ユーザーガイド

AWS Certificate Manager

Version 1.0

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Certificate Manager: ユーザーガイド

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していない他のすべての商標は、それぞれの所有者の所有物であり、Amazon と提携、接続、または後援されている場合とされていない場合があります。

Table of Contents

とは AWS Certificate Manager 1
サポート対象のリージョン 1
料金 2
概念 2
ACM 証明書 3
ACM ルート CA 5
Apex ドメイン 5
非対称キー暗号 6
認証局 6
証明書の透明性ログ記録 6
ドメインネームシステム 7
ドメイン名 7
暗号化と復号 8
完全修飾ドメイン名 (FQDN) 9
ハイパーテキスト転送プロトコル (HTTP) 9
パブリックキーインフラストラクチャ (PKI) 9
ルート証明書 10
Secure Sockets Layer (SSL) 10
安全な HTTPS 10
SSL サーバー証明書 11
対称キー暗号化 11
Transport Layer Security (TLS) 11
信頼 11
ニーズに適した AWS 証明書サービスは何ですか? 11
開始方法 13
セットアップする 14
にサインアップする AWS アカウント 14
管理アクセスを持つユーザーを作成する 15
ドメイン名を登録する 16
(オプション) CAA レコードの設定 16
パブリック証明書 20
特徴と制限 21
パブリック証明書のリクエスト 26
コンソールを使用してパブリック証明書をリクエストする 27

CLI を使用してパブリック証明書をリクエストする 29
エクスポート可能なパブリック証明書 30
利点 30
ACM エクスポート可能なパブリック証明書の仕組み 30
セキュリティに関する考慮事項 31
制限事項 31
料金 31
ベストプラクティス 31
証明書をエクスポートする 32
Kubernetes ワークロードの保護 34
証明書の取り消し 40
自動更新イベントを設定する 42
証明書の更新を強制する 42
証明書の検証 43
DNS での検証 44
E メール検証 50
HTTP 検証 56
プライベート証明書 62
使用条件 63
プライベート証明書のリクエスト 64
プライベート証明書のリクエスト (コンソール) 64
プライベート証明書のリクエスト (CLI) 66
証明書をエクスポートする 68
プライベート証明書のエクスポート (コンソール) 68
プライベート証明書 (CLI) をエクスポートします。 69
インポートされた証明書 71
前提条件 72
証明書の形式 73
証明書のインポート 75
インポート (コンソール) 76
インポート (AWS CLI) 76
証明書の再インポート 77
再インポート (コンソール) 78
再インポート(AWS CLI) 78
証明書管理 80
証明書の一覧表示 80

証明書の詳細を表示する 83
証明書を削除する 87
マネージド証明書の更新 88
パブリック証明書 90
DNS 検証済みドメイン 90
E メール検証済みドメイン 91
HTTP 検証済みドメイン 92
プライベート証明書 93
更新された証明書のエクスポートの自動化 94
マネージド更新のテスト 95
更新ステータスの確認 96
ステータスの確認 (コンソール) 98
ステータスの確認 (API) 98
ステータスの確認 (CLI) 98
Personal Health Dashboard (PHD) を使用してステータスを確認する 98
リソースにタグを付ける 100
タグの制限 100
タグの管理 101
タグの管理 (コンソール) 101
タグを管理する (CLI) 103
タグの管理 103
統合サービス 104
セキュリティ 110
データ保護 110
証明書の秘密鍵のセキュリティ 111
Identity and Access Management 112
オーディエンス 113
アイデンティティを使用した認証 113
ポリシーを使用したアクセスの管理 115
が IAM と AWS Certificate Manager 連携する方法 116
アイデンティティベースのポリシーの例 122
ACM API のアクセス許可リファレンス 128
AWS マネージドポリシー 130
条件キーの使用 131
サービスリンクロールを使用する 137
トラブルシューティング 141

耐障害性 143
インフラストラクチャセキュリティ 144
ACM へのプログラムによるアクセス権の管理 144
ベストプラクティス 146
アカウントレベルの分離 147
AWS CloudFormation 148
カスタム信頼ストア 148
証明書のピンニング 148
ドメイン検証 149
ドメイン名の追加または削除 149
証明書の透明性ログ記録のオプトアウト 150
オンにする AWS CloudTrail 152
モニタリングとロギング 153
Amazon EventBridge 153
サポートされるイベント 153
アクションの例 159
CloudTrail 169
サポートされている API アクション 170
統合サービスの API 呼び出し 184
CloudWatch メトリクス 190
SDK for Java AWS Certificate Manager で を使用する 191
AddTagsToCertificate 191
DeleteCertificate 193
DescribeCertificate 195
ExportCertificate 198
GetCertificate 201
ImportCertificate 203
ListCertificates 207
RenewCertificate 209
ListTagsForCertificate 211
RemoveTagsFromCertificate 213
RequestCertificate 215
ResendValidationEmail 218
トラブルシューティング 221
証明書のリクエスト 221
リクエストのタイムアウト 221

