ユーザーガイド

AWS Verified Access

AWS Verified Access: ユーザーガイド

Table of Contents

AWS Verified Accessとは

AWS Verified Accessを使用すると、仮想プライベートネットワーク (VPN) を使用しなくても、アプリケーションへの安全なアクセスを提供できます。Verified Access は各アプリケーションリクエストを評価し、指定されたセキュリティ要件を満たす場合にのみユーザーが各アプリケーションにアクセスできるようにサポートします。

Verified Access の利点

- セキュリティ状態の向上 — 従来のセキュリティモデルでは、アクセスを一度評価すると、すべてのアプリケーションへのアクセス権がユーザーに付与されます。Verified Access では、各アプリケーションのアクセスリクエストがリアルタイムで評価されます。これにより、脅威アクターがあるアプリケーションから別のアプリケーションに移動することが困難になります。
- セキュリティサービスとの統合 — Verified Access は、AWS とサードパーティーサービスの両方を含むデバイス管理サービスと ID を統合しています。Verified Access は、これらのサービスのデータを使用して、一連のセキュリティ要件に照らしてユーザーとデバイスの信頼性を検証し、ユーザーがアプリケーションに対するアクセス権を所有すべきかどうかを判断します。
- ユーザーエクスペリエンスの向上 — Verified Accessにより、ユーザーは VPN を使用してアプリケーションにアクセスする必要がなくなります。これにより、VPN 関連の問題から生じるサポートケースの数を減らすことができます。
- トラブルシューティングと監査の簡素化 — Verified Access はすべてのアクセス試行を記録し、アプリケーションへのアクセスを一元的に把握できるため、セキュリティインシデントや監査請求に迅速に対応できます。

AWS Verified Access へのアクセス

次のいずれかのインターフェイスを使用して Verified Access を操作できます。

- AWS Management Console – Verified Accessリソースの作成と管理に使用できるウェブインターフェイスを提供します。AWS Management Console にサインインして、Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
- AWS Command Line Interface (AWS CLI) — AWS Verified Access を始めとする一連のさまざまな AWS のサービス 用のコマンドを提供します。AWS CLI は、Windows、macOS、Linux でサポートされています。AWS CLI を取得する方法については、「AWS Command Line Interface」を参照してください。

- AWS SDK — 言語固有の API を提供します。AWS SDK は、署名の計算、リクエストの再試行処理およびエラー処理など、接続のさまざまな詳細を処理します。詳細については、AWSSDK をご参照ください。
- クエリ API – HTTPS リクエストを使用して呼び出す低レベル API アクションを提供します。クエリ API の使用は、Verified Access にアクセスする最も直接的な方法です。ただし、この方法では、リクエストに署名するハッシュの生成やエラー処理など、低レベルの詳細な作業をアプリケーションで処理する必要があります。詳細については、「Amazon EC2 API リファレンス」の「 Verified Access アクション」を参照してください。

このガイドでは、AWS Management Console を使用したVerified Access リソースの作成、アクセス、管理方法を説明しています。

料金

Verified Access 上のアプリケーションごとに時間単位で課金され、Verified Access で処理されたデータ量に対して課金されます。詳細については、「AWSVerified Access 料金」を参照してください。

Verified Access のしくみ

AWS Verified Access は、ユーザーからの各アプリケーションリクエストを評価し、以下に基づいてアクセスを許可します。

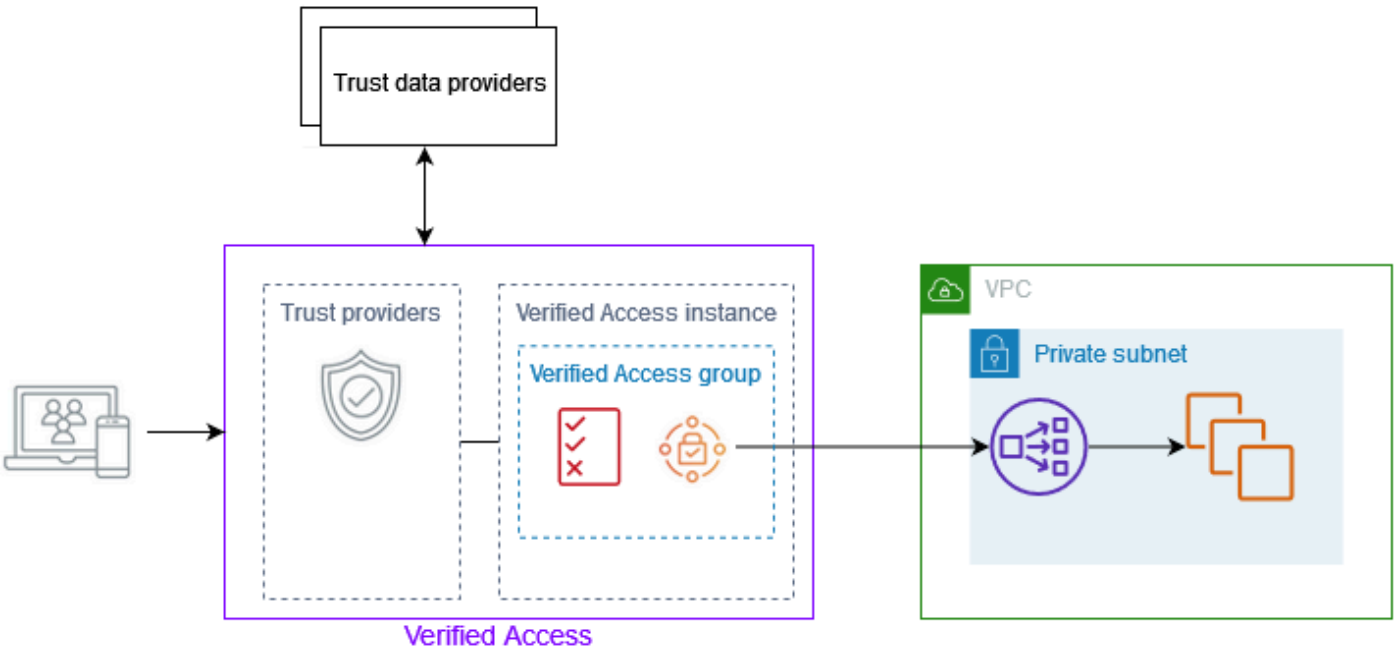
- 選択した信頼プロバイダーにより (AWS またはサードパーティから) 送信された信頼データ。
- Verified Access で作成したアクセスポリシー。

ユーザーがアプリケーションにアクセスしようとすると、Verified Access は信頼プロバイダーからデータを取得し、そのデータをアプリケーションに設定したポリシーと照らし合わせて評価します。Verified Access は、指定されたセキュリティ要件をユーザーが満たしている場合にのみ、要求されたアプリケーションへのアクセスを許可します。ポリシーが定義されるまで、すべてのアプリケーションリクエストはデフォルトで拒否されます。

さらに、Verified Access はすべてのアクセス試行を口グに記録するため、セキュリティインシデントや監査請求に迅速に対応できます。

Verified Access の主要コンポーネント

次の図は、Verified Access の仕組みの大まかな概要を示しています。ユーザーはアプリケーションへのアクセス要求を送信します。Verified Access は、グループのアクセスポリシーおよびアプリケーション固有のエンドポイントポリシーと照らし合わせてリクエストを評価します。Access が許可されている場合、リクエストはエンドポイントを介してアプリケーションに送信されます。



- Verified Access インスタンス — インスタンスはアプリケーションリクエストを評価し、セキュリティ要件が満たされた場合にのみアクセス権を付与します。
- Verified Access エンドポイント — 各エンドポイントはアプリケーションを表します。ロードバランサーエンドポイントまたはネットワークインターフェースエンドポイントを作成できます。
- Verified Access グループ — Verified Access エンドポイントのコレクション。同様のセキュリティ要件を持つアプリケーションのエンドポイントをグループ化し、ポリシー管理を簡素化することをお勧めします。たとえば、すべての営業アプリケーションのエンドポイントをグループ化できます。
- アクセスポリシー — アプリケーションへのアクセスを許可するか拒否するかを決定するユーザー定義のルール。ユーザー ID やデバイスのセキュリティ状態など、さまざまな要素を組み合わせて指定できます。Verified Access グループごとにグループアクセスポリシーを作成します。このポリシーは、グループ内のすべてのエンドポイントに継承されます。オプションでアプリケーション固有のポリシーを作成し、特定のエンドポイントに添付できます。
- 信頼プロバイダー — ユーザー ID やデバイスのセキュリティ状態を管理するサービス。Verified Access は、AWS およびサードパーティの信頼プロバイダーの両方を使用します。各 Verified Access インスタンスには少なくとも 1 つの信頼プロバイダーを接続する必要があります。各 Verified Access インスタンスには、1 つの ID 信頼プロバイダーと複数のデバイス信頼プロバイダーを接続できます。
- トラストデータ — 信頼プロバイダーが Verified Access に送信する、ユーザーまたはデバイスのセキュリティ関連データ。ユーザークレームまたはトラストコンテキストとも呼ばれます。たと

えば、ユーザーの電子メールアドレスやデバイスのオペレーティングシステムバージョンなどです。Verified Access は、アプリケーションへのアクセス要求を受信すると、このデータをアクセスポリシーと照らし合わせて評価します。

チュートリアル：Verified Access 入門

このチュートリアルを使用して、AWS Verified Access の使用を開始してください。Verified Access リソースを作成および設定する方法について学習します。

このアプリケーションを Verified Access に追加する前は、アプリケーションにアクセス可能なのはプライベートネットワークだけでした。このチュートリアル終了時には、特定のユーザーが VPN を使用せずにインターネット経由で同じアプリケーションにアクセスできるようになります。

> **Note**
>
> この例では、デバイスベースの信頼プロバイダーとの統合については説明していません。この例では、ID ベースの信頼プロバイダーのみを取り扱っています。

タスク

前提条件

このチュートリアルの前提条件は次の通りです。

- Verified Access の使用例を示すために、2つの AWS アカウント を使用します。1 つのアカウントがターゲットアプリケーションをホストし、もう 1 つのアカウントで Verified Access リソースを作成します。
- 現在作業している AWS リージョン で AWS IAM Identity Center を有効にします。その後、IAM アイデンティティセンターを Verified Access の信頼プロバイダーとして使用できます。詳細については、「AWS IAM Identity Centerユーザーガイド」の「IAM アイデンティティセンターを有効にする」を参照してください。
- パブリックホストドメインと、ドメインの DNS レコードを更新するために必要なアクセス許可。
- AWS アカウント 内の内部ロードバランサーの背後で実行されているアプリケーション。使用するアプリケーションドメイン名の例は、`www.myapp.example.com` です。
- IAM ポリシーに、ここに記載されている AWS Verified Access インスタンス Verified Access インスタンスを作成するためのポリシー を作成するために必要なすべてのアクセス許可が含まれていることを確認してください。

ステップ 1： Verified Access インスタンスを作成する

以下の手順に従って Verified Access インスタンスを作成します。

Verified Access インスタンスを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. Amazon VPC のナビゲーションペインで、[Verified Access インスタンス] を選択し、[Verified Access インスタンスの作成] を選択します。
3. (オプション) [名前] と [説明] に、Verified Access インスタンスの名前と説明を入力します。
4. [信頼プロバイダー] については、デフォルトのオプションをそのまま使用してください。
5. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。
6. [Verified Access インスタンスの作成] を選択します。

ステップ 2：信頼プロバイダーを設定する

AWS IAM Identity Centerを信頼プロバイダーとして設定できます。

IAM アイデンティティセンターの信頼プロバイダーを作成するには

1. Amazon VPC のナビゲーションペインで、[Verified Access 信頼プロバイダー] を選択し、[Verified Access 信頼プロバイダーの作成] を選択します。
2. (オプション) [名前タグ] と [説明] に、Verified Access 信頼プロバイダーの名前と説明を入力します。
3. 後でポリシー参照名のポリシールールを利用するときに使用するカスタム ID を入力します。例えば、「**idc**」 と入力します。
4. [信頼プロバイダのタイプ] で、[ユーザー信頼プロバイダー] を選択します。
5. [ユーザー信頼プロバイダのタイプ] で [IAM アイデンティティセンター] を選択します。
6. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。
7. [Verified Access 信頼プロバイダーの作成] を選択します。

ステップ 3：信頼プロバイダーをインスタンスに接続します。

次の手順に従って、Verified Access インスタンスに信頼プロバイダーを接続します。

信頼プロバイダーをインスタンスに接続するには

1. Amazon VPC ナビゲーションペインで、[Verified Access インスタンス] を選択します。
2. インスタンスを選択します。
3. [アクション]、[Verified Access 信頼プロバイダーを接続] を選択します。
4. [Verified Access 信頼プロバイダー] で、信頼プロバイダーを選択します。
5. [Verified Access 信頼プロバイダーを接続] を選択します。

ステップ 4：Verified Access グループを作成する

次のステップで作成するエンドポイントに使用できるグループを作成しましょう。

Verified Access グループを作成するには

1. Amazon VPC ナビゲーションペインで、[Verified Access グループ] を選択し、次に [Verified Access グループの作成] を選択します。
2. (オプション) [名前タグ] と [説明] に、グループの名前と説明を入力します。

3. [Verified Access インスタンス] で、Verified Access インスタンスを選択します。
4. [ポリシー定義] については、空白のままにしてください。このチュートリアルの後半で、ポリシーを作成します。
5. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。
6. [Verified Access グループの作成] を選択します。

ステップ 5：AWS Resource Access Manager で Verified Access グループを共有する

このステップでは、作成したグループを、ターゲットアプリケーションが実行されているAWS アカウント と共有します。Verified Access グループを共有するには、リソース共有に追加する必要があります。リソース共有がない場合は、まずリソース共有を作成する必要があります。

AWS Organizations に属していて、組織内での共有が有効になっている場合、組織内のコンシューマーには共有 Verfied Access に対するアクセス許可が自動的に付与されます。これに該当しない場合、コンシューマーはリソース共有への参加の招待を受け取り、その招待を受け入れた後で、共有 Verified Access グループに対するアクセス許可が付与されます。

「AWS RAM ユーザーガイド」の「リソース共有を作成する」の手順に従います。[リソースタイプを選択]で、[Verified Access グループ] を選択し、Verified Access グループのチェックボックスを選択します。

詳細については、「AWS RAM ユーザーガイド」の「使用開始」を参照してください。

ステップ 6：エンドポイントを作成してアプリケーションを追加する

次の手順に従ってエンドポイントを作成します。このステップは、Elastic Load Balancing の内部ロードバランサーの背後でアプリケーションを実行していることを前提としています。

Verified Access エンドポイントを作成するには

1. Amazon VPC ナビゲーションペインで、[Verified Access エンドポイント] を選択し、[Verified Access エンドポイントの作成] を選択します。
2. (オプション) [名前タグ] と [説明] に、エンドポイントの名前と説明を入力します。

3. [Verified Access グループ] では、Verified Access グループを選択します。
4. [アプリケーション詳細] では、次の操作を行います。
   a. [アプリケーションドメイン] には、アプリケーションの DNS 名を入力します。
   b. [ドメイン証明書の Amazon リソースネーム（ARN）] で、パブリック TLS 証明書の Amazon リソースネーム（ARN）を選択します。
5. [エンドポイント詳細] では、次の操作を行います。
   a. [添付タイプ] で、[VPC] を選択します。
   b. (オプション) [セキュリティグループ]で、エンドポイントに関連付けるセキュリティグループを選択します。
   c. [エンドポイントのドメインプレフィックス] には、カスタム ID を入力します。これは Verified Access が生成する DNS 名の前に付加されます。この例では、**my-ava-app** を使用します。
   d. [エンドポイントタイプ] で、[ロードバランサー] を選択します。
   e. [プロトコル] で [HTTPS または HTTP] を選択します。これはロードバランサーの設定によって異なります。
   f. [Port (ポート)] に、ポート番号を入力します。これはロードバランサーの設定によって異なります。
   g. [ロードバランサー ARN] では、ロードバランサーを選択します。
   h. [サブネット] では、ロードバランサーに関連付けられているサブネットを選択します。
6. [ポリシー定義] には、現時点ではポリシーを入力しないでください。これについては、このチュートリアルの後半で説明します。
7. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。
8. [Verified Access エンドポイントの作成] を選択します。

ステップ 7：DNS を設定する

このステップでは、アプリケーションのドメイン名 (www.myapp.example.com など) を Verified Access エンドポイントのドメイン名にマッピングします。DNS のマッピングを完了するには、DNS プロバイダーで Canonical Name Record (CNAME)を作成します。CNAME レコードを作成すると、ユーザーからアプリケーションへのすべてのリクエストが Verified Access に送信されます。

エンドポイントの ドメイン名を入手するためには

1. Amazon VPC のナビゲーションペインで、[Verified Acceess エンドポイント] を選択します。
2. 前に作成したエンドポイントを選択します。
3. エンドポイントの [詳細] タブを選択します。
4. エンドポイントドメインを [エンドポイントドメイン] からコピーします。

このチュートリアルでは、エンドポイントのドメイン名は「`my-ava-app.edge-1a2b3c4d5e6f7g.vai-1a2b3c4d5e6f7g.prod.verified-access.us-west-2.amazonaws.com`」になります。

DNS プロバイダーで CNAME レコードを作成します。

レコード名	タイプ	値
www.myapp.example.com	CNAME	my-ava-app.edge-1a2b3c4d5e6f7g.vai-1a2b3c4d5e6f7g.prod.verified-access.us-west-2.amazonaws.com

ステップ 8 : アプリケーションへの接続をテストする

これで、アプリケーションへの接続をテストできます。アプリケーションのドメイン名をウェブブラウザに入力します。Verified Access ポリシーのデフォルト動作では、すべてのリクエストが拒否されます。誰もがアクセスできるようにするポリシーをまだ制定していないため、すべてのリクエストは拒否されるはずです。

ステップ 9:グループレベルのアクセスポリシーを設定する

以下の手順に従って Verified Access グループを変更し、アプリケーションへの接続を許可するアクセスポリシーを設定します。ポリシーの詳細は、IAM アイデンティティセンターに設定されているユーザーとグループによって異なります。ポリシーの作成の詳細については、「Verified Access ポリシー」を参照してください。

Verified Access グループを変更するには

1. Amazon VPC ナビゲーションペインで、[Verified Access グループ] を選択します。
2. グループを選択します。
3. [アクション]、[Verified Access グループポリシーの変更] を選択します。
4. ポリシーを入力します。
5. [Verified Access グループポリシーの変更] を選択します。

ステップ 10：接続を再テストする

グループポリシーが設定されたので、アプリケーションにアクセスできます。アプリケーションのドメイン名をウェブブラウザに入力します。リクエストが許可され、アプリケーションにリダイレクトされるはずです。

クリーンアップ

テストが終了したら、以下の手順に従って作成されたリソースを削除します。

このチュートリアルで作成した Verified Access リソースを削除するには

1. Amazon VPC のナビゲーションペインで、[Verified Acceess エンドポイント] を選択します。削除するエンドポイントを選択します。[アクション]、[Verified Access エンドポイントの削除] を選択します。
2. ナビゲーションペインで、[Verified Access グループ] を選択します。削除するグループを選択します。[アクション]、[Verified Access グループの削除] を選択します。注 - エンドポイントの削除処理が完了するまで数分かかる場合があります。
3. Amazon VPC ナビゲーションペインで、[Verified Access インスタンス] を選択します。このチュートリアル用に作成したインスタンスを選択します。[アクション]、[Verified Access 信頼プロバイダーを切り離す] を選択します。ドロップダウンリストから信頼プロバイダーを選択し、[Verified Access 信頼プロバイダーを切り離す] を選択します。
4. Amazon VPC ナビゲーションペインで、[Verified Access 信頼プロバイダー] を選択します。このチュートリアルで作成した信頼プロバイダーを選択します。[アクション]、[Verified Access 信頼プロバイダーの削除] を選択します。
5. Amazon VPC ナビゲーションペインで、[Verified Access インスタンス] を選択します。このチュートリアル用に作成したインスタンスを選択します。[アクション]、[Verified Acess インスタンスの削除] を選択します。

Verified Access インスタンス

AWS Verified Access インスタンスは、AWS信頼プロバイダーと Verified Access グループを整理するのに役立つリソースです。

トピック

Verified Access インスタンスの作成

以下の手順に従って Verified Access インスタンスを作成します。

Verified Access インスタンスを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [Verified Access インスタンス] を選択し、[Verified Access インスタンスの作成] を選択します。
3. (オプション) [名前] と [説明] に、Verified Access インスタンスの名前と説明を入力します。
4. (オプション) Verified Access を FIPS に準拠させる必要がある場合は、[連邦情報処理標準 (FIPS)] を [有効にする] を選択します。
5. (オプション) [信頼プロバイダー] では、Verified Access インスタンスに添付する信頼プロバイダーを選択します。
6. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。
7. [Verified Access インスタンスの作成] を選択します。

インスタンスに信頼プロバイダーを添付する

以下の手順に従って Verified Access インスタンスに信頼プロバイダーを添付します。

Verified Access インスタンスに信頼プロバイダーを添付するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. インスタンスを選択します。
4. [アクション]、[Verified Access 信頼プロバイダーを添付] を選択します。
5. [Verified Access 信頼プロバイダー] では、信頼プロバイダーを選択します。
6. [Verified Access 信頼プロバイダーを添付] を選択します。

インスタンスから信頼プロバイダーを切り離す

以下の手順に従って、Verified Access インスタンスから信頼プロバイダーを切り離すします。

Verified Access インスタンスから信頼プロバイダーを切り離すには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. インスタンスを選択します。
4. [アクション]、[Verified Access 信頼プロバイダーを切り離す] を選択します。
5. [Verified Access 信頼プロバイダー] で、信頼プロバイダーを選択します。
6. [Verified Access 信頼プロバイダーを切り離す] を選択します。

