-southeast-2 (シドニー) で Windows Nvidia インスタンスを起動するように設定された AMI ID は現在正しくありません。

このタイプのインスタンス設定ami-0e190f8939a996cafの AMI-ID は、ap-southeast-2 (シドニー) に誤ってリストされています。代わりに AMI ID ami-027cf6e71e2e442f4 を使用する必要があります。

デフォルトの AMI ami-0e190f8939a996caf でインスタンスを起動しようとすると、次のエラーが発生します。

An error occurred (InvalidAMIID.NotFound) when calling the RunInstances operation: The image id '[ami-0e190f8939a996caf]' does not exist

設定ファイルの例を含む、バグを再現する手順：

- ap-southeast-2 リージョンに RES をデプロイします。
- Windows-NVIDIA のデフォルトソフトウェアスタック (AMI ID) を使用してインスタンスを起動しますami-0e190f8939a996caf。

影響を受けるバージョン

すべての RES バージョン 2024.04.02 以前が影響を受けます

緩和策

以下の緩和策は RES バージョン 2024.01.01 でテストされています。

- 次の設定で新しいソフトウェアスタックを登録する
 - AMI ID: ami-027cf6e71e2e442f4
 - オペレーティングシステム: Windows
 - GPU 製造元: NVIDIA
 - 最小 ストレージサイズ (GB): 30
 - 最小 RAM (GB): 4
- このソフトウェアスタックを使用して Windows-NVIDIA インスタンスを起動する

.....

(2024.04 および 2024.04.01) GovCloud での RES 削除の失敗

問題

RES 削除ワークフロー中、UnprotectCognitoUserPoolLambda は後で削除される Cognito ユーザープールの削除保護を無効にします。Lambda の実行は、 によって開始されずInstallerStateMachine。

商用リージョンと GovCloud リージョンでデフォルトの AWS CLI バージョンが異なるため、Lambda のupdate_user_pool呼び出しは GovCloud リージョンで失敗します。

GovCloud リージョンで RES を削除しようとする、次のエラーが表示されます。

```
Parameter validation failed: Unknown parameter in input: \"DeletionProtection\", must be one of: UserPoolId, Policies, LambdaConfig, AutoVerifiedAttributes, SmsVerificationMessage, EmailVerificationMessage, EmailVerificationSubject, VerificationMessageTemplate, SmsAuthenticationMessage, MfaConfiguration, DeviceConfiguration, EmailConfiguration, SmsConfiguration, UserPoolTags, AdminCreateUserConfig, UserPoolAddOns, AccountRecoverySetting
```

バグを再現するステップ：

- GovCloud リージョンに RES をデプロイする
- RES スタックを削除する

影響を受けるバージョン

RES バージョン 2024.04 および 2024.04.01

緩和策

RES バージョン 2024.04 では、次の緩和策がテストされています。

- UnprotectCognitoUserPool Lambda を開く
 - 命名規則: `<env-name>-InstallerTasksUnprotectCognitoUserPool-...`
- ランタイム設定 -> 編集 -> ランタイム -> 保存 Python 3.11 を選択します。
- CloudFormation を開きます。
- Delete RES stack -> leave Retain Installer Resource UNCHECKED -> Delete。

.....

(2024.04 - 2024.04.02) Linux 仮想デスクトップは再起動時に「RESUMING」ステータスのままになる可能性があります

問題

Linux 仮想デスクトップは、手動またはスケジュールによる停止後に再起動すると、「RESUMING」ステータスのままになることがあります。

インスタンスを再起動した後、AWS Systems Manager は新しい DCV セッションを作成するためのリモートコマンドを実行せず、次のログメッセージが vdc-controller CloudWatch ログ (<environment-name>/vdc/controllerCloudWatch ロググループの下) に欠落しています。

```
Handling message of type DCV_HOST_REBOOT_COMPLETE_EVENT
```

影響を受けるバージョン

2024 年 4 月 - 2024.04.02

緩和策

「RESUMING」状態でスタックしている仮想デスクトップを復旧するには：

1. EC2 コンソールから問題インスタンスに SSH 接続します。
2. インスタンスで次のコマンドを実行します。

```
sudo su -  
/bin/bash /root/bootstrap/latest/virtual-desktop-host-linux/  
configure_post_reboot.sh  
sudo reboot
```

3. インスタンスが再起動するのを待ちます。

新しい仮想デスクトップが同じ問題に陥らないようにするには：

1. パッチスクリプトとパッチファイル ([patch.py](#) および [vdi_stuck_in_resuming_status.patch](#)) をダウンロードするには、次のコマンドを実行し、をファイルを配置するディレクトリ<output-directory>に置き換えます。

