aws

ユーザーガイド

AWS 検証済みアクセス

AWS 検証済みアクセス: ユーザーガイド

Table of Contents

とは AWS Verified Access

を使用すると AWS Verified Access、仮想プライベートネットワーク (VPN) を使用することなく、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、 とサードパーティーサービスの両方を含む ID AWS およびデバイス管理サービスと統合されます。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。

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 マネジメントコンソール – 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。にサインイン AWS マネジメントコンソール し、https://console.aws.amazon.com/vpc/ で Amazon VPC コンソールを開きます。
- AWS Command Line Interface （ AWS CLI ） – AWS のサービスを含む、幅広い のセットのコマンドを提供します AWS Verified Access。 AWS CLI は、Windows、macOS、および Linux でサポートされています。を取得するには AWS CLI、「」を参照してくださいAWS Command Line Interface。

- AWS SDKs – 言語固有の APIs。 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、AWS SDK を参照してください。
- 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、 を使用して Verified Access リソース AWS マネジメントコンソール を作成、アクセス、管理する方法について説明します。

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、AWS Verified Access の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。

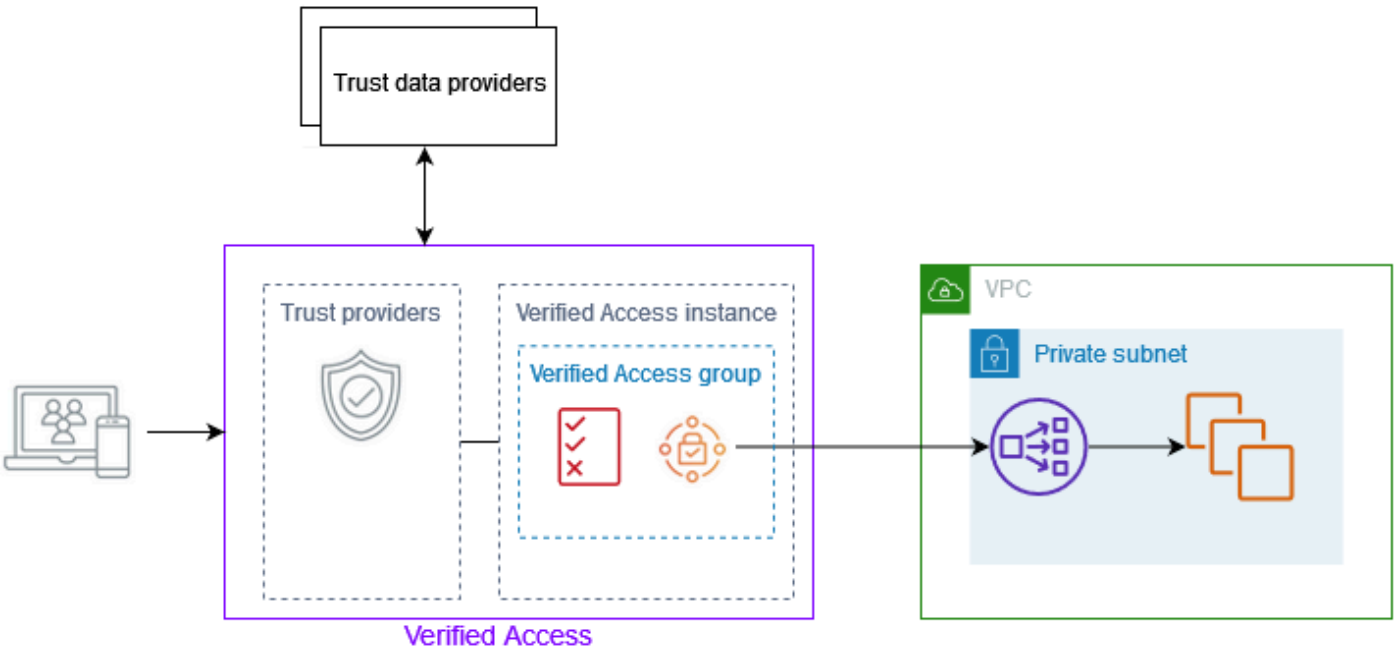
- 選択した信頼プロバイダー (AWS またはサードパーティー) によって送信された信頼データ。
- Verified Access で作成したアクセスポリシー。

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。上記の図では、アプリケーションはロードバランサーのターゲットである EC2 インスタンスでホストされています。
- 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 とサードパーティーの信頼プロバイダーの両方で動作します。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 の使用を開始する

の使用を開始するには、このチュートリアルを使用します AWS Verified Access。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。

このチュートリアルの一部として、Verified Access にアプリケーションを追加します。チュートリアルの終了時には、VPN を使用せずに、そのアプリケーションに特定のユーザーがインターネット経由でアクセスできるようになります。代わりに、 を ID 信頼プロバイダー AWS IAM アイデンティティセンター として使用します。このチュートリアルではデバイス信頼プロバイダーも使用しません。

タスク

Verified Access チュートリアルの前提条件

このチュートリアルを完了するための前提条件は次のとおりです。

- AWS IAM アイデンティティセンター は、作業 AWS リージョン している で有効になっています。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 アイデンティティセンター ユーザーガイド」の「有効化 AWS IAM アイデンティティセンター」を参照してください。
- アプリケーションへのアクセスを制御するセキュリティグループ。VPC CIDR からのすべてのインバウンドトラフィックと、すべてのアウトバウンドトラフィックが許可されていること。
- Elastic Load Balancing の内部ロードバランサーの背後で動作するアプリケーション。セキュリティグループがロードバランサーに関連付けられていること。

- 自己署名またはパブリック TLS 証明書 AWS Certificate Manager。キー長が 1,024 または 2,048 の RSA 証明書を使用してください。
- 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 Verified Access インスタンスの作成に必要なアクセス許可を持つ IAM ポリシー。詳細については、「Verified Access インスタンスを作成するためのポリシー」を参照してください。

ステップ 1: Verified Access 信頼プロバイダーを作成する

信頼プロバイダー AWS IAM アイデンティティセンター として を設定するには、次の手順に従います。

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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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 信頼プロバイダーの作成] を選択します。
4. (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。
5. 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。
6. [信頼プロバイダータイプ] で、[ユーザー信頼プロバイダー] を選択します。
7. [ユーザーの信頼プロバイダータイプ] で、[IAM アイデンティティセンター] を選択します。
8. [Verified Access 信頼プロバイダーの作成] を選択します。

ステップ 2： 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. 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. [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. [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 インスタンスの作成] を選択します。

ステップ 3：Verified Access グループを作成する

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. ナビゲーションペインで、[Verified Access グループ] を選択します。
2. [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] で、Verified Access インスタンスを選択します。
5. [ポリシー定義] は空白のままにします。後のステップでグループレベルのポリシーを追加します。
6. [Verified Access グループの作成] を選択します。

ステップ 4: Verified Access エンドポイントを作成する

以下の手順に従って、Verified Access エンドポイントを作成します。このステップでは、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が実行されていること、および AWS Certificate Managerにパブリックドメイン証明書があ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. 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。
2. [Verified Access エンドポイントの作成] を選択します。
3. (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。
4. [Verified Access グループ] では、Verified Access グループを選択します。
5. [エンドポイント詳細] では、次の操作を行います。

    a. [プロトコル] で、ロードバランサーの設定に応じて [HTTPS] または [HTTP] を選択します。
    b. [添付タイプ] で、[VPC] を選択します。
    c. [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。
    d. [ポート] に、ロードバランサーリスナーで使用されるポート番号を入力します。例えば、HTTPS の場合は 443、HTTP の場合は 80 になります。

e. [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。

f. [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。

g. [セキュリティグループ] で、セキュリティグループを選択します。ロードバランサーとエンドポイントに同じセキュリティグループを使用すると、それらの間のトラフィックが許可されます。同じセキュリティグループを使用しない場合は、ロードバランサーのエンドポイントセキュリティグループを参照して、エンドポイントからのトラフィックが受け入れられるようにしてください。

h. [エンドポイントドメインプレフィックス] には、カスタム ID を入力します。例えば、**my-ava-app**。このプレフィックスは、Verified Access が生成する DNS 名の前に付加されます。

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。このドメインは、ドメイン証明書内のドメインと一致する必要があります。

b. [ドメイン証明書の ARN] で、 AWS Certificate Managerにあるドメイン証明書の Amazon リソースネーム (ARN) を選択します。

7. [ポリシーの詳細] は空白のままにします。後のステップでグループレベルのアクセスポリシーを追加します。

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

ステップ 5: Verified Access エンドポイントの 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. 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。

2. エンドポイントを選択します。

3. [詳細] タブを選択します。

4. [エンドポイントドメイン] からドメインをコピーします。エンドポイントドメイン名は、例えば `my-ava-app.edge-1a2b3c4d5e6f7g.vai-1a2b3c4d5e6f7g.prod.verified-access.us-west-2.amazonaws.com` のようになります。

DNS プロバイダーの指示に従って CNAME レコードを作成します。アプリケーションのドメイン名をレコード名として使用し、Verified Access エンドポイントのドメイン名をレコード値として使用します。

ステップ 6: 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 のデフォルトの動作では、すべてのリクエストが拒否されます。グループにもエンドポイントにも Verified Access ポリシーを追加しなかったため、すべてのリクエストは拒否されます。

ステップ 7: Verified Access グループレベルのアクセスポリシーを追加する

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。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. 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. [ポリシーを有効にする] を有効にします。
5. IAM アイデンティティセンターのユーザーがアプリケーションにアクセスすることを許可するポリシーを入力します。例については「the section called “ポリシーの例”」を参照してください。
6. [Verified Access グループポリシーの変更] を選択します。
7. これでグループポリシーが設定されたため、前のステップのテストを繰り返して、リクエストが許可されることを確認します。リクエストが許可されると、IAM アイデンティティセンターのサインインページからサインインするように求められます。ユーザー名とパスワードを指定したら、アプリケーションにアクセスできます。

Verified Access リソースのクリーンアップ

このチュートリアルが完了したら、次の手順を使用して Verified Access リソースを削除します。

Verified Access リソースを削除するには

1. 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。エンドポイントを選択し、[アクション]、[Verified Access エンドポイントを削除] の順に選択します。
2. ナビゲーションペインで、[Verified Access グループ] を選択します。グループを選択し、[アクション]、[Verified Access グループを削除] の順に選択します。エンドポイントの削除処理が完了するまで待つ必要がある場合があります。
3. 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。インスタンスを選択し、[アクション]、[Verified Access 信頼プロバイダーをデタッチ] の順に選択します。信頼プロバイダーを選択し、[Verified Access 信頼プロバイダーをデタッチ] を選択します。
4. 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。信頼プロバイダーを選択し、[アクション]、[Verified Access 信頼プロバイダーを削除] の順に選択します。
5. 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。インスタンスを選択し、[アクション]、[Verified Access インスタンスを削除] の順に選択します。

Verified Access インスタンス

AWS Verified Access インスタンスは、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 AWS リソースです。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。

タスク

- Verified Access インスタンスの作成と管理
- Verified Access インスタンスの削除
- Verified Access を と統合する AWS WAF
- Verified Access の FIPS 準拠