リクエストの失敗 222
証明書の検証 223
DNS での検証 224
E メール検証 227
HTTP 検証 228
証明書の更新 230
自動ドメイン検証の準備 230
マネージド証明書更新のエラーを処理する 230
E メール検証済みの証明書のマネージド証明書更新 230
DNS で検証済みの証明書のマネージド型の証明書更新 231
HTTP 検証済みの証明書のマネージド型の証明書更新 232
更新のタイミングについて 233
その他の問題 233
CAA レコード 233
証明書のインポート 234
証明書のピンニング 235
API ゲートウェイ 235
予期しない障害 236
ACM のサービスにリンクされたロール (SLR) に関する問題 236
例外処理 237
プライベート証明書の例外処理 237
クォータ 240
一般的なクォータ 240
API レートクォータ 242
ドキュメント履歴 245
..... cclii

とは AWS Certificate Manager

AWS Certificate Manager (ACM) は、 AWS ウェブサイトとアプリケーションを保護するパブリックおよびプライベート SSL/TLS X.509 証明書とキーの作成、保存、更新の複雑さを処理します。統合 AWS サービス の証明書は、ACM で直接発行するか、サードパーティーの証明書を ACM 管理システムに インポート することで提供できます。ACM 証明書は、単一のドメイン名、複数の特定のドメイン名、ワイルドカードドメイン、またはこれらの組み合わせを保護できます。ACM ワイルドカード 証明書は、無制限の数のサブドメインを保護できます。によって署名された ACM 証明書をエクスポート AWS Private CA して、内部 PKI の任意の場所で使用することもできます。

> **Note**
>
> ACM は、スタンドアロン Web サーバでの使用を目的としていません。Amazon EC2 インスタンスにスタンドアロンのセキュアサーバーをセットアップする場合は、次のチュートリアル「Configure SSL/TLS on Amazon Linux 2023」に手順が記載されています。

トピック

- サポート対象のリージョン
- の料金 AWS Certificate Manager
- AWS Certificate Manager の概念
- ニーズに適した AWS 証明書サービスは何ですか?

サポート対象のリージョン

ACM は、パブリックエンドポイントで IPv4 と IPv6 をサポートします。AWS 全般のリファレンスの「AWS リージョンとエンドポイント」、または「AWS リージョン表」で ACM を利用できるリージョンを参照してください。

ACM の証明書はリージョナルリソースです。複数の AWS リージョンで同じ完全修飾ドメイン名 (FQDN) または一連の FQDNs に対して Elastic Load Balancing で証明書を使用するには、リージョンごとに証明書をリクエストまたはインポートする必要があります。ACM が提供する証明書においては、これは、リージョンごとに証明書の各ドメイン名を取得する必要があることを意味します。リージョン間で証明書をコピーすることはできません。

Amazon CloudFront で ACM 証明書を使用するには、米国東部 (バージニア北部) リージョン の証明書をリクエスト (またはインポート) していることを確認します。CloudFront ディストリビューションに関連づけられたこのリージョンの ACM 証明書は、このディストリビューションに設定されたすべての地域に分配されます。

の料金 AWS Certificate Manager

で管理する SSL/TLS 証明書には追加料金はかかりません AWS Certificate Manager。ウェブサイトまたはアプリケーションを実行するために作成した AWS リソースに対してのみ料金が発生します。最新の ACM 料金情報については、 AWS ウェブサイトのAWS Certificate Manager 「サービス料金」ページを参照してください。

AWS Certificate Manager の概念

このセクションでは、 で使用される概念の定義について説明します AWS Certificate Manager。

トピック

- ACM 証明書
- ACM ルート CA
- Apex ドメイン
- 非対称キー暗号
- 認証局
- 証明書の透明性ログ記録
- ドメインネームシステム
- ドメイン名
- 暗号化と復号
- 完全修飾ドメイン名 (FQDN)
- ハイパーテキスト転送プロトコル (HTTP)
- パブリックキーインフラストラクチャ (PKI)
- ルート証明書
- Secure Sockets Layer (SSL)
- 安全な HTTPS
- SSL サーバー証明書
- 対称キー暗号化

- Transport Layer Security (TLS)
- 信頼

ACM 証明書

ACM は X.509 バージョン 3 証明書を生成します。各 は 198 日間有効で、次の拡張機能が含まれています。

- 基本的な制約 - 証明書の対象が認証局 (CA) かどうかを指定します
- 認証キー識別子 - 証明書に署名するために使用されるプライベートキーに対応するパブリックキーの識別を可能にします。
- サブジェクトキー識別子 - 特定のパブリックキーを含む証明書を識別することができます。
- キーの使用状況 - 証明書に埋め込まれたパブリックキーの目的を定義します。
- 拡張キー用途 - キーの使用状況拡張機能で指定された目的に加えて、パブリックキーを使用する目的を 1 つ以上指定します。

> **Important**
>
> 2025 年 6 月 11 日以降、ウェブサイト証明書の新しいブラウザ要件に合わせて、「TLS Web Client Authentication」 (clientAuth) 拡張キー使用法 (EKU) で証明書が発行され AWS Certificate Manager なくなりました。