Note

- パッチは RES 2024.04.02 にのみ適用されます。
- パッチスクリプトには、[AWS CLI v2](#)、Python 3.9.16 以降、および [Boto3](#) が必要です。
- RES AWS がデプロイされているアカウントとリージョンの CLI を設定し、RES によって作成されたバケットに書き込む S3 アクセス許可があることを確認します。

```
OUTPUT_DIRECTORY=<output-directory>
```

```
curl https://research-engineering-studio-us-east-1.s3.amazonaws.com/
releases/2024.04.02/patch_scripts/patch.py --output ${OUTPUT_DIRECTORY}/patch.py
```

```
curl https://research-engineering-studio-us-east-1.s3.amazonaws.com/
releases/2024.04.02/patch_scripts/patches/vdi_stuck_in_resuming_status.patch --
output ${OUTPUT_DIRECTORY}/vdi_stuck_in_resuming_status.patch
```

2. パッチスクリプトとパッチファイルがダウンロードされるディレクトリに移動します。次のパッチコマンドを実行し、を RES 環境の名前<environment-name>に、を RES がデプロイされているリージョン<aws-region>に置き換えます。

```
python3 patch.py --environment-name <environment-name> --res-version 2024.04.02
--module virtual-desktop-controller --patch vdi_stuck_in_resuming_status.patch --
region <aws-region>
```

3. 環境の " Controller インスタンスを再起動するには、次のコマンドを実行し、を RES 環境の名前<environment-name>に置き換えます。

```
ENVIRONMENT_NAME=<environment-name>
```

```
INSTANCE_ID=$(aws ec2 describe-instances \
  --filters \
    Name=tag:Name,Values=${ENVIRONMENT_NAME}-vdc-controller \
    Name=tag:res:EnvironmentName,Values=${ENVIRONMENT_NAME} \
  --query "Reservations[0].Instances[0].InstanceId" \
  --output text)
```

```
aws ec2 terminate-instances --instance-ids ${INSTANCE_ID}
```

.....

(2024.04.02 以前) SAMAccountName 属性に大文字または特殊文字が含まれている AD ユーザーの同期に失敗しました

問題

SSO が少なくとも 2 時間 (2 つの AD 同期サイクル) セットアップされると、RES は AD ユーザーの同期に失敗します。クラスターマネージャーの CloudWatch ログ (<environment-name>/cluster-manager ロググループの下) には、AD 同期中に次のエラーが含まれます。

```
Error: [INVALID_PARAMS] Invalid params: user.username must match regex: ^(?=[3,20]$)(?![_.])(?!.*[_.]{2})[a-z0-9._]+(?<![_.])$
```

エラーは、以下の要件を満たす SAMAccount ユーザー名のみを RES が受け入れることが原因です。

- 小文字の ASCII 文字、数字、ピリオド (.)、アンダースコア (_) のみを含めることができます。
- ピリオドまたはアンダースコアは、最初または最後の文字として使用できません。
- 2 つの連続したピリオドまたはアンダースコア (...、__、._、_ など) を含めることはできません。

影響を受けるバージョン

2024.04.02 以前

緩和策

1. パッチスクリプトとパッチファイル ([patch.py](#) と [samaccountname_regex.patch](#)) をダウンロードするには、次のコマンドを実行し、 をファイルを配置するディレクトリ<output-directory>に置き換えます。

Note

- パッチは RES 2024.04.02 にのみ適用されます。
- パッチスクリプトには、[AWS CLI v2](#)、Python 3.9.16 以降、および [Boto3](#) が必要です。
- RES AWS がデプロイされているアカウントとリージョンの CLI を設定し、RES によって作成されたバケットに書き込む S3 アクセス許可があることを確認します。

```
OUTPUT_DIRECTORY=<output-directory>
```

```
curl https://research-engineering-studio-us-east-1.s3.amazonaws.com/releases/2024.04.02/patch_scripts/patch.py --output ${OUTPUT_DIRECTORY}/patch.py
```

```
curl https://research-engineering-studio-us-east-1.s3.amazonaws.com/
releases/2024.04.02/patch_scripts/patches/samaccountname_regex.patch --output
${OUTPUT_DIRECTORY}/samaccountname_regex.patch
```

2. パッチスクリプトとパッチファイルがダウンロードされるディレクトリに移動します。次のパッチコマンドを実行し、を RES 環境の名前<environment-name>に置き換えます。

```
python3 patch.py --environment-name <environment-name> --res-version 2024.04.02 --
module cluster-manager --patch samaccountname_regex.patch
```