Verified Access インスタンスの削除

浮揚になった Verified Access インスタンスは、削除することができます。インスタンスを削除する前に、関連付けられている信頼プロバイダーまたは Verified Access グループをすべて削除する必要があります。

Verified Access インスタンスを削除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. Verified Access インスタンスを選択します。
4. [アクション]、[Verified Acess インスタンスの削除] を選択します。
5. 確認を求められたら、「**delete**」と入力してから、[Delete] (削除) を選択します。

AWS WAF との統合

Verified Access によって適用される認証ルールと承認ルールに加えて、ペリメータ保護を適用する場合もあります。これにより、アプリケーションを他の脅威から保護することができます。AWS WAF を Verified Access デプロイに統合することでこれを実現できます。AWS WAF は、保護されたウェブアプリケーションリソースに転送される HTTP(S) リクエストをモニタリングできるウェブアプリケーションファイアウォールです。AWS WAF の詳細については、「AWS WAF デベロッパーガイド」の「AWS WAF」を参照してください。

AWS WAF ウェブアクセスコントロールリスト（ACL）を Verified Access インスタンスに関連付けることで、AWS WAF と Verified Access を統合することができます。ウェブ ACL は、保護されたリソースが応答するすべての HTTP(S) ウェブリクエストをきめ細かく制御できるようにする AWS WAF のリソースです。AWS WAF 関連付けまたは関連付け解除のリクエストが処理されている間、インスタンスに接続されている Verified Access エンドポイントのステータスは `updating` として表示されます。リクエストが完了すると、ステータスは `active` に戻ります。ステータスは、AWS Management Console または、AWS CLI でエンドポイントを説明することで確認できます。

> **Note**
>
> AWS WAF コンソールまたは API を使用してこの統合を行うこともできます。Verified Access インスタンスの Amazon リソースネーム (ARN) が必要になります。この ARN は、次の形式を使用して作成できます：`arn:${Partition}:ec2:${Region}:${Account}:verified-access-instance/${VerifiedAccessInstanceId}`。

トピック

- AWS WAF の統合に必要な IAM アクセス許可
- AWS WAF ウェブ ACL を関連付ける
- AWS WAF インテグレーションのステータスの確認
- AWS WAFウェブ ACL の関連付けを解除する

AWS WAF の統合に必要な IAM アクセス許可

AWS WAF と Verified Access の統合には、API 操作に直接対応しないアクセス許可限定のアクションが含まれています。これらのアクションについては、「`[permission only]` によるAWS Identity and Access Management サービス認証リファレンス」に記載されています。詳細について

は、「サービス認証リファレンス」の「Amazon EC2 のアクション、リソース、および条件キー」を参照してください。

ウェブ ACL を使用するには、AWS Identity and Access Management プリンシパルに以下のアクセス許可が必要です。

- `ec2:AssociateVerifiedAccessInstanceWebAcl`
- `ec2:DisassociateVerifiedAccessInstanceWebAcl`
- `ec2:DescribeVerifiedAccessInstanceWebAclAssociations`
- `ec2:GetVerifiedAccessInstanceWebAcl`

AWS WAF ウェブ ACL を関連付ける

次の手順は、AWS Management Console を使用して AWS WAF ウェブアクセスコントロールリスト (ACL) を Verified Access インスタンスに関連付ける方法を示しています。

> **Tip**
>
> 以下の手順を完了するには、既存の AWS WAF ウェブ ACL が必要です。ウェブ ACL の詳細については、「AWS WAF デベロッパーガイド」の「ウェブ ACL の管理と使用 」を参照してください。

AWS WAF ウェブ ACL を Verified Access インスタンスに関連付けるためには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. Verified Access インスタンスを選択します。
4. [統合] タブを選択します。
5. [アクション]、[ウェブ ACL の関連付け] の順に選択します。
6. [ウェブ ACL] では、既存のウェブ ACL を選択し、[ウェブ ACL を関連付ける] を選択します。

AWS Management Console を使用して AWS WAF がこのタスクを実行することもできます。詳細については、「AWS WAF デベロッパーガイド」の「ウェブ ACL と AWS リソースの関連付けまたは関連付け解除」を参照してください。

AWS WAF インテグレーションのステータスの確認

AWS WAF ウェブアクセスコントロールリスト（ACL）が Verified Access インスタンスに関連付けられているかどうかは、AWS Management Console を使用して確認できます。

Verified Access インスタンスと AWS WAF との統合ステータスを表示するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. Verified Access インスタンスを選択します。
4. [統合] タブを選択します。
5. WAF 統合ステータスにリストされている詳細を確認します。ステータスは、関連付けされた状態の場合、ウェブ ACL 識別子と共に、 [関連付け済み]または [関連付けなし] として表示されます。

AWS WAFウェブ ACL の関連付けを解除する

次の手順は、AWS WAFを使用してAWS Management Consoleウェブアクセスコントロールリスト (ACL) と Verified Access インスタンスの関連付けを解除する方法を示しています。

AWS WAFウェブ ACL と Verified Access インスタンスの関連付けを解除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. Verified Access インスタンスを選択します。
4. [統合] タブを選択します。
5. [アクション] を選択し、[ウェブ ACL の関連付け解除] を選択します。
6. [ウェブ ACL の関連付け解除] を選択して確定します。

AWS Management Console を使用して AWS WAF がこのタスクを実行することもできます。詳細については、「AWS WAF デベロッパーガイド」の「ウェブ ACL と AWS リソースの関連付けまたは関連付け解除」を参照してください。

Verified Access の FIPS 準拠

連邦情報処理規格（Federal Information Processing Standards/FIPS）は、機密情報を保護する暗号モジュールのセキュリティ要件を規定する米国およびカナダ政府の基準です。AWS Verified Access は、FIPS 刊行物140-2 に順守するように環境を設定するオプションを提供します。Verified Access の FIPS 準拠は、次のAWSリージョンで利用できます。

- 米国東部 (オハイオ)
- 米国東部 (バージニア北部)
- 米国西部 (北カリフォルニア)
- 米国西部 (オレゴン)
- カナダ (中部)

このページでは、新規または既存の Verified Access 環境を FIPS 準拠するように設定する方法を説明します。

トピック

既存の Verified Access 環境を FIPS に準拠するように設定する

既存の Verified Access 環境があり、それを FIPS に準拠するように設定したい場合、一部のリソースを削除し再作成して FIPS コンプライアンスを有効にする必要があります。

既存の AWS Verified Access 環境を FIPS に準拠するように再構成するには、次の手順に従います。

1. 元の Verified Access エンドポイント、グループ、インスタンスを削除します。設定した信頼プロバイダーは再利用できます。
2. Verified Access インスタンスを作成します。作成時には必ず連邦情報処理標準 (FIPS) を有効にしてください。また、作成時に、使用する [Verified Access 信頼プロバイダー] をドロップダウンリストから選択して添付します。
3. Verified Access グループを作成します。グループの作成時、作成したばかりの Verified Access インスタンスにそのグループを関連付けます。
4. 1 つ以上の Verified Access エンドポイント を作成します。エンドポイントの作成時に、前のステップで作成したグループにエンドポイントを関連付けます。

新しい Verified Access 環境を FIPS に準拠するように設定する

FIPS に準拠する新しい AWS Verified Access 環境を設定するには、次の手順に従います。

1. 信頼プロバイダーを設定します。必要に応じて、ユーザー ID 信頼プロバイダー、(オプションで) デバイスベースの 信頼プロバイダーを作成する必要があります。
2. Verified Access インスタンスを作成します。処理中は必ず連邦情報処理標準 (FIPS) を有効にしてください。また、作成時に、前のステップで作成した Verified Access 信頼プロバイダー をドロップダウンリストから選択して添付します。
3. Verified Access グループを作成します。グループの作成時、作成したばかりの Verified Access インスタンスにそのグループを関連付けます。
4. 1 つ以上の Verified Access エンドポイント を作成します。エンドポイントの作成時に、前のステップで作成したグループにエンドポイントを関連付けます。

Verified Access 信頼プロバイダー

信頼プロバイダーは、ユーザーとデバイスに関する情報を AWS Verified Access に送信するサービスです。この情報はトラストコンテキストと呼ばれます。これには、メールアドレスや「営業」組織のメンバーなどのユーザー ID に基づく属性や、インストール済みのセキュリティパッチやウイルス対策ソフトウェアのバージョンなどのデバイス管理情報が含まれる場合があります。

Verified Access は、以下のカテゴリの信頼プロバイダーをサポートします。

- ユーザー ID — ユーザーのデジタルアイデンティティを保存および管理する ID プロバイダー (IdP) サービス。
- デバイス管理 — ラップトップ、タブレット、スマートフォンなどのデバイス用のデバイス管理システム。

目次

ユーザー ID 信頼プロバイダー

AWS IAM Identity Center あるいは、OpenID Connect 互換のユーザー ID 信頼プロバイダーのどちらを使用するかを選択できます。

目次

IAM アイデンティティセンター を信頼プロバイダーとして使用

AWS Verified Access では、ユーザー ID 信頼プロバイダーとしてAWS IAM Identity Centerを使用できます。

前提条件と考慮事項

- IAM アイデンティティセンターのインスタンスは AWS Organizations インスタンスでなければなりません。スタンドアロンの AWS アカウントの IAM アイデンティティセンターインスタンスは機能しません。
- IAM アイデンティティセンターインスタンスは、Verified Access 信頼プロバイダーを作成する AWS リージョンと同じリージョンで有効にする必要があります。

さまざまなインスタンスタイプの詳細については、AWS IAM Identity Center ユーザーガイドの「IAM アイデンティティセンターの組織とアカウントインスタンスの管理」を参照してください。

IAM アイデンティティセンター信頼プロバイダーの作成

AWS アカウントで IAM アイデンティティセンターが有効になると、以下の手順に従って IAM アイデンティティセンターを Verified Access の信頼プロバイダーとして設定できます。

IAM アイデンティティセンターの信頼プロバイダーを作成するには (AWSコンソール)

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、「 Verified Access 信頼プロバイダー」を選択し、[Verified Access 信頼プロバイダーの作成] を選択します。
3. (オプション) [名前タグ] と [説明] に、信頼プロバイダーの名前と説明を入力します。
4. [ポリシー参照名] には、後でポリシールールを利用するときに使用する識別子を入力します。
5. [信頼プロバイダのタイプ] で、[ユーザー信頼プロバイダー] を選択します。
6. [ユーザー信頼プロバイダのタイプ] で [IAM アイデンティティセンター] を選択します。
7. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。
8. [Verified Access 信頼プロバイダーの作成] を選択します。

IAM アイデンティティセンターの信頼プロバイダー (AWSCLI) を作成するには

- create-verified-access-trust-provider (AWS CLI)

IAM アイデンティティセンターの信頼プロバイダーの削除

信頼プロバイダーを削除する前に、信頼プロバイダーが添付されているインスタンスからすべてのエンドポイントとグループ設定を削除する必要があります。

IAM アイデンティティセンターの信頼プロバイダーを削除するには (AWS コンソール)

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [Verified Access 信頼プロバイダー] を選択し、[Verified Access 信頼プロバイダー] で削除する信頼プロバイダーを選択します。
3. 「アクション」、「 Verified Access 信頼プロバイダの削除」の順に選択します。
4. テキストボックスに「delete」と入力して削除を確定します。
5. [Delete] (削除) を選択します。

IAM アイデンティティセンターの信頼プロバイダーを削除するには (AWS CLI)

- delete-verified-access-trust-provider (AWS CLI)

OpenID Connect 信頼プロバイダーの使用

AWS Verified Access は、標準の OpenID Connect (OIDC) メソッドを使用する ID プロバイダーをサポートします。Vefiried Access では、OIDC 互換プロバイダーをユーザー ID 信頼プロバイダーとして使用できます。ただし、可能性のある OIDC プロバイダーは多岐にわたるため、AWS では Verified Access と各 OIDC との統合をテストすることはできません。

Verified Access は、評価対象のトラストデータを OIDC プロバイダーの `UserInfo Endpoint` から取得します。この `Scope` パラメータは、検索するトラストデータのセットを決定するために使用されます。トラストデータを受信すると、Verified Access ポリシーがそのデータに対して評価されます。

> **Note**
>
> Verified Access は、Verified Access ポリシーを評価する際に OIDC プロバイダーにより送信された `ID token` からのトラストデータを使用しません。`UserInfo Endpoint` からのトラストデータのみがポリシーに照らして評価されます。

目次

OIDC 信頼プロバイダーを作成するための前提条件

信頼プロバイダーサービスから次の情報を直接収集する必要があります。

- Issuer
- 認可エンドポイント
- トークンエンドポイント
- UserInfo エンドポイント
- クライアント ID
- クライアントシークレット
- スコープ

OIDC 信頼プロバイダーの作成

以下の手順に従って、信頼プロバイダーとして OIDC を作成します。

OIDC 信頼プロバイダーを作成するには (AWS コンソール)

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、「 Verified Access 信頼プロバイダー」を選択し、[Verified Access 信頼プロバイダーの作成] を選択します。
3. (オプション) [名前タグ] と [説明] に、信頼プロバイダーの名前と説明を入力します。
4. [ポリシー参照名] には、後でポリシールールを利用するときに使用する識別子を入力します。
5. [信頼プロバイダのタイプ] で、[ユーザー信頼プロバイダー] を選択します。
6. [ユーザ信頼プロバイダのタイプ] で、[OIDC (OpenID Connect)] を選択します。
7. [Issuer] には、OIDC 発行者の ID を入力します。
8. [認証エンドポイント] には、認証エンドポイントの完全な URL を入力します。

9. [トークンエンドポイント] には、トークンエンドポイントの完全な URL を入力します。
10. [ユーザーエンドポイント] には、ユーザーエンドポイントの完全な URL を入力します。
11. [クライアント ID]に、OAuth 2.0 クライアント ID を入力します。
12. [クライアントシークレット] に OAuth 2.0 クライアントシークレットを入力します。
13. ID プロバイダーで定義されている対象範囲のスペースで区切られたリストを入力します。対象範囲には少なくとも「openid」対象範囲が必要です。
14. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。
15. [Verified Access 信頼プロバイダーの作成] を選択します。

> **Note**
>
> OIDC プロバイダーの許可リストにリダイレクト URI を追加する必要があります。このためは、Verified Access エンドポイントの `ApplicationDomain` を使用します。これは、Verified Access エンドポイントの [詳細] タブにあるAWS Management Console、または、エンドポイントを説明する AWS CLI を使用して確認できます。OIDC プロバイダーの許可リストに以下を追加してください。https://`ApplicationDomain`/oauth2/idpresponse

OIDC 信頼プロバイダーを作成するには (AWS CLI)

- create-verified-access-trust-provider (AWS CLI)

OIDC 信頼プロバイダーの変更

信頼プロバイダーの作成後、その設定を更新できます。

OIDC 信頼プロバイダーを変更するには (AWS コンソール)

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [Verified Access 信頼プロバイダー] を選択し、[Verified Access 信頼プロバイダー] で変更する信頼プロバイダーを選択します。
3. [アクション]、[Verified Access 信頼プロバイダの変更] の順に選択します。
4. オプションを変更します。
5. [Verified Access 信頼プロバイダーの変更] を選択します。

OIDC 信頼プロバイダー (AWSCLI) を変更するには

- modify-verified-access-trust-provider (AWS CLI)

OIDC 信頼プロバイダーの削除

ユーザーの信頼プロバイダーを削除する前に、まず信頼プロバイダーが添付されているインスタンスからすべてのエンドポイントとグループ設定を削除する必要があります。

OIDC 信頼プロバイダーを削除するには (AWSコンソール)

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [Verified Access 信頼プロバイダー] を選択し、[Verified Access 信頼プロバイダー] で削除する信頼プロバイダーを選択します。
3. 「アクション」、「 Verified Access 信頼プロバイダの削除」の順に選択します。
4. テキストボックスに「`delete`」と入力して削除を確定します。
5. [Delete] (削除) を選択します。

OIDC 信頼プロバイダーを削除するには (AWS CLI)

- delete-verified-access-trust-provider (AWS CLI)

デバイスベースの信頼プロバイダー

AWS Verified Access では、デバイス信頼プロバイダーを使用できます。Verified Access インスタンスでは 1 つまたは複数のデバイス信頼プロバイダーを使用できます。

目次

サポートされているデバイス信頼プロバイダー

以下のデバイス信頼プロバイダーは Verified Access と統合できます。

- CrowdStrike — CrowdStrike と Verified Access でのプライベートアプリケーションの保護
- Jamf — Verified Access と Jamf デバイス ID の統合
- JumpCloud — JumpCloud と AWS Verified Access の統合

デバイスベースの信頼プロバイダーの作成

これらの手順に従って、Verified Access で使用するデバイス信頼プロバイダーを作成して設定します。

Verified Access デバイス信頼プロバイダーを作成するには (AWS コンソール)

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、「 Verified Access 信頼プロバイダー」を選択し、[Verified Access 信頼プロバイダーの作成] を選択します。
3. (オプション) [名前タグ] と [説明] に、信頼プロバイダーの名前と説明を入力します。
4. ポリシー参照名のポリシールールを後で利用する際に使用する識別子を入力します。
5. [信頼プロバイダーのタイプ] には、[デバイス ID] を選択します。
6. [デバイス ID タイプ] には、[Jamf] または [CrowdStrike] または [JumpCloud] を選択します。
7. [テナント ID] には、テナントアプリケーションの識別子を入力します。
8. (オプション) [パブリック署名キー URL] には、デバイス信頼プロバイダーが共有する一意のキー URL を入力します。(このパラメータは Jamf、CrowdStrike、または JumpCloud では必要ありません。)
9. [Verified Access 信頼プロバイダーの作成] を選択します。

> **Note**
>
> OIDC プロバイダーの許可リストにリダイレクト URI を追加する必要があります。このためは、Verified Access エンドポイントの `DeviceValidationDomain` を使用します。これは、Verified Access エンドポイントの [詳細] タブにあるAWS Management Console、または、エンドポイントを説明する AWS CLI を使用して確認できます。OIDC プロバイダーの許可リストに以下を追加してください。https://`DeviceValidationDomain`/oauth2/idpresponse

Verified Access デバイスの信頼プロバイダーを作成するには (AWS CLI)

- create-verified-access-trust-provider (AWS CLI)

デバイスベースの信頼プロバイダーの変更

信頼プロバイダーの作成後、その設定を更新できます。

Verified Access デバイスの信頼プロバイダーを変更するには (AWS コンソール)

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access 信頼プロバイダー] を選択します。
3. 信頼プロバイダーを選択します。
4. [アクション] を選択し、[Verified Access 信頼プロバイダーの変更] を選択します。
5. 必要に応じて説明を変更します。
6. (オプション) [パブリック署名キー URL] では、デバイス信頼プロバイダーが共有する一意のキー URL を変更します。(ご使用のデバイス信頼プロバイダーが Jamf、CrowdStrike、または JumpCloud の場合、このパラメータは必要ありません。)
7. [Verified Access 信頼プロバイダーの変更] を選択します。

Verified Access デバイスの信頼プロバイダーを変更するには (AWS CLI)

- modify-verified-access-trust-provider (AWS CLI)

デバイスベースの信頼プロバイダーの削除

不要になった信頼プロバイダーは、削除することができます。

Verified Access デバイスの信頼プロバイダーを削除するには (AWS コンソール)

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access 信頼プロバイダー] を選択します。
3. 「Verified Access 信頼プロバイダー」で、削除する信頼プロバイダーを選択します。
4. [アクション] を選択し、[Verified Access 信頼プロバイダーの削除] を選択します。
5. 確認を求められたら、「**delete**」と入力してから、[Delete] (削除) を選択します。

Verified Access デバイスの信頼プロバイダーを削除するには（AWS CLI）

- delete-verified-access-trust-provider (AWS CLI)

Verified Access グループ

AWS Verified Access グループは、Verified Access エンドポイントとグループレベルの Verified Access ポリシーのコレクションです。グループ内の各エンドポイントは、Verified Access ポリシーを共有します。グループを使用して、共通のセキュリティ要件を持つエンドポイントをまとめることができます。これにより、1 つのポリシーで複数のアプリケーションのセキュリティニーズに対応できるため、ポリシー管理の簡素化に役立ちます。

たとえば、すべての営業アプリケーションをグループ化して、グループ全体のアクセスポリシーを設定できます。その後、このポリシーを使用して、すべての営業アプリケーションに共通の最低限のセキュリティ要件を定義できます。このアプローチは、ポリシー管理の簡素化に役立ちます。

グループを作成する際、グループを Verified Access インスタンスに関連付ける必要があります。エンドポイントを作成する過程で、エンドポイントをグループに関連付けます。

タスク

- Verified Access グループの作成
- Verified Access グループポリシーの変更
- Verified Access グループを削除する

Verified Access グループの作成

以下の手順に従って、Verified Access グループを作成します。

Verified Access グループを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで, [Verified Access グループ] を選択し、次に [Verified Access グループの作成] を選択します。
3. (オプション) [名前タグ] と [説明] に、グループの名前と説明を入力します。
4. [Verified Access インスタンス] には、グループに関連付ける Verified Access インスタンスを選択します。
5. (オプション)[ポリシー定義] には、グループに適用する Verified Access ポリシーを入力します。
6. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。

7. [Verified Access グループの作成] を選択します。

Verified Access グループポリシーの変更

以下の手順に従って、Verified Access グループポリシーを変更します。

Verified Access グループポリシーを変更するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access グループ]を選択し、ポリシーを変更するグループを選択します。
3. [アクション] を選択し、[Verified Access グループポリシーの変更] を選択します。
4. (オプション) 現在の目標に応じて [ポリシーを有効にする]をオンまたはオフにします。
5. (オプション)[ポリシー] には、グループに適用する Verified Access ポリシーを入力します。
6. [Verified Access グループポリシーの変更] を選択します。