- CRL ディストリビューションポイント - CRL 情報を取得できる場所を指定します。

ACM によって発行された証明書のプレーンテキストは、次の例のようになります。

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            f2:16:ad:85:d8:42:d1:8a:3f:33:fa:cc:c8:50:a8:9e
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: O=Example CA
        Validity
            Not Before: Jan 30 18:46:53 2018 GMT
            Not After : Jan 31 19:46:53 2018 GMT
        Subject: C=US, ST=VA, L=Herndon, O=Amazon, OU=AWS, CN=example.com
        Subject Public Key Info:
```

```
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:ba:a6:8a:aa:91:0b:63:e8:08:de:ca:e7:59:a4:
                    69:4c:e9:ea:26:04:d5:31:54:f5:ec:cb:4e:af:27:
                    e3:94:0f:a6:85:41:6b:8e:a3:c1:c8:c0:3f:1c:ac:
                    a2:ca:0a:b2:dd:7f:c0:57:53:0b:9f:b4:70:78:d5:
                    43:20:ef:2c:07:5a:e4:1f:d1:25:24:4a:81:ab:d5:
                    08:26:73:f8:a6:d7:22:c2:4f:4f:86:72:0e:11:95:
                    03:96:6d:d5:3f:ff:18:a6:0b:36:c5:4f:78:bc:51:
                    b5:b6:36:86:7c:36:65:6f:2e:82:73:1f:c7:95:85:
                    a4:77:96:3f:c0:96:e2:02:94:64:f0:3a:df:e0:76:
                    05:c4:56:a2:44:72:6f:8a:8a:a1:f3:ee:34:47:14:
                    bc:32:f7:50:6a:e9:42:f5:f4:1c:9a:7a:74:1d:e5:
                    68:09:75:19:4b:ac:c6:33:90:97:8c:0d:d1:eb:8a:
                    02:f3:3e:01:83:8d:16:f6:40:39:21:be:1a:72:d8:
                    5a:15:68:75:42:3e:f0:0d:54:16:ed:9a:8f:94:ec:
                    59:25:e0:37:8e:af:6a:6d:99:0a:8d:7d:78:0f:ea:
                    40:6d:3a:55:36:8e:60:5b:d6:0d:b4:06:a3:ac:ab:
                    e2:bf:c9:b7:fe:22:9e:2a:f6:f3:42:bb:94:3e:b7:
                    08:73
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Authority Key Identifier:
                keyid:84:8C:AC:03:A2:38:D9:B6:81:7C:DF:F1:95:C3:28:31:D5:F7:88:42
            X509v3 Subject Key Identifier:
                97:06:15:F1:EA:EC:07:83:4C:19:A9:2F:AF:BA:BB:FC:B2:3B:55:D8
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 CRL Distribution Points:
                Full Name:
                  URI:http://example.com/crl

    Signature Algorithm: sha256WithRSAEncryption
         69:03:15:0c:fb:a9:39:a3:30:63:b2:d4:fb:cc:8f:48:a3:46:
         69:60:a7:33:4a:f4:74:88:c6:b6:b6:b8:ab:32:c2:a0:98:c6:
         8d:f0:8f:b5:df:78:a1:5b:02:18:72:65:bb:53:af:2f:3a:43:
         76:3c:9d:d4:35:a2:e2:1f:29:11:67:80:29:b9:fe:c9:42:52:
         cb:6d:cd:d0:e2:2f:16:26:19:cd:f7:26:c5:dc:81:40:3b:e3:
         d1:b0:7e:ba:80:99:9a:5f:dd:92:b0:bb:0c:32:dd:68:69:08:
```