3. 環境の Cluster Manager インスタンスを再起動するには、次のコマンドを実行し、を RES 環境の名前<environment-name>に置き換えます。Amazon EC2 マネジメントコンソールからインスタンスを終了することもできます。

```
ENVIRONMENT_NAME=<environment-name>

INSTANCE_ID=$(aws ec2 describe-instances \
  --filters \
    Name=tag:Name,Values=${ENVIRONMENT_NAME}-cluster-manager \
    Name=tag:res:EnvironmentName,Values=${ENVIRONMENT_NAME} \
  --query "Reservations[0].Instances[0].InstanceId" \
  --output text)

aws ec2 terminate-instances --instance-ids ${INSTANCE_ID}
```

.....

(2024.04.02 以前) 踏み台ホストにアクセスするためのプライベートキーが無効です

問題

ユーザーがプライベートキーをダウンロードして RES ウェブポータルから踏み台ホストにアクセスすると、キーの形式が正しくありません。複数の行が 1 行としてダウンロードされるため、キーが無効になります。ダウンロードしたキーを使用して踏み台ホストにアクセスしようとすると、次のエラーが表示されます。

```
Load key "<downloaded-ssh-key-path>": error in libcrypto
<user-name>@<bastion-host-public-ip>: Permission denied (publickey,gssapi-keyex,gssapi-
with-mic)
```

影響を受けるバージョン

2024.04.02 以前

緩和策

このブラウザは影響を受けないため、Chrome を使用してキーをダウンロードすることをお勧めします。

または、の後に新しい行を作成し、の直前に-----BEGIN PRIVATE KEY-----別の行を作成して、キーファイルを再フォーマットすることもできます-----END PRIVATE KEY-----。

.....

(2024.06 以前) AD 同期中に RES に同期されていないグループメンバー

バグの説明

GroupOU が UserOU と異なる場合、グループメンバーは RES に正しく同期されません。UserOU

RES は、AD グループからユーザーを同期しようとする、ldapsearch フィルターを作成します。現在のフィルターは、GroupOU パラメータの代わりに UserOU GroupOU パラメータを誤って使用します。その結果、検索はユーザーを返すことができません。この動作は UsersOU と GroupOU が異なるインスタンスでのみ発生します。

影響を受けるバージョン

この問題は、すべての RES バージョン 2024.06 以前に影響します。

緩和策

問題を解決するには、次の手順に従います。

1. patch.py スクリプトと group_member_sync_bug_fix.patch ファイルをダウンロードするには、次のコマンドを実行し、をファイルをダウンロードするローカルディレクトリ<output-directory>に置き換え、をパッチを適用する RES のバージョン<res_version>に置き換えます。

Note

- パッチスクリプトには、[AWS CLI v2](#)、Python 3.9.16 以降、および [Boto3](#) が必要です。

- RES AWS がデプロイされているアカウントとリージョンの CLI を設定し、RES によって作成されたバケットに書き込む S3 アクセス許可があることを確認します。
- パッチは RES バージョン 2024.04.02 と 2024.06 のみをサポートしています。2024.04 または 2024.04.01 を使用している場合は、「」に記載されている手順に従って[マイナーバージョンの更新](#)、パッチを適用する前にまず環境を 2024.04.02 に更新できます。
- RES バージョン: RES 2024.04.02

パッチダウンロードリンク: [2024.04.02_group_member_sync_bug_fix.patch](#)

- RES バージョン: RES 2024.06

パッチダウンロードリンク: [2024.06_group_member_sync_bug_fix.patch](#)

```
OUTPUT_DIRECTORY=<output-directory>
RES_VERSION=<res_version>
mkdir -p ${OUTPUT_DIRECTORY}

curl https://research-engineering-studio-us-east-1.s3.amazonaws.com/releases/
${RES_VERSION}/patch_scripts/patch.py --output ${OUTPUT_DIRECTORY}/patch.py

curl https://research-engineering-studio-us-east-1.s3.amazonaws.com/releases/
${RES_VERSION}/patch_scripts/patches/${RES_VERSION}_group_member_sync_bug_fix.patch
--output ${OUTPUT_DIRECTORY}/${RES_VERSION}_group_member_sync_bug_fix.patch
```

2. パッチスクリプトとパッチファイルがダウンロードされるディレクトリに移動します。次のパッチコマンドを実行し、を RES 環境の名前<environment-name>に置き換えます。

```
cd ${OUTPUT_DIRECTORY}
ENVIRONMENT_NAME=<environment-name>

python3 patch.py --environment-name ${ENVIRONMENT_NAME} --res-
version ${RES_VERSION} --module cluster-manager --patch $PWD/
${RES_VERSION}_group_member_sync_bug_fix.patch
```