Verified Access グループを削除する

不要になった Verified Access グループは、削除することができます。

Verified Access グループを削除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access グループ] を選択します。
3. グループを選択します。
4. [アクション]、[Verified Access グループの削除] を選択します。
5. 確認を求められたら、「**delete**」と入力してから、[Delete] (削除) を選択します。

Verified Access エンドポイント

Verified Access エンドポイントはアプリケーションを表します。各エンドポイントは Verified Access グループに関連付けられ、グループのアクセスポリシーを継承します。オプションで、アプリケーション固有のエンドポイントポリシーを各エンドポイントに添付することができます。

目次

Verified Access エンドポイントタイプ

可能なエンドポイントの種類は次のとおりです。

- ロードバランサー — アプリケーションリクエストはロードバランサーに送信され、アプリケーションに配布されます。
- ネットワークインターフェース — アプリケーションリクエストは、指定されたプロトコルとポートを使用してネットワークインターフェースに送信されます。

共有 VPC およびサブネット

共有 VPC サブネットに関する動作は次のとおりです。

- Verified Access エンドポイントは VPC サブネット共有でサポートされています。参加者は共有サブネットに Verified Access エンドポイントを作成できます。
- エンドポイントを作成した参加者がエンドポイントの所有者となり、エンドポイントを変更できるのはその参加者だけです。VPC 所有者はエンドポイントの変更を許可されません。

- Verified Access エンドポイントは AWS Local Zones では作成できないため、Local Zones 経由で共有することはできません。

詳細については、「Amazon VPC ユーザーガイド」の「他のアカウントと VPCを共有する」を参照してください。

Verified Access 用のロードバランサーエンドポイントを作成する

次の手順に従って、ロードバランサーエンドポイントを作成します。ロードバランサーの詳細については、「Elastic Load Balancing ユーザーガイド」を参照してください。

要件

- サポートされているのはIPv4 トラフィックのみです。
- サポートされているのは、HTTP プロトコルと HTTPS プロトコルのみです。
- ロードバランサーは、Application Load Balancer または Network Load Balancer のいずれかで、内部ロードバランサーである必要があります。
- ロードバランサーとサブネットは同じ仮想プライベートクラウド (VPC) に属している必要があります。
- HTTPS ロードバランサーは、自己署名 TLS 証明書またはパブリック TLS 証明書のどちらでも使用できます。
- アプリケーションのドメイン名を入力する必要があります。これは、ユーザーがアプリケーションにアクセスするために使用するパブリック DNS 名です。また、このドメイン名と一致する CN を含むパブリック SSL 証明書を入力する必要があります。AWS Certificate Manager を使用して証明書を作成またはインポートできます。

ロードバランサーエンドポイントを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access エンドポイント] を選択します。
3. [Verified Access エンドポイントの作成] を選択します。
4. (オプション) [名前タグ] と [説明] に、エンドポイントの名前と説明を入力します。
5. [Verified Access グループ] では、エンドポイントの Verified Access グループを選択します。
6. [アプリケーション詳細] では、次の操作を行います。

a. [アプリケーションドメイン] には、アプリケーションの DNS 名を入力します。

b. [ドメイン証明書 ARN] で、パブリック TLS 証明書を選択します。

7. [エンドポイント詳細] では、次の操作を行います。

a. [添付タイプ] で、[VPC] を選択します。

b. [セキュリティグループ] で、VPC エンドポイントのセキュリティグループを選択します。Verified Access エンドポイントからロードバランサーに入るトラフィックは、このセキュリティグループに関連付けられます。

c. [エンドポイントのドメインプレフィックス] には、Verified Access がエンドポイント用に生成する DNS 名の頭にカスタム識別子を入力します。

d. [エンドポイントタイプ] で、[ロードバランサー] を選択します。

e. [プロトコル] で、[HTTP] または [HTTPS] を選択します。

f. [ポート] に、ポート番号を入力します。

g. [ロードバランサー ARN] で、ロードバランサーを選択します。

h. [サブネット] で、ロードバランサーのサブネットを選択します。

8. (オプション) [ポリシー定義] には、エンドポイントの Verified Access ポリシーを入力します。

9. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。

10. [Verified Access エンドポイントの作成] を選択します。

Verified Access のネットワークインターフェイスのエンドポイントを作成する

次の手順に従って、 ネットワークインターフェイスエンドポイントを作成します。

要件

- サポートされているのはIPv4 トラフィックのみです。
- サポートされているのは、HTTP プロトコルと HTTPS プロトコルのみです。
- ネットワークインターフェイスは、セキュリティグループと同じ仮想プライベートクラウド (VPC) に属している必要があります。
- ネットワークインターフェースのプライベート IP を使用してトラフィックを転送します。

- アプリケーションのドメイン名を入力する必要があります。これは、ユーザーがアプリケーションにアクセスするために使用するパブリック DNS 名です。また、このドメイン名と一致する CN を含むパブリック SSL 証明書を入力する必要があります。AWS Certificate Manager を使用して証明書を作成またはインポートできます。

ネットワークインターフェイスエンドポイントを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access エンドポイント] を選択します。
3. [Verified Access エンドポイントの作成] を選択します。
4. (オプション) [名前タグ] と [説明] に、エンドポイントの名前と説明を入力します。
5. [Verified Access グループ] では、エンドポイントの Verified Access グループを選択します。
6. [アプリケーション詳細] では、次の操作を行います。
   a. [アプリケーションドメイン] には、アプリケーションの DNS 名を入力します。
   b. [ドメイン証明書 ARN] で、パブリック TLS 証明書を選択します。
7. [エンドポイント詳細] では、次の操作を行います。
   a. [添付タイプ] で、[VPC] を選択します。
   b. [セキュリティグループ] で、VPC エンドポイントのセキュリティグループを選択します。Verified Access エンドポイントからネットワークインターフェースに入るトラフィックは、このセキュリティグループに関連付けられます。
   c. [エンドポイントのドメインプレフィックス] には、Verified Access がエンドポイント用に生成する DNS 名の頭にカスタム識別子を入力します。
   d. [エンドポイントタイプ] で、[ネットワークインターフェイス] を選択します。
   e. [プロトコル] で、[HTTP] または [HTTPS] を選択します。
   f. [ポート] に、ポート番号を入力します。
   g. [ネットワークインターフェイス] でネットワークインターフェイスを選択します。
8. (オプション) [ポリシー定義] には、エンドポイントの Verified Access ポリシーを入力します。
9. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力します。
10. [Verified Access エンドポイントの作成] を選択します。

Verified Access エンドポイントから発信されるトラフィックを許可する

Verified Access エンドポイントから発信されるトラフィックを許可するように、アプリケーションのセキュリティグループを設定できます。そのためには、エンドポイントのセキュリティグループをソースとして指定するインバウンドルールを追加します。アプリケーションが Verified Access エンドポイントからのトラフィックのみを受信するように、その他のインバウンドルールを削除することをお勧めします。

既存のアウトバウンドルールを維持することをお勧めします。

アプリケーションのセキュリティグループルールを更新するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access エンドポイント] を選択します。
3. Verified Access エンドポイントを選択し、「詳細」タブでセキュリティグループ ID を探し、エンドポイントのセキュリティグループの ID をコピーします。
4. ナビゲーションペインで、[セキュリティグループ] を選択します。
5. ターゲットに関連付けられているセキュリティグループのチェックボックスを選択し、[アクション]、[インバウンドルールの編集] を選択します。
6. Verified Access エンドポイントから発信するトラフィックを許可するセキュリティグループルールを追加するには、次の操作を行います。
   a. [ルールの追加] を選択します。
   b. [タイプ] で、[すべてのトラフィック]、または許可する特定のトラフィックを選択します。
   c. [ソース] で、[カスタム] を選択し、エンドポイントのセキュリティグループの ID を貼り付けます。
7. (オプション) トラフィックが Verified Access エンドポイントからのみ発信するようにするには、他のインバウンドセキュリティグループルールをすべて削除します。
8. [Save Rules] (ルールの保存) を選択します。

Verified Access エンドポイントの変更

Verified Access エンドポイントを作成したら、その設定を更新できます。

Verified Access エンドポイントを変更するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access エンドポイント] を選択します。
3. エンドポイントを選択します。
4. [アクション]、[Verified Access エンドポイントの変更] を選択します。
5. 必要に応じてエンドポイントの詳細を変更します。
6. [Verified Access エンドポイントの変更] を選択します。

Verified Access エンドポイントポリシーの変更

Verified Access エンドポイントを作成した後、そのポリシーを変更できます。

Verified Access エンドポイントポリシーを変更するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access エンドポイント] を選択します。
3. ポリシーを変更するエンドポイントを選択します。
4. [アクション]、[Verified Access エンドポイントポリシーの変更] を選択します。
5. (オプション) 現在の目標に応じて [ポリシーを有効にする]をオンまたはオフにします。
6. (オプション)[ポリシー]には、エンドポイントに適用する Verified Access ポリシーを入力します。
7. [Verified Access エンドポイントポリシーの変更]を選択します。

Verified Access エンドポイントの削除

不要になった VPC Access エンドポイントは、削除することができます。

Verified Access エンドポイントを削除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access エンドポイント] を選択します。
3. エンドポイントを選択します。
4. [アクション]、[Verified Access エンドポイントの削除] を選択します。

5. 確認を求められたら、「**delete**」と入力し、[削除] を選択します。

信頼プロバイダーからのトラストデータ

トラストデータは、信頼プロバイダーからAWS Verified Access に送信されるデータです。「ユーザークレーム」や「トラストコンテキスト」と呼ばれることもあります。データには通常、ユーザーまたはデバイスに関する情報が含まれます。信頼データの例には、ユーザーの電子メール、グループメンバーシップ、デバイスのオペレーティングシステムのバージョン、デバイスのセキュリティ状態などがあります。送信される情報は信頼プロバイダーによって異なるため、トラストデータの完全かつ最新のリストについては、信頼プロバイダーのドキュメントを参照してください。

ただし、Verified Access のログ記録機能を使用すれば、信頼プロバイダーからどのようなトラストデータが送信されているかを確認することもできます。これは、アプリケーションへのアクセスを許可または拒否するポリシーを定義する際に非常に役立ちます。ログにトラストコンテキストを含める方法については、トラストコンテキストを含めるを参照してください。

このセクションでは、ポリシー作成を開始するためのトラストデータのサンプルと例を示します。ここに記載されている情報は、例示を目的とするもので、正式なレファレンスではありません。

目次

Verified Access デフォルトコンテキスト

AWSVerified Access には、設定されている信頼プロバイダーに関係なく、すべての Cedar 評価にデフォルトで現在の HTTP リクエストに関するいくつかの要素が含まれます。ポリシーが評価されると、Verified Access は現在の HTTP リクエストに関するデータを `context.http_request` key の下の Cedar コンテキストに含めます。必要に応じて、データに対して評価を行うポリシーを作成できます。次の JSON スキーマは、評価に含まれるデーを示しています。

```
{
     "title": "HTTP Request data included by Verified Access",
     "type": "object",
     "properties": {
         "user_agent": {
```

```
            "type": "string",
            "description": "The value of the User-Agent request header"
        },
        "x_forwarded_for": {
            "type": "string",
            "description": "The value of the X-Forwarded-For request header"
        },
        "http_method": {
            "type": "string",
            "description": "The HTTP Method provided (e.g. GET or POST)"
        },
        "hostname": {
            "type": "string",
            "description": "The value of the Host request header"
        },
        "port": {
           "type": "integer",
           "description": "The value of the verified access endpoint port"
        },
        "client_ip": {
            "type": "string",
            "description": "User ip connecting to the verified access endpoint"
        }
    }
}
```

以下は、HTTP リクエストデータに対して評価を行うポリシーの例です。

```
forbid(principal, action, resource) when {
   context.http_request.http_method == "POST"
   && !(context.identity.roles.contains("Administrator"))
};
```

AWS IAM アイデンティティセンター

ポリシーが評価される際に、AWS IAM Identity Centerを信頼プロバイダーとして定義すると、AWS Verified Access は、信頼プロバイダー設定で「ポリシーレファレンス名」として指定するキーの下の Cedar コンテキスト内のトラストデータを含めます。必要に応じて、トラストデータに対して評価するポリシーを作成できます。

> **Note**
>
> 信頼プロバイダーのコンテキストキーは、信頼プロバイダーの作成時に設定したポリシーレファレンス名から取得されます。たとえば、ポリシーレファレンスを「idp123」と設定した場合、コンテキストキーは「context.idp123」となります。ポリシーを作成する際は、正しいコンテキストキーを使用していることを確認してください。

次の JSON スキーマは、評価に含まれるデーを示しています。

```
{
   "title": "AWS IAM Identity Center context specification",
   "type": "object",
   "properties": {
     "user": {
       "type": "object",
       "properties": {
         "user_id": {
           "type": "string",
           "description": "a unique user id generated by AWS IdC"
         },
         "user_name": {
           "type": "string",
           "description": "username provided in the directory"
         },
         "email": {
           "type": "object",
           "properties": {
             "address": {
               "type": "email",
               "description": "email address associated with the user"
             },
             "verified": {
               "type": "boolean",
               "description": "whether the email address has been verified by AWS IdC"
             }
           }
         }
       }
     },
     "groups": {
       "type": "object",
```

```
       "description": "A list of groups the user is a member of",
       "patternProperties": {
         "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]
{12}$": {
           "type": "object",
           "description": "The Group ID of the group",
           "properties": {
             "group_name": {
               "type": "string",
               "description": "The customer-provided name of the group"
             }
           }
         }
       }
     }
   }
 }
```

以下は、AWS IAM アイデンティティセンター が提供するトラストデータに対して評価を行うポリシーの例です。

```
permit(principal, action, resource) when {
   context.idc.user.email.verified == true
   // User is in the "sales" group with specific ID
   && context.idc.groups has "c242c5b0-6081-1845-6fa8-6e0d9513c107"
 };
```

> **Note**
>
> グループ名は変更できるため、IAM アイデンティティセンターはグループ ID を使用してグループを参照します。これにより、グループの名前を変更する際にポリシーステートメントが破られるのを防ぐことができます。

サードパーティの信頼プロバイダー

このセクションでは、サードパーティーの信頼プロバイダーがAWS Verified Access に提供するトラストデータについて説明します。

> **Note**
>
> 信頼プロバイダーのコンテキストキーは、信頼プロバイダーの作成時に設定したポリシーレファレンス名から取得されます。たとえば、ポリシーレファレンスを「idp123」と設定した場合、コンテキストキーは「context.idp123」となります。ポリシーを作成する際は、正しいコンテキストキーを使用していることを確認してください。

目次

ブラウザ拡張

デバイスのトラストコンテキストをアクセスポリシーに組み込む予定の場合は、AWS Verified Access ブラウザ拡張機能または別のパートナーのブラウザ拡張機能が必要になります。Verified Access は、現在 Google Chrome と Mozilla Firefox ブラウザをサポートしています。

現在、Jamf (macOS デバイスをサポート)、CrowdStrike (Windows 11 デバイスと Windows 10 デバイスをサポート)、および JumpCloud (Windows と MacOS の両方をサポート) の 3 つのデバイス信頼プロバイダーをサポートしています。

- ポリシーで Jamf トラストデータを使用している場合、ユーザーは Chrome ウェブストア または Firefox アドオンサイトからAWS Verified Access ブラウザ拡張機能をダウンロードしてデバイスにインストールする必要があります。
- ポリシーで CrowdStrike トラストデータを使用している場合は、まずユーザーは AWSVerified Access Native Messaging Host (ダイレクトダウンロードリンク) をインストールする必要があります。このコンポーネントは、ユーザーのデバイスで実行されている CrowdStrike エージェントからトラストデータを取得するために必要です。次に、このコンポーネントをインストールした後、ユーザーは Chrome ウェブストア または Firefox アドオンサイトAWS から Verified Access ブラウザ拡張機能をデバイスにインストールする必要があります。
- JumpCloud を使用している場合、ユーザーのデバイスには Chrome ウェブストアまたは Firefox アドオンサイトの JumpCloud ブラウザ拡張機能がインストールされている必要があります。

Jamf

Jamf はサードパーティー信頼プロバイダーです。ポリシーが評価される際に、Jamf を信頼プロバイダーとして定義すると、Verified Access は、信頼プロバイダー設定で「ポリシーレファレンス名」として指定するキーの下の Cedar コンテキスト内のトラストデータを含めます。必要に応じて、トラストデータに対して評価するポリシーを作成できます。次の JSON スキーマは、評価に含まれるデーを示しています。

AWS Veriried Access で Jamf を使用する方法の詳細については、Jamf ウェブサイトの「AWS Verified Access と Jamf デバイス ID の統合」を参照してください。

```
{
    "title": "Jamf device data specification",
    "type": "object",
    "properties": {
        "iss": {
            "type": "string",
            "description": "\"Issuer\" - the Jamf customer ID"
        },
        "iat": {
            "type": "integer",
            "description": "\"Issued at Time\" - a unixtime (seconds since epoch) value
 of when the device information data was generated"
        },
        "exp": {
            "type": "integer",
            "description": "\"Expiration\" - a unixtime (seconds since epoch) value for
 when this device information is no longer valid"
        },
        "sub": {
            "type": "string",
            "description": "\"Subject\" - either the hardware UID or a value generated
 based on device location"
        },
        "groups": {
            "type": "array",
            "description": "Group IDs from UEM connector sync",
            "items": {
                "type": "string"
            }
        },
        "risk": {
```

```
            "type": "string",
            "enum": [
                "HIGH",
                "MEDIUM",
                "LOW",
                "SECURE",
                "NOT_APPLICABLE"
            ],
            "description": "a Jamf-reported level of risk associated with the device."
        },
        "osv": {
            "type": "string",
            "description": "The version of the OS that is currently running, in Apple
 version number format (https://support.apple.com/en-us/HT201260)"
        }
    }
}
```

以下は、Jamf が提供するトラストデータに対して評価を行うポリシーの例です。

```
permit(principal, action, resource) when {
   context.jamf.risk == "LOW"
};
```

Cedar には、Jamf のリスクスコアなどの列挙型を使用する際に役立つ便利な `.contains()` 機能があります。

```
permit(principal, action, resource) when {
   ["LOW", "SECURE"].contains(context.jamf.risk)
};
```

CrowdStrike

CrowdStrike はサードパーティー信頼プロバイダーです。ポリシーが評価される際に、CrowdStrike を信頼プロバイダーとして定義すると、Verified Access は、信頼プロバイダー設定で「ポリシーレファレンス名」として指定するキーの下の Cedar コンテキスト内のトラストデータを含めます。必要に応じて、トラストデータに対して評価するポリシーを作成できます。次の JSON スキーマは、評価に含まれるデーを示しています。

AWS Verified Access で CrowdStrike を使用する方法の詳細については、GitHub ウェブサイトの「CrowdStrike と AWS Verified Access によるプライベートアプリケーションの保護」を参照してください。

```
{
  "title": "CrowdStrike device data specification",
  "type": "object",
  "properties": {
    "assessment": {
      "type": "object",
      "description": "Data about CrowdStrike's assessment of the device",
      "properties": {
        "overall": {
          "type": "integer",
          "description": "A single metric, between 1-100, that accounts as a weighted
 average of the OS and and Sensor Config scores"
        },
        "os": {
          "type": "integer",
          "description": "A single metric, between 1-100, that accounts for the OS-
specific settings monitored on the host"
        },
        "sensor_config": {
          "type": "integer",
          "description": "A single metric, between 1-100, that accounts for the
 different sensor policies monitored on the host"
        },
        "version": {
          "type": "string",
          "description": "The version of the scoring algorithm being used"
        }
      }
    },
    "cid": {
      "type": "string",
      "description": "Customer ID (CID) unique to the customer's environemnt"
    },
    "exp": {
      "type": "integer",
      "description": "unixtime, The expiration time of the token"
    },
    "iat": {
      "type": "integer",
```

```
      "description": "unixtime, The issued time of the token"
    },
    "jwk_url": {
      "type": "string",
      "description": "URL that details the JWT signing"
    },
    "platform": {
      "type": "string",
      "enum": ["Windows 10", "Windows 11", "macOS"],
      "description": "Operating system of the endpoint"
    },
    "serial_number": {
      "type": "string",
      "description": "The serial number of the device derived by unique system
 information"
    },
    "sub": {
      "type": "string",
      "description": "Unique CrowdStrike Agent ID (AID) of machine"
    },
    "typ": {
      "type": "string",
      "enum": ["crowdstrike-zta+jwt"],
      "description": "Generic name for this JWT media. Client MUST reject any other
 type"
    }
  }
}
```

以下は、CrowdStrike が提供するトラストデータに対して評価を行うポリシーの例です。

```
permit(principal, action, resource) when {
   context.crowdstrike.assessment.overall > 50
};
```

JumpCloud

JumpCloud はサードパーティー信頼プロバイダーです。ポリシーが評価される際に、JumpCloud を信頼プロバイダーとして定義すると、 Verified Access は、信頼プロバイダー設定で「ポリシーレファレンス名」として指定するキーの下の Cedar コンテキスト内のトラストデータを含めます。必要に応じて、トラストデータに対して評価するポリシーを作成できます。次の JSON スキーマは、評価に含まれるデーを示しています。