Verified Access インスタンスの作成と管理

Verified Access インスタンスは、信頼プロバイダーと Verified Access グループを体系化するために使用します。Verified Access インスタンスを作成し、Verified Access に信頼プロバイダーをアタッチしたり、Verified Access から信頼プロバイダーをデタッチしたりするには、次の手順を使用します。

タスク

- Verified Access インスタンスの作成
- Verified Access インスタンスへの信頼プロバイダーのアタッチ
- Verified Access インスタンスからの信頼プロバイダーのデタッチ
- カスタムサブドメインを追加する

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。

コンソールを使用して Verified Access インスタンスを作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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. (ネットワーク CIDR エンドポイント) ネットワーク CIDR エンドポイントのカスタムサブドメインには、カスタムサブドメインを入力します。
5. (オプション) Verified Access が FIPS に準拠する必要がある場合は、連邦情報プロセス標準 (FIPS) を有効にするを選択します。
6. (オプション) Verified Access 信頼プロバイダーの場合は、Verified Access インスタンスにアタッチする信頼プロバイダーを選択します。
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てください。
8. [Verified Access インスタンスの作成] を選択します。

を使用して Verified Access インスタンスを作成するには AWS CLI

create-verified-access-instance コマンドを使用します。

Verified Access インスタンスへの信頼プロバイダーのアタッチ

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。

コンソールを使用して Verified Access インスタンスに信頼プロバイダーをアタッチするには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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てください。
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。

を使用して Verified Access インスタンスに信頼プロバイダーをアタッチするには AWS CLI

attach-verified-access-trust-provider コマンドを使用します。

Verified Access インスタンスからの信頼プロバイダーのデタッチ

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。

コンソールを使用して Verified Access インスタンスから信頼プロバイダーをデタッチするには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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てください。
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。

を使用して Verified Access インスタンスから信頼プロバイダーをデタッチするには AWS CLI

detach-verified-access-trust-provider コマンドを使用します。

カスタムサブドメインを追加する

カスタムサブドメインを追加または更新するには、次の手順に従います。このサブドメインは、ネットワーク CIDR エンドポイントを作成する場合にのみ使用されます。

コンソールを使用してカスタムサブドメインを追加するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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てください。
4. アクション、検証済みアクセスインスタンスの変更を選択します。
5. ネットワーク CIDR エンドポイントのカスタムサブドメインには、カスタムサブドメインを入力します。
6. Verified Access インスタンスの変更を選択します。
7. サブドメインのネームサーバーを更新し、Verified Access が提供するネームサーバーを入力します。このリストは、インスタンスの詳細タブの Nameservers で使用できます。

を使用してカスタムサブドメインを追加するには AWS CLI

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

Verified Access インスタンスの削除

不要になった 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。

コンソールを使用して Verified Access インスタンスを削除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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、[削除] を選択します。

を使用して Verified Access インスタンスを削除するには AWS CLI

delete-verified-access-instance コマンドを使用します。

Verified Access を と統合する AWS WAF

Verified Access によって適用される認証ルールと認可ルールに加えて、境界保護を適用することもできます。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。これを実現するには、Verified Access デプロイ AWS WAF に統合します。 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 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。詳細については、AWS WAF デベロッパーガイドを参照してください。

Verified Access AWS WAF と統合するには、 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ます。ウェブ ACL は、保護された AWS WAF リソースが応答するすべての HTTP ウェブリクエストをきめ細かく制御できるリソースです。 AWS WAF 関連付けまたは関連付け解除リクエストの処理中に、インスタンスにアタッチされた Verified Access エンドポイントのステータスが として表示されますupdating。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、 または でエンドポイントを記述 AWS マネジメントコンソール することで表示できます AWS CLI。

ユーザー ID 信頼プロバイダーは、 がトラフィックをいつ AWS WAF 検査するかを決定します。IAM Identity Center を使用する場合、 はユーザー認証の前にトラフィック AWS WAF を検査します。OpenID Connect (OIDC) を使用する場合、 AWS WAF はユーザー認証後にトラフィックを検査します。

内容

必要な IAM 許可

Verified Access AWS WAF との統合には、API オペレーションに直接対応しないアクセス許可のみのアクションが含まれます。このようなアクションは、 AWS Identity and Access Management の「サービス認可リファレンス」で [permission only] として示されています。「サービス認可リファレンス」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 プリンシパルに次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 の関連付け

次の手順は、Verified Access コンソールを使用して 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。

前提条件

開始する前に、 AWS WAF ウェブ ACL を作成します。詳細については、「AWS WAF デベロッパーガイド」の「ウェブ ACL の作成」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には

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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。

または、 AWS WAF コンソールを使用することもできます。 AWS WAF コンソールまたは API を使用する場合は、Verified Access インスタンスの Amazon リソースネーム (ARN) が必要です。AVA ARN は `arn:${Partition}:ec2:${Region}:${Account}:verified-access-instance/${VerifiedAccessInstanceId}` という形式になります。詳細については、「 AWS WAF デベロッパーガイド」の「ウェブ ACL を AWS リソースに関連付ける」を参照してください。

関連付けのステータスの確認

Verified Access コンソールを使用して、 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を確認できます。

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

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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、Verified Access コンソールを使用して Verified Access インスタンスから AWS WAF ウェブアクセスコントロールリスト (ACL) の関連付けを解除する方法を示しています。

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

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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。

または、 AWS WAF コンソールを使用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「 AWS リソースからウェブ ACL の関連付けを解除する」を参照してください。

Verified Access の FIPS 準拠

連邦情報処理規格 (FIPS) は、機密情報を保護する暗号化モジュールのセキュリティ要件を指定する米国およびカナダ政府の規格です。 は、FIPS 出版物 140-2 に準拠するように環境を設定するオプション AWS Verified Access を提供します。Verified Access の FIPS コンプライアンスは、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)
- AWS GovCloud (US) 西部
- AWS GovCloud (US) 東部

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。

内容

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設定するには、以下の手順に従います。

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 準拠の新しい AWS Verified Access 環境を設定するには、以下の手順に従います。

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 に送信するサービスです AWS Verified Access。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。

内容

Verified Access のユーザー ID 信頼プロバイダー

AWS IAM アイデンティティセンター または OpenID Connect 互換のユーザー ID 信頼プロバイダーのいずれかを使用できます。

内容

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 AWS IAM アイデンティティセンター として を使用できます。

前提条件と考慮事項

- IAM Identity Center インスタンスは AWS Organizations インスタンスである必要があります。スタンドアロン AWS アカウントの IAM Identity Center インスタンスは機能しません。

- IAM Identity Center インスタンスは、Verified Access 信頼プロバイダーを作成するリージョンと同じ AWS リージョンで有効にする必要があります。
- Verified Access は、最大 1,000 のグループに割り当てられている IAM アイデンティティセンターのユーザーにアクセスを提供できます。

さまざまなインスタンスタイプの詳細については、AWS IAM アイデンティティセンター 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 Identity Center を有効にしたら、次の手順を使用して、Verified Access の信頼プロバイダーとして IAM Identity Center を設定できます。

IAM Identity Center 信頼プロバイダーを作成するには (AWS コンソール)

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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てください。
8. [Verified Access 信頼プロバイダーの作成] を選択します。

IAM Identity Center 信頼プロバイダーを作成するには (AWS CLI)

- 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 Identity Center 信頼プロバイダーを削除するには (AWS コンソール)

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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. [削除] を選択します。

IAM Identity Center 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ています。Verified 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、潜在的な OIDC プロバイダーが多数存在するため、 AWS は Verified Access との各 OIDC 統合をテストできません。

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。

2025 年 2 月 24 日に作成された信頼プロバイダーでは、OIDC 信頼プロバイダーからの ID トークンクレームが`addition_user_context`キーに含まれます。

2025 年 2 月 24 日より前に作成された信頼プロバイダーでは、Verified Access は OIDC プロバイダーによって`ID token`送信された からの信頼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。

2025 年 2 月 24 日に作成された信頼プロバイダーの場合、デフォルトのセッション期間は 1 日です。2025 年 2 月 24 日より前に作成された信頼プロバイダーの場合、デフォルトのセッション期間は 7 日間です。

更新トークンが指定されている場合、Verified Access は更新トークンの有効期限をセッション期間として使用します。更新トークンがない場合は、デフォルトのセッション期間が使用されます。

内容

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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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. OIDC (OpenID Connect) の場合は、信頼プロバイダーを選択します。
8. [Issuer] には、OIDC 発行者の ID を入力します。

9. [認可エンドポイント] には、認可エンドポイントの完全な URL を入力します。
10. [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。
11. [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。
12. (ネイティブアプリケーション OIDC) パブリック署名キー URL には、パブリック署名キーエンドポイントの完全な URL を入力します。
13. [クライアント ID]に、OAuth 2.0 クライアント ID を入力します。
14. [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。
15. 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。openid スコープには、少なくともスコープが必要です。
16. (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てください。
17. [Verified Access 信頼プロバイダーの作成] を選択します。
18. 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。
    - HTTP アプリケーション – 次の URI を使用します: **`https://application_domain/oauth2/idpresponse`**。コンソールでは、Verified Access エンドポイントの詳細タブにアプリケーションドメインがあります。 AWS CLI または AWS SDK を使用すると、Verified Access エンドポイントを記述するときに、アプリケーションドメインが出力に含まれます。
    - TCP アプリケーション – 次の URI を使用します: **`http://localhost:8000`**。

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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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 信頼プロバイダーを変更するには (AWS CLI)

- 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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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. [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)

Verified Access のデバイスベースの信頼プロバイダー

AWS Verified Access で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。

内容

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 AWS Verified Access による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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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 AWS マネジメントコンソール、Verified Access エンドポイントの詳細タブの 、または AWS CLI を使用してエンドポイントを記述することで確認できます。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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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

Verified Access グループは、複数の Verified Access エンドポイントと、グループ内のすべてのエンドポイントに適用される 1 つの Verified Access ポリシーで構成されます。共通のセキュリティ要件を持つエンドポイントをグループ化して、複数のエンドポイントの最小セキュリティ要件を満たす単一のグループポリシーを定義できます。このため、エンドポイントごとにポリシーを作成して維持する必要がなくなります。

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。

Verified Access グループのもう 1 つの機能は、 を使用して他のAWSアカウントと共有できることですAWS RAM。これにより、1 つのアカウントでグループの作成と管理を一元的に行い、それらのグループを複数のアカウントと共有することができます。

タスク

- Verified Access グループの作成と管理
- Verified Access グループポリシーの変更
- Verified Access グループを別のグループと共有するAWS アカウント
- Verified Access グループを削除する

Verified Access グループの作成と管理

Verified Access グループを使用して、セキュリティ要件に基づいてエンドポイントを整理します。Verified Access エンドポイントを作成するときは、エンドポイントをグループと関連付けます。

タスク

- Verified Access グループの作成
- Verified Access グループを変更する

Verified Access グループの作成

Verified Access グループを作成するには、次の手順に従います。Verified Access グループを作成する前に、Verified Access インスタンスを作成する必要があります。詳細については、「the section called “Verified Access インスタンスの作成”」を参照してください。