3. 環境の cluster-manager インスタンスを再起動するには、次のコマンドを実行します。

```
INSTANCE_ID=$(aws ec2 describe-instances \
--filters \
Name=tag:Name,Values=${ENVIRONMENT_NAME}-cluster-manager \
```

```
Name=tag:res:EnvironmentName,Values=${ENVIRONMENT_NAME}\  
--query "Reservations[0].Instances[0].InstanceId" \  
--output text)  
  
aws ec2 terminate-instances --instance-ids ${INSTANCE_ID}
```

(2024.06 以前) CVE-2024-6387、RegreSSHion、RHEL9 および Ubuntu VDIs のセキュリティ脆弱性

バグの説明

regreSSHion と呼ばれる [CVE-2024-6387](#) は、OpenSSH サーバーで識別されています。この脆弱性により、リモートの認証されていない攻撃者はターゲットサーバーで任意のコードを実行し、安全な通信に OpenSSH を利用するシステムに重大なリスクをもたらします。

RES の場合、標準設定は踏み台ホストを経由し、仮想デスクトップに SSH されます。踏み台ホストはこの脆弱性の影響を受けません。ただし、すべての RES バージョンで RHEL9 および Ubuntu2024 VDIs (仮想デスクトップインフラストラクチャ) に提供するデフォルトの AMI (Amazon マシンイメージ) は、セキュリティの脅威に対して脆弱な OpenSSH バージョンを使用します。

つまり、既存の RHEL9 および Ubuntu2024 VDIs は悪用される可能性があります。攻撃者は踏み台ホストにアクセスする必要があります。

問題の詳細については、[こちら](#)を参照してください。

影響を受けるバージョン

この問題は、すべての RES バージョン 2024.06 以前に影響します。

緩和策

RHEL9 と Ubuntu の両方が OpenSSH のパッチをリリースし、セキュリティの脆弱性を修正しました。これらは、プラットフォームのそれぞれのパッケージマネージャーを使用してプルできます。

既存の RHEL9 または Ubuntu VDIs がある場合は、以下の PATCH EXISTING VDIs の手順に従うことをお勧めします。今後の VDIs パッチを適用するには、PATCH FUTURE VDIs の手順に従うことをお勧めします。以下の手順では、スクリプトを実行してプラットフォームの更新を VDIs に適用する方法について説明します。

既存の VDI にパッチを適用する

1. 既存のすべての Ubuntu および RHEL9 VDI にパッチを適用する次のコマンドを実行します。
 - a. パッチスクリプトには [AWS CLI v2](#) が必要です。
 - b. RES がデプロイされているアカウントとリージョンの AWS CLI を設定し、AWS Systems Manager Run Command を送信するための Systems Manager アクセス許可があることを確認します。

```
aws ssm send-command \  
  --document-name "AWS-RunRemoteScript" \  
  --targets "Key=tag:res:NodeType,Values=virtual-desktop-dcv-host" \  
  --parameters '{"sourceType":["S3"],"sourceInfo":[{"path":"https://research-engineering-studio-us-east-1.s3.amazonaws.com/releases/2024.06/patch_scripts/scripts/patch_openssh.sh"}],"commandLine":["bash patch_openssh.sh"]}'
```

2. Run [Command ページ](#) でスクリプトが正常に実行されたことを確認できます。コマンド履歴タブをクリックし、最新のコマンド ID を選択し、すべてのインスタンス IDs に SUCCESS メッセージがあることを確認します。

将来の VDI パッチを適用する

1. パッチスクリプトとパッチファイル ([patch.py](#) および [update_openssh.patch](#)) をダウンロードするには、 をファイルをダウンロードするディレクトリ<output-directory>に、 を RES 環境の名前<environment-name>に置き換えて、次のコマンドを実行します。

Note

- パッチは RES 2024.06 にのみ適用されます。
- パッチスクリプトには、[AWS CLI v2](#)、Python 3.9.16 以降、および [Boto3](#) が必要です。
- RES がデプロイされているアカウントとリージョンに AWS CLI のコピーを設定し、RES によって作成されたバケットに書き込む S3 アクセス許可があることを確認します。

```
OUTPUT_DIRECTORY=<output-directory>
```

```
ENVIRONMENT_NAME=<environment-name>
```

```
curl https://research-engineering-studio-us-east-1.s3.amazonaws.com/
releases/2024.06/patch_scripts/patch.py --output ${OUTPUT_DIRECTORY}/patch.py
```

```
curl https://research-engineering-studio-us-east-1.s3.amazonaws.com/
releases/2024.06/patch_scripts/patches/update_openssh.patch --output
${OUTPUT_DIRECTORY}/update_openssh.patch
```