AWS Verified Access で JumpCloud を使用する方法について詳しくは、JumpCloud ウェブサイトの「JumpCloud ウェブサイトで JumpCloud と AWS Verified Access を統合する」を参照してください。

```
{
  "title": "JumpCloud device data specification",
  "type": "object",
  "properties": {
    "device": {
      "type": "object",
      "description": "Properties of the device",
      "properties": {
        "is_managed": {
          "type": "boolean",
          "description": "Boolean to indicate if the device is under management"
        }
      }
    },
    "exp": {
      "type": "integer",
      "description": "Expiration. Unixtime of the token's expiration."
    },
    "durt_id": {
      "type": "string",
      "description": "Device User Refresh Token ID. Unique ID that represents the
 device + user."
    },
    "iat": {
      "type": "integer",
      "description": "Issued At. Unixtime of the token's issuance."
    },
    "iss": {
      "type": "string",
      "description": "Issuer. This will be 'go.jumpcloud.com'"
    },
    "org_id": {
      "type": "string",
      "description": "The JumpCloud Organization ID"
    },
    "sub": {
      "type": "string",
      "description": "Subject. The managed JumpCloud user ID on the device."
    },
```

```
    "system": {
      "type": "string",
      "description": "The JumpCloud system ID"
    }
  }
}
```

以下は、JumpCloud が提供するトラストコンテキストに対して評価を行うポリシーの例です。

```
permit(principal, action, resource) when {
   context.jumpcloud.org_id = 'Unique_orgnaization_identifier'
};
```

ユーザークレームの引き渡しと署名の検証

AWSVerified Access インスタンスがユーザーの認証に成功すると、IdP から受け取ったユーザークレームが Verified Access エンドポイントに送信されます。ユーザークレームには署名が付けられているため、アプリケーションは署名の検証および、そのクレームが Verified Access から送信されたものであることの検証を行うことができます。この処理中に、以下の HTTP ヘッダーが追加されます。

`x-amzn-ava-user-context`

このヘッダーには、JSON Web Token (JWT) 形式のユーザークレームが含まれます。JWT 形式にはヘッダー、ペイロード、および Base64 URL でエンコードされた署名が含まれています。Verified Access は ES384 (SHA-384 ハッシュアルゴリズムを使用する ECDSA 署名アルゴリズム) を使用して JWT 署名を生成します。

アプリケーションはこれらのクレームをパーソナライズやその他のユーザー固有のエクスペリエンスに使用できます。アプリケーションデベロッパーは、使用前に ID プロバイダーから提供される各クレームの一意性と検証のレベルについて十分に理解しておく必要があります。一般に、sub クレームは、特定のユーザーを識別する最良の方法です。

目次

例：OIDC ユーザークレーム用の署名付き JWT

以下の例は、OIDC ユーザークレームのヘッダーとペイロードが JWT 形式でどのように表示されるかを示しています。

ヘッダーの例：

```
{
"alg": "ES384",
"kid": "12345678-1234-1234-1234-123456789012",
"signer": "arn:aws:ec2:us-east-1:123456789012:verified-access-instance/vai-
abc123xzy321a2b3c",
"iss": "OIDC Issuer URL"
"exp": "expiration" (120 secs)
}
```

ペイロードの例：

```
{
"sub": "xyzsubject",
"email": "xxx@amazon.com",
"email_verified": true,
"groups": [
"Engineering",
"finance"
]
}
```

例：IAM アイデンティティセンターのユーザークレーム用署名付き JWT

以下の例は、IAM アイデンティティセンターユーザークレームのヘッダーとペイロードが JWT 形式でどのように表示されるかを示しています。

> **Note**
>
> IAM アイデンティティセンターについては、ユーザー情報のみがクレームに含まれます。

ヘッダーの例：

```
{
```

```
"alg": "ES384",
"kid": "12345678-1234-1234-1234-123456789012",
"signer": "arn:aws:ec2:us-east-1:123456789012:verified-access-instance/vai-
abc123xzy321a2b3c",
"iss": "arn:aws:ec2:us-east-1:123456789012:verified-access-trust-provider/vatp-
abc123xzy321a2b3c",
"exp": "expiration" (120 secs)
}
```

ペイロードの例：

```
{
"user": {
        "user_id": "f478d4c8-a001-7064-6ea6-12423523",
        "user_name": "test-123",
        "email": {
            "address": "test@amazon.com",
            "verified": false
        }
    }
}
```

パブリックキー

Verified Access インスタンスではユーザークレームが暗号化されないため、Verified Access エンドポイントが HTTPS を使用するように設定することをお勧めします。Verified Access エンドポイントが HTTP を使用するように設定する場合は、エンドポイントへのトラフィックをセキュリティグループを使用するトラフィックのみに制限してください。

また、クレームに基づいて認証を行う前に、署名を検証することをお勧めします。パブリックキーを取得するには、JWT ヘッダーからキー ID を取得し、それを使用して次のエンドポイントからパブリックキーを検索します。それぞれの AWS リージョン の場合、エンドポイントは次のとおりです：

`https://public-keys.prod.verified-access.<region>.amazonaws.com/<key-id>`

例：JWT の取得とデコード

次のコードは、Python 3.9 でキー ID、公開キー、およびペイロードを取得する方法を示しています。

```
import jwt
import requests
import base64
import json

# Step 1: Get the key id from JWT headers (the kid field)
encoded_jwt = headers.dict['x-amzn-ava-user-context']
jwt_headers = encoded_jwt.split('.')[0]
decoded_jwt_headers = base64.b64decode(jwt_headers)
decoded_jwt_headers = decoded_jwt_headers.decode("utf-8")
decoded_json = json.loads(decoded_jwt_headers)
kid = decoded_json['kid']

# Step 2: Get the public key from Regional endpoint
url = 'https://public-keys.prod.verified-access.' + region + '.amazonaws.com/' + kid
req = requests.get(url)
pub_key = req.text

# Step 3: Get the payload
payload = jwt.decode(encoded_jwt, pub_key, algorithms=['ES384'])
```

Verified Access ポリシー

AWS Verified Access ポリシーにより、AWS にホストされているアプリケーションにアクセスするためのルールを定義できます。AWS ポリシー言語である Cedar で記述されています。Cedar を使用すると、Verified Access で使用するように設定する ID またはデバイスベースの信頼プロバイダーから送信されたトラストコンテキストに基づいて評価されるポリシーを作成できます。

Cedar ポリシー言語の詳細については、「Cedar リファレンスガイド」をご覧ください。

このセクションでは、Verified Access ポリシーの構造、内容、定義方法について説明し、例をいくつか示します。

目次

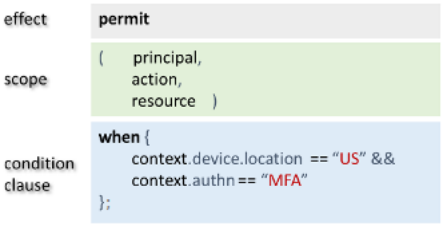
Verified Access ポリシーの使用

Verified Access グループを作成する時、または Verified Access エンドポイントを作成する時、オプションとして Verified Access ポリシーを定義できます。Verified Access ポリシーを定義しなくてもグループまたはエンドポイントを作成できますが、ポリシーを定義するまですべてのアクセス要求はブロックされます。

作成後に既存の Verified Access グループまたはエンドポイントにポリシーを追加または変更する場合は、Verified Access グループポリシーの変更 または Verified Access エンドポイントポリシーの変更 をご覧ください。

ポリシーステートメントの構造

このセクションでは、AWS Verified Access ポリシーステー トメントとその評価方法について説明します。ひとつの Verified Access ポリシーに複数のステートメントを設定できます。Verified Access ポリシーの構造は、以下の図の通りです。



ポリシーには、次の部分が含まれます。

- 効果 — ポリシーステートメントが `permit` (`Allow`) か `forbid` (`Deny`) かを指定します。
- 対象範囲 — 効果を適用するプリンシパル、アクション、リソースを指定します。特定のプリンシパル、アクション、またはリソースを特定しないことで、Cedar の対象範囲を未定義のままにしておくことができます (前の例を参照)。この場合、ポリシーはすべてのプリンシパル、アクション、リソースに適用されます。
- 条件節 — 効果が適用されるコンテキストを指定します。

> **Important**
>
> Verified Access では、条件節に含まれるトラストコンテキストを参照することでポリシーが完全に表現されます。ポリシーの対象範囲は、常に未定義のままにしておく必要があります。その後、条件節に含まれる ID とデバイスのトラストコンテキストを使用してアクセスを指定できます。

簡単なポリシーの例

```
permit(principal,action,resource)
when{
 context.<policy-reference-name>.<attribute> &&
 context.<policy-reference-name>.<attribute2>
};
```

前の例では、&&演算子を使用して 1 つのポリシーステートメントに複数の条件節を使用できることにご注意ください。Cedar のポリシー言語を使うと、カスタムできめ細かな広範囲にわたるポリシーステートメントを作成する表現力が得られます。その他の例については、「ポリシーの例」を参照してください。

ポリシーの評価

ポリシードキュメントは 1 つ以上のポリシーステートメント (`permit` または `forbid` ステートメント) のセットです。ポリシーは、条件節 (when ステートメント) が true である場合に適用されます。ポリシードキュメントがアクセスを許可するには、ドキュメント内の少なくとも 1 つの許可ポリシーが適用されている必要があり、禁止ポリシーを適用することはできません。許可ポリシーが適用されない場合や、1 つ以上の禁止ポリシーが適用されている場合、ポリシードキュメントはアクセスを拒否します。Verified Access グループと Verified Access エンドポイントの両方に定義されたポリシードキュメントがある場合、両方のドキュメントがアクセスを許可する必要があります。Verified Access エンドポイントのポリシードキュメントを定義していない場合は、アクセスを許可する必要があるのは Verified Access グループポリシーのみです。

> **Note**
>
> AWS ポリシー作成時に Verified Access は構文を検証しますが、条件節に入力したデータは検証しません。

ビルトイン演算子

ポリシーステートメントの構造 で説明したように、さまざまな条件を使用して AWS Verified Access ポリシーのコンテキストを作成する場合、&& 演算子を使用して追加条件を加えることができます。ポリシー条件にさらに表現力を加えるために使用できるビルトイン演算子は他にも多数あります。参照用にすべてのビルトイン演算子を下表に示します。

演算子	タイプとオーバーロード	説明
!	Boolean → Boolean	論理否定。
==	any → any	等価。タイプが一致しなくても、いずれかのタイプの引数で機能します。異なるタイプ

演算子	タイプとオーバーロード	説明
		の値が互いに等しくなることはありません。
!=	any → any	不等価。等価の正反対 (上記参照)。
<	(long, long) → Boolean	Long integer less-than.
<=	(long, long) → Boolean	Long integer less-than-or-equal-to.
>	(long, long) → Boolean	Long integer greater-than.
>=	(long, long) → Boolean	Long integer greater-than-or-equal-to.
in	(entity, entity) → Boolean	階層メンバーシップ (再帰形：A の A は常にトゥルー)。
	(entity, set(entity)) → Boolean	階層メンバーシップ：(A and B) || (A in C) || であれば A in [B, C, ...] はトゥルー... セットに non-entity が含まれている場合はエラーになります。
&&	(Boolean, Boolean) → Boolean	Logical and (short-circuiting).
||	(Boolean, Boolean) → Boolean	Logical or (short-circuiting).
.exists()	entity → Boolean	Entity existence.

演算子	タイプとオーバーロード	説明
has	(entity, attribute) → Boolean	中置演算子。`e has f` レコードまたはエンティティに e 属性 f へのバインディングがあるかどうかをテストします。e が存在しないか、e が存在しても属性 f を持たない場合は `false` を返します。属性は識別子または文字列リテラルとして表現できます。
like	(string, string) → Boolean	中置演算子。`t like p` テキスト t がパターン p と一致するかどうかを確認します。パターンには、0 個以上の文字と一致するワイルドカード文字 * が含まれる場合があります。t のリテラルスター文字と一致させるには、p で特殊なエスケープ文字シーケンス * を使用できます。
.contains()	(set, any) → Boolean	メンバーシップ (B が A の要素かどうか) を設定します。
.containsAll()	(set, set) → Boolean	セット A にセット B のすべての要素が含まれているかどうかをテストします。
.containsAny()	(set, set) → Boolean	セット A にセット B の要素のいずれかが含まれているかどうかをテストします。

ポリシーコメント

AWS Verified Access ポリシーにコメントステートメントを含めることができます。コメントは、// で始まり改行で終わる行として定義されます。

次の例は、ポリシー内のコメントステートメントを示しています。

```
// this policy grants access to users in a given domain with trusted devices
permit(principal, action, resource)
when {
  // the user's email address is in the @example.com domain
  context.idc.user.email.address.contains("@example.com")
  // Jamf thinks the user's computer is low risk or secure.
  && ["LOW", "SECURE"].contains(context.jamf.risk)
};
```

ポリシーロジックのショートサーキット

特定のコンテキストに存在する、または存在しないデータを評価する AWS Verified Access ポリシーを作成する場合があります。存在しないコンテキスト内のデータを参照すると、意図に関係なく、Cedar はエラーを作成し、ポリシーでアクセスが拒否されるよう評価します。例えば、fake_provider と bogus_key はこのコンテキストでは存在しないため、結果的に拒否されます。

```
permit(principal, action, resource) when {
  context.fake_provider.bogus_key > 42
};
```

このような状況を回避するには、has 演算子を使用してキーが存在するかどうかを確認します。has 演算子が false を返すと、連鎖ステートメントのさらなる評価は停止し、Cedar は存在しない項目の参照を試みることによるエラーを生成しません。

```
permit(principal, action, resource) when {
  context.identity.user has "some_key" && context.identity.user.some_key > 42
};
```

これは、2 つの異なる信頼プロバイダーを参照するポリシーを指定する場合に最も有用です。

```
permit(principal, action, resource) when {
```

```
  // user is in an allowed group
  context.aws_idc.groups has "c242c5b0-6081-1845-6fa8-6e0d9513c107"
  &&(
    (
      // if CrowdStrike data is present,
      // permit if CrowdStrike's overall assessment is over 50
      context has "crowdstrike" && context.crowdstrike.assessment.overall > 50
    )
    ||
    (
      // if Jamf data is present,
      // permit if Jamf's risk score is acceptable
      context has "jamf" && ["LOW", "NOT_APPLICABLE", "MEDIUM",
 "SECURE"].contains(context.jamf.risk)
    )
  )
};
```

ポリシーの例

例 1：IAM アイデンティティセンターのポリシーの作成

> **Note**
>
> グループ名は変更できるため、IAM アイデンティティセンターはグループ ID を使用してグループを参照します。これにより、グループの名前を変更する際にポリシーステートメントが破られるのを防ぐことができます。

以下のポリシー例では、ユーザーが finance グループ (グループ ID は c242c5b0-6081-1845-6fa8-6e0d9513c107) に属し、検証済みメールアドレスを持っている場合にのみアクセスを許可します。

```
permit(principal,action,resource)
when {
    context.<policy-reference-name>.groups has "c242c5b0-6081-1845-6fa8-6e0d9513c107"
    && context.<policy-reference-name>.user.email.verified == true
};
```

例 1b：IAM アイデンティティセンターのポリシーステートメントにさらに条件を追加する

以下のポリシー例では、ユーザーが finance グループ (グループ ID は c242c5b0-6081-1845-6fa8-6e0d9513c107) に属し、検証済みメールアドレスを持っていて、Jamf デバイスリスクスコアが LOW の場合にのみアクセスを許可します。

```
permit(principal,action,resource)
when {
    context.<policy-reference-name>.groups has "c242c5b0-6081-1845-6fa8-6e0d9513c107"
    && context.<policy-reference-name>.user.email.verified == true
    && context.jamf.risk == "LOW"
};
```

例 2：サードパーティ OIDC プロバイダーを対象としたポリシーと同じポリシー

以下のポリシー例では、ユーザーが「finance」グループに属し、検証済みメールアドレスを持っていて、Jamf デバイスのリスクスコアが LOW の場合にのみアクセスを許可します。

```
permit(principal,action,resource)
when {
     context.<policy-reference-name>.groups.contains("finance")
     && context.<policy-reference-name>.email_verified == true
     && context.jamf.risk == "LOW"
};
```

例 3: CrowdStrike の使用

次のポリシー例では、全体評価のスコアが 50 を超えるとアクセスが許可されます。

```
permit(principal,action,resource)
when {
    context.crwd.assessment.overall > 50
};
```

例 4：特殊文字の使用

次の例では、コンテキストプロパティがポリシー言語の予約文字である：(セミコロン) を使用している場合のポリシーの記述方法を示しています。

```
permit(principal, action, resource)
when {
    context.<policy-reference-name>["namespace:groups"].contains("finance")
```

```
};
```

例 5：特定の IP アドレスの許可

以下は、特定の IP アドレスのみを許可するポリシーの例です。

```
permit(principal, action, resource)
when {
    context.http_request.client_ip == "192.0.2.1"
};
```

例 5a：特定の IP アドレスのブロック

以下は、特定の IP アドレスをブロックするポリシーの例です。

```
forbid(principal,action,resource)
when {
ip(context.http_request.client_ip).isInRange(ip("192.0.2.1/32"))
};
```

Verified Access ポリシーアシスタント

Verified Access ポリシーアシスタントは、ポリシーのテストと開発に使用できる Verified Access コンソールに含まれるツールです。エンドポイントポリシー、グループポリシー、およびトラストコンテキストが 1 つの画面に表示され、そこでポリシーをテストしたり編集したりできます。

トラストコンテキストの形式は信頼プロバイダーごとに異なり、Verified Access 管理者は特定の信頼プロバイダーの使用する正確な形式を知らない場合があります。そのため、テストや開発の目的で、信頼コンテキストとグループポリシーとエンドポイントポリシーの両方を 1 か所で確認できると非常に便利です。

以下のセクションでは、ポリシーエディターを使用するうえでの基本事項を説明します。

タスク

- ステップ 1: リソースを指定する
- ステップ 2: ポリシーをテストおよび編集する
- ステップ 3: 変更を確認して適用する

ステップ 1: リソースを指定する

ポリシーアシスタントの最初のページで、使用する Verified Access エンドポイントを指定します。また、ユーザー (E メールアドレスで識別) を指定し、オプションでユーザーの名前および/またはデバイス ID を指定します。デフォルトでは、指定したユーザーの Verified Access ログから最新の認証決定が抽出されます。オプションで、最新の許可または拒否の決定を具体的に選択できます。

最後に、トラストコンテキスト、認証決定、エンドポイントポリシー、およびグループポリシーがすべて次の画面に表示されます。

ポリシーアシスタントを開いてリソースを指定するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [Verified Access インスタンス] を選択し、操作するインスタンスの [Verified Access インスタンス ID] をクリックします。
3. [ポリシーアシスタントを起動] を選択します。
4. [ユーザーの E メールアドレス] で、ユーザーのメールアドレスを入力します。
5. [Verified Access エンドポイント] では、ポリシーを編集してテストするエンドポイントを選択します。
6. (オプション) [名前] には、ユーザーの名前を入力します。
7. (オプション) [デバイス識別子] に、一意のデバイス識別子を指定します。
8. (オプション)[認証結果] では、使用する最新の認証結果の種類を選択します。デフォルトでは、最新の認証結果が使用されます。
9. [次へ] を選択します。

ステップ 2: ポリシーをテストおよび編集する

このページには、次で作業するための情報が表示されます。

- 信頼プロバイダーがユーザーと (オプションで) 前のステップで指定したデバイスのために送信したトラストコンテキスト。
- 前のステップで指定された Verified Access エンドポイントの Cedar ポリシー。
- エンドポイントが属する Verified Access グループの Cedar ポリシー。

Verified Access エンドポイントとグループの Cedar ポリシーはこのページで編集できますが、トラストコンテキストは静的です。このページを使用して、Cedar ポリシーと一緒にトラストコンテキストを表示できるようになりました。

[ポリシーをテスト] ボタンを選択して、トラストコンテキストに対し、ポリシーをテストすると、認証結果が画面に表示されます。必要に応じてこのプロセスを繰り返すことで、ポリシーを編集して変更を再テストできます。

ポリシーの変更に問題がなければ、[次へ] を選択してポリシーアシスタントの次の画面に進みます。

ステップ 3: 変更を確認して適用する

ポリシーアシスタントの最後のページでは、ポリシーに加えた変更が強調表示され、簡単に確認できます。これで、変更内容を最後に確認し、[変更を適用]] を選択して変更を確定できます。

また、[前へ] を選択して前のページに戻るか、[キャンセル] を選択してポリシーアシスタントを完全にキャンセルすることもできます。

AWS Verfied Access のセキュリティ

AWS では、クラウドセキュリティを最優先事項としています。AWS のユーザーは、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャを利用できます。

セキュリティは、AWS とユーザーの間の責任共有です。責任共有モデル では、これをクラウド のセキュリティおよびクラウド 内の セキュリティとして説明しています。

- クラウドのセキュリティ - AWS は、AWS クラウド で AWS のサービスを実行するインフラストラクチャを保護する責任を担います。また、AWS は、ユーザーが安全に使用できるサービスも提供します。AWS コンプライアンスプログラム の一環として、サードパーティーの監査が定期的にセキュリティの有効性をテストおよび検証しています。AWS Verified Access に適用するコンプライアンスプログラムの詳細については、コンプライアンスプログラムによる対象範囲内の AWS のサービスをご参照ください。
- クラウド内のセキュリティ - ユーザーの責任は、使用する AWS のサービスに応じて異なります。またお客様は、データの機密性、企業要件、適用法令と規制などのその他の要因に対しても責任を担います。

このドキュメントは、Verified Access を使用する際に責任共有モデルを適用する方法を理解するのに役立ちます。以下のトピックでは、セキュリティおよびコンプライアンスの目的を達成するために Verified Access を設定する方法を示します。Verified Access リソースのモニタリングやセキュリティ確保に役立つ他の AWS のサービスの使用方法についても学習します。

目次

AWS Verified Access でのデータ保護

AWS 責任共有モデルは、AWS Verified Access におけるデータ保護に適用されます。このモデルで説明されているように、AWS は、AWS クラウド のすべてを実行するグローバルインフラストラクチャを保護するがあります。お客様は、このインフラストラクチャでホストされているコンテンツ