```
        e9:3c:41:2f:15:a7:53:78:4d:33:45:17:3e:f2:f1:45:6b:e7:
        17:d4:80:41:15:75:ed:c3:d4:b5:e3:48:8d:b5:0d:86:d4:7d:
        94:27:62:84:d8:98:6f:90:1e:9c:e0:0b:fa:94:cc:9c:ee:3a:
        8a:6e:6a:9d:ad:b8:76:7b:9a:5f:d1:a5:4f:d0:b7:07:f8:1c:
        03:e5:3a:90:8c:bc:76:c9:96:f0:4a:31:65:60:d8:10:fc:36:
        44:8a:c1:fb:9c:33:75:fe:a6:08:d3:89:81:b0:6f:c3:04:0b:
        a3:04:a1:d1:1c:46:57:41:08:40:b1:38:f9:57:62:97:10:42:
        8e:f3:a7:a8:77:26:71:74:c2:0a:5b:9e:cc:d5:2c:c5:27:c3:
        12:b9:35:d5
```

ACM ルート CA

ACM によって発行されたパブリックエンドエンティティ証明書は、次の Amazon ルート CA から信頼を得ます。

識別名	暗号化アルゴリズム
CN=Amazon Root CA 1,O=Amazon,C=US	2048 ビット RSA (RSA_2048)
CN=Amazon Root CA 2,O=Amazon,C=US	4096 ビット RSA (RSA_4096)
CN=Amazon Root CA 3,O=Amazon,C=US	Elliptic Prime Curve 256 ビット (EC_prime256v1)
CN=Amazon Root CA 4,O=Amazon,C=US	Elliptic Prime Curve 384 ビット (EC_secp384r1)

ACM が発行した証明書のデフォルトの信頼の基点は CN=Amazon Root CA 1,O=Amazon,C=US で、2048 ビット RSA セキュリティを提供します。他の基点は、将来の使用のために予約されています。すべての基点は、Starfield Services Root Certificate Authority 証明書によってクロス署名されています。

詳細については、Amazon Trust Services を参照してください。

Apex ドメイン

ドメイン名 を参照してください。

非対称キー暗号

非対称暗号化は、対称キー暗号化 と異なり、異なるが数学的に関連しているキーを使用して、コンテンツの暗号化と復号を行います。キーの 1 つはパブリックで、通常、X.509 v3 証明書で使用できます。もう 1 つのキーはプライベートで、安全に保存されます。X.509 証明書は、ユーザー、コンピュータ、その他のリソース (証明書件名) のアイデンティティをパブリックキーにバインドします。

ACM 証明書は、ウェブサイトのアイデンティティと組織の詳細を、証明書に含まれているパブリックキーにバインドする X.509 SSL/TLS 証明書です。ACM は AWS KMS key を使用してプライベートキーを暗号化します。詳細については、「証明書の秘密鍵のセキュリティ」を参照してください。

認証局

認証局 (CA) とは、デジタル証明書を発行するエンティティです。流通している最も一般的なタイプのデジタル証明書は、ISO X.509 規格に基づいています。CA は、証明書件名の ID を肯定するデジタル証明書を発行し、その ID を証明書に含まれるパブリックキーにバインドします。また、通常の場合、CA は証明書の失効も管理します。

証明書の透明性ログ記録

誤って発行された SSL/TLS 証明書や漏洩した CA から保護されるようにするために、一部のブラウザではドメインで発行されたパブリック証明書を証明書の透明性ログに記録する必要があります。ドメイン名が記録されます。プライベートキーは記録されません。通常、ログに記録されない証明書は、ブラウザでエラーを生成します。

ログをモニタリングして、承認した証明書のみがドメインに対して発行されていることを確認できます。Certificate Search などのサービスを使用してログを確認することができます。

Amazon CA がドメインにパブリックに信頼できる SSL/TLS 証明書を発行する前に、少なくとも 3 つの証明書の透明性ログサーバーに証明書を送信します。これらのサーバーは、パブリックデータベースに証明書を追加し、署名付き証明書タイムスタンプ (SCT) を Amazon CA に返します。CA は SCT を証明書に埋め込み、証明書に署名して発行します。タイムスタンプは他の X.509 拡張機能に含まれています。

```
X509v3 extensions:

  CT Precertificate SCTs:
    Signed Certificate Timestamp:
```

```
        Version   : v1(0)
          Log ID    : BB:D9:DF:...8E:1E:D1:85
          Timestamp : Apr 24 23:43:15.598 2018 GMT
          Extensions: none
          Signature : ecdsa-with-SHA256
                      30:45:02:...18:CB:79:2F
      Signed Certificate Timestamp:
        Version   : v1(0)
          Log ID    : 87:75:BF:...A0:83:0F
          Timestamp : Apr 24 23:43:15.565 2018 GMT
          Extensions: none
          Signature : ecdsa-with-SHA256
                      30:45:02:...29:8F:6C
```

オプトアウトする場合を除き、証明書をリクエストまたは更新すると、証明書の透明性ログが自動的に記録されます。オプトアウトの詳細については、証明書の透明性ログ記録のオプトアウト を参照してください。

ドメインネームシステム

ドメインネームシステム (DNS) は、コンピュータおよびインターネットやプライベートネットワークに接続するその他のリソースの階層的な分散命名方式です。DNS は主に、aws.amazon.com などの文字形式のドメイン名を、111.122.133.144 形式の数値的な IP (インターネットプロトコル) アドレスに変換するために使用されます。ただし、ドメインの DNS データベースには多数のレコードが含まれており、他の目的に使用できます。たとえば、ACM では、CNAME レコードを使用して、証明書をリクエストする際にドメインの所有者または管理者であることを検証できます。詳しくは、AWS Certificate Manager DNS 検証 を参照してください。

ドメイン名

ドメイン名は、ドメインネームシステム (DNS) により IP アドレスに変換できる、www.example.com などのテキスト文字列です。インターネットを含むコンピュータネットワークでは、テキスト名ではなく IP アドレスが使用されます。ドメイン名はピリオドで区切られた個別のラベルで構成されます。

TLD

1 番右のラベルは最上位ドメイン (TLD) と呼ばれます。一般的な例には、.com、.net、.edu などがあります。また、一部の国で登録されたエンティティの TLD は国名の略であり、国コードと呼ばれます。たとえば、英国は .uk、ロシアは .ru、フランスは .fr です。国コードを使用する場合、

登録されたエンティティの種類を識別するために、TLD の第 2 レベルの階層がよく使用されます。たとえば、.co.uk という TLD は、英国の営利事業を識別します。

Apex ドメイン

Apex ドメイン名には最上位ドメインが含まれ、そこで拡張されます。国コードを含むドメイン名の場合、apex ドメインには、登録されたエンティティのタイプを識別するコードとラベル (ある場合) が含まれます。apex ドメインには、サブドメインは含まれません (次の段落を参照)。www.example.com では、apex ドメインの名前は example.com です。www.example.co.uk では、apex ドメインの名前は example.co.uk です。apex の代わりによく使用されるその他の名前には、base、bare、root、root apex、zone apex があります。

サブドメイン

サブドメイン名は apex ドメイン名の前に付き、ピリオドでドメイン名から区切られるとともに、相互にも区切られます。最も一般的なサブドメイン名は www ですが、任意の名前を付けることができます。また、サブドメイン名には複数のレベルを含めることができます。たとえば、jake.dog.animals.example.com で、サブドメインは jake、dog、animals の順になります。

スーパードメイン

サブドメインが属するドメイン。

FQDN

完全修飾ドメイン名 (FQDN) は、ネットワークまたはインターネットに接続されるコンピュータ、ウェブサイト、またはその他のリソースの完全な DNS 名です。例 aws.amazon.com は、Amazon Web Services の FQDN です。FQDN には、最上位ドメインまでのすべてのドメインが含まれます。たとえば、[$subdomain_1$].[$subdomain_2$]...[$subdomain_n$].[apex domain].[top-level domain] は FQDN の一般的な形式を表します。

PQDN

完全修飾されていないドメイン名は、部分修飾ドメイン名 (PQDN) と呼ばれ、あいまいです。[$subdomain_1$.$subdomain_2$.] などの名前は、ルートドメインを特定できないため、PQDN です。

暗号化と復号

暗号化とは、データに機密性を提供する処理です。復号化では、このプロセスを逆転して元のデータを復元します。暗号化されていないデータは、テキストであるかどうかに関係なく一般的にプレー

ンテキストと呼ばれます。暗号化されたデータは、通常、暗号文と呼ばれます。クライアントとサーバー間のメッセージの HTTPS 暗号化にはアルゴリズムとキーが使用されます。アルゴリズムは段階的な手順で定義され、これによってプレーンテキストのデータは暗暗号文に変換され (暗号化)、暗号文は元のプレーンテキストに復元されます (復号化)。暗号化または復号化プロセス中にアルゴリズムによってキーが使用されます。キーはプライベートまたはパブリックとなります。

完全修飾ドメイン名 (FQDN)

ドメイン名 を参照してください。

ハイパーテキスト転送プロトコル (HTTP)

ハイパーテキスト転送プロトコル (HTTP) は、World Wide Web 上のデータ通信の基盤です。これは、さまざまな種類のコンテンツの交換を可能にするアプリケーション層プロトコルです。HTTP はクライアントサーバーモデルで動作し、ウェブブラウザーは通常、ウェブサーバーからリソースをリクエストするクライアントとして機能します。HTTP はステートレスプロトコルであり、以前のリクエストの情報を保持せずに各リクエストを個別に処理します。

ACM のコンテキストでは、SSL/TLS 証明書の発行時に HTTP をドメイン検証に使用できます。このプロセスでは、ACM が特定の HTTP リクエストを送信してドメインの所有権を検証します。これらのリクエストに正しく応答するサーバーの機能は、ドメインに対するコントロールを示しています。