コンソールを使用して Verified Access グループを作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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てください。
7. [Verified Access グループの作成] を選択します。

を使用して Verified Access グループを作成するにはAWS CLI

create-verified-access-group コマンドを使用します。

Verified Access グループを変更する

Verified Access グループを変更するには、次の手順に従います。

コンソールを使用して Verified Access グループを変更するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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. グループを選択し、アクション、検証済みアクセスグループの変更を選択します。
4. (オプション) 説明を更新します。
5. [Verified Access グループの作成] を選択します。

6. グループに関連付ける Verified Access インスタンスを選択します。

を使用して Verified Access グループを変更するにはAWS CLI

modify-verified-access-group コマンドを使用します。

Verified Access グループポリシーの変更

AWS Verified Accessは、作成したアクセスポリシーに基づいてアプリケーションへのアクセスを許可します。グループにアタッチした Verified Access 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特定のエンドポイントにアタッチできます。

Verified Access グループのポリシーを変更するには、次の手順に従います。変更を行った後、変更が有効になるまでには数分かかります。

コンソールを使用して Verified Access グループポリシーを変更するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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 グループポリシーの変更] を選択します。
5. (オプション) 必要に応じて [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、グループに適用する Verified Access ポリシーを入力します。
7. [Verified Access グループポリシーの変更] を選択します。

を使用して Verified Access グループポリシーを変更するにはAWS CLI

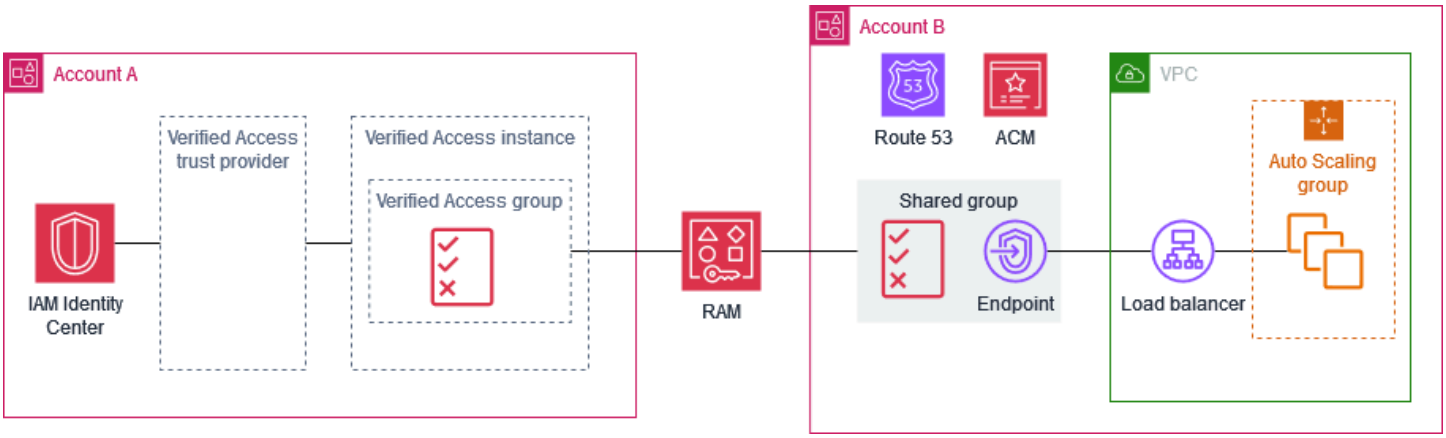
modify-verified-access-group-policy コマンドを使用します。

Verified Access グループを別のグループと共有するAWS アカウント

所有している Verified Access グループを他のAWSアカウントと共有する場合、それらのアカウントがグループに Verified Access エンドポイントを作成できるようにします。Verified Access グルー

プを作成したアカウントは、所有者アカウントと呼ばれます。共有グループを使用するアカウントは、コンシューマーアカウントと呼ばれます。

以下の図は、Verified Access グループを共有する利点を示しています。中央セキュリティチームはアカウント A を所有しています。 のユーザーとグループを管理しAWS IAM アイデンティティセンター、Verified Access 信頼プロバイダー、Verified Access インスタンス、Verified Access グループ、Verified Access ポリシーなどの内部アプリケーションへのアクセスを提供するために必要な Verified Access リソースを管理します。アプリケーションチームはアカウント B を所有しています。ロードバランサー、Auto Scaling グループ、Amazon Route 53 の DNS 設定、 AWS Certificate Manager(ACM) の TLS 証明書など、内部アプリケーションの実行に必要なリソースを管理します。中央セキュリティチームが Verified Access グループをアカウント B と共有したら、アプリケーションチームは、共有されたグループを使用して Verified Access エンドポイントを作成できます。アプリケーションへのアクセスは、中央セキュリティチームが Verified Access グループに作成したポリシーに基づいて許可または拒否されます。



考慮事項

共有 Verified Access グループには、以下の考慮事項が適用されます。

Owners

- Verified Access グループを共有するには、ユーザーに `ec2:PutResourcePolicy` および `ec2:DeleteResourcePolicy` アクセス許可が必要です。
- Verified Access グループを共有するには、そのグループを所有している必要があります。自分に共有された Verified Access グループを共有することはできません。
- 組織内のアカウントとの共有を有効にすると、Verified Access グループなどのリソースを、招待を使用せずに共有することができます。有効にしない場合、コンシューマーは招待を受け取り、共有グループにアクセスするには招待を受け入れる必要があります。共有を有効にするには、組

織の管理アカウントから、AWS RAMコンソールの設定ページを開き、共有を有効にするAWS Organizationsを選択します。

- 関連付けられた Verified Access エンドポイントがある場合は、グループを削除することはできません。コンシューマーアカウントによって作成されたエンドポイントは、所有者アカウントの [Verified Access エンドポイント] ページで確認できます。エンドポイントの所有者のアカウント ID は、エンドポイントの証明書の Amazon リソースネーム (ARN) に反映されます。

コンシューマー

- 自分と共有されている Verified Access グループを確認するには、コンソールで [Verified Access グループ] ページを開くか、describe-verified-access-groups を呼び出します。所有者のアカウント ID は、[所有者] フィールドとグループの Amazon リソースネーム (ARN) に反映されます。
- Verified Access エンドポイントを作成するときは、自分に共有された Verified Access グループをどれでも指定できます。
- 共有グループに関連付けられているエンドポイントのうち、自分が所有していないエンドポイントは表示できません。
- Verified Access グループの所有者がリソース共有を削除した場合、グループ内に新しい Verified Access エンドポイントを作成することはできません。リソース共有を削除する前に作成した Verified Access エンドポイントは、リソース共有の削除の影響を受けません。ただし、共有グループの所有者はコンシューマーのエンドポイントを削除できます。

リソース共有

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。リソース共有は、共有するリソースと、共有されたリソースを使用できるコンシューマーを指定するものです。

コンソールを使用して Verified Access グループを共有するには

1. https://console.aws.amazon.com/ram/home でAWS RAMコンソールを開きます。
2. 組織にリソース共有がない場合は、リソース共有を作成します。プリンシパルでは、組織全体、組織単位、または特定のAWSアカウントを選択できます。
3. リソース共有を選択し、[変更] を選択します。
4. `Resources` で、リソースタイプとして [Verified Access グループ] を選択し、共有するリソースグループを選択します。
5. [確認と更新にスキップ] を選択します。

6. [リソース共有を更新] を選択します。

詳細については、「AWS RAM ユーザーガイド」の「Create a resource share」を参照してください。

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。関連付けられた Verified Access エンドポイントがある場合は、グループを削除することはできません。

コンソールを使用して Verified Access グループを削除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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、[削除] を選択します。

を使用して Verified Access グループを削除するにはAWS CLI

delete-verified-access-group コマンドを使用します。

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。

内容

Verified Access エンドポイントタイプ

Verified Access エンドポイントには次のタイプがあります。

- 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。詳細については、「ロードバランサーエンドポイントの作成」を参照してください。
- 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。詳細については、「ネットワークインターフェイスエンドポイントの作成」を参照してください。
- ネットワーク CIDR – アプリケーションリクエストは、指定された CIDR ブロックに送信されます。詳細については、「ネットワーク CIDR エンドポイントを作成する」を参照してください。
- Amazon Relational Database Service (RDS) – アプリケーションリクエストは、RDS インスタンス、RDS クラスター、または RDS DB プロキシに送信されます。詳細については、「Amazon Relational Database Service エンドポイントを作成する」を参照してください。

Verified Access と共有 VPC およびサブネットの動作

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 ローカルゾーンで作成できないため、ローカルゾーンを介した共有はできません。

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

Verified Access のロードバランサーエンドポイントを作成するには、次の手順に従います。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。
- WebSocket 接続などの存続期間の長い HTTPS 接続は、TCP を介してのみサポートされます。
- 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。キー長が 1,024 または 2,048 の RSA 証明書を使用してください。
- Verified Access エンドポイントを作成する前に、Verified Access グループを作成する必要があります。詳細については、「the section called “Verified Access グループの作成”」を参照してください。
- 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。を使用して証明書を作成またはインポートできます AWS Certificate Manager。

コンソールを使用して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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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、Verified Access グループを選択します。
6. [エンドポイント詳細] では、次の操作を行います。
   a. Protocol で、プロトコルを選択します。
   b. [添付タイプ] で、[VPC] を選択します。
   c. [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。
   d. (HTTP/HTTPS) ポートには、ポート番号を入力します。(TCP) ポート範囲にポート範囲を入力し、ポートの追加を選択します。
   e. ロードバランサー ARN で、ロードバランサーを選択します。
   f. Subnet で、サブネットを選択します。アベイラビリティーゾーンごとに 1 つだけサブネットを指定できます。
   g. [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。これらのセキュリティグループは、Verified Access エンドポイントのインバウンドトラフィックとアウトバウンドトラフィックを制御します。
   h. [エンドポイント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。
7. (HTTP/HTTPS) アプリケーションの詳細については、以下を実行します。
   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。
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てください。
10. [Verified Access エンドポイントの作成] を選択します。

を使用して Verified Access エンドポイントを作成するには AWS CLI

create-verified-access-endpoint コマンドを使用します。

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。
- 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。
- Verified Access エンドポイントを作成する前に、Verified Access グループを作成する必要があります。詳細については、「the section called “Verified Access グループの作成”」を参照してください。
- 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。を使用して証明書を作成またはインポートできます AWS Certificate Manager。

コンソールを使用し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するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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、Verified Access グループを選択します。
6. [エンドポイント詳細] では、次の操作を行います。
    a. Protocol で、プロトコルを選択します。
    b. [添付タイプ] で、[VPC] を選択します。
    c. [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。
    d. (HTTP/HTTPS) ポートには、ポート番号を入力します。(TCP) ポート範囲にポート範囲を入力し、ポートの追加を選択します。
    e. 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。

f. [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。これらのセキュリティグループは、Verified Access エンドポイントのインバウンドトラフィックとアウトバウンドトラフィックを制御します。

g. [エンドポイント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。

7. (HTTP/HTTPS) アプリケーションの詳細については、以下を実行します。

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。

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てください。

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

を使用して Verified Access エンドポイントを作成するには AWS CLI

create-verified-access-endpoint コマンドを使用します。

Verified Access 用のネットワーク CIDR エンドポイントを作成する

ネットワーク CIDR エンドポイントを作成するには、次の手順に従います。たとえば、ネットワーク CIDR エンドポイントを使用して、ポート 22 (SSH) 経由で特定のサブネット内の EC2 インスタンスへのアクセスを有効にすることができます。