に対する管理を維持する責任があります。また、使用する AWS のサービスのセキュリティ設定と管理タスクもユーザーの責任となります。データプライバシーの詳細については、「データプライバシーのよくある質問」を参照してください。欧州でのデータ保護の詳細については、「AWS セキュリティブログ」に投稿された「AWS 責任共有モデルおよび GDPR」のブログ記事を参照してください。

データを保護するため、AWS アカウント の認証情報を保護し、AWS IAM Identity Center または AWS Identity and Access Management (IAM) を使用して個々のユーザーをセットアップすることをお勧めします。この方法により、それぞれのジョブを遂行するために必要な権限のみを各ユーザーに付与できます。また、次の方法でデータを保護することをおすすめします。

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 が必須です。TLS 1.3 が推奨されます。
- AWS CloudTrail で API とユーザーアクティビティロギングをセットアップします。
- AWS のサービス内でデフォルトである、すべてのセキュリティ管理に加え、AWS の暗号化ソリューションを使用します。
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- コマンドラインインターフェイスまたは API により AWS にアクセスするときに FIPS 140−2 検証済み暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。利用可能な FIPS エンドポイントの詳細については、「連邦情報処理規格 (FIPS) 140-2」を参照してください。

お客様の E メールアドレスなどの機密情報やセンシティブ情報は、タグや [名前] フィールドなどの自由形式のフィールドに配置しないことを強くお勧めします。これは、コンソール、API、AWS CLI、または AWS SDK を使用して Veriied Access や他の AWS のサービスを操作する場合も同様です。名前に使用する自由記述のテキストフィールドやタグに入力したデータは、課金や診断ログに使用される場合があります。外部サーバーへ URL を提供する場合は、そのサーバーへのリクエストを有効にするために認証情報を URL に含めないことを強くお勧めします。

転送中の暗号化

Verified Access では、Transport Layer Security (TLS) 1.2 以降を使用してインターネット経由でエンドユーザーから Verified Access エンドポイントに転送中のデータをすべて暗号化します。

ネットワーク間トラフィックのプライバシー

VPC 内の特定のリソースへのアクセスを制限するように Verified Access を設定できます。ユーザーベースの認証の場合、 エンドポイントにアクセスするユーザーグループに基づいて、ネットワークの一部へのアクセスを制限することもできます。詳細については、「Verified Access ポリシー」を参照してください。

AWS Verified Access の保管時のデータ暗号化

AWS Verified Access は、デフォルトで、AWS が所有する KMS キーを使用して保管中のデータを暗号化します。デフォルトで保管中のデータを暗号化すると、機密データの保護に伴う運用上のオーバーヘッドや複雑さを軽減できます。同時に、暗号化のコンプライアンスと規制の厳格な要件を満たす、安全なアプリケーションを構築することもできます。以下のセクションでは、Verified Access が保管中のデータ暗号化に KMS キーを使用する方法について詳しく説明します。

コンテンツ

Verified Access と KMS キー

AWS 所有キー

Verified Access では、KMS キーを使用して、個人を特定できる情報 (PII) を自動的に暗号化します。これはデフォルトで発生し、AWS が所有するキーの使用を自分で表示、管理、使用、または監査することはできません。ただし、データを暗号化するキーを保護するための行動やプログラムを操作したり変更したりする必要はありません。詳細については、「AWS Key Management Service デベロッパーガイド」の「AWS所有キー」を参照してください。

この暗号化レイヤーを無効にしたり、別の暗号化タイプを選択したりすることはできませんが、Verified Access リソースを作成する際にカスタマーマネージドキーを選択することで、既存の AWS 所有暗号化キーに 2 番目の暗号化レイヤーを追加できます。

カスタマーマネージドキー

Verified Access では、お客様が作成して管理する対称カスタマーマネージドキーを使用して、既存のデフォルトの暗号化に 2 番目の暗号化レイヤーを追加することができます。この暗号化レイヤーはユーザーが完全に制御できるため、次のようなタスクを実行できます。

- キーポリシーの策定と維持
- IAM ポリシーとグラントの策定と維持
- キーポリシーの有効化と無効化
- 暗号化素材のローテーション
- タグの追加
- キーエイリアスの作成
- キー削除のスケジュール設定

詳細については、「AWS Key Management Service デベロッパーガイド」の「カスタマーマネージドキー」を参照してください。

> **Note**
>
> Verified Access では、個人を特定できるデータを無料で保護するために、AWS所有キーを使用して保管中の暗号化を自動的に有効にします。
> ただし、カスタマーマネージドキーの使用には AWS KMS 料金が適用されます。料金の詳細については、「AWS Key Management Service の料金」を参照してください。

個人を特定できる情報

次の表では、Verified Access が使用する個人を特定できる情報 (PII) と、その暗号化方法をまとめます。

データタイプ	AWS 所有のキー暗号化	カスタマーマネージドキーの暗号化 (オプション)
`Trust provider (user-type)`	有効	有効

データタイプ	AWS 所有のキー暗号化	カスタマーマネージドキーの暗号化 (オプション)
ユーザータイプの信頼プロバイダーには、PII と見なされる AuthorizationEndpoint、UserInfoEndpoint ClientId、ClientSecretなどの OIDC オプションが含まれています。		
`Trust provider (device-type)` デバイスタイプの信頼プロバイダーには TenantId、PII と見なされる が含まれています。	有効	有効
`Group policy` Verified Access グループの作成または変更時に提供されます。アクセス要求を承認するためのルールが含まれています。ユーザー名やメールアドレスなどの PII が含まれる場合があります。	有効	有効
`Endpoint policy` Verified Access エンドポイントの作成または変更時に提供されます。アクセス要求を承認するためのルールが含まれています。ユーザー名やメールアドレスなどの PII が含まれる場合があります。	有効	有効

AWS KMS における AWS Verified Access のグラント使用方法

Verified Access では、カスタマーマネージドキーを使用するためのグラントが必要です。

カスタマーマネージドキーで暗号化された Verified Access リソースを作成すると、Verified Access は にCreateGrantリクエストを送信することで、ユーザーに代わって許可を作成しますAWS KMS。AWS KMSでのグラントを使用して、Verified Access にアカウント内のカスタマーマネージドキーへのアクセス権を与えます。

Verified Access では、以下の内部オペレーションでカスタマーマネージドキーを使用するためにグラントが必要です。

- 暗号化されたデータキーを復号して、それらのキーによるデータを復号化できるようにするには、Decrypt リクエストを AWS KMS に送信します。
- グラントを削除するRetireGrantリクエストを に送信AWS KMSします。

任意のタイミングで、許可に対するアクセス権を取り消したり、カスタマーマネージドキーに対するサービスからのアクセス権を削除したりできます。これを行うと、Verified Access はカスタマーマネージドキーによって暗号化されたすべてのデータにアクセスできなくなり、そのデータに依存しているオペレーションが影響を受けます。

Verified Access でカスタマーマネージドキーを使用する

対称カスタマーマネージドキーを作成するには、AWS Management Console または AWS KMS API を使用します。「AWS Key Management Service デベロッパーガイド」にある「対称カスタマーマネージドキーの作成」ステップを実行します。

キーポリシー

キーポリシーは、カスタマーマネージドキーへのアクセスを制御します。すべてのカスタマーマネージドキーには、キーポリシーが 1 つだけ必要です。このポリシーには、そのキーを使用できるユーザーとその使用方法を決定するステートメントが含まれています。カスタマーマネージドキーを作成する際に、キーポリシーを指定することができます。詳細については、「AWS Key Management Service デベロッパーガイド」の「カスタマーマネージドキーへのアクセスの管理」を参照してください。

カスタマーマネージドキーを Verified Access リソースで使用するには、キーポリシーで次の API オペレーションを許可する必要があります。

- kms:CreateGrant - カスタマーマネージドキーに許可を追加します。グラントは指定した KMS キーへのアクセスを制御します。これにより、必要な許可の付与オペレーションに対し Verified Access がアクセスができるようになります。。詳細については、「AWS Key Management Service デベロッパーガイド」の「グラントの使用」を参照してください。

  これにより、Verified Access が以下を実行できるようになります。

  - GenerateDataKeyWithoutPlainText を呼び出して暗号化されたデータキーを生成して保存します。データキーは暗号化にすぐには使用されないからです。
  - Decrypt を呼び出して、保存されている暗号化データキーを使用して暗号化されたデータにアクセスします。
  - 廃止するプリンシパルを設定して、サービスがRetireGrantを実行できるようにします。
- kms:DescribeKey — カスタマーマネージドキーの詳細を提供し、Verified Access がキーを検証できるようにします。
- kms:GenerateDataKey — Verified Access がキーを使用してデータを暗号化できるようにします。
- kms:Decrypt — Verified Access が暗号化されたデータを復号できるようにします。

以下は、Verified Access で使用できるキーポリシーの例です。

```
"Statement" : [
    {
      "Sid" : "Allow access to principals authorized to use Verified Access",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "*"
      },
      "Action" : [
        "kms:DescribeKey",
        "kms:CreateGrant",
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "kms:ViaService" : "verified-access.region.amazonaws.com",
          "kms:CallerAccount" : "111122223333"
        }
    },
```

```
    {
      "Sid": "Allow access for key administrators",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
       },
      "Action" : [
        "kms:*"
       ],
      "Resource": "arn:aws:kms:region:111122223333:key/key_ID"
    },
    {
      "Sid" : "Allow read-only access to key metadata to the account",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "arn:aws:iam::111122223333:root"
      },
      "Action" : [
        "kms:Describe*",
        "kms:Get*",
        "kms:List*",
        "kms:RevokeGrant"
      ],
      "Resource" : "*"
    }
  ]
```

ポリシーでの許可の指定に関する詳細については、「AWS Key Management Service デベロッパーガイド」を参照してください。

キーアクセスのトラブルシューティングの詳細については、「AWS Key Management Service デベロッパーガイド」を参照してください。

Verified Access リソースのカスタマーマネージドキーを指定する

カスタマーマネージドキーを指定して、以下のリソースに 2 番目の暗号化レイヤを提供できます。

- Verified Access グループ
- Verified Access エンドポイント
- Verified Access 信頼プロバイダー

AWS Management Consoleを使用してこれらのリソースを作成する場合、「追加の暗号化 -- オプション」セクションでカスタマーマネージドキーを指定できます。作成中に [暗号化設定のカスタマイズ (詳細)] チェックボックスを選択し、私用するAWS KMSキー ID を入力します。これは既存のリソースを変更する場合や、AWS CLI を使用して行うこともできます。

> **Note**
>
> 上記のリソースのいずれかに暗号化を追加するために使用するカスタマーマネージドキーが失われた場合、リソースの設定値にアクセスできなくなります。ただし、リソースは、AWS Management Consoleまたは を使用して新しいカスタマーマネージドキーを適用しAWS CLI、設定値をリセットすることで変更できます。

AWS Verified Access 暗号化コンテキスト

暗号化コンテキストとは、データに関する追加のコンテキスト情報を含むために、使用する (オプションの) キーと値のペアのセットです。AWS KMS は、暗号化コンテキストを追加の認証済みデータとして使用し、暗号化の認証をサポートします。データの暗号化リクエストに暗号化コンテキストを組み込むと、AWS KMS は暗号化コンテキストを暗号化後のデータにバインドします。データを復号化するには、そのリクエストに (暗号化時と) 同じ暗号化コンテキストを含めます。

AWS Verifed Access 暗号化コンテキスト

Verified Access は、AWS KMS での暗号化処理のすべてで、同じ暗号化コンテキストを使用します。この際キーには `aws:verified-access:arn` を、値にはリソースの Amazon リソースネーム (ARN) を指定します。Verified Access リソースの暗号化コンテキストは次のとおりです。

Verified Access 信頼プロバイダー

```
"encryptionContext": {
    "aws:verified-access:arn":
    "arn:aws:ec2:region:111122223333:VerifiedAccessTrustProviderId"
}
```

Verified Access グループ

```
"encryptionContext": {
    "aws:verified-access:arn":
    "arn:aws:ec2:region:111122223333:VerifiedAccessGroupId"
}
```

Verified Access エンドポイント

```
"encryptionContext": {
    "aws:verified-access:arn":
    "arn:aws:ec2:region:111122223333:VerifiedAccessEndpointId"
}
```

グラントあるいはポリシー内の暗号化コンテキスト使用の詳細については、「AWS Key Management Service デベロッパーガイド」の「暗号化コンテキスト」を参照してください。

AWS Verified Access の暗号化キーのモニタリング

AWS Verified Access リソースでカスタマーマネージド KMS　キーを使用する場合、AWS CloudTrailを使用して、 Verified Access が AWS KMS に送信するリクエストを追跡できます。

次の例は CreateGrant、RetireGrant、Decrypt、DescribeKey および GenerateDataKey の AWS CloudTrail イベントであり、カスタマーマネージド KMS キーで暗号化されたデータにアクセスするために Verified Access が呼び出す KMS オペレーションがモニタリングされます。

CreateGrant

カスタマーマネージドキーを使用してリソースを暗号化すると、Verified Access はユーザーに代わって AWS アカウント内のキーにアクセスする CreateGrant 要求を送信します。Verified Access が作成するグラントは、カスタマーマネージドキーに関連付けられているリソースに固有のものです。

以下のイベント例は、CreateGrant オペレーションを記録したものです。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
```

```
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-09-11T16:27:12Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "verified-access.amazonaws.com"
    },
    "eventTime": "2023-09-11T16:41:42Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "ca-central-1",
    "sourceIPAddress": "verified-access.amazonaws.com",
    "userAgent": "verified-access.amazonaws.com",
    "requestParameters": {
        "operations": [
            "Decrypt",
            "RetireGrant",
            "GenerateDataKey"
        ],
        "keyId": "arn:aws:kms:ca-central-1:111122223333:key/5ed79e7f-88c9-420c-
ae1a-61ee87104dae",
        "constraints": {
            "encryptionContextSubset": {
                "aws:verified-access:arn": "arn:aws:ec2:ca-
central-1:111122223333:verified-access-trust-provider/vatp-0e54f581e2e5c97a2"
            }
        },
        "granteePrincipal": "verified-access.ca-central-1.amazonaws.com",
        "retiringPrincipal": "verified-access.ca-central-1.amazonaws.com"
    },
    "responseElements": {
        "grantId":
 "e5a050fff9893ba1c43f83fddf61e5f9988f579beaadd6d4ad6d1df07df6048f",
        "keyId": "arn:aws:kms:ca-central-1:111122223333:key/5ed79e7f-88c9-420c-
ae1a-61ee87104dae"
    },
    "requestID": "0faa837e-5c69-4189-9736-3957278e6444",
    "eventID": "1b6dd8b8-cbee-4a83-9b9d-d95fa5f6fd08",
    "readOnly": false,
    "resources": [
        {
```

```
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:ca-central-1:111122223333:key/5ed79e7f-88c9-420c-
ae1a-61ee87104dae"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

RetireGrant

Verified Access では、リソースを削除するときに、RetireGrant オペレーションを使用してグラントを削除します。

以下のイベント例は、RetireGrant オペレーションを記録したものです。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-09-11T16:42:33Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "verified-access.amazonaws.com"
    },
    "eventTime": "2023-09-11T16:47:53Z",
```

```
    "eventSource": "kms.amazonaws.com",
    "eventName": "RetireGrant",
    "awsRegion": "ca-central-1",
    "sourceIPAddress": "verified-access.amazonaws.com",
    "userAgent": "verified-access.amazonaws.com",
    "requestParameters": null,
    "responseElements": {
        "keyId": "arn:aws:kms:ca-central-1:111122223333:key/5ed79e7f-88c9-420c-
ae1a-61ee87104dae"
    },
    "additionalEventData": {
        "grantId":
 "b35e66f9bacb266cec214fcaa353c9cf750785e28773e61ba6f434d8c5c7632f"
    },
    "requestID": "7d4a31c2-d426-434b-8f86-336532a70462",
    "eventID": "17edc343-f25b-43d4-bbff-150d8fff4cf8",
    "readOnly": false,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:ca-central-1:111122223333:key/5ed79e7f-88c9-420c-
ae1a-61ee87104dae"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Decrypt

Verified Access は、保存されている暗号化データキーを使用して暗号化されたデータにアクセスするために Decrypt オペレーションを呼び出します。

以下のイベント例は、Decrypt オペレーションを記録したものです。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/",
```

```
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-09-11T17:19:33Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "verified-access.amazonaws.com"
    },
    "eventTime": "2023-09-11T17:47:05Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "ca-central-1",
    "sourceIPAddress": "verified-access.amazonaws.com",
    "userAgent": "verified-access.amazonaws.com",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e",
        "encryptionContext": {
            "aws:verified-access:arn": "arn:aws:ec2:ca-
central-1:111122223333:verified-access-trust-provider/vatp-00f20a4e455e9340f",
            "aws-crypto-public-key": "AkK+vi1W/
acBKv7OR8p2DeUrA8EgpTffSrjBqNucODuBYhyZ3hlMuYYJz9x7CwQWZw=="
        }
    },
    "responseElements": null,
    "requestID": "2e920fd3-f2f6-41b2-a5e7-2c2cb6f853a9",
    "eventID": "3329e0a3-bcfb-44cf-9813-8106d6eee31d",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
```

```
            "ARN": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

DescribeKey

Verified Access は DescribeKey オペレーションを使用して、リソースに関連付けられているカスタマーマネージドキーがアカウントおよびリージョンに存在するかどうかを確認します。

以下のイベント例は、DescribeKey オペレーションを記録したものです。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-09-11T17:19:33Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "verified-access.amazonaws.com"
    },
    "eventTime": "2023-09-11T17:46:48Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
```

```
    "awsRegion": "ca-central-1",
    "sourceIPAddress": "verified-access.amazonaws.com",
    "userAgent": "verified-access.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e"
    },
    "responseElements": null,
    "requestID": "5b127082-6691-48fa-bfb0-4d40e1503636",
    "eventID": "ffcfc2bb-f94b-4c00-b6fb-feac77daff2a",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


GenerateDataKey

以下のイベント例は、GenerateDataKey オペレーションを記録したものです。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAI44QH8DHBEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
```

```
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-09-11T17:19:33Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "verified-access.amazonaws.com"
    },
    "eventTime": "2023-09-11T17:46:49Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "ca-central-1",
    "sourceIPAddress": "verified-access.amazonaws.com",
    "userAgent": "verified-access.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws:verified-access:arn": "arn:aws:ec2:ca-
central-1:111122223333:verified-access-trust-provider/vatp-00f20a4e455e9340f",
            "aws-crypto-public-key": "A/ATGxaYatPUlOtM+l/mfDndkzHUmX5Hav+29IlIm
+JRBKFuXf24ulztmOIsqFQliw=="
        },
        "numberOfBytes": 32,
        "keyId": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e"
    },
    "responseElements": null,
    "requestID": "06535808-7cce-4ae1-ab40-e3afbf158a43",
    "eventID": "1ce79601-5a5e-412c-90b3-978925036526",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:ca-
central-1:111122223333:key/380d006e-706a-464b-99c5-68768297114e"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

AWS Verified Access の Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つ です。IAM 管理者は、誰を認証 (サインイン) し、誰に Verified Access リソースの使用を 許可する (アクセス許可を持たせる) かをコントロールします。IAM は、追加料金なしで AWS のサービス 使用できる です。

トピック

対象者

AWS Identity and Access Management (IAM) の使用方法は、Verified Access で行う作業によって異なります。

サービスユーザー – Verified Access サービスを使用してジョブを実行する場合は、必要な認証情報とアクセス許可を管理者が提供します。作業を実行するためにさらに多くの Verified Access の機能を使用するとき、追加の許可が必要になる場合があります。アクセスの管理方法を理解しておくと、管理者に適切な許可をリクエストするうえで役立ちます。Veriied Access の機能にアクセスできない場合は、「AWS Verified Access のアイデンティティとアクセスのトラブルシューティング」を参照してください。

サービス管理者 – 社内の Verified Access リソースを担当している場合は、通常、Verified Access に完全にアクセスすることができます。サービスのユーザーがどの Verified Access 機能やリソースにアクセスするかを決めるのは管理者の仕事です。その後、IAM 管理者にリクエストを送信して、サービスユーザーの権限を変更する必要があります。このページの情報を点検して、IAM の基本概念を理解してください。自社で Verified Access で IAM を使用する方法の詳細については、「AWS Verified Access と IAM の連携方法」を参照ください。

IAM 管理者 – IAM 管理者は、Verified Access へのアクセスを管理するポリシーの作成方法の詳細について確認する場合があります。IAM で使用できる Verified Access アイデンティティベースのポリシーの例を表示するには、「AWS Verified Access のアイデンティティベースのポリシーの例」を参照してください。

アイデンティティを使用した認証

認証とは、ID 認証情報 AWS を使用して にサインインする方法です。として、IAM ユーザーとして AWS アカウントのルートユーザー、または IAM ロールを引き受けて認証 (にサインイン AWS) される必要があります。

ID ソースを介して提供された認証情報を使用して、フェデレーティッド ID AWS として にサインインできます。 AWS IAM Identity Center (IAM Identity Center) ユーザー、会社のシングルサインオン認証、Google または Facebook の認証情報は、フェデレーション ID の例です。フェデレーティッド ID としてサインインする場合、IAM ロールを使用して、前もって管理者により ID フェデレーションが設定されています。フェデレーション AWS を使用して にアクセスすると、間接的にロールを引き受けることになります。

ユーザーのタイプに応じて、 AWS Management Console または AWS アクセスポータルにサインインできます。へのサインインの詳細については AWS、「 ユーザーガイド」の「 へのサインイン AWS アカウント方法AWS サインイン 」を参照してください。