E メール証明書や DNS 検証済みの証明書とは異なり、ACM のお客様は ACM から直接 HTTP 検証済みの証明書を発行することはできません。代わりに、これらの証明書は CloudFront プロビジョニングプロセスの一部として自動的に発行および管理されます。お客様は ACM を使用してこれらの証明書を表示、監視、管理できますが、最初の発行は ACM と CloudFront の統合によって処理されます。

HTTP は広く使用されていますが、プレーンテキストでデータを送信することにご注意ください。安全な通信のために、SSL/TLS プロトコルを使用してデータを暗号化する HTTPS (HTTP Secure) が使用されます。安全な内部通信の詳細については、安全な HTTPS を参照してください。

パブリックキーインフラストラクチャ (PKI)

パブリックキーインフラストラクチャ (PKI) は、パブリックネットワーク上で安全な通信を可能にするプロセス、テクノロジ、およびポリシーのシステムです。ACM のコンテキストでは、PKI はデジタル証明書の発行、管理、検証において重要な役割を果たします。PKI は、自由に配布されるパブリックキーと、所有者によって秘密に保持されるプライベートキーの 2 つの暗号化キーを使用しま

す。このシステムにより、安全なデータ転送、デジタル署名、デジタルエンティティの認証が可能になります。

ACM は、PKI のいくつかの重要なコンポーネントを実装します。これは、デジタル証明書を発行し、ドメインや組織などのエンティティにパブリックキーをバインドする信頼できるサードパーティーである認証局 (CA) として機能します。ACM は、エンティティ、そのパブリックキー、および証明書の有効期間に関する情報を含む X.509 証明書を発行します。また、発行、更新、失効など、証明書のライフサイクル全体を処理します。証明書リクエストの正当性を保証するために、ACM は DNS 検証や HTTP 検証など、ドメインの所有権を検証するさまざまな方法をサポートしています。

PKI を活用することで、ACM は AWS リソースとアプリケーションの安全な HTTPS 接続、デジタル署名、暗号化された通信を可能にします。このインフラストラクチャは、インターネット経由で送信されるデータの機密性、整合性、および信頼性を維持する上で不可欠です。ACM が PKI を実装する方法の詳細については、AWS Certificate Manager 証明書の開始方法 を参照してください。

ルート証明書

認証局 (CA) は通常、相互の親子関係を明確に定義した他の複数の CA を含む段階構造内に存在します。子または下位の CA はその親 CA に証明され、証明書チェーンを造り上げます。階層の最上部にある CA はルート CA とされ、その証明書はルート証明書と呼ばれます。この証明書は通常、自己署名されています。

Secure Sockets Layer (SSL)

Secure Sockets Layer (SSL) と Transport Layer Security (TLS) は、コンピュータのネットワークにおける通信の安全性を提供する暗号化プロトコルです。TLS は SSL の後続です。両方とも X.509 証明書を使用してサーバーを認証します。どちらのプロトコルもクライアントとサーバーの間を通信するデータの暗号化に使用される対称キーを使用します。

安全な HTTPS

HTTPS とは、SSL/TLS 経由の HTTP であり、すべての主要なブラウザとサーバーにサポートされる HTTP の安全な形式です。すべての HTTP リクエストとレスポンスは、ネットワーク経由で送信される前に暗号化されます。HTTPS は、HTTP プロトコルに対称、非対称、および X.509 証明書ベースの暗号化を組み合わせた手法を採用しています。HTTPS は、開放型システム間相互接続 (OSI) モデルで HTTP アプリケーション層の下位と TCP トランスポート層の上位に暗号化されたセキュリティ層を挿入することで動作します。このセキュリティ層は、Secure Sockets Layer (SSL) プロトコルまたは Transport Layer Security (TLS) プロトコルを使用します。

SSL サーバー証明書

HTTPS トランザクションは、サーバーを認証するためにサーバー証明書を要求します。サーバー証明書とは、証明書のパブリックキーを証明書の件名にバインドする X.509 v3 データ構造です。SSL/TLS 証明書は認証局 (CA) によって署名され、サーバーの名前、有効期間、パブリックキー、署名アルゴリズムなどを含有します。

対称キー暗号化

対称キー暗号化では、デジタルデータの暗号化と復号化で同じキーが使用されます。非対称キー暗号も参照してください。

Transport Layer Security (TLS)

Secure Sockets Layer (SSL) を参照してください。

信頼

ウェブブラウザがウェブサイトのアイデンティティを信頼するために、ウェブサイトの証明書がブラウザによって検証できる必要があります。しかし、ブラウザは CA ルート証明書と呼ばれるほんのわずかな証明書のみを信頼します。認証機関 (CA) と呼ばれる信頼されるサードパーティは、ウェブサイトのアイデンティティを検証し、ウェブサイト運営者に署名されたデジタル証明書を発行します。こうしてブラウザはデジタル署名を確認して、ウェブサイトのアイデンティティを検証します。検証に成功した場合は、ブラウザのアドレスバーにロックアイコンが表示されます。

ニーズに適した AWS 証明書サービスは何ですか?

AWS は、マネージド X.509 証明書をデプロイするお客様に 2 つのオプションを提供します。ニーズに最適なオプションを選択してください。

1. AWS Certificate Manager (ACM) — このサービスは、TLS を使用して安全なウェブプレゼンスを必要とするエンタープライズのお客様を対象としています。ACM 証明書は、Elastic Load Balancing、Amazon CloudFront、Amazon API Gateway、およびその他の統合 AWS サービスを通じてデプロイされます。この種の代表的なアプリケーションは、重要なトラフィック要件を持つセキュアなパブリックウェブサイトです。ACM は、有効期限が切れる証明書の更新を自動化することで、セキュリティ管理を簡素化します。お客様は、このサービスの対象となります。
2. AWS Private CA—このサービスは、 AWS クラウド内にパブリックキーインフラストラクチャ (PKI) を構築するエンタープライズのお客様を対象とし、組織内でのプライベートな使用を目的と

しています。を使用すると AWS Private CA、独自の認証機関 (CA) 階層を作成し、ユーザー、コンピュータ、アプリケーション、サービス、サーバー、その他のデバイスを認証するための証明書を発行できます。プライベート CA によって発行された証明書をインターネットで使用することはできません。詳細については、「AWS Private CA ユーザーガイド」を参照してください。