要件

- TCP プロトコルのみがサポートされています。
- Verified Access は、リソースで使用される CIDR 範囲内の各 IP アドレスの DNS レコードを提供します。リソースを削除すると、その IP アドレスは使用されなくなり、Verified Access は対応する DNS レコードを削除します。
- カスタムサブドメインを指定した場合、Verified Access は、指定された CIDR 範囲内にあり、サブドメインで使用されるエンドポイントサブネット内の各 IP アドレスの DNS レコードを提供し、その DNS サーバーの IP アドレスを提供します。Verified Access DNS サーバーを指すようにサブドメインの転送ルールを設定できます。ドメイン内のレコードに対して行われたリクエスト

は、Verified Access DNS サーバーによって、リクエストされたリソースの IP アドレスに解決されます。

- Verified Access エンドポイントを作成する前に、Verified Access グループを作成する必要があります。詳細については、「the section called “Verified Access グループの作成”」を参照してください。
- エンドポイントを作成し、 を使用してアプリケーションに接続します接続クライアント。

コンソールを使用してネットワーク CIDR エンドポイントを作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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. [エンドポイント詳細] では、次の操作を行います。
   a. [プロトコル] で [TCP] を選択します。
   b. [添付タイプ] で、[VPC] を選択します。
   c. エンドポイントタイプで、ネットワーク CIDR を選択します。
   d. ポート範囲にポート範囲を入力し、ポートの追加を選択します。
   e. Subnet で、サブネットを選択します。
   f. [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。これらのセキュリティグループは、Verified Access エンドポイントのインバウンドトラフィックとアウトバウンドトラフィックを制御します。
   g. (オプション) エンドポイントドメインプレフィックスに、Verified Access がエンドポイントに対して生成する DNS 名の前に追加するカスタム識別子を入力します。
7. (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。
8. (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てください。
9. [Verified Access エンドポイントの作成] を選択します。

を使用して Verified Access エンドポイントを作成するには AWS CLI

create-verified-access-endpoint コマンドを使用します。

Verified Access 用の Amazon Relational Database Service エンドポイントを作成する

Amazon Relational Database Service (RDS) エンドポイントを作成するには、次の手順に従います。

要件

- TCP プロトコルのみがサポートされています。
- RDS インスタンス、RDS クラスター、または RDS DB プロキシを作成します。
- Verified Access エンドポイントを作成する前に、Verified Access グループを作成する必要があります。詳細については、「the section called “Verified Access グループの作成”」を参照してください。
- エンドポイントを作成し、 を使用してアプリケーションに接続します接続クライアント。

コンソールを使用して Amazon Relational Database Service エンドポイントを作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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. [エンドポイント詳細] では、次の操作を行います。
    a. [プロトコル] で [TCP] を選択します。
    b. [添付タイプ] で、[VPC] を選択します。
    c. エンドポイントタイプで、Amazon Relational Database Service (RDS) を選択します。
    d. RDS ターゲットタイプの場合は、次のいずれかを実行します。
        - RDS インスタンスを選択し、RDS インスタンスから RDS インスタンスを選択します。
        - RDS クラスターを選択し、RDS クラスターから RDS クラスターを選択します。
        - RDS DB プロキシを選択し、RDS DB プロキシから RDS DB プロキシを選択します。
    e. RDS エンドポイントで、前のステップで選択した RDS リソースに関連する RDS エンドポイントを選択します。

f. [ポート] に、ポート番号を入力します。

g. Subnet で、サブネットを選択します。アベイラビリティーゾーンごとに 1 つだけサブネットを指定できます。

h. [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。これらのセキュリティグループは、Verified Access エンドポイントのインバウンドトラフィックとアウトバウンドトラフィックを制御します。

i. (オプション) エンドポイントドメインプレフィックスに、Verified Access がエンドポイントに対して生成する DNS 名の前に追加するカスタム識別子を入力します。

7. (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。
8. (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てください。
9. [Verified Access エンドポイントの作成] を選択します。

を使用して Verified Access エンドポイントを作成するには AWS CLI

create-verified-access-endpoint コマンドを使用します。

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は維持することをお勧めします。

コンソールを使用して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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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. [ルールを追加] を選択してください。
   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. [ルールの保存] を選択します。

を使用して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 AWS CLI

describe-verified-access-endpoints コマンドを使用してセキュリティグループの ID を取得し、authorize-security-group-ingress コマンドを使用してインバウンドルールを追加します。

Verified Access エンドポイントの変更

Verified Access エンドポイントを変更するには、次の手順に従います。

コンソールを使用して Verified Access エンドポイントを変更するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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。

を使用して Verified Access エンドポイントを変更するには AWS CLI

modify-verified-access-endpoint コマンドを使用します。

Verified Access エンドポイントポリシーの変更

Verified Access エンドポイントのポリシーを変更するには、次の手順に従います。変更を行った後、変更が有効になるまでには数分かかります。

コンソールを使用して Verified Access エンドポイントポリシーを変更するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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 エンドポイントポリシーの変更] を選択します。
5. (オプション) 必要に応じて [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、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。

を使用して Verified Access エンドポイントポリシーを変更するには AWS CLI

modify-verified-access-endpoint-policy コマンドを使用します。

Verified Access エンドポイントの削除

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

コンソールを使用して Verified Access エンドポイントを削除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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。

を使用して Verified Access エンドポイントを削除するには AWS CLI

delete-verified-access-endpoint コマンドを使用します。

信頼プロバイダーから Verified Access に送信されるトラストデータ

信頼データは、信頼プロバイダー AWS Verified Access から に送信されるデータです。トラストデータは、「ユーザークレーム」または「信頼コンテキスト」とも呼ばれます。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。トラストデータの例には、ユーザーの E 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、最新のトラストデータの完全なリストについては、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ときに役立つ可能性があります。ログにトラストコンテキストを含める方法については、Verified Access 信頼コンテキストの有効化または無効化 を参照してください。

このセクションでは、ポリシーの作成を開始するときに役立つ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。

内容

Verified Access トラストデータのデフォルトコンテキスト

AWS Verified Access には、設定された信頼プロバイダーに関係なく、すべての Cedar 評価に現在のリクエストに関するいくつかの要素がデフォルトで含まれています。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。

以下は、評価に含まれるデータの例です。

例

- HTTP リクエスト
- TCP フロー

HTTP リクエスト

ポリシーが評価されると、Verified Access は `context.http_request`キーの Cedar コンテキストに現在の HTTP リクエストに関するデータを含めます。

```
{
    "title": "HTTP Request data included by Verified Access",
    "type": "object",
    "properties": {
        "http_method": {
            "type": "string",
            "description": "The HTTP method",
            "example": "GET"
        },
        "hostname": {
            "type": "string",
            "description": "The host subcomponent of the authority component of the
 URI",
            "example": "example.com"
        },
        "path": {
            "type": "string",
            "description": "The path component of the URI",
            "example": "app/images"
        },
        "query": {
            "type": "string",
            "description": "The query component of the URI",
            "example": "value1=1&value2=2"
        },
        "x_forwarded_for": {
            "type": "string",
            "description": "The value of the X-Forwarded-For request header",
            "example": "17.7.7.1"
        },
        "port": {
           "type": "integer",
           "description": "The endpoint port",
           "example": 443
```

```
        },
        "user_agent": {
            "type": "string",
            "description": "The value of the User-Agent request header",
            "example": "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:47.0)
 Gecko/20100101 Firefox/47.0"
        },
        "client_ip": {
            "type": "string",
            "description": "The IP address connecting to the endpoint",
            "example": "15.248.6.6"
        }
    }
}
```

ポリシーの例

以下は、HTTP リクエストデータを使用する Cedar ポリシーの例です。

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

TCP フロー

ポリシーが評価されると、Verified Access は`context.tcp_flow`キーの下の Cedar コンテキストに現在の TCP フローに関するデータを含めます。

```
{
    "title": "TCP flow data included by Verified Access",
    "type": "object",
    "properties": {
        "destination_ip": {
            "type": "string",
            "description": "The IP address of the target",
            "example": "192.100.1.3"
        },
        "destination_port": {
            "type": "string",
            "description": "The target port",
            "example": 22
```

```
        },
        "client_ip": {
            "type": "string",
            "description": "The IP address connecting to the endpoint",
            "example": "172.154.16.9"
        }
    }
}
```

AWS IAM アイデンティティセンター Verified Access 信頼データのコンテキスト

ポリシーが評価されると、 を信頼プロバイダー AWS IAM アイデンティティセンター として定義すると、 AWS Verified Access は、信頼プロバイダー設定で「ポリシー参照名」として指定したキーの Cedar コンテキストに信頼データを含めます。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。

Verified Access トラストデータのサードパーティー信頼プロバイダーのコンテキスト

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 AWS Verified Access から に提供される信頼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。

内容

ブラウザ拡張

アクセスポリシーにデバイスの信頼コンテキストを組み込む場合は、 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です。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 信頼データを使用している場合、ユーザーは AWS Verified Access ブラウザ拡張機能を Chrome ウェブストアまたは Firefox アドオンサイトからデバイスにダウンロードして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、ユーザーはまず AWS Verified Access Native Messaging Host (直接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 アドオンサイトから AWS 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。

Verified 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 Device Identity との統合」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。

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「Securing private applications with CrowdStrike and AWS Verified Access」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 environment"
    },
    "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。

JumpCloud と AWS Verified Access の使用の詳細については、JumpCloud ウェブサイトの JumpCloud と AWS Verified Access の統合」を参照してください。 JumpCloud

```
{
  "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_organization_identifier'
};
```

Verified Access でのユーザークレームの引き渡しと署名の検証

AWS Verified Access インスタンスがユーザーを正常に認証すると、IdP から受信し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を検証するとともに、そのクレームが Verified Access から送信されたものであることを検証できます。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。

内容

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"
   ],
   "additional_user_context": {
      "aud": "xxx",
```

```
      "exp": 1000000000,
      "groups": [
         "group-id-1",
         "group-id-2"
      ],
      "iat": 1000000000,
      "iss": "https://oidc-tp.com/",
      "sub": "xyzsubject",
      "ver": "1.0"
   }
}
```

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。

セキュリティを確保するには、クレームに基づいて認可を行う前に署名を検証し、JWT ヘッダーの `signer` フィールドに、想定される Verified Access インスタンス ARN が含まれていることを確認する必要があります。

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。

それぞれのエンドポイント AWS リージョン 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: Validate the signer
expected_verified_access_instance_arn = 'arn:aws:ec2:region-code:account-id:verified-
access-instance/verified-access-instance-id'

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)
received_verified_access_instance_arn = decoded_json['signer']

assert expected_verified_access_instance_arn == received_verified_access_instance_arn,
 "Invalid Signer"

# Step 2: Get the key id from JWT headers (the kid field)
kid = decoded_json['kid']

# Step 3: 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 4: Get the payload
payload = jwt.decode(encoded_jwt, pub_key, algorithms=['ES384'])
```