AWS プログラムで にアクセスする場合、 は Software Development Kit (SDK) とコマンドラインインターフェイス (CLI) AWS を提供し、認証情報を使用してリクエストに暗号で署名します。 AWS ツールを使用しない場合は、リクエストに自分で署名する必要があります。推奨される方法を使用してリクエストを自分で署名する方法の詳細については、IAM ユーザーガイドの API AWS リクエストの署名を参照してください。

使用する認証方法を問わず、追加セキュリティ情報の提供をリクエストされる場合もあります。例えば、 AWS では、多要素認証 (MFA) を使用してアカウントのセキュリティを向上させることをお勧めします。詳細については、「AWS IAM Identity Center ユーザーガイド」の「Multi-factor authentication」(多要素認証) および「IAM ユーザーガイド」の「AWSでの多要素認証 (MFA) の使用」を参照してください。

AWS アカウント ルートユーザー

を作成するときは AWS アカウント、アカウント内のすべての AWS のサービス およびリソースへの完全なアクセス権を持つ 1 つのサインインアイデンティティから始めます。この ID は AWS アカウント ルートユーザーと呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでサインインすることでアクセスできます。日常的なタスクには、ルートユーザーを使用しないことを強く

お勧めします。ルートユーザーの認証情報は保護し、ルートユーザーでしか実行できないタスクを実行するときに使用します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、IAM ユーザーガイドのルートユーザー認証情報が必要なタスクを参照してください。

フェデレーティッドアイデンティティ

ベストプラクティスとして、管理者アクセスを必要とするユーザーを含む人間のユーザーに、一時的な認証情報を使用して にアクセスするための ID プロバイダーとのフェデレーションの使用を要求 AWS のサービス します。

フェデレーティッド ID は、エンタープライズユーザーディレクトリ、ウェブ ID プロバイダー、AWS Directory Service、アイデンティティセンターディレクトリのユーザー、または ID ソースを通じて提供された認証情報 AWS のサービス を使用して にアクセスするユーザーです。フェデレーティッド ID が にアクセスすると AWS アカウント、ロールを引き受け、ロールは一時的な認証情報を提供します。

アクセスを一元管理する場合は、 AWS IAM Identity Centerを使用することをお勧めします。IAM Identity Center でユーザーとグループを作成することも、独自の ID ソース内のユーザーとグループのセットに接続して同期して、すべての AWS アカウント とアプリケーションで使用することもできます。IAM Identity Center の詳細については、「AWS IAM Identity Center ユーザーガイド」の「What is IAM Identity Center?」(IAM Identity Center とは) を参照してください。

IAM ユーザーとグループ

IAM ユーザーは、単一のユーザーまたはアプリケーションに対して特定のアクセス許可 AWS アカウント を持つ 内のアイデンティティです。可能であれば、パスワードやアクセスキーなどの長期的な認証情報を保有する IAM ユーザーを作成する代わりに、一時的な認証情報を使用することをお勧めします。ただし、IAM ユーザーでの長期的な認証情報が必要な特定のユースケースがある場合は、アクセスキーをローテーションすることをお勧めします。詳細については、IAM ユーザーガイドの長期的な認証情報を必要とするユースケースのためにアクセスキーを定期的にローテーションするを参照してください。

IAM グループは、IAM ユーザーの集団を指定するアイデンティティです。グループとしてサインインすることはできません。グループを使用して、複数のユーザーに対して一度に権限を指定できます。多数のユーザーグループがある場合、グループを使用することで権限の管理が容易になります。例えば、IAMAdminsという名前のグループを設定して、そのグループにIAM リソースを管理する許可を与えることができます。

ユーザーは、ロールとは異なります。ユーザーは 1 人の人または 1 つのアプリケーションに一意に関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。ユー

ザーには永続的な長期の認証情報がありますが、ロールでは一時的な認証情報が提供されます。詳細については、「IAM ユーザーガイド」の「IAM ユーザー (ロールではなく) の作成が適している場合」を参照してください。

IAM ロール

IAM ロールは、特定のアクセス許可 AWS アカウント を持つ 内のアイデンティティです。これは IAM ユーザーに似ていますが、特定のユーザーには関連付けられていません。ロール を切り替える AWS Management Console ことで、 で IAM ロールを一時的に引き受けることができます。ロールを引き受けるには、 または AWS API AWS CLI オペレーションを呼び出すか、カスタム URL を使用します。ロールを使用する方法の詳細については、「IAM ユーザーガイド」の「IAM ロールの使用」を参照してください。

IAM ロールと一時的な認証情報は、次の状況で役立ちます:

- フェデレーションユーザーアクセス – フェデレーティッド ID に許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーティッド ID が認証されると、その ID はロールに関連付けられ、ロールで定義されている許可が付与されます。フェデレーションの詳細については、「IAM ユーザーガイド」の「Creating a role for a third-party Identity Provider」(サードパーティーアイデンティティプロバイダー向けロールの作成) を参照してください。IAM Identity Center を使用する場合は、許可セットを設定します。アイデンティティが認証後にアクセスできるものを制御するため、IAM Identity Center は、権限セットを IAM のロールに関連付けます。アクセス許可セットの詳細については、「AWS IAM Identity Center ユーザーガイド」の「アクセス許可セット」を参照してください。
- 一時的な IAM ユーザー権限 - IAM ユーザーまたはロールは、特定のタスクに対して複数の異なる権限を一時的に IAM ロールで引き受けることができます。
- クロスアカウントアクセス - IAM ロールを使用して、自分のアカウントのリソースにアクセスすることを、別のアカウントの人物 (信頼済みプリンシパル) に許可できます。クロスアカウントアクセス権を付与する主な方法は、ロールを使用することです。ただし、一部の では AWS のサービス、 (ロールをプロキシとして使用する代わりに) ポリシーをリソースに直接アタッチできます。クロスアカウントアクセスにおけるロールとリソースベースのポリシーの違いについては、「IAM ユーザーガイド」の「IAM でのクロスアカウントのリソースへのアクセス」を参照してください。
- クロスサービスアクセス — 一部の は、他の の機能 AWS のサービス を使用します AWS のサービス。例えば、あるサービスで呼び出しを行うと、通常そのサービスによって Amazon EC2 でアプリケーションが実行されたり、Amazon S3 にオブジェクトが保存されたりします。サービスで

は、呼び出し元プリンシパルの許可、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。

- 転送アクセスセッション (FAS) – IAM ユーザーまたはロールを使用して でアクションを実行する場合 AWS、ユーザーはプリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストのリクエストと組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。
- サービスロール - サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAM ユーザーガイド」の「AWS のサービスにアクセス許可を委任するロールの作成」を参照してください。
- サービスにリンクされたロール – サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。

- Amazon EC2 で実行されているアプリケーション – IAM ロールを使用して、EC2 インスタンスで実行され、 AWS CLI または AWS API リクエストを行うアプリケーションの一時的な認証情報を管理できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。 AWS ロールを EC2 インスタンスに割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時的な認証情報を取得できます。詳細については、IAM ユーザーガイドのAmazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用して許可を付与するを参照してください。

IAM ロールと IAM ユーザーのどちらを使用するかについては、IAM ユーザーガイドの(IAM ユーザーではなく) IAM ロールをいつ作成したら良いのか?を参照してください。

ポリシーを使用したアクセスの管理

でアクセスを制御する AWS には、ポリシーを作成し、 AWS ID またはリソースにアタッチします。ポリシーは AWS 、アイデンティティまたはリソースに関連付けられているときにアクセス許可を

定義する のオブジェクトです。 は、プリンシパル (ユーザー、ルートユーザー、またはロールセッション) がリクエストを行うときに、これらのポリシー AWS を評価します。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。ほとんどのポリシーは JSON ドキュメント AWS として に保存されます。JSON ポリシードキュメントの構造と内容の詳細については、IAM ユーザーガイドのJSON ポリシー概要を参照してください。

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

デフォルトでは、ユーザーやロールに権限はありません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き継ぐことができます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションの許可を定義します。例えば、`iam:GetRole`アクションを許可するポリシーがあるとします。そのポリシーを持つユーザーは、 AWS Management Console、、 AWS CLIまたは AWS API からロール情報を取得できます。

アイデンティティベースのポリシー

アイデンティティベースポリシーは、IAM ユーザーグループ、ユーザーのグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースのポリシーを作成する方法については、IAM 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。

アイデンティティベースのポリシーは、さらにインラインポリシーまたはマネージドポリシーに分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれています。管理ポリシーは、 内の複数のユーザー、グループ、ロールにアタッチできるスタンドアロンポリシーです AWS アカウント。管理ポリシーには、 AWS 管理ポリシーとカスタマー管理ポリシーが含まれます。マネージドポリシーまたはインラインポリシーのいずれかを選択する方法については、IAM ユーザーガイドのマネージドポリシーとインラインポリシーの比較を参照してください。

リソースベースのポリシー

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシー や Amazon S3 バケットポリシー があげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。プ

リンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーでは、IAM の AWS マネージドポリシーを使用できません。

アクセスコントロールリスト (ACL)

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Amazon S3、 AWS WAF、および Amazon VPC は、ACLs。ACL の詳細については、Amazon Simple Storage Service デベロッパーガイドのアクセスコントロールリスト (ACL) の概要を参照してください。

その他のポリシータイプ

AWS は、一般的ではない追加のポリシータイプをサポートします。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の権限を設定できます。

- アクセス許可の境界 - アクセス許可の境界は、アイデンティティベースのポリシーによって IAM エンティティ (IAM ユーザーまたはロール) に付与できる権限の上限を設定する高度な機能です。エンティティにアクセス許可の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。`Principal` フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。アクセス許可の境界の詳細については、IAM ユーザーガイドのIAM エンティティのアクセス許可の境界を参照してください。
- サービスコントロールポリシー (SCPs) – SCPs は、 の組織または組織単位 (OU) に対する最大アクセス許可を指定する JSON ポリシーです AWS Organizations。 AWS Organizations は、 AWS アカウント ビジネスが所有する複数の をグループ化して一元管理するサービスです。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCP) を一部またはすべてのアカウントに適用できます。SCP は、各 を含むメンバーアカウントのエンティティのアクセス許可を制限します AWS アカウントのルートユーザー。Organizations と SCP の詳細については、AWS Organizations ユーザーガイドの「SCP の仕組み」を参照してください。
- セッションポリシー - セッションポリシーは、ロールまたはフェデレーションユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの権限は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポ

リシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。詳細については、IAM ユーザーガイドのセッションポリシーを参照してください。

複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成される権限を理解するのがさらに難しくなります。複数のポリシータイプが関与する場合にリクエストを許可するかどうか AWS を決定する方法については、IAM ユーザーガイドの「ポリシー評価ロジック」を参照してください。

AWS Verified Access と IAM の連携方法

IAM を使用して Verified Access へのアクセスを管理する前に、Verified Access で利用できる IAM の機能について学びます。

AWS Verified Access で使用できる IAM の機能

IAM 機能	Verified Access サポート
アイデンティティベースのポリシー	あり
リソースベースのポリシー	なし
ポリシーアクション	あり
ポリシーリソース	Yes
ポリシー条件キー	Yes
ACL	なし
ABAC (ポリシー内のタグ)	部分的
一時的な認証情報	あり
プリンシパル権限	あり
サービスロール	いいえ

IAM 機能	Verified Access サポート
サービスリンクロール	あり

Verified Access およびその他の AWS のサービスがほとんどの IAM 機能と連携する方法の概要を把握するには、「IAM ユーザーガイド」のAWS 「IAM と連携する のサービス」を参照してください。

Verified Access のアイデンティティベースのポリシー

アイデンティティベースポリシーをサポートする	あり

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースのポリシーを作成する方法については、IAM 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。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。JSON ポリシーで使用できるすべての要素について学ぶには、IAM ユーザーガイドのIAM JSON ポリシーの要素のリファレンスを参照してください。

Verified Access のアイデンティティベースポリシーの例

Verified Access アイデンティティベースポリシーの例を表示するには、「AWS Verified Access のアイデンティティベースのポリシーの例」を参照してください。

Verified Access 内のリソースベースのポリシー

リソースベースのポリシーのサポート	なし

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシー や Amazon S3 バケットポリシー があ

げられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

クロスアカウントアクセスを有効にするには、アカウント全体、または別のアカウントの IAM エンティティをリソースベースのポリシーのプリンシパルとして指定します。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる にある場合 AWS アカウント、信頼されたアカウントの IAM 管理者は、プリンシパルエンティティ (ユーザーまたはロール) にリソースへのアクセス許可も付与する必要があります。IAM 管理者は、アイデンティティベースのポリシーをエンティティにアタッチすることで権限を付与します。ただし、リソースベースのポリシーで、同じアカウントのプリンシパルへのアクセス権が付与されている場合は、アイデンティティベースのポリシーをさらに付与する必要はありません。詳細については、「IAM ユーザーガイド」の「IAM でのクロスアカウントリソースアクセス」を参照してください。

Verified Access のポリシーアクション

ポリシーアクションに対するサポート	あり

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

JSON ポリシーの`Action`要素には、ポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。ポリシーアクションの名前は通常、関連付けられた AWS API オペレーションと同じです。一致する API オペレーションのない許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは、依存アクションと呼ばれます。

このアクションは、関連付けられたオペレーションを実行するための権限を付与するポリシーで使用されます。

Verified Access アクションのリストを確認するには、「サービス認証リファレンス」の「Amazon EC2 で定義されるアクション」を参照してください。

Verified Access のポリシーアクションは、アクションの前に以下のプレフィックスを使用します。

```
ec2
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。

```
"Action": [
    "ec2:action1",
    "ec2:action2"
]
```

Verified Access アイデンティティベースポリシーの例を表示するには、「AWS Verified Access のアイデンティティベースのポリシーの例」を参照してください。

Verified Access のポリシーリソース

ポリシーリソースに対するサポート	あり

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースにどのような条件でアクションを実行できるかということです。

`Resource` JSON ポリシー要素は、アクションが適用されるオブジェクトを指定します。ステートメントには、`Resource` または `NotResource` 要素を含める必要があります。ベストプラクティスとして、Amazon リソースネーム (ARN) を使用してリソースを指定します。これは、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

```
"Resource": "*"
```

Verified Access のリソースのタイプとその ARN のリストを確認するには、「サービス認証リファレンス」の「Amazon EC2 で定義されるリソース」を参照してください。どのアクションで各リソースの ARN を指定できるかについては、「Amazon EC2 で定義されるアクション」を参照してください。

Verified Access アイデンティティベースポリシーの例を表示するには、「AWS Verified Access のアイデンティティベースのポリシーの例」を参照してください。

Verified Access の条件キー

サービス固有のポリシー条件キーのサポート	あり

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

Condition 要素 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。イコールや未満などの 条件演算子 を使用して条件式を作成することで、ポリシーの条件とリクエスト内の値を一致させることができます。

1 つのステートメントに複数の Condition 要素を指定する場合、または 1 つの Condition 要素に複数のキーを指定する場合、 AWS では AND 論理演算子を使用してそれらを評価します。1 つの条件キーに複数の値を指定すると、 は論理ORオペレーションを使用して条件 AWS を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、IAM ユーザー名がタグ付けされている場合のみリソースにアクセスできる権限を付与することができます。詳細については、IAM ユーザーガイドのIAM ポリシーの要素: 変数およびタグを参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートします。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」のAWS 「 グローバル条件コンテキストキー」を参照してください。

Verified Access での条件キーの一覧については、「サービス認証リファレンス」の「Amazon EC2 の条件キー」を参照してください。どのアクションおよびリソースと条件キーを使用できるかについては、「Amazon EC2 で定義されるアクション」を参照してください。

Verified Access アイデンティティベースポリシーの例を表示するには、「AWS Verified Access のアイデンティティベースのポリシーの例」を参照してください。

Verified Access の ACL

ACL のサポート	なし

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可を持つかをコントロールします。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Verified Access での ABAC

ABAC (ポリシー内のタグ) のサポート	部分的

属性ベースのアクセス制御 (ABAC) は、属性に基づいてアクセス許可を定義する認可戦略です。では AWS、これらの属性はタグ と呼ばれます。タグは、IAM エンティティ (ユーザーまたはロール) および多くの AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、ABAC の最初の手順です。その後、プリンシパルのタグがアクセスしようとしているリソースのタグと一致した場合にオペレーションを許可するように ABAC ポリシーをします。

ABAC は、急成長する環境やポリシー管理が煩雑になる状況で役立ちます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値はありです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、IAM ユーザーガイドのABAC とは?を参照してください。ABAC をセットアップするステップを説明するチュートリアルについては、「IAM ユーザーガイド」の「属性ベースのアクセス制御 (ABAC) を使用する」を参照してください。

Verified Access での一時的認証情報の使用

一時的な認証情報のサポート	あり

一部の AWS のサービス は、一時的な認証情報を使用してサインインすると機能しません。一時的な認証情報 AWS のサービス を使用する機能などの詳細については、IAM ユーザーガイドのAWS のサービス 「IAM と連携する 」を参照してください。

ユーザー名とパスワード以外の AWS Management Console 方法で にサインインする場合、一時的な認証情報を使用します。例えば、会社の Single Sign-On (SSO) リンク AWS を使用して にアクセ

スすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えに関する詳細については、IAM ユーザーガイドのロールへの切り替え (コンソール)を参照してください。

一時的な認証情報は、 AWS CLI または AWS API を使用して手動で作成できます。その後、これらの一時的な認証情報を使用して . AWS recommends にアクセスできます AWS。これは、長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成することを推奨しています。詳細については、IAM の一時的セキュリティ認証情報を参照してください。

Verified Access のクロスサービスプリンシパル許可

フォワードアクセスセッション (FAS) をサポート	あり

IAM ユーザーまたはロールを使用して でアクションを実行すると AWS、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストリクエストリクエストと組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。

Verified Access のサービスロール

サービスロールのサポート	いいえ

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAM ユーザーガイド」の「AWS のサービスにアクセス許可を委任するロールの作成」を参照してください。

Verified Access 用のサービスにリンクされたロール

サービスリンクロールのサポート	あり

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。

Verified Access のサービスにリンクされたロールの作成または管理の詳細については、「Verified Access のサービスにリンクされたロールを使用する」を参照してください。

AWS Verified Access のアイデンティティベースのポリシーの例

デフォルトでは、ユーザーおよびロールには、Verified Access リソースを作成または変更するアクセス許可はありません。また、、 AWS Command Line Interface （ AWS CLI ） AWS Management Console、または AWS API を使用してタスクを実行することはできません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き受けることができます。

これらサンプルの JSON ポリシードキュメントを使用して、IAM アイデンティティベースのポリシーを作成する方法については、「IAM 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。

Verified Access が定義するアクションとリソースタイプ (リソースタイプごとの ARN の形式を含む) の詳細については、「サービス認証リファレンス」の「Amazon EC2 のアクション、リソース、および条件キー」を参照してください。

トピック

- ポリシーのベストプラクティス
- Verified Access インスタンスを作成するためのポリシー
- 自分の権限の表示をユーザーに許可する

ポリシーのベストプラクティス

アイデンティティベースのポリシーは、ユーザーのアカウントで誰かが Verified Access リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションを実行すると、 AWS アカウントに料金が発生する可能性があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可を付与するには、多くの一般的なユースケースにアクセス許可を付与する AWS 管理ポリシーを使用します。これらは で使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義して、アクセス許可をさらに減らすことをお勧めします。詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」または「ジョブ機能のAWS マネージドポリシー」を参照してください。
- 最小特権を適用する – IAM ポリシーで許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、IAM ユーザーガイドのIAM でのポリシーとアクセス許可を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。例えば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。条件を使用して、 などの特定の を介してサービスアクションが使用される場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます AWS CloudFormation。詳細については、「IAM ユーザーガイド」の [IAM JSON policy elements: Condition] (IAM JSON ポリシー要素:条件) を参照してください。
- IAM Access Analyzer を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する - IAM Access Analyzer は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、IAM ユーザーガイドのIAM Access Analyzer ポリシーの検証を参照してください。
- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、セキュリティを強化するために MFA を有効にします。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、IAM ユーザーガイドのMFA 保護 API アクセスの設定を参照してください。

IAM でのベストプラクティスの詳細については、IAM ユーザーガイドのIAM でのセキュリティのベストプラクティスを参照してください。

Verified Access インスタンスを作成するためのポリシー

Verifid Access のインスタンスを作成するには、IAM プリンシパルは IAM ポリシーにこの追加ステートメントを追加する必要があります。

```
{
    "Effect": "Allow",
    "Action": "verified-access:AllowVerifiedAccess",
    "Resource": "*"
}
```

> **Note**
>
> verified-access:AllowVerifiedAccess はアクションのみの仮想 API です。リソース、タグ、または条件キーベースの認証はサポートされていません。ec2:CreateVerifiedAccessInstance API アクションでは、リソース、タグ、または条件キーベースの認証を使用します。

Verified Access インスタンスを作成するためのポリシーの例。この例では、 123456789012は AWS アカウント番号、 us-east-1は AWS リージョンです。