AWS Certificate Manager 証明書の開始方法

ACM は、パブリック証明書、プライベート証明書、およびインポートした証明書を管理します。証明書は、インターネットまたは内部ネットワーク内で安全な通信を確立するために使用されます。パブリックに信頼できる証明書を ACM (「ACM 証明書」) から直接リクエストしたり、サードパーティーによって発行されたパブリックに信頼できる証明書をインポートしたりできます。自己署名証明書もサポートされています。組織の内部 PKI をプロビジョニングするには、AWS Private CA によって作成され、管理されているプライベート認証機関 (CA) が署名した ACM 証明書を発行できます。CA は、お客様のアカウントに存在するか、別のアカウントによってお客様と共有される場合があります。

> **Note**
>
> パブリック ACM 証明書は、Nitro Enclave に接続されている Amazon EC2 インスタンスにインストールできます。任意の Amazon EC2 インスタンスで使用する[export a public certificate] (パブリック証明書をエクスポート) することもできます。Nitro Enclave に接続されていない Amazon EC2 インスタンスでのスタンドアロンウェブサーバーのセットアップについては、チュートリアル:「Amazon Linux 2 に LAMP ウェブサーバーをインストールする」またはチュートリアル:「Amazon Linux AMI を使用して LAMP ウェブサーバーをインストールする」を参照してください。

> **Note**
>
> プライベート CA によって署名された証明書はデフォルトでは信頼されないため、管理者はそれらをクライアントの信頼ストアにインストールする必要があります。

証明書の発行を開始するには、 AWS マネジメントコンソールにサインインし、https://console.aws.amazon.com/acm/home で ACM コンソールを開きます。入門者向けページが表示される場合は、[Get Started Now] を選択します。それ以外の場合は、左側のナビゲーションペインの [Certificate Manager] または [プライベート CA] を選択します。

トピック

- を使用するように をセットアップする AWS Certificate Manager

を使用するように をセットアップする AWS Certificate Manager

AWS Certificate Manager (ACM) を使用すると、 AWS ベースのウェブサイトとアプリケーションの SSL/TLS 証明書をプロビジョニングおよび管理できます。ACM を使用して、証明書を作成またはインポートしてから管理します。証明書をウェブサイトまたはアプリケーションにデプロイするには、他の AWS サービスを使用する必要があります。ACM に統合されるサービスについての詳細は、サービスと ACM の統合 を参照してください。次のセクションでは、ACM を使用する前に実行する必要のある手順について説明します。

トピック

- にサインアップする AWS アカウント
- 管理アクセスを持つユーザーを作成する
- ACM のドメイン名を登録する
- (オプション) CAA レコードの設定

にサインアップする AWS アカウント

がない場合は AWS アカウント、次の手順を実行して作成します。

にサインアップするには AWS アカウント

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

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

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

AWS サインアッププロセスが完了すると、 から確認メールが送信されます。https://aws.amazon.com/ の [マイアカウント] をクリックして、いつでもアカウントの現在のアクティビティを表示し、アカウントを管理することができます。

管理アクセスを持つユーザーを作成する

にサインアップしたら AWS アカウント、日常的なタスクにルートユーザーを使用しないように AWS アカウントのルートユーザー、 のセキュリティを確保し AWS IAM アイデンティティセンター、 を有効にして管理ユーザーを作成します。

を保護する AWS アカウントのルートユーザー

1. ルートユーザーを選択し、 AWS アカウント E メールアドレスを入力して、アカウント所有者AWS マネジメントコンソールとして にサインインします。次のページでパスワードを入力します。

   ルートユーザーを使用してサインインする方法については、「AWS サインイン ユーザーガイド」の「ルートユーザーとしてサインインする」を参照してください。
2. ルートユーザーの多要素認証 (MFA) を有効にします。

   手順については、IAM ユーザーガイドの AWS アカウント 「ルートユーザー (コンソール) の仮想 MFA デバイスを有効にする」を参照してください。

管理アクセスを持つユーザーを作成する

1. IAM アイデンティティセンターを有効にします。

   手順については、「AWS IAM アイデンティティセンター ユーザーガイド」の「AWS IAM アイデンティティセンターの有効化」を参照してください。
2. IAM アイデンティティセンターで、ユーザーに管理アクセスを付与します。

   を ID ソース IAM アイデンティティセンターディレクトリ として使用する方法のチュートリアルについては、「 AWS IAM アイデンティティセンター ユーザーガイド」の「デフォルトを使用してユーザーアクセスを設定する IAM アイデンティティセンターディレクトリ」を参照してください。

管理アクセス権を持つユーザーとしてサインインする

- IAM アイデンティティセンターのユーザーとしてサインインするには、IAM アイデンティティセンターのユーザーの作成時に E メールアドレスに送信されたサインイン URL を使用します。

  IAM Identity Center ユーザーを使用してサインインする方法については、AWS サインイン「 ユーザーガイド」の AWS 「 アクセスポータルにサインインする」を参照してください。

追加のユーザーにアクセス権を割り当てる

1. IAM アイデンティティセンターで、最小特権のアクセス許可を適用するというベストプラクティスに従ったアクセス許可セットを作成します。

   手順については、「AWS IAM アイデンティティセンター ユーザーガイド」の「アクセス許可セットを作成する」を参照してください。
2. グループにユーザーを割り当て、そのグループにシングルサインオンアクセス権を割り当てます。