Verified Access ポリシー

AWS Verified Access ポリシーを使用すると、 で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。 AWS ポリシー言語である Cedar で記述されます。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データ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。

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。Verified Access グループまたはエンドポイントを作成してから、既存のグループやエンドポイントのポリシーを追加または変更することもできます。

内容

Verified Access ポリシーステートメントの構造

次の表は、Verified Access ポリシーの構造を示しています。

コンポーネント	構文
effect	`permit | forbid`
scope	`(principal, action, resource)`
条件句	`when { context.policy-reference-name .attribute-name`

コンポーネント	構文
	};

ポリシーの構成要素

Verified Access ポリシーには、次の構成要素が含まれています。

- 効果 - アクセスの `permit` (許可) または `forbid` (拒否)。
- スコープ - 効果を適用するプリンシパル、アクション、リソース。特定のプリンシパル、アクション、リソースを指定しないでおくと、Cedar のスコープを未定義のままにすることができます。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。
- 条件句 - 効果が適用されるコンテキスト。

> **Important**
>
> Verified Access では、条件節に含まれるトラストデータを参照することでポリシーが完全に表現されます。ポリシーのスコープは、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。

コメント

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

以下の例は、ポリシー内のコメントを示しています。

```
// grants access to users in a specific domain using 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)
};
```

複数の句

&& 演算子を使用すると、1 つのポリシーステートメントに複数の条件句を含めることができます。

```
permit(principal,action,resource)
when{
 context.policy-reference-name.attribute1 &&
 context.policy-reference-name.attribute2
};
```

その他の例については、「Verified Access ポリシーの例」を参照してください。

予約文字

次の例は、コンテキストのプロパティでポリシー言語の予約文字である：(セミコロン) が使用され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")
};
```

Verified Access ポリシーのビルトイン演算子

「」で説明されているように、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Verified Access ポリシーステートメントの構造、 &&演算子を使用して条件を追加できます。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。異なるタイプの値が互いに等しくなることはありません。

演算子	タイプとオーバーロード	説明
!=	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.
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。

演算子	タイプとオーバーロード	説明
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。

Verified Access 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。

AWS Verified Access はポリシーの作成時に構文を検証しますが、条件句に入力したデータは検証しません。

Verified Access ポリシーロジックでの問題回避

特定のコンテキストに存在するかどうかにかかわらず、データを評価する AWS Verified Access ポリシーを記述できます。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できます。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)
    )
```

```
  )
};
```

Verified Access ポリシーの例

Verified Access ポリシーを使用して、特定のユーザーとデバイスにアプリケーションへのアクセス権を付与できます。

ポリシーの例

- 例 1: IAM アイデンティティセンターのグループにアクセス権を付与する
- 例 2: サードパーティープロバイダーのグループにアクセス権を付与する
- 例 3: CrowdStrike を使用してアクセス権を付与する
- 例 4 : 特定の IP アドレスを許可または拒否する

例 1: IAM アイデンティティセンターのグループにアクセス権を付与する

を使用する場合は AWS IAM アイデンティティセンター、IDs を使用してグループを参照することをお勧めします。これにより、グループの名前を変更した場合にポリシーステートメントが機能しなくなるのを防ぐことができます。

次のポリシー例では、指定されたグループに所属する、検証済みの E メールアドレスを持つユーザーにのみアクセスを許可します。グループ ID は c242c5b0-6081-1845-6fa8-6e0d9513c107 です。

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

次のポリシー例では、ユーザーが指定のグループに所属し、検証済みの E 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"
```

```
};
```

トラストデータの詳細については、「the section called “AWS IAM アイデンティティセンター コンテキスト”」を参照してください。

例 2: サードパーティープロバイダーのグループにアクセス権を付与する

次のポリシー例では、ユーザーが指定のグループに所属し、検証済みの E 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。グループの名前は「finance」です。

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

トラストデータの詳細については、「the section called “サードパーティーのコンテキスト”」を参照してください。

例 3: CrowdStrike を使用してアクセス権を付与する

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：特定の IP アドレスを許可または拒否する

次のポリシー例では、指定された IP アドレスからの HTTP リクエストを許可します。

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

次のポリシー例では、指定された IP アドレスからの HTTP リクエストを拒否します。

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

次のポリシー例では、指定された IP アドレスからの TCP リクエストを許可します。

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

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 ログから最新の認可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、認可決定、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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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、使用する最近の認可結果の種類を選択します。デフォルトでは、最新の認可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] ボタンを選択して信頼コンテキストに対してポリシーをテストすると、画面に認可結果が表示されます。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。

の接続クライアント AWS Verified Access

AWS Verified Access は Connectivity Client を提供するため、ユーザーデバイスと非 HTTP アプリケーション間の接続を有効にできます。クライアントは、ユーザートラフィックを安全に暗号化し、ユーザー ID 情報とデバイスコンテキストを追加し、ポリシーの適用のために Verified Access にルーティングします。アクセスポリシーでアクセスが許可されている場合、ユーザーはアプリケーションに接続されます。Connectivity Client が接続されている限り、ユーザーアクセスは継続的に承認されます。

クライアントはシステムサービスとして実行され、クラッシュに対して回復力があります。接続が不安定になると、クライアントは接続を再確立します。

クライアントは、エフェメラル OAuth アクセストークンを使用してセキュアトンネルを確立します。トンネルは、ユーザーがクライアントからサインアウトすると切断されます。

アクセストークンと更新トークンは、暗号化された SQLite データベースのユーザーデバイスにローカルに保存されます。

内容

前提条件

開始する前に、次の前提条件を完了します。

- 信頼プロバイダーを使用して Verified Access インスタンスを作成します。
- アプリケーションの TCP エンドポイントを作成します。
- ルーティングの問題を回避するために、VPN クライアントからコンピュータを切断します。

- コンピュータで IPv6 を有効にします。手順については、コンピュータで実行されているオペレーティングシステムのドキュメントを参照してください。
- Windows コンピュータで、トラステッドプラットフォームモジュール (TPM) がサポートされていることを確認し、WebView2 ランタイムをインストールします。

接続クライアントをダウンロードする

以前のバージョンのクライアントをアンインストールします。クライアントをダウンロードし、インストーラが署名されていることを確認し、インストーラを実行します。署名なしインストーラを使用してクライアントをインストールしないでください。

- Apple Silicon バージョン 1.0.4 を使用した Mac 用 Connectivity Client
- Intel バージョン 1.0.4 を搭載した Mac 用 Connectivity Client
- x64 バージョン 1.0.6 を使用する Windows 用 Connectivity Client

クライアント設定ファイルをエクスポートする

Verified Access インスタンスからクライアントに必要な設定情報をエクスポートするには、次の手順に従います。

コンソールを使用してクライアント設定ファイルをエクスポートするには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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、エクスポートクライアント設定ファイルを選択します。

を使用してクライアント設定ファイルをエクスポートするには AWS CLI

export-verified-access-instance-client-configuration コマンドを使用します。出力を .json ファイルに保存します。ファイル名は `ClientConfig-` プレフィックスで始まる必要があります。

アプリケーションに接続する

クライアントを使用してアプリケーションに接続するには、次の手順に従います。

クライアントを使用してアプリケーションに接続するには

1. 次の場所にあるユーザーのデバイスにクライアント設定ファイルをデプロイします。
   - Windows – `C:\ProgramData\Connectivity Client`
   - macOS – `/Library/Application\ Support/Connectivity\ Client`
2. クライアント設定ファイルが root (macOS) または Admin (Windows) によって所有されていることを確認します。
3. Connectivity Client を起動します。
4. Connectivity Client がロードされると、ユーザーは IdP によって認証されます。
5. 認証後、ユーザーは Verified Access が提供する DNS 名を使用して、選択したクライアントを使用してアプリケーションにアクセスできます。

クライアントをアンインストールする

Connectivity Client の使用が終了したら、アンインストールできます。

macOS

バージョン 1.0.1 以降

`/Applications/Connectivity Client` に移動して `Connectivity Client Uninstaller.app` を実行します。

バージョン 1.0.0

Mac with Apple Silicon または Mac with Intel の`connectivity_client_cleanup.sh`スクリプトをダウンロードし、スクリプトの実行許可を設定し、次のようにスクリプトを実行します。