```
{
      "Version": "2012-10-17",
      "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateVerifiedAccessInstance",
            "Resource": "arn:aws:ec2:us-east-1:123456789012:verified-access-instance/*"
        },
        {
            "Effect": "Allow",
            "Action": "verified-access:AllowVerifiedAccess",
            "Resource": "*"
        }
    ]
}
```

自分の権限の表示をユーザーに許可する

この例では、ユーザーアイデンティティにアタッチされたインラインおよびマネージドポリシーの表示を IAM ユーザーに許可するポリシーの作成方法を示します。このポリシーには、コンソールで、または AWS CLI または AWS API を使用してプログラムでこのアクションを実行するアクセス許可が含まれています。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS Verified Access のアイデンティティとアクセスのトラブルシューティング

次の情報は、Verified Access と IAM を使用する際に発生する可能性がある一般的な問題の診断や修復に役立ちます。

問題

- Verified Access でアクションを実行する権限がない

- iam を実行する権限がありません。PassRole
- 自分の 以外のユーザーに Verified Access リソース AWS アカウント へのアクセスを許可したい

Verified Access でアクションを実行する権限がない

「I am not authorized to perform an action in Amazon Bedrock」というエラーが表示された場合、そのアクションを実行できるようにポリシーを更新する必要があります。

次のエラー例は、mateojackson IAM ユーザーがコンソールを使用して、ある *my-example-widget* リソースに関する詳細情報を表示しようとしたことを想定して、その際に必要なec2:*GetWidget* アクセス許可を持っていない場合に発生するものです。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 ec2:GetWidget on resource: my-example-widget
```

この場合、ec2:*GetWidget* アクションを使用して *my-example-widget*リソースへのアクセスを許可するように、mateojackson ユーザーのポリシーを更新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。サインイン認証情報を提供した担当者が管理者です。

iam を実行する権限がありません。PassRole

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 Verified Access 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 のサービス 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。

以下の例のエラーは、marymajor という IAM ユーザーがコンソールを使用して Verified Access でアクションを実行しようする場合に発生します。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。サインイン認証情報を提供した担当者が管理者です。

自分の 以外のユーザーに Verified Access リソース AWS アカウント 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が、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:

- Verified Access でこれらの機能がサポートされるかどうかを確認するには、「AWS Verified Access と IAM の連携方法」を参照してください。
- 所有 AWS アカウント している の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 AWS アカウント している別の 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。
- リソースへのアクセスをサードパーティー に提供する方法については AWS アカウント、IAM ユーザーガイドの「サードパーティー AWS アカウント が所有する 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介し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) へのアクセスの許可を参照してください。
- クロスアカウントアクセスでのロールとリソースベースのポリシーの使用の違いについては、「IAM ユーザーガイド」の「IAM でのクロスアカウントリソースアクセス」を参照してください。

Verified Access のサービスにリンクされたロールを使用する

AWS Verified Access は、AWS Identity and Access Management (IAM) サービスにリンクされたロールを使用します。 サービスにリンクされたロールは、 Verified Access に直接リンクされた一意のタイプの IAM ロールです。サービスにリンクされたロールは、Verified Access によって事前定義されており、お客様の代わりにサービスから他の AWS のサービス を呼び出す必要のある許可がすべて含まれています。

サービスにリンクされたロールを使用することで、必要なアクセス権限を手動で追加する必要がなくなるため、Verified Access の設定が簡単になります。Verified Access は、サービスにリンクされた

ロールのアクセス許可を定義します。特に定義されている場合を除き、Verified Access のみがそのロールを引き受けることができます。定義した許可には、トラスポリシーと許可ポリシーが含まれます。この許可ポリシーを他のIAM エンティティに添付することはできません。

サービスにリンクされたロールをサポートする他のサービスについては、「IAM と連動する AWS のサービス」を参照し、[Service-linked roles] (サービスにリンクされたロール) の列内で [Yes] (はい) と表記されたサービスを確認してください。サービスにリンクされたロールに関するドキュメントをサービスで表示するには、[はい] リンクを選択します。

Verified Access のためのサービスにリンクされたロールの許可

Verified Access は、AWSServiceRoleForVPCVerifiedAccess という名前のサービスにリンクされたロールを使用して、サービスの使用に必要なリソースをアカウントにプロビジョニングします。

AWSServiceRoleForVPCVerifiedAccess サービスにリンクされたロールは、以下のサービスを信頼してロールを引き受けます。

- `verified-access.amazonaws.com`

AWSVPCVerifiedAccessServiceRolePolicy という名前のロールのアクセス許可ポリシーでは、Verified Access は、指定されたリソースで次のアクションを完了することができます。

- アクション `ec2:CreateNetworkInterface` すべてのサブネット、セキュリティグループ、およびタグ `VerifiedAccessManaged=true` が付いたすべてのネットワークインターフェイスで
- アクション `ec2:CreateTags` 作成時のすべてのネットワークインターフェースで
- アクション `ec2:DeleteNetworkInterface` タグ `VerifiedAccessManaged=true` が付いたすべてのネットワークインターフェースで
- アクション `ec2:ModifyNetworkInterfaceAttribute` すべてのセキュリティグループ、およびタグ`VerifiedAccessManaged=true`が付いたすべてのネットワークインターフェースで

このポリシーのアクセス許可は、 AWS Management Consoleの AWSVPCVerifiedAccessServiceRolePolicy で確認することもできます。または、AWS Managed Policy Reference Guide の AWSVPCVerifiedAccessServiceRolePolicy ポリシーで確認することができます。

サービスにリンクされたロールの作成、編集、削除をIAM エンティティ (ユーザー、グループ、ロールなど) に許可するには、許可を設定する必要があります。詳細については、「IAM User Guide」

(IAM ユーザーガイド) の「Service-linked role permissions」(サービスにリンクされたロールのアクセス権限) を参照してください。

Verified Access のサービスにリンクされたロールを作成する

サービスにリンクされたロールを手動で作成する必要はありません。AWS Management Console、AWS CLI、または AWS API でCreateVerifiedAccessEndpointを呼び出すと、Verified Access によってサービスにリンクされたロールが作成されます。

このサービスにリンクされたロールを削除した後で再度作成する必要が生じた場合は、同じ方法でアカウントにロールを再作成できます。CreateVerifiedAccessEndpoint を再度呼び出すと、 によって、Verified Access によってサービスにリンクされたロールが再度作成されます。

Verified Access のサービスにリンクされたロールを編集する

Verified Access では、サービスにリンクされたロールである AWSServiceRoleForVPCVerifiedAccess を編集できません。サービスにリンクされたロールを作成すると、多くのエンティティによってロールが参照される可能性があるため、ロール名を変更することはできません。ただし、IAM を使用したロールの説明の編集はできます。詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの編集」を参照してください。

Verified Access のサービスにリンクされたロールを削除する

AWSServiceRoleForVPCVerifiedAccess ロールを手動で削除する必要はありません。AWS Management Console、AWS CLI、または AWS API 内で DeleteVerifiedAccessEndpoint を呼び出すと、Verified Access がリソースをクリーンアップし、サービスにリンクされたロールを削除します。

IAM を使用してサービスリンクロールを手動で削除するには

IAM コンソール、AWS CLI、または AWS API を使用して、AWSServiceRoleForVPCVerifiedAccess サービスリンクロールを削除します。詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの削除」を参照してください。

Verified Access サービスにリンクされたロールをサポートするリージョン

Verified Access では、このサービスが利用可能なすべての AWS リージョン で、サービスリンクロールの使用をサポートしています。詳細については、「AWS リージョンとエンドポイント」を参照してください。

AWS Verified Access のAWS マネージドポリシー

AWS マネージドポリシーは、AWS が作成および管理するスタンドアロンポリシーです。AWS マネージドポリシーは、多くの一般的なユースケースで権限を提供できるように設計されているため、ユーザー、グループ、ロールへの権限の割り当てを開始できます。

AWS マネージドポリシーは、ご利用の特定のユースケースに対して最小特権の権限を付与しない場合があることにご注意ください。AWS のすべてのお客様が使用できるようになるのを避けるためです。ユースケース別にカスタマー管理ポリシーを定義することで、権限を絞り込むことをお勧めします。

AWS マネージドポリシーで定義したアクセス権限は変更できません。AWS が AWS マネージドポリシーに定義されている権限を更新すると、更新はポリシーがアタッチされているすべてのプリンシパルアイデンティティ (ユーザー、グループ、ロール) に影響します。新しい AWS のサービスを起動するか、既存のサービスで新しい API オペレーションが使用可能になると、AWS が AWS マネージドポリシーを更新する可能性が最も高くなります。

詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」を参照してください。

AWS マネージドポリシー：AWSVPCVerifiedAccessServiceRolePolicy

このポリシーは、ユーザーに代わって Verified Access がアクションを実行することを許可する、サービスにリンクされたロールに添付されます。詳細については、「サービスにリンクされたロールの使用」を参照してください。このポリシーのアクセス権限を確認するには、AWS Management Consoleの AWSVPCVerifiedAccessServiceRolePolicyを確認するか、「AWS マネージドポリシーレファレンスガイド」の AWSVPCVerifiedAccessServiceRolePolicy ポリシーを確認してください。

Verified Access の AWS マネージドポリシーに対する更新

Verified Access の AWS マネージドポリシーの更新に関する詳細を、このサービスがこれらの変更の追跡を開始した以降の分について表示します。このページの変更に関する自動通知については、Verified Access [Document history] (ドキュメントの履歴) ページの RSS フィードをサブスクライブしてください。

変更	説明	日付
AWSVPCVerifiedAccessServiceRolePolicy - ポリシー更新	Verified Access は、すべてのアクションの説明を「sid」フィールドに追加するように	2023 年 11 月 17 日

変更	説明	日付
	マネージドポリシーを更新しました。	
AWSVPCVerifiedAccessServiceRolePolicy - ポリシー更新	Verified Access がマネージドポリシーを更新し、セキュリティグループリソースを`ec2:CreateNetworkInterface` のアクセス権限に追加しました。	2023 年 5 月 31 日
AWSVPCVerifiedAccessServiceRolePolicy - 新ポリシー	Verified Access に、サービスの使用に必要なリソースをアカウントにプロビジョニングできるようにする新しいポリシーが追加されました。	2022 年 11 月 29 日
Verified Access は変更の追跡を開始	Verified Access が AWS マネージドポリシーの変更の追跡を開始しました。	2022 年 11 月 29 日

AWS Verified Access のコンプライアンス検証

AWS Verified Access は、連邦情報処理標準 (FIPS) のコンプライアンスをサポートするように設定できます。Verified Access の FIPS コンプライアンスの設定に関する情報と詳細については、Verified Access の FIPS 準拠を参照してください。

AWS のサービス が特定のコンプライアンスプログラムの範囲内にあるかどうかを確認するには、コンプライアンスプログラムAWS のサービス による対象範囲内のコンプライアンスプログラムを参照し、関心のあるコンプライアンスプログラムを選択します。一般的な情報については、AWS 「コンプライアンスプログラム」を参照してください。

を使用して、サードパーティーの監査レポートをダウンロードできます AWS Artifact。詳細については、「 でのレポートのダウンロード AWS Artifact」の」を参照してください。

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

- セキュリティとコンプライアンスのクイックスタートガイド – これらのデプロイガイドでは、アーキテクチャ上の考慮事項について説明し、セキュリティとコンプライアンスに重点を置いたベースライン環境 AWS を にデプロイする手順について説明します。
- アマゾン ウェブ サービスにおける HIPAA セキュリティとコンプライアンスのアーキテクチャ — このホワイトペーパーでは、企業が AWS を使用して HIPAA 対象アプリケーションを作成する方法について説明します。

> **Note**
>
> すべて AWS のサービス HIPAA の対象となるわけではありません。詳細については、「HIPAA 対応サービスのリファレンス」を参照してください。

- AWS コンプライアンスリソース – このワークブックとガイドのコレクションは、お客様の業界や地域に適用される場合があります。
- AWS カスタマーコンプライアンスガイド — コンプライアンスの観点から責任共有モデルを理解します。このガイドでは、ガイダンスを保護し AWS のサービス 、複数のフレームワーク (米国国立標準技術研究所 (NIST)、Payment Card Industry Security Standards Council (PCI)、国際標準化機構 (ISO) を含む) のセキュリティコントロールにマッピングするためのベストプラクティスをまとめています。
- 「 デベロッパーガイド」の「 ルールによるリソースの評価」 – この AWS Config サービスは、リソース設定が社内プラクティス、業界ガイドライン、および規制にどの程度準拠しているかを評価します。 AWS Config
- AWS Security Hub – これにより AWS のサービス 、 内のセキュリティ状態を包括的に確認できます AWS。Security Hub では、セキュリティコントロールを使用して AWS リソースを評価し、セキュリティ業界標準とベストプラクティスに対するコンプライアンスをチェックします。サポートされているサービスとコントロールのリストについては、「Security Hub のコントロールリファレンス」を参照してください。
- Amazon GuardDuty – これにより AWS アカウント、疑わしいアクティビティや悪意のあるアクティビティがないか環境を監視することで、、、ワークロード、コンテナ、データに対する潜在的な脅威 AWS のサービス を検出します。 GuardDuty は、特定のコンプライアンスフレームワークで義務付けられている侵入検知要件を満たすことで、PCI DSS などのさまざまなコンプライアンス要件への対応に役立ちます。

- AWS Audit Manager – これにより AWS のサービス 、 AWS 使用状況を継続的に監査し、リスクの管理方法と規制や業界標準への準拠を簡素化できます。

AWS Verified Access における耐障害性

AWS グローバルインフラストラクチャは AWS リージョン およびアベイラビリティーゾーンを中心に構築されています。AWS リージョン には、低レイテンシー、高いスループット、そして高度の冗長ネットワークで接続されている物理的に独立· 隔離された複数のアベイラビリティーゾーンがあります。アベイラビリティーゾーンを使用すると、中断することなくゾーン間で自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用できます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性が高く、フォールトトレラントで、スケーラブルです。

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

Verified Access では、AWSグローバルインフラストラクチャに加えて、高可用性のニーズに対応できるように以下の機能を提供しています。

高可用性対応の複数のサブネット

ロードバランサータイプの Verified Access エンドポイントを作成する際には、エンドポイントに複数のサブネットを関連付けることができます。エンドポイントに関連付ける各サブネットは、異なるアベイラビリティーゾーンに属している必要があります。複数のサブネットを関連付けることで、複数のアベイラビリティーゾーンを使用して高い可用性を確保できます。

AWS Verified Access のモニタリング

モニタリングは、AWS Verified Access の信頼性、可用性、およびパフォーマンスの維持における重要な要素です。 AWS は、 Verified Access をモニタリングし、問題が発生した場合には報告を行い、必要に応じて自動アクションを実行するために以下のモニタリングツールを提供しています。

- アクセスログ — アプリケーションへのアクセス要求に関する詳細情報を取得します。詳細については、「the section called “Verified Access ログ”」を参照してください。
- AWS CloudTrail - AWS アカウント により、またはそのアカウントに代わって行われた API コールおよび関連イベントを取得し、指定した Amazon S3 バケットにログファイルを配信します。AWS を呼び出したユーザーとアカウント、呼び出し元の IP アドレス、および呼び出し日時を特定できます。詳細については、「the section called “CloudTrail ログ”」を参照してください。

Verified Access ログ

AWS Verified Access は、各アクセスリクエストを評価すると、すべてのアクセス試行をログに記録します。これにより、アプリケーションへのアクセスを一元的に把握でき、セキュリティインシデントや監査請求に迅速に対応できます。Verified Access は、オープンサイバーセキュリティスキーマフレームワーク (OCSF) ロギング形式をサポートしています。

ロギングを有効にする場合は、ログの送信先を設定する必要があります。ロギング先の設定に使用する IAM プリンシパルには、ロギングが正しく機能するための特定のアクセス権限が必要です。各ロギング先に必要な IAM アクセス権限は、ロギングのアクセス権限 セクションで確認できます。Verified Access は、以下のアクセスログのパブリッシュ先をサポートします。

- Amazon CloudWatch Logs ロググループ
- Amazon S3 バケット
- Amazon Data Firehose 配信ストリーム

コンテンツ

- Verified Access ログのログエントリの例

ロギングバージョン

デフォルトで、Verified Access ロギングシステムはオープンサイバーセキュリティスキーマフレームワーク（OCSF）バージョン 0.1 を使用します。バージョン 0.1 を使用したサンプルログは、OCSF バージョン 0.1 の例 セクションで確認できます。

最新のロギングバージョンは OCSF バージョン 1.0.0-rc.2 と互換性があります。スキーマの詳細については、OCSF スキーマをご覧ください。バージョン 1.0.0-rc.2 を使用したサンプルログは、OCSF バージョン 1.0.0-rc.2 の例 セクションで確認できます。

ログバージョンのアップグレード

使用しているロギングバージョンをアップグレードする場合は、以下の手順に従ってください。

コンソールを使用してロギングバージョンをアップグレードするには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. 適切な Verified Access インスタンスを選択します。
4. [Verified Access インスタンスのロギング設定] タブで、[Verified Access インスタンスのロギング設定の変更] を選択します。
5. 「ログバージョンの更新」ドロップダウンリストから ocsf-1.0.0-rc.2 を選択します。
6. [Verified Access インスタンスのロギング設定の変更] を選択します。

を使用してログ記録バージョンをアップグレードするには AWS CLI

modify-verified-access-instance-logging-configuration コマンドを使用します。

ロギングのアクセス権限

ロギング先の設定に使用する IAM プリンシパルには、ロギングが正しく機能するための特定のアクセス権限が必要です。以下に、各ロギング先に必要なアクセス権限を以下に示します。

CloudWatch ログへの配信：

- Verified Access インスタンスの `ec2:ModifyVerifiedAccessInstanceLoggingConfiguration`

- すべてのリソースの logs:CreateLogDelivery、logs:DeleteLogDelivery、logs:GetLogDelivery、logs:ListLo および logs:UpdateLogDelivery
- 送信先ロググループの logs:DescribeLogGroups、logs:DescribeResourcePolicies および logs:PutResourcePolicy

Amazon S3 への配信：

- Verified Access インスタンスの ec2:ModifyVerifiedAccessInstanceLoggingConfiguration
- すべてのリソースの logs:CreateLogDelivery、logs:DeleteLogDelivery、logs:GetLogDelivery、logs:ListLo および logs:UpdateLogDelivery
- 送信先バケットの s3:GetBucketPolicy および s3:PutBucketPolicy

Firehose への配信の場合：

- Verified Access インスタンスの ec2:ModifyVerifiedAccessInstanceLoggingConfiguration
- すべてのリソースの firehose:TagDeliveryStream
- すべてのリソースの iam:CreateServiceLinkedRole
- すべてのリソースの logs:CreateLogDelivery、logs:DeleteLogDelivery、logs:GetLogDelivery、logs:ListLo および logs:UpdateLogDelivery

Enable or disable logs

ロギングを有効にする場合は、ログの送信先を設定する必要があります。ロギング先の設定に使用する IAM プリンシパルには、ロギングが正しく機能するための特定のアクセス権限が必要です。各ロギング先に必要な IAM アクセス権限は、ロギングのアクセス権限 セクションで確認できます。

コンテンツ

アクセスログの有効化

Verified Access ログを有効にするには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. Verified Access インスタンスを選択します。
4. [Verified Access インスタンスのロギング設定] タブで、[Verified Access インスタンスのロギング設定の変更] を選択します。
5. (オプション) 信頼プロバイダーから送信されるトラストデータをログに含めるには、次の操作を行います。
   a. 「ログバージョンの更新」ドロップダウンリストから ocsf-1.0.0-rc.2 を選択します。
   b. [トラストコンテキストを含める] を選択します。
6. 次のいずれかを行います。
   - Amazon CloudWatch Logs への配信を有効にします。送信先ロググループを選択します。
   - [Amazon S3 に配信] をオンにします。送信先バケットの名前、所有者、プレフィックスを入力します。
   - Firehose への配信 をオンにします。送信先の配信ストリームを選択します。
7. [Verified Access インスタンスのロギング設定の変更] を選択します。

を使用して Verified Access ログを有効にするには AWS CLI

modify-verified-access-instance-logging-configuration コマンドを使用します。

アクセスログの無効化

Verified Access インスタンスのアクセスログは、いつでも無効化できます。アクセスログを無効にした後は、削除するまでログデータはログ送信先に残ります。

Verified Access ログを無効にするには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. Verified Access インスタンスを選択します。

4. [Verified Access インスタンスのロギング設定] タブで、[Verified Access インスタンスのロギング設定の変更] を選択します。
5. ログ配信をオフにします。
6. [Verified Access インスタンスのロギング設定の変更] を選択します。

を使用して Verified Access ログを無効にするには AWS CLI

modify-verified-access-instance-logging-configuration コマンドを使用します。

トラストコンテキストを含める

信頼プロバイダーから送信されるトラストコンテキストは、オプションで Verified Access ログに含めることができます。これは、アプリケーションへのアクセスを許可または拒否するポリシーを定義する際に非常に役立ちます。有効にすると、トラストコンテキストはログの `data` フィールドの下に表示されます。無効にすると、`data` フィールドは `null` に設定されます。トラストコンテキストをログに含めるように Verified Access を設定するには、以下の手順に従います。

> **Note**
>
> Verified Access ログにトラストコンテキストを含めるには、最新のロギングバージョン `ocsf-1.0.0-rc.2` にアップグレードする必要があります。以下の手順は、ロギングがすでに有効になっていることを前提としています。そうでない場合、手順の詳細については アクセスログの有効化 を参照してください。

コンテンツ

トラストコンテキストを有効にする

コンソールを使用して Verified Access ログにトラストコンテキストを含めるには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. 適切な Verified Access インスタンスを選択します。

4. [Verified Access インスタンスのロギング設定] タブで、[Verified Access インスタンスのロギング設定の変更] を選択します。
5. 「ログバージョンの更新」ドロップダウンリストから [ocsf-1.0.0-rc.2] を選択します。
6. [トラストコンテキストを含める] をオンにします。
7. [Verified Access インスタンスのロギング設定の変更] を選択します。

を使用して Verified Access ログに信頼コンテキストを含めるには AWS CLI

modify-verified-access-instance-logging-configuration コマンドを使用します。

トラストコンテキストを無効にする

トラストコンテキスト内のログが不要になった場合は、以下の手順で削除できます。

コンソールを使用して Verified Access ログからトラストコンテキストを削除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Verified Access インスタンス] を選択します。
3. 適切な Verified Access インスタンスを選択します。
4. [Verified Access インスタンスのロギング設定] タブで、[Verified Access インスタンスのロギング設定の変更] を選択します。
5. [トラストコンテキストを含める] をオフにします。
6. [Verified Access インスタンスのロギング設定の変更] を選択します。