2. 次のパッチコマンドを実行します。

```
python3 ${OUTPUT_DIRECTORY}/patch.py --environment-name ${ENVIRONMENT_NAME} --res-
version 2024.06 --module virtual-desktop-controller --patch ${OUTPUT_DIRECTORY}/
update_openssh.patch
```

3. 次のコマンドを使用して、環境の " Controller インスタンスを再起動します。

```
INSTANCE_ID=$(aws ec2 describe-instances \
  --filters \
    Name=tag:Name,Values=${ENVIRONMENT_NAME}-vdc-controller \
    Name=tag:res:EnvironmentName,Values=${ENVIRONMENT_NAME}\
  --query "Reservations[0].Instances[0].InstanceId" \
  --output text)

aws ec2 terminate-instances --instance-ids ${INSTANCE_ID}
```

 Important

将来の VDI パッチ適用は、RES バージョン 2024.06 以降でのみサポートされています。2024.06 より前のバージョンの RES 環境の将来の VDI にパッチを適用するには、まず [手順を使用して RES 環境を 2024.06 にアップグレードします](#) [メジャーバージョンの更新](#)。

.....

注意

各 Amazon EC2 インスタンスには、管理目的で 2 つのリモートデスクトップサービス (ターミナルサービス) ライセンスが付属しています。この[情報は](#)、これらのライセンスを管理者にプロビジョニングするのに役立ちます。また、[を使用することもできます](#)。これにより[AWS Systems Manager Session Manager](#)、RDP を使用せずに、RDP ライセンスを必要とせずに Amazon EC2 インスタンスにリモート接続できます。追加のリモートデスクトップサービスライセンスが必要な場合は、リモートデスクトップユーザー CALs を Microsoft または Microsoft ライセンスリセラーから購入する必要があります。アクティブなソフトウェアアシュアランスを持つリモートデスクトップユーザー CALs にはライセンスモビリティの利点があり、デフォルト (共有) テナント環境に移行 AWS できます。ソフトウェアアシュアランスまたはライセンスモビリティのメリットなしでライセンスを持ち込む方法については、FAQ の[このセクション](#)を参照してください。

お客様は、本書に記載されている情報を独自に評価する責任を負うものとします。このドキュメント：(a) は情報提供のみを目的としています。(b) 現在の製品の提供とプラクティスを表します AWS 。 予告なしに変更される可能性があります。 および (c) は、AWS およびその関連会社からのコミットメントまたは保証を作成しません。 サプライヤーまたは licensors. AWS products または サービスは、保証なしで「現状有姿」で提供されます。 表現、またはあらゆる種類の条件、明示的か黙示的にかかわらず、お客様に対する AWS 責任と責任は AWS 契約によって管理されます。 このドキュメントは の一部ではありません。 も変更されません。 AWS とその顧客との間の契約。

の Research and Engineering Studio AWS は、Apache [Software Foundation で利用可能な Apache License Version 2.0](#) の条項に基づいてライセンスされます。

リビジョン

詳細については、GitHub リポジトリの [CHANGELOG.md](#) ファイルを参照してください。

日付	変更
2024 年 8 月	- リリースバージョン 2024.08 — - Amazon S3 バケットを Linux Virtual Desktop Infrastructure (VDI) インスタンスにマウントするサポートが追加されました。「Amazon S3 バケット」を参照してください。 - カスタムプロジェクトアクセス許可のサポート、既存のロールのカスタマイズとカスタムロールの追加を可能にする拡張アクセス許可モデルが追加されました。「アクセス許可プロファイル」を参照してください。 - ユーザーガイド: トラブルシューティング セクションを展開しました。
2024 年 6 月	- リリースバージョン 2024.06 — Ubuntu サポート、プロジェクト所有者のアクセス許可。 - ユーザーガイド: を追加 デモ環境を作成する
2024 年 4 月	リリースバージョン 2024.04 — RES 対応 AMIsとプロジェクト起動テンプレート
2024 年 3 月	その他のトラブルシューティングトピック、CloudWatch Logs の保持、マイナーバージョンのアンインストール
2024 年 2 月	リリースバージョン 2024.01.01 — デプロイテンプレートを更新

日付	変更
2024 年 1 月	リリースバージョン 2024.01
2023 年 12 月	GovCloud の指示とテンプレートを追加
2023 年 11 月	初回リリース

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。