```
sudo ./connectivity_client_cleanup.sh
```

Windows

Windows でクライアントをアンインストールするには、インストーラを実行し、削除を選択します。

ベストプラクティス

以下のベストプラクティスを考慮します。

- クライアントの最新バージョンをインストールします。
- 署名なしインストーラを使用してクライアントをインストールしないでください。
- IT 管理者が提供する信頼できる設定でない限り、ユーザーは設定を使用しないでください。信頼できない設定は、フィッシングページにリダイレクトされる可能性があります。
- ユーザーは、ワークステーションをアイドル状態のままにする前に、クライアントからサインアウトする必要があります。
- OIDC 設定に`offline_access`スコープを追加します。これにより、更新トークンのリクエストが許可されます。更新トークンは、ユーザーが再認証しなくても、より多くのアクセストークンを取得するために使用します。

トラブルシューティング

以下の情報は、クライアントに関する問題のトラブルシューティングに役立ちます。

問題

- サインインすると、ブラウザが開いて IdP による認証を完了しません
- 認証後、クライアントのステータスは「接続されていません」になります。
- Chrome または Edge ブラウザを使用して接続できない

サインインすると、ブラウザが開いて IdP による認証を完了しません

考えられる原因: 設定ファイルがないか、形式が正しくありません。

解決策: システム管理者に連絡して、更新された設定ファイルをリクエストします。

認証後、クライアントのステータスは「接続されていません」になります。

考えられる原因: 、Cisco AnyConnect AWS Client VPN、OpenVPN Connect などの他の VPN ソフトウェアの実行。

解決策: 他の VPN ソフトウェアから切断します。それでも接続できない場合は、診断レポートを生成し、システム管理者と共有します。

考えられる原因: Windows プラットフォームでは、クライアントはコントロールプレーン通信にポート 80 で HTTP を使用します。TCP ポート 80 をブロックするファイアウォールルールは、コントロールプレーンの通信を防ぎます。

解決策: Windows Firewall ルールでポート 80 で TCP をブロックする明示的なアウトバウンドルールを確認し、無効にします。

Chrome または Edge ブラウザを使用して接続できない

考えられる原因: Chrome または Edge ブラウザを使用してウェブアプリケーションに接続すると、ブラウザは IPv6 ドメイン名の解決に失敗します。

解決策: にお問い合わせくださいAWS サポート。

バージョン履歴

次の表に、 クライアントのバージョン履歴を示します。

バージョン	変更	ダウンロード	日付
1.0.6	Windows • 軽微なバグを修正	• x64 を使用する Windows	2026 年 6 月 1 日
1.0.5	Windows • 軽微なバグを修正	• x64 を使用する Windows	2026 年 4 月 20 日
1.0.4	macOS • 軽微なバグを修正	• Mac と Apple Silicon • インテル搭載 Mac	2026 年 4 月 9 日
1.0.4	Windows • 軽微なバグを修正	• x64 を使用する Windows	2026 年 2 月 10 日
1.0.3	macOS • 軽微なバグを修正	• Mac と Apple Silicon • インテル搭載 Mac	2026 年 1 月 29 日

バージョン	変更	ダウンロード	日付
1.0.3	Windows • 軽微なバグ修正とセキュリティ体制の改善	• x64 を使用する Windows	2025 年 12 月 11 日
1.0.2	macOS • バグ修正および安定性の向上 • UI の機能強化 Windows • バグ修正および安定性の向上 • UI の機能強化	• Mac と Apple Silicon • インテル搭載 Mac • x64 を使用する Windows	2025 年 6 月 9 日
1.0.1	macOS • 安定性の向上 • アンインストーラアプリケーション Windows • 安定性の向上	• Mac と Apple Silicon • インテルを搭載した Mac • x64 を使用する Windows	2025 年 2 月 5 日
1.0.0	パブリックプレビュー	• Mac と Apple Silicon • インテル搭載 Mac • x64 を使用する Windows	2024 年 12 月 1 日

Verfied Access のセキュリティ

のクラウドセキュリティが最優先事項 AWS です。お客様は AWS 、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャを活用できます。

セキュリティは、 AWS お客様とお客様の間の責任共有です。責任共有モデルではこれをクラウドのセキュリティおよびクラウド内のセキュリティと説明しています。

- クラウドのセキュリティ – AWS は、 で AWS サービスを実行するインフラストラクチャを保護する責任を担います AWS クラウド。 は、お客様が安全に使用できるサービス AWS も提供します。サードパーティーの監査者は、AWS コンプライアンスプログラムコンプライアンスプログラムの一環として、当社のセキュリティの有効性を定期的にテストおよび検証。 AWS Verified Access に適用されるコンプライアンスプログラムの詳細については、「コンプライアンスプログラムAWS による対象範囲内のサービスコンプライアンスプログラム」を参照してください。
- クラウドのセキュリティ – お客様の責任は、使用する AWS サービスによって決まります。また、ユーザーは、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

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。また、Verified Access リソースのモニタリングや保護に役立つ他の AWS サービスの使用方法についても説明します。

内容

Verified Access でのデータ保護

責任 AWS 共有モデル、 AWS Verified Access でのデータ保護に適用されます。このモデルで説明されているように、 AWS はすべての を実行するグローバルインフラストラクチャを保護する責任があります AWS クラウド。ユーザーは、このインフラストラクチャでホストされ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、「Data Privacy FAQChina」を参照してください。 欧州におけるデータ保護に関する情報については、General Data Protection Regulation (GDPR) Center を参照してください。

データ保護の目的で、認証情報を保護し AWS アカウント 、 AWS IAM アイデンティティセンターまたは AWS Identity and Access Management (IAM) を使用して個々のユーザーを設定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が各ユーザーに付与されます。また、次の方法でデータを保護することもお勧めします:

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 は必須ですが、TLS 1.3 を推奨します。
- で API とユーザーアクティビティのログ記録を設定します AWS CloudTrail。CloudTrail 証跡を使用して AWS アクティビティをキャプチャする方法については、「 AWS CloudTrail ユーザーガイド」のCloudTrail 証跡の使用」を参照してください。
- AWS 暗号化ソリューションと、その中のすべてのデフォルトのセキュリティコントロールを使用します AWS のサービス。
- Amazon Macie などの高度な管理された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 AWS を介して にアクセスするときに FIPS 140-3 検証済みの暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。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) 140-3」を参照してください。

お客様の E メールアドレスなどの極秘または機密情報を、タグ、または [名前] フィールドなどの自由形式のテキストフィールドに含めないことを強くお勧めします。これは、コンソール、API、または SDK を使用して Verified Access AWS CLIまたは他の AWS のサービス を使用する場合も同様です。 AWS SDKs タグ、または名前に使用される自由記述のテキストフィールドに入力したデータは、請求または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、 AWS 所有の KMS キーを使用して、デフォルトで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。

内容

- Verified Access と KMS キー
- 個人を特定できる情報
- AWS Verified Access が で許可を使用する方法 AWS KMS
- Verified Access でカスタマーマネージドキーを使用する
- Verified Access リソースのカスタマーマネージドキーを指定する
- AWS Verified Access 暗号化コンテキスト
- AWS Verified Access の暗号化キーのモニタリング

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、代替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 リソースの作成時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 所有の暗号化キーに 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
- キー暗号化マテリアルのローテーション
- タグを追加する
- キーエイリアスの作成
- 削除のためのキースケジューリング

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 は AWS 、所有キーを使用して保管時の暗号化を自動的に有効にし、個人を特定できるデータを無償で保護します。
> ただし、カスタマーマネージドキーを使用する場合、 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。

データ型	AWS 所有キーの暗号化	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

データ型	AWS 所有キーの暗号化	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、AuthorizationEndpoint、UserInfoEndpoint、ClientId、ClientSecret などの OIDC オプションが含まれており、これらは PII と見なされます。		
`Trust provider (device-type)` デバイスタイプの信頼プロバイダーには PII と見なされる TenantID が含まれます。	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 Verified Access が で許可を使用する方法 AWS KMS

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 CreateGrant リクエストを送信してユーザーに代わって許可を作成します AWS KMS。の権限 AWS KMS は、Verified Access にアカウントのカスタマーマネージドキーへのアクセスを許可するために使用されます。

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。

- に Decrypt リクエストを送信 AWS KMS して、暗号化されたデータキーを復号化して、データの復号化に使用できるようにします。
- RetireGrant リクエストを に送信 AWS KMS して、許可を削除します。

グラントへのアクセスの取り消しや、カスタマーマネージドキーに対するサービスのアクセスの取り消しは、いつでもできます。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は AWS マネジメントコンソール、、または AWS KMS APIs を使用して作成できます。「 AWS Key Management Service デベロッパーガイド」の「対称暗号化キーの作成」の手順に従います。

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。キーポリシーは、カスタマーマネージドキーの作成時に指定できます。詳細については、「AWS Key Management Service デベロッパーガイド」の「キーポリシー」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「 Grants」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、保存された暗号化データキーを使用して暗号化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" : "*"
    }
  ]
```

詳細については、「 デベロッパーガイド」の「キーポリシーの作成」および「キー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。 AWS Key Management Service

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

を使用してこれらのリソースを作成する場合は AWS マネジメントコンソール、「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。プロセス中に、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、使用する 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追加の暗号化を提供しているカスタマーマネージドキーが失われた場合、そのリソースの設定値にはアクセスできなくなります。ただし、リソースは、AWS マネジメントコンソール 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は、データに関する追加のコンテキスト情報を含むキーと値のペアのオプションセットです。 は、認証された暗号化をサポートする追加の認証済みデータとして暗号化コンテキスト AWS KMS を使用します。データを暗号化するリクエストに暗号化コンテキストを含めると、は暗号化コンテキストを暗号化されたデータに AWS KMS 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 Verified Access 暗号化コンテキスト

Verified Access は、すべての AWS KMS 暗号化オペレーションで同じ暗号化コンテキストを使用します。キーは で、`aws:verified-access:arn`値はリソース Amazon リソースネーム (ARN) です。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"
}
```

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 が送信するリクエストを追跡できます AWS KMS。

次の例はCreateGrant、Verified Access によって呼び出される KMS RetireGrant DecryptオペレーションをモニタリングしてGenerateDataKey、カスタマーマネージド KMS キーによって暗号化されたデータにアクセスする 、DescribeKey、、、および の AWS CloudTrail イベントです。

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"
}
```


Verified Access のID およびアクセス管理

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、追加料金なしで使用できる AWS のサービス です。

トピック

オーディエンス

AWS Identity and Access Management (IAM) の使用方法は、ロールによって異なります。

- サービスユーザー - 機能にアクセスできない場合は、管理者にアクセス許可をリクエストします (「Verified Access アイデンティティとアクセスに関するトラブルシューティング」を参照)。
- サービス管理者 - ユーザーアクセスを決定し、アクセス許可リクエストを送信します (「Verified Access と IAM の仕組み」を参照)
- IAM 管理者 - アクセスを管理するためのポリシーを作成します (「Verified Access のアイデンティティベースポリシーの例」を参照)

アイデンティティを使用した認証

認証は、ID 認証情報 AWS を使用して にサインインする方法です。、IAM ユーザー AWS アカウントのルートユーザー、または IAM ロールを引き受けることで認証される必要があります。

AWS IAM アイデンティティセンター (IAM Identity Center)、シングルサインオン認証、Google/Facebook 認証情報などの ID ソースからの認証情報を使用して、フェデレーティッド ID としてサインインできます。サインインの詳細については、「AWS サインイン ユーザーガイド」の「AWS アカウントにサインインする方法」を参照してください。

プログラムによるアクセスの場合、 は SDK と CLI AWS を提供してリクエストを暗号化して署名します。詳細については、「IAM ユーザーガイド」の「API リクエストに対するAWS 署名バージョン 4」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、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 AWS アカウント root ユーザーと呼ばれる 1 つのサインインアイデンティティから始めます。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。ルートユーザー認証情報を必要とするタスクについては、「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。

フェデレーテッドアイデンティティ

ベストプラクティスとして、人間のユーザーが一時的な認証情報 AWS のサービス を使用して にアクセスするには、ID プロバイダーとのフェデレーションを使用する必要があります。

フェデレーティッド ID は、エンタープライズディレクトリ、ウェブ ID プロバイダー、または ID Directory Service ソースの認証情報 AWS のサービス を使用して にアクセスするユーザーです。フェデレーテッドアイデンティティは、一時的な認証情報を提供するロールを引き受けます。

アクセスを一元管理する場合は、 AWS IAM アイデンティティセンターをお勧めします。詳細については、「AWS IAM アイデンティティセンター ユーザーガイド」の「IAM アイデンティティセンターとは」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、特定の個人やアプリケーションに対する特定のアクセス許可を持つアイデンティティです。長期認証情報を持つ IAM ユーザーの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。詳細については、IAM ユーザーガイドの「ID プロバイダーとのフェデレーションを使用して にアクセスすることを人間のユーザーに要求する AWS」を参照してください。

IAM グループは、IAM ユーザーの集合を指定し、大量のユーザーに対するアクセス許可の管理を容易にします。詳細については、「IAM ユーザーガイド」の「IAM ユーザーに関するユースケース」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を持つアイデンティであり、一時的な認証情報を提供します。ユーザーから IAM ロール (コンソール) に切り替えるか、 または API オペレーションを呼び出す

ことで、ロールを引き受けることができます。 AWS CLI AWS 詳細については、「IAM ユーザーガイド」の「ロールを引き受けるための各種方法」を参照してください。

IAM ロールは、フェデレーションユーザーアクセス、一時的な IAM ユーザーのアクセス許可、クロスアカウントアクセス、クロスサービスアクセス、および Amazon EC2 で実行するアプリケーションに役立ちます。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。

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、ID AWS またはリソースにアタッチします。ポリシーは、アイデンティティまたはリソースに関連付けられたときにアクセス許可を定義します。は、プリンシパル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。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、ポリシーを使用して、どのプリンシパルがどのリソースに対して、どのような条件でアクションを実行できるかを定義することで、誰が何にアクセスできるかを指定します。