を使用して Verified Access ログから信頼コンテキストを削除するには AWS CLI

modify-verified-access-instance-logging-configuration コマンドを使用します。

Verified Access ログのログエントリの例

以下にログエントリの例を示します。

コンテンツ

OCSF バージョン 0.1 の例

以下は、デフォルトのロギング OCSF バージョン 0.1 を使用したサンプルログです。

例

- OIDC によるアクセス許可
- OIDC と JAMF によるアクセス許可
- OIDC と で付与されるアクセス CrowdStrike
- Cookie の欠落によるアクセスの拒否
- ポリシーによるアクセス拒否
- 不明なログエントリ

OIDC によるアクセス許可

このログエントリの例では、Verified Access は OIDC ユーザトラストプロバイダーでエンドポイントへのアクセスを許可します。


```
{
    "activity": "Access Granted",
    "activity_id": "1",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
    "class_uid": "208001",
    "device": {
        "ip": "10.2.7.68",
        "type": "Unknown",
        "type_id": 0
    },
    "duration": "0.004",
    "end_time": "1668580194344",
    "time": "1668580194344",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "https",
            "text": "https://hello.app.example.com:443/"
```

```
        },
        "user_agent": "python-requests/2.28.1",
        "version": "HTTP/1.1"
    },
    "http_response": {
        "code": 200
    },
    "identity": {
        "authorizations": [
            {
                "decision": "Allow",
                "policy": {
                    "name": "inline"
                }
            }
        ],
        "idp": {
            "name": "user",
            "uid": "vatp-09bc4cbce2EXAMPLE"
        },
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
            "uuid": "00u6wj48lbxTAEXAMPLE"
        }
    },
    "message": "",
    "metadata": {
        "uid": "Root=1-63748362-6408d24241120b942EXAMPLE",
        "logged_time": 1668580281337,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-16T06:29:54.344948Z",
    "proxy": {
        "ip": "192.168.34.167",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-002fa341aeEXAMPLE"
    },
```

```
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "172.24.57.68",
        "port": "48234"
    },
    "start_time": "1668580194340",
    "status_code": "100",
    "status_details": "Access Granted",
    "status_id": "1",
    "status": "Success",
    "type_uid": "20800101",
    "type_name": "AccessLogs: Access Granted",
    "unmapped": null
}
```

OIDC と JAMF によるアクセス許可

この例では、Verified Access は OIDC と JAMF の両方のデバイス信頼プロバイダーでエンドポイントへのアクセスを許可します。

```
{
    "activity": "Access Granted",
    "activity_id": "1",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
    "class_uid": "208001",
    "device": {
        "ip": "10.2.7.68",
        "type": "Unknown",
        "type_id": 0,
        "uid": "41b07859-4222-4f41-f3b9-97dc1EXAMPLE"
    },
    "duration": "0.347",
    "end_time": "1668804944086",
    "time": "1668804944086",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
```

```
            "scheme": "h2",
            "text": "https://hello.app.example.com:443/"
        },
        "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36",
        "version": "HTTP/2.0"
    },
    "http_response": {
        "code": 304
    },
    "identity": {
        "authorizations": [
            {
                "decision": "Allow",
                "policy": {
                    "name": "inline"
                }
            }
        ],
        "idp": {
            "name": "oidc",
            "uid": "vatp-9778003bc2EXAMPLE"
        },
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
            "uuid": "4f040d0f96becEXAMPLE"
        }
    },
    "message": "",
    "metadata": {
        "uid": "Root=1-321318ce-6100d340adf4fb29dEXAMPLE",
        "logged_time": 1668805278555,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-18T20:55:44.086480Z",
    "proxy": {
        "ip": "10.5.192.96",
        "port": 443,
```

```
        "svc_name": "Verified Access",
        "uid": "vai-3598f66575EXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "192.168.20.246",
        "port": 61769
    },
    "start_time": "1668804943739",
    "status_code": "100",
    "status_details": "Access Granted",
    "status_id": "1",
    "status": "Success",
    "type_uid": "20800101",
    "type_name": "AccessLogs: Access Granted",
    "unmapped": null
}
```

OIDC と で付与されるアクセス CrowdStrike

このログエントリの例では、Verified Access は OIDC と CrowdStrikeデバイスの両方の信頼プロバイダーを使用してエンドポイントへのアクセスを許可します。

```
{
    "activity": "Access Granted",
    "activity_id": "1",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
    "class_uid": "208001",
    "device": {
        "ip": "10.2.173.3",
        "os": {
            "name": "Windows 11",
            "type": "Windows",
            "type_id": 100
        },
        "type": "Unknown",
        "type_id": 0,
        "uid": "122978434f65093aee5dfbdc0EXAMPLE",
        "hw_info": {
            "serial_number": "751432a1-d504-fd5e-010d-5ed11EXAMPLE"
```

```
        }
    },
    "duration": "0.028",
    "end_time": "1668816620842",
    "time": "1668816620842",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "test.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "h2",
            "text": "https://test.app.example.com:443/"
        },
        "user_agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36",
        "version": "HTTP/2.0"
    },
    "http_response": {
        "code": 304
    },
    "identity": {
        "authorizations": [
            {
                "decision": "Allow",
                "policy": {
                    "name": "inline"
                }
            }
        ],
        "idp": {
            "name": "oidc",
            "uid": "vatp-506d9753f6EXAMPLE"
        },
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
            "uuid": "23bb45b16a389EXAMPLE"
        }
    },
    "message": "",
    "metadata": {
        "uid": "Root=1-c16c5a65-b641e4056cc6cb0eeEXAMPLE",
```

```
        "logged_time": 1668816977134,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-19T00:10:20.842295Z",
    "proxy": {
        "ip": "192.168.144.62",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-2f80f37e64EXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "10.14.173.3",
        "port": 55706
    },
    "start_time": "1668816620814",
    "status_code": "100",
    "status_details": "Access Granted",
    "status_id": "1",
    "status": "Success",
    "type_uid": "20800101",
    "type_name": "AccessLogs: Access Granted",
    "unmapped": null
}
```


Cookie の欠落によるアクセスの拒否

このログエントリの例では、認証 Cookie の欠落により Verified Access がアクセスを拒否します。


```
{
    "activity": "Access Denied",
    "activity_id": "2",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
    "class_uid": "208001",
    "device": null,
    "duration": "0.0",
    "end_time": "1668593568259",
```

```
    "time": "1668593568259",
    "http_request": {
        "http_method": "POST",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/dns-query",
            "port": 443,
            "scheme": "h2",
            "text": "https://hello.app.example.com:443/dns-query"
        },
        "user_agent": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML",
        "version": "HTTP/2.0"
    },
    "http_response": {
        "code": 302
    },
    "identity": null,
    "message": "",
    "metadata": {
        "uid": "Root=1-5cf1c832-a565309ce20cc7dafEXAMPLE",
        "logged_time": 1668593776720,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-16T10:12:48.259762Z",
    "proxy": {
        "ip": "192.168.34.167",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-108ed7a672EXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "10.7.178.16",
        "port": "46246"
    },
    "start_time": "1668593568258",
    "status_code": "200",
    "status_details": "Authentication Denied",
    "status_id": "2",
```

```
    "status": "Failure",
    "type_uid": "20800102",
    "type_name": "AccessLogs: Access Denied",
    "unmapped": null
}
```

ポリシーによるアクセス拒否

この口グエントリの例では、認証されたリクエストがアクセスポリシーで許可されていないため、Verified Access は認証されたリクエストを拒否します。

```
{
    "activity": "Access Denied",
    "activity_id": "2",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
    "class_uid": "208001",
    "device": {
        "ip": "10.4.133.137",
        "type": "Unknown",
        "type_id": 0
    },
    "duration": "0.023",
    "end_time": "1668573630978",
    "time": "1668573630978",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "h2",
            "text": "https://hello.app.example.com:443/"
        },
        "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36",
        "version": "HTTP/2.0"
    },
    "http_response": {
        "code": 401
    },
    "identity": {
```

```
        "authorizations": [],
        "idp": {
            "name": "user",
            "uid": "vatp-e048b3e0f8EXAMPLE"
        },
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
            "uuid": "0e1281ad3580aEXAMPLE"
        }
    },
    "message": "",
    "metadata": {
        "uid": "Root=1-531a036a-09e95794c7b96aefbEXAMPLE",
        "logged_time": 1668573773753,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-16T04:40:30.978732Z",
    "proxy": {
        "ip": "3.223.34.167",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-021d5eaed2EXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "10.4.133.137",
        "port": "31746"
    },
    "start_time": "1668573630955",
    "status_code": "300",
    "status_details": "Authorization Denied",
    "status_id": "2",
    "status": "Failure",
    "type_uid": "20800102",
    "type_name": "AccessLogs: Access Denied",
    "unmapped": null
```

```
}
```

不明なログエントリ

このログエントリの例では、Verified Access では完全なログエントリを生成できないため、不明なログエントリが出力されます。これにより、すべてのリクエストがアクセスログに表示されることが保証されます。

```
{
    "activity": "Unknown",
    "activity_id": "0",
    "category_name": "Application Activity",
    "category_uid": "8",
    "class_name": "Access Logs",
    "class_uid": "208001",
    "device": null,
    "duration": "0.004",
    "end_time": "1668580207898",
    "time": "1668580207898",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "https",
            "text": "https://hello.app.example.com:443/"
        },
        "user_agent": "python-requests/2.28.1",
        "version": "HTTP/1.1"
    },
    "http_response": {
        "code": 200
    },
    "identity": null,
    "message": "",
    "metadata": {
        "uid": "Root=1-435eb955-6b5a1d529343f5adaEXAMPLE",
        "logged_time": 1668580579147,
        "version": "0.1",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
```

```
        }
    },
    "ref_time": "2022-11-16T06:30:07.898344Z",
    "proxy": {
        "ip": "10.1.34.167",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-6c32b53b3cEXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "172.28.57.68",
        "port": "47220"
    },
    "start_time": "1668580207893",
    "status_code": "000",
    "status_details": "Unknown",
    "status_id": "0",
    "status": "Unknown",
    "type_uid": "20800100",
    "type_name": "AccessLogs: Unknown",
    "unmapped": null
}
```

OCSF バージョン 1.0.0-rc.2 の例

コンテンツ

トラストコンテキストを含むアクセス許可

```
{
    "activity_name": "Access Grant",
    "activity_id": "1",
    "actor": {
        "authorizations": [{
            "decision": "Allow",
            "policy": {
                "name": "inline"
            }
```

```
        }],
        "idp": {
            "name": "user",
            "uid": "vatp-09bc4cbce2EXAMPLE"
        },
        "invoked_by": "",
        "process": {},
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
            "uuid": "00u6wj48lbxTAEXAMPLE"
        },
        "session": {}
    },
    "category_name": "Audit Activity",
    "category_uid": "3",
    "class_name": "Access Activity",
    "class_uid": "3006",
    "device": {
        "ip": "10.2.7.68",
        "type": "Unknown",
        "type_id": 0
    },
    "duration": "0.004",
    "end_time": "1668580194344",
    "time": "1668580194344",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "https",
            "text": "https://hello.app.example.com:443/"
        },
        "user_agent": "python-requests/2.28.1",
        "version": "HTTP/1.1"
    },
    "http_response": {
        "code": 200
    },
    "message": "",
    "metadata": {
```

```
        "uid": "Root=1-63748362-6408d24241120b942EXAMPLE",
        "logged_time": 1668580281337,
        "version": "1.0.0-rc.2",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-16T06:29:54.344948Z",
    "proxy": {
        "ip": "192.168.34.167",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-002fa341aeEXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "172.24.57.68",
        "port": "48234"
    },
    "start_time": "1668580194340",
    "status_code": "100",
    "status_detail": "Access Granted",
    "status_id": "1",
    "status": "Success",
    "type_uid": "300601",
    "type_name": "Access Activity: Access Grant",
    "data": {
        "context": {
            "oidc": {
                "family_name": "Last",
                "zoneinfo": "America/Los_Angeles",
                "exp": 1670631145,
                "middle_name": "Middle",
                "given_name": "First",
                "email_verified": true,
                "name": "Test User Display",
                "updated_at": 1666305953,
                "preferred_username": "johndoe-user@test.com",
                "profile": "http://www.example.com",
                "locale": "US",
                "nickname": "Tester",
                "email": "johndoe-user@test.com"
```

```
            },
            "http_request": {
                "x_forwarded_for": "1.1.1.1,2.2.2.2",
                "http_method": "GET",
                "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36",
                "port": "80",
                "hostname": "hostname.net"
            }
        }
    }
}
```

トラストコンテキストを省略したアクセス許可

```
{
    "activity_name": "Access Grant",
    "activity_id": "1",
    "actor": {
        "authorizations": [{
            "decision": "Allow",
            "policy": {
                "name": "inline"
            }
        }],
        "idp": {
            "name": "user",
            "uid": "vatp-09bc4cbce2EXAMPLE"
        },
        "invoked_by": "",
        "process": {},
        "user": {
            "email_addr": "johndoe@example.com",
            "name": "Test User Display",
            "uid": "johndoe@example.com",
            "uuid": "00u6wj48lbxTAEXAMPLE"
        },
        "session": {}
    },
    "category_name": "Audit Activity",
    "category_uid": "3",
    "class_name": "Access Activity",
    "class_uid": "3006",
```

```
    "device": {
        "ip": "10.2.7.68",
        "type": "Unknown",
        "type_id": 0
    },
    "duration": "0.004",
    "end_time": "1668580194344",
    "time": "1668580194344",
    "http_request": {
        "http_method": "GET",
        "url": {
            "hostname": "hello.app.example.com",
            "path": "/",
            "port": 443,
            "scheme": "https",
            "text": "https://hello.app.example.com:443/"
        },
        "user_agent": "python-requests/2.28.1",
        "version": "HTTP/1.1"
    },
    "http_response": {
        "code": 200
    },
    "message": "",
    "metadata": {
        "uid": "Root=1-63748362-6408d24241120b942EXAMPLE",
        "logged_time": 1668580281337,
        "version": "1.0.0-rc.2",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "ref_time": "2022-11-16T06:29:54.344948Z",
    "proxy": {
        "ip": "192.168.34.167",
        "port": 443,
        "svc_name": "Verified Access",
        "uid": "vai-002fa341aeEXAMPLE"
    },
    "severity": "Informational",
    "severity_id": "1",
    "src_endpoint": {
        "ip": "172.24.57.68",
```

```
        "port": "48234"
    },
    "start_time": "1668580194340",
    "status_code": "100",
    "status_detail": "Access Granted",
    "status_id": "1",
    "status": "Success",
    "type_uid": "300601",
    "type_name": "Access Activity: Access Grant",
    "data": null
}
```

AWS CloudTrailを使用して AWS Verified Access API コールをログに記録します。

AWS Verified Access は、Verified Access のユーザー、ロール、または AWS のサービス によって実行されたアクションをレコードするサービスである AWS CloudTrail と統合されています。CloudTrail は、Verified Access のすべての API コールをイベントとしてキャプチャします。キャプチャされた呼び出しには、Verified Access コンソールからの呼び出しと、Verified Access の API オペレーションへのコードの呼び出しが含まれます。証跡を作成する場合は、Verified Access のイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的な配信を有効にすることができます。追跡を設定しない場合でも、CloudTrail コンソールの [Event history] (イベント履歴) で最新のイベントを表示できます。CloudTrail で収集された情報を使用して、Verified Access に対するリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト日時などの詳細を確認できます。

CloudTrail の詳細については、「AWS CloudTrail ユーザーガイド」を参照してください。

CloudTrail での Verified Access 情報

CloudTrail は、アカウント作成時に AWS アカウント で有効になります。Verified Access でアクティビティが発生すると、そのアクティビティはイベント履歴内の他の AWS のサービス のイベントと共に、CloudTrail イベントに記録されます。最近のイベントは、AWS アカウント で表示、検索、ダウンロードできます。詳細については、 CloudTrail イベント履歴でのイベントの表示を参照してください。

Verified Access のイベントなど、AWS アカウント のイベントを継続的に記録するには、証跡を作成します。追跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォ

ルトでは、コンソールで証跡を作成するときに、証跡がすべての AWS リージョン に適用されます。証跡は、AWS パーティションのすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集されたイベントデータをより詳細に分析し、それを基にアクションを取るために他の AWS のサービス を設定できます。詳細については、次を参照してください。

- 追跡を作成するための概要
- CloudTrail がサポートされているサービスと統合
- CloudTrail の Amazon SNS 通知の設定
- 複数のリージョンから CloudTrail ログファイルを受け取るおよび複数のアカウントから CloudTrail ログファイルを受け取る

すべての Verified Access のアクションは CloudTrail によってログ記録され、[Amazon EC2 API リファレンス] に記録されます。例えば、`CreateVerifiedAccessInstance`、`DeleteVerifiedAccessInstance`、`ModifyVerifiedAcce` の各アクションを呼び出すと、CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。同一性情報は次の判断に役立ちます。

- リクエストが、ルートユーザーまたは AWS Identity and Access Management (IAM) ユーザーのどちらの認証情報を使用して送信されたかどうか。
- リクエストがロールまたはフェデレーションユーザーの一時的なセキュリティ認証情報を使用して行われたかどうか。
- リクエストが、別の AWS のサービス によって送信されたかどうか。

詳細については、「CloudTrail userIdentity 要素」を参照してください。

Verified Access のログファイルエントリを理解する

「トレイル」は、指定した Simple Storage Service (Amazon S3) バケットにイベントをログファイルとして配信するように設定できます。CloudTrail ログファイルには、1 つ以上のログエントリがあります。イベントは、任意の送信元からの単一のリクエストを表します。これには、リクエストされたアクション、アクションの日時、リクエストパラメータなどに関する情報が含まれます。CloudTrail ログファイルは、パブリック API コールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

次の例は、CreateVerifiedAccessInstance アクションの CloudTrail ログエントリを示しています。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAIKK4OOINJWEXAMPLE:jdoe",
        "arn": "arn:aws:iam::123456789012:user/jdoe",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "jdoe"
    },
    "eventTime": "2022-11-18T20:44:04Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "CreateVerifiedAccessInstance",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "console.amazonaws.com",
    "requestParameters": {
        "CreateVerifiedAccessInstanceRequest": {
            "Description": "",
            "ClientToken": "85893b1e-49f6-4d24-97de-280c664edf1b"
        }
    },
    "responseElements": {
        "CreateVerifiedAccessInstanceResponse": {
            "verifiedAccessInstance": {
                "creationTime": "2022-11-18T20:44:04",
                "description": "",
                "verifiedAccessInstanceId": "vai-0d79d91875542c549",
                "verifiedAccessTrustProviderSet": ""
            },
            "requestId": "2eae195d-6bfd-46d7-b46e-a68cb791de09"
        }
    },
    "requestID": "2eae195d-6bfd-46d7-b46e-a68cb791de09",
    "eventID": "297d6529-1144-40f6-abf8-3a76f18d88f0",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
```

```
}
```

AWS Verified Access のクォータ

AWS アカウント には、各 AWS のサービス にデフォルトのクォータ（以前は制限と呼ばれたもの）があります。特に明記されていない限り、クォータはリージョンごとに存在します。

AWS アカウント レベルのクォータ

お客様の AWS アカウント には、Verified Access に関連する以下のクォータがあります。

名前	デフォルト	引き上げ可能	説明
Verified Access インスタンス	5	はい	お客様が現在のリージョンで作成できる Verified Access インスタンスの最大数。
Verified Access グループ	10	はい	お客様が現在のリージョンで作成できる Verified Access グループの最大数。
Verified Access 信頼プロバイダー	15	はい	お客様が現在のリージョンで作成できる Verified Access 信頼プロバイダーの最大数。
Verified Access エンドポイント	50	はい	お客様が現在のリージョンで作成できる Verified Access エンドポイントの最大数。

HTTP ヘッダー

HTTP ヘッダーには次のようなサイズ制限があります。

名前	デフォルト	引き上げ可能
リクエスト行	16 K	いいえ
単一ヘッダー	16 K	いいえ

名前	デフォルト	引き上げ可能
レスポンスのヘッダー全体	32 K	いいえ
リクエストのヘッダー全体	64 K	いいえ

OIDC クレームサイズ

OIDC クレームサイズ制限は以下のとおりです。

名前	デフォルト	引き上げ可能
OIDC クレームサイズ	11 K	いいえ

「Verified Access ユーザーガイド」のドキュメント履歴

次の表は、「Verified Access」のドキュメントリリースの内容をまとめたものです。

変更	説明	日付
AWS マネージドポリシー更新	AWS Verified Access のマネージド IAM ポリシーが更新されました。	2023 年 11 月 17 日
保管時のデータ暗号化	AWS Verified Access は、デフォルトで、AWSが所有する KMS キーを使用して保管中のデータを暗号化します。	2023 年 9 月 28 日
FIPS コンプライアンスのサポート	FIPS に準拠するように Verified Access を設定します。	2023 年 9 月 26 日
高度なログ記録	ログにトラストコンテキストを追加するログ記録機能の追加。	2023 年 6 月 19 日
AWS マネージドポリシー更新	AWS Verified Access のマネージド IAM ポリシーが更新されました。	2023 年 5 月 31 日
GA リリース	「Verified Access ユーザーガイド」の GA リリース。AWS WAF 統合を含んでいます。	2023 年 4 月 27 日
プレビューリリース	「Verified Access ユーザーガイド」のプレビューリリース	2022 年 11 月 29 日

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