   手順については、「AWS IAM アイデンティティセンター ユーザーガイド」の「グループを追加する」を参照してください。

ACM のドメイン名を登録する

完全修飾ドメイン名 (FQDN) は、`.com` や `.org` などのトップレベルのドメイン拡張子を末尾に付けた、組織や個人のインターネット上の独自の名前です。登録したドメイン名をまだ保持していない場合には、Amazon Route 53 やそのほか多くの商業レジストラからドメイン名を登録できます。一般的には、レジストラのウェブサイトからドメイン名をリクエストします。ドメイン名の登録の有効期限は通常、1〜2 年間となり、期限内に更新する必要があります。

Amazon Route 53 でドメイン名を登録する方法についての詳細は、Amazon Route 53 開発者ガイドの「Amazon Route 53 を使用したドメイン名の登録」を参照してください。

(オプション) CAA レコードの設定

CAA レコードでは、ドメインまたはサブドメインの証明書の発行を許可する認証機関 (CA) を指定します。ACM で使用する CAA レコードを作成すると、間違った CA がドメインの証明書を発行することを防止できます。CAA レコードは、ドメインの所有者であることを検証する要件など、認証機関によって指定されたセキュリティ要件に代わるものではありません。

ACM は、証明書リクエストプロセス中にドメインを検証した後、CAA レコードの有無をチェックして、証明書を発行できるかどうかを確認します。CAA レコードの設定はオプションです。

CAA レコードを設定するときは、次の値を使用します。

flags

ACM で [tag] フィールドの値がサポートされるかどうかを指定します。この値は 0 に設定します。

tag

[tag] フィールドの値は次のいずれかになります。現時点では iodef フィールドは無視されます。

問題

[value] フィールドに指定した ACM CA が、ドメインまたはサブドメインの証明書の発行を許可されていることを示します。

issuewild

[value] フィールドに指定した ACM CA が、ドメインまたはサブドメインのワイルドカード証明書の発行を許可されていることを示します。ワイルドカード証明書はドメインまたはサブドメイン、およびそのすべてのサブドメインに適用されます。HTTP 検証を使用する予定の場合、HTTP 検証ではワイルドカード証明書がサポートされていないため、この設定は適用されません。ワイルドカード証明書の代わりに、DNS または E メール検証を使用します。

値

このフィールドの値は、[tag] フィールドの値によって異なります。この値は、引用符 ("") で囲む必要があります。

[tag] が [issue] の場合

[value] フィールドには CA ドメイン名を指定します。このフィールドには、Amazon CA 以外の CA の名前を指定することができます。ただし、次の 4 つの Amazon CA のいずれかを指定する CAA レコードがない場合、ACM は、ドメインまたはサブドメインに証明書を発行することはできません。

- amazon.com
- amazontrust.com
- awstrust.com
- amazonaws.com

[value] フィールドにセミコロン (;) を指定して、ドメインまたはサブドメインの証明書を発行することを許可された CA はないことを示すことができます。このフィールドは、特定のドメインに対する証明書の発行が不要になった時点で使用します。

[tag] が [issuewild] の場合

[value] フィールドは、値がワイルドカード証明書に適用されること以外は [tag] が [issue] の場合と同じです。

ACM CA 値を含まない issuewild CAA レコードが存在する場合、ACM はワイルドカードを発行できません。issuewildが存在しないが、ACM の発行 CAA レコードがある場合、ワイルドカードが ACM によって発行される場合があります。

Example CAA レコードの例

次の例では、ドメイン名が先頭にあり、その後にレコードタイプ (CAA) が続いています。[flags] フィールドは常に 0 です。[tags] フィールドは、[issue] または [issuewild] にすることができます。フィールドが [issue] のときに、[value] フィールドに CA サーバーのドメイン名を入力した場合、その CAA レコードは、リクエストされた証明書のサーバーによる発行を許可したことを示します。[value] フィールドにセミコロン ";" を入力した場合、その CAA レコードは、証明書の発行を許可された CA はないことを示します。CAA レコードの設定は、DNS プロバイダーによって異なります。

> **Important**
>
> CloudFront で HTTP 検証を使用する場合、HTTP 検証はワイルドカード証明書をサポートしていないため、 issuewild レコードを設定する必要はありません。ワイルドカード証明書の場合は、代わりに DNS または E メール検証を使用します。

```
Domain       Record type  Flags  Tag      Value
example.com.   CAA            0        issue      "SomeCA.com"
```

```
Domain       Record type  Flags  Tag      Value
example.com.   CAA            0        issue      "amazon.com"
```

```
Domain       Record type  Flags  Tag      Value
example.com.   CAA            0        issue      "amazontrust.com"
```

```
Domain       Record type  Flags  Tag      Value
example.com.   CAA            0        issue      "awstrust.com"
```

```
Domain       Record type  Flags  Tag      Value
example.com.   CAA            0        issue      "amazonaws.com"
```

```
Domain       Record type  Flags  Tag      Value
```

```
example.com    CAA            0        issue      ";"
```

DNS レコードを追加または変更する方法の詳細については、DNS プロバイダーに確認してください。Route 53 は CAA レコードをサポートしています。