デフォルトでは、ユーザーやロールにアクセス許可はありません。IAM 管理者は IAM ポリシーを作成してロールに追加し、このロールをユーザーが引き受けられるようにします。IAM ポリシーは、オペレーションの実行方法を問わず、アクセス許可を定義します。

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のポリシーは、アイデンティティ (ユーザー、グループ、またはロール) にアタッチできる JSON アクセス許可ポリシードキュメントです。これらのポリシーは、アイデンティティがどのリソースに対してどのような条件下でどのようなアクションを実行できるかを制御します。アイデンティティベースポリシーの作成方法については、IAM ユーザーガイド の カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する 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、インラインポリシー (単一の ID に直接埋め込む) または管理ポリシー (複数の ID にアタッチされたスタンドアロンポリシー) にすることができます。管理ポリシーと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「管理ポリシーとインラインポリシーのいずれかを選択する」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。例としては、IAM ロール信頼ポリシーや Amazon S3 バケットポリシーなどがあります。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。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。

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。

その他のポリシータイプ

AWS は、より一般的なポリシータイプによって付与されるアクセス許可の最大数を設定できる追加のポリシータイプをサポートしています。

- アクセス許可の境界 – アイデンティティベースのポリシーで IAM エンティティに付与することのできるアクセス許可の数の上限を設定します。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 (SCP) - AWS Organizations内の組織または組織単位の最大のアクセス許可を指定します。詳細については、「AWS Organizations ユーザーガイド」の「サービスコントロールポリシー」を参照してください。
- リソースコントロールポリシー (RCP) – は、アカウント内のリソースで利用できる最大数のアクセス許可を定義します。詳細については、「AWS Organizations ユーザーガイド」の「リソースコントロールポリシー (RCP)」を参照してください。
- セッションポリシー – ロールまたはフェデレーションユーザーの一時セッションを作成する際にパラメータとして渡される高度なポリシーです。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アクセス許可を理解するのがさらに難しくなります。が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。

Verified Access と IAM の仕組み

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。

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	あり
ポリシー条件キー	あり
ACL	なし
ABAC (ポリシー内のタグ)	部分的
一時認証情報	あり
プリンシパルアクセス権限	あり
サービスロール	いいえ
サービスリンクロール	はい

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については、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のポリシーのサポート: 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ポリシーの作成方法については、「IAM ユーザーガイド」の「カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する」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。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、「Verified Access のアイデンティティベースポリシーの例」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コントロールできます。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、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、全体のアカウント、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。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のサポート: あり

管理者は JSON AWS 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対してどのような条件下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。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アクセス許可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、「サービス認可リファレンス」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、「Verified Access のアイデンティティベースポリシーの例」を参照してください。

Verified Access のポリシーリソース

ポリシーリソースのサポート: あり

管理者は JSON AWS 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対してどのような条件下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。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。リソースレベルのアクセス許可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 のリソースタイプとその ARN のリストを確認するには、「サービス認可リファレンス」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、「Verified Access のアイデンティティベースポリシーの例」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 JSON AWS 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対してどのような条件下でアクションを実行できるかということです。

Condition 要素は、定義された基準に基づいてステートメントが実行される時期を指定します。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して、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。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 の条件キーのリストを確認するには、「サービス認可リファレンス」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、「Verified Access のアイデンティティベースポリシーの例」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アクセス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。

Verified Access での ABAC

ABAC (ポリシー内のタグ) のサポート: 一部

属性ベースのアクセスコントロール (ABAC) は、タグと呼ばれる属性に基づいてアクセス許可を定義する認可戦略です。IAM エンティティと AWS リソースにタグをアタッチし、プリンシパルのタグがリソースのタグと一致するときにオペレーションを許可するように ABAC ポリシーを設計できます。

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 認可でアクセス許可を定義する」を参照してください。ABAC をセットアップする手順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コントロール (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

一時的な認証情報は AWS 、リソースへの短期的なアクセスを提供し、フェデレーションまたはスイッチロールの使用時に自動的に作成されます。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 AWS 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「IAM の一時的な認証情報」および「AWS のサービス と IAM との連携」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

転送アクセスセッション (FAS) のサポート: あり

転送アクセスセッション (FAS) は、 を呼び出すプリンシパルのアクセス許可と AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストをリクエストする を使用します。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: なし

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 のサービスに許可を委任するロールを作成する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、サービスリンク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。

Verified Access のアイデンティティベースポリシーの例

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。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。

これらのサンプルの 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 ポリシーを作成する (コンソール)」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の詳細 (リソースタイプごとの ARN の形式を含む) については、「サービス認可リファレンス」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

- ポリシーに関する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

ポリシーに関する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では、 AWS アカウントに費用が発生する場合があります。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の付与を開始するには、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することで、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 ポリシーでアクセス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。たとえば、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、サービスアクションが などの特定の を通じて使用されている場合に 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 CloudFormation。詳細については、IAM ユーザーガイド の IAM JSON ポリシー要素:条件 を参照してください。
- IAM アクセスアナライザー 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 アクセスアナライザー 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 でポリシーを検証する 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、MFA をオンにしてセキュリティを強化します。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 を使用した安全な API アクセス 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可はサポートされていません。リソース、タグ、または条件キーベースの認可は、ec2:CreateVerifiedAccessInstance API アクションで使用します。

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 アカウント番号で、 us-east-1 は AWS リージョンです。

JSON

```
{
      "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": "*"
        }
    ]
}
```

Verified Access アイデンティティとアクセスに関する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:PassRole を実行する権限がありません
- 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

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、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:PassRole を実行する権限がありません

iam:PassRole アクションを実行する権限がないというエラーが表示された場合は、ポリシーを更新して 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、既存のロールをそのサービスに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アクセス許可が必要です。

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アクセス許可が必要です。Mary には、ロールをサービスに渡すアクセス許可がありません。

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

この場合、Mary のポリシーを更新してメアリーに iam:PassRole アクションの実行を許可する必要があります。

サポートが必要な場合は、 AWS 管理者にお問い合わせください。サインイン資格情報を提供した担当者が管理者です。

自分の 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、「Verified Access と IAM の仕組み」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におけるロールとリソースベースのポリシーの使用方法の違いについては、「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。

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、 サービスに直接リンクされた IAM ロールの一種である IAM AWS サービスにリンクされたロールを使用します。Verified Access のサービスにリンクされたロールは、Verified Access によって定義され、サービスが AWS のサービス ユーザーに代わって他の を呼び出すために必要なすべてのアクセス許可が含まれます。

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。

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 「 マネージドポリシーリファレンスガイド」でも確認できます。AWSVPCVerifiedAccessServiceRolePolicy」を参照してください。

サービスリンク役割の作成、編集、削除を IAM エンティティ (ユーザー、グループ、役割など) に許可するにはアクセス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

サービスリンクロールを手動で作成する必要はありません。 AWS マネジメントコンソール、、AWS CLIまたは API で CreateVerifiedAccessEndpoint を AWS 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。

このサービスリンク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。サービスリンクロールの作成後は、さまざまなエンティティがロールを参照す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 を使用してロールの説明を編集することはできます。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 マネジメントコンソール、、 AWS CLIまたは API で DeleteVerifiedAccessEndpoint を AWS 呼び出すと、Verified Access は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。

サービスリンクロールを IAM で手動削除するには

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 は、サービスを利用できるすべての で AWS リージョン 、サービスにリンクされた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 マネージドポリシー

AWS 管理ポリシーは、 によって作成および管理されるスタンドアロンポリシーです AWS。 AWS 管理ポリシーは、ユーザー、グループ、ロールにアクセス許可の割り当てを開始できるように、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ように設計されています。

AWS 管理ポリシーは、すべての AWS お客様が使用できるため、特定のユースケースに対して最小特権のアクセス許可を付与しない場合があることに注意してください。ユースケースに固有のカスタマー管理ポリシーを定義して、アクセス許可を絞り込むことをお勧めします。

AWS 管理ポリシーで定義されているアクセス許可は変更できません。が AWS マネージドポリシーで定義されたアクセス許可 AWS を更新すると、ポリシーがアタッチされているすべてのプリンシパ

ル ID (ユーザー、グループ、ロール) に影響します。 AWS は、新しい が起動されるか、新しい API オペレーション AWS のサービス が既存のサービスで使用できるようになったときに、 AWS マネージドポリシーを更新する可能性が高くなります。

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、「サービスリンクロールを使用する」を参照してください。このポリシーのアクセス許可を表示するには、「」のAWSVPCVerifiedAccessServiceRolePolicy」を参照するか AWS マネジメントコンソール、AWS「 マネージドポリシーリファレンスガイド」のAWSVPCVerifiedAccessServiceRolePolicy」ポリシーを参照してください。

AWS マネージドポリシーに対する Verified Access の更新

このサービスがこれらの変更の追跡を開始してからの Verified Access の AWS マネージドポリシーの更新に関する詳細を表示します。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 のマネージドポリシーが更新され、すべてのアクションの記述に「sid」フィールドが追加されました。	2023 年 11 月 17 日
AWSVPCVerifiedAccessServiceRolePolicy - ポリシー更新	Verified Access のマネージドポリシーが更新され、`ec2:CreateNetworkInterface` アクセス許可にセキュリティグループリソースが追加されました。	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	2022 年 11 月 29 日

変更	説明	日付
	できるようにする新しいポリシーが追加されました。	
Verified Access は変更の追跡を開始	Verified Access は、 AWS 管理ポリシーの変更の追跡を開始しました。	2022 年 11 月 29 日

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AWS のサービス プログラムによるスコープ」の「コンプライアンス」を参照して、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、「Downloading Reports in AWS Artifact」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。を使用する際のコンプライアンス責任の詳細については AWS のサービス、AWS 「 セキュリティドキュメント」を参照してください。

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。 は、低レイテンシー、高スループット、および高度に冗長なネットワークで接続された、物理的に分離および分離された複数のアベイラビリティーゾーン AWS リージョン を提供します。アベイラビリティーゾーンでは、ゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、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 および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 は、 AWS グローバルインフラストラクチャに加えて、高可用性のニーズをサポートするため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。

モニタリング AWS Verified Access

モニタリングは、 の信頼性、可用性、パフォーマンスを維持する上で重要な部分です AWS Verified Access。 は、Verified Access をモニタリングし、問題が発生したときに報告し、必要に応じて自動アクションを実行するために、次のモニタリングツール AWS を提供します。

- 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 – によって、または に代わって行われた API コールおよび関連イベントをキャプチャ AWS アカウント し、指定した Amazon S3 バケットにログファイルを配信します。が呼び出したユーザーとアカウント AWS、呼び出し元のソース IP アドレス、および呼び出しの発生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

が各アクセスリクエスト AWS Verified Access を評価すると、すべてのアクセス試行がログに記録されます。これにより、アプリケーションへのアクセスが一元的に視覚化され、セキュリティインシデントや監査リクエスト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。ロギングを正しく機能させるには、ロギング先の設定に使用される IAM プリンシパルに特定のアクセス許可が必要です。各ロギング先に必要な IAM アクセス権限は、Verified Access のロギングのアクセス許可 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

内容

- Verified Access の OCSF バージョン 1.0.0-rc.2 ログの例

Verified Access の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 を使用するサンプルログについては、「」を参照してくださいVerified Access の OCSF バージョン 0.1 ログの例。

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」を参照してください。バージョン 1.0.0-rc.2 を使用するサンプルログについては、「」を参照してくださいVerified Access の OCSF バージョン 1.0.0-rc.2 ログの例。

Verified Access エンドポイントが TCP プロトコルを使用している場合、OCSF バージョン 0.1 を使用することはできません。

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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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。

Verified Access のロギングのアクセス許可

ロギングを正しく機能させるには、ロギング先の設定に使用される IAM プリンシパルに特定のアクセス許可が必要です。各ロギング先に必要なアクセス許可を以下のセクションに示します。

CloudWatch Logs 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:

- 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

Verified Access ログの有効化または無効化

ロギングを有効または無効にするには、このセクションの手順に従います。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。ロギングを正しく機能させるには、ロギング先の設定に使用される IAM プリンシパルに特定のアクセス許可が必要です。各ロギング先に必要な IAM アクセス権限は、Verified Access のロギングのアクセス許可 セクションで確認できます。

内容

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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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] をオンにします。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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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。

Verified Access 信頼コンテキストの有効化または無効化

必要に応じて、信頼プロバイダーから送信された信頼コンテキストを 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ときに役立つ可能性があります。有効にすると、信頼コンテキストがログの `data` フィールドに含められます。信頼コンテキストが無効になっている場合、`data`フィールドは `null` に設定されます。信頼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、以下の手順を実行します。

> **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、ロギングが既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。

内容

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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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

信頼コンテキストをログに含める必要がなくなった場合は、次の手順に従って削除できます。

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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
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 の OCSF バージョン 0.1 ログの例

OCSF バージョン 0.1 を使用したサンプルログを次に示します。

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 CrowdStrike によるアクセス許可
- 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

このログエントリの例では、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 CrowdStrike 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 と CrowdStrike 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": "10.2.173.3",
        "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
}
```


Verified Access の OCSF バージョン 1.0.0-rc.2 ログの例

OCSF バージョン 1.0.0-rc.2 を使用したサンプルログを次に示します。

例

- トラストコンテキストを含むアクセス許可
- トラストコンテキストを省略したアクセス許可
- ネットワーク CIDR エンドポイントで権限を割り当てる

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",
                "additional_user_context": {
                    "aud": "xxx",
                    "exp": 1000000000,
                    "groups": [
                        "group-id-1",
                        "group-id-2"
                    ],
                    "iat": 1000000000,
                    "iss": "https://oidc-tp.com/",
                    "sub": "xyzsubject",
                    "ver": "1.0"
                }
            },
            "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
}
```

ネットワーク CIDR エンドポイントで権限を割り当てる

```
{
    "activity_id": "1",
    "activity_name": "Assign Privileges",
    "category_name": "Audit Activity",
    "category_uid": "3",
    "class_name": "Authorization",
    "class_uid": "3003",
    "data": {
        "endpoint_type": "cidr",
        "protocol": "tcp",
        "access_path": "public",
        "idp": {
            "name": "my-oidc-instance",
            "uid": "vatp-09bc4cbce2EXAMPLE"
        },
```

```
        "authorizations": [{
            "decision": "Allow",
            "policy": {
                "name": "inline"
            }
        }],
        "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",
                "additional_user_context": {
                    "aud": "xxx",
                    "exp": 1000000000,
                    "groups": [
                        "group-id-1",
                        "group-id-2"
                    ],
                    "iat": 1000000000,
                    "iss": "https://oidc-tp.com/",
                    "sub": "xyzsubject",
                    "ver": "1.0"
                }
            },
            "tcp_flow": {
                "destination_ip": "10.0.0.1",
                "destination_port": 22,
                "client_ip": "10.2.7.68"
            }
        }
    },
    "device": {
        "ip":  "10.2.7.68",
        "port": 1002,
```

```
        "type": "Unknown",
        "type_id": 0
    },
    "duration": "0.004",
    "end_time": "1668580194344",
    "time": "1668580194344",
    "metadata": {
        "uid": "",
        "logged_time": 1668580281337,
        "version": "1.0.0-rc.2",
        "product": {
            "name": "Verified Access",
            "vendor_name": "AWS"
        }
    },
    "severity": "Informational",
    "severity_id": "1",
    "start_time": "1668580194340",
    "status_code": "200",
    "status_id": "1",
    "status": "Success",
    "type_uid": "300301",
    "type_name": "Authorization: Assign Privileges",
    "count": 1,
    "dst_endpoint": {
        "ip": "107.22.231.155",
        "port": 22
    },
    "privileges": [
        "vae-12345cbce2EXAMPLE"
    ],
    "user": {
        "email_addr": "johndoe-user@test.com",
        "uid": "johndoe-user",
        "uuid": "9bcce02a-fc15-4091-a0b7-874d157c67b8"
    }
}
```

を使用して Verified Access API コールをログに記録する AWS CloudTrail

AWS Verified Access は AWS CloudTrail、Verified Access のユーザー、ロール、または によって実行されたアクションを記録するサービスである と統合 AWS のサービス されています。CloudTrail は、Verified Access の API コールをイベントとしてキャプチャします。キャプチャされる呼び出しには、Verified Access コンソールからの呼び出しと、コードからの Verified Access API オペレーションの呼び出しが含まれます。CloudTrail で収集した情報を使用して、Verified Access に対するリクエスト、リクエスト元の IP アドレス、リクエストの作成日時、その他の詳細を確認できます。

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。アイデンティティ情報は、以下を判別するのに役立ちます。

- ルートユーザーまたはユーザー認証情報のどちらを使用してリクエストが送信されたか。
- リクエストが IAM Identity Center ユーザーに代わって行われたかどうか。
- 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テンポラリ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。

アカウント AWS アカウント を作成するとCloudTrail が でアクティブになり、CloudTrail イベント履歴に自動的にアクセスできます。CloudTrail の [イベント履歴] では、 AWS リージョンで過去 90 日間に記録された 管理イベントの表示、検索、およびダウンロードが可能で、変更不可能な記録を確認できます。詳細については、「AWS CloudTrail ユーザーガイド」の「CloudTrail イベント履歴の使用」を参照してください。[イベント履歴] の閲覧には CloudTrail の料金はかかりません。

AWS アカウント 過去 90 日間のイベントの継続的な記録については、証跡または CloudTrail Lake イベントデータストアを作成します。

CloudTrail 証跡

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。を使用して作成された証跡はすべてマルチリージョン AWS マネジメントコンソール です。 AWS CLIを使用する際は、単一リージョンまたは複数リージョンの証跡を作成できます。 AWS リージョン アカウントのすべての でアクティビティをキャプチャするため、マルチリージョン証跡を作成することをお勧めします。単一リージョンの証跡を作成する場合、証跡の AWS リージョンに記録されたイベントのみを表示できます。証跡の詳細については、「AWS CloudTrail ユーザーガイド」の「AWS アカウントの証跡の作成」および「組織の証跡の作成」を参照してください。

証跡を作成すると、進行中の管理イベントのコピーを 1 つ無料で CloudTrail から Amazon S3 バケットに配信できますが、Amazon S3 ストレージには料金がかかります。CloudTrail の料金の詳細については、「AWS CloudTrail の料金」を参照してください。Amazon S3 の料金に関する詳細については、「Amazon S3 の料金」を参照してください。

CloudTrail Lake イベントデータストア

[CloudTrail Lake] を使用すると、イベントに対して SQL ベースのクエリを実行できます。CloudTrail Lake は、行ベースの JSON 形式の既存のイベントを Apache ORC 形式に変換します。ORC は、データを高速に取得するために最適化された単票ストレージ形式です。イベントは、イベントデータストアに集約されます。イベントデータストアは、高度なイベントセレクタを適用することによって選択する条件に基づいた、イベントのイミュータブルなコレクションです。どのイベントが存続し、クエリに使用できるかは、イベントデータストアに適用するセレクタが制御します。CloudTrail Lake の詳細については、「 AWS CloudTrail ユーザーガイド」の「Working with AWS CloudTrail Lake」を参照してください。

CloudTrail Lake のイベントデータストアとクエリにはコストがかかります。イベントデータストアを作成する際に、イベントデータストアに使用する料金オプションを選択します。料金オプションによって、イベントの取り込みと保存にかかる料金、および、そのイベントデータストアのデフォルトと最長の保持期間が決まります。CloudTrail の料金の詳細については、「AWS CloudTrail の料金」を参照してください。

Verified Access の管理イベント

管理イベントは、 のリソースで実行される管理オペレーションに関する情報を提供します AWS アカウント。これらのイベントは、コントロールプレーンオペレーションとも呼ばれます。CloudTrail は、デフォルトで管理イベントをログ記録します。

Verified Access は、コントロールプレーンオペレーションを管理イベントとしてログに記録します。一覧については、「Amazon EC2 API リファレンス」を参照してください。

Verified Access イベントの例

以下の例は、`CreateVerifiedAccessInstance` アクションを示す CloudTrail イベントエントリです。

```
{
    "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"
}
```


CloudTrail レコードの内容については、「AWS CloudTrail ユーザーガイド」の「CloudTrail record contents」を参照してください。

のクォータ AWS Verified Access

AWS アカウント には、各 の制限と呼ばれるデフォルトのクォータがあります AWS のサービス。特に明記していない限り、クォータはリージョン固有です。

AWS アカウントレベルのクォータ

AWS アカウント には、Verified Access に関連する次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	あり	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	あり	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	あり	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

HTTP トラフィック

接続アイドルタイムアウトは 60 秒です。アプリケーションが HTTP リクエストに応答するのに 60 秒以上かかる場合、クライアントは HTTP 504 ゲートウェイタイムアウトエラーを受け取ります。Verified Access ログが有効になっている場合、HTTP 504 エラーはすべてログに記録されます。

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

IAM アイデンティティセンター

Verified Access は、最大 1,000 のグループに割り当てられている IAM アイデンティティセンターのユーザーにアクセスを提供できます。

接続クライアント

Connectivity Client には次の制限があります。

名前	デフォルト	引き上げ可能
デバイスあたりの同時 Verified Access インスタンス接続数	5	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
信頼コンテキストでのアクセストークンのサポート	を更新して OIDC ユーザークレームに追加`additional_user_context` します。	2025 年 2 月 24 日
HTTP プロトコル以外のリソースのサポート	HTTP 以外のプロトコルを介したリソースへのアクセスのリリース。	2025 年 2 月 5 日
プレビューリリース	HTTP 以外のプロトコルを介したリソースへのアクセスのプレビューリリース。	2024 年 12 月 1 日
AWS マネージドポリシーの更新	Verified Access の AWS マネージド IAM ポリシーを更新しました。	2023 年 11 月 17 日
保管時のデータ暗号化	AWS Verified Access は、AWS 所有の KMS キーを使用して、デフォルトで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の更新	Verified Access の AWS マネージド IAM ポリシーを更新しました。	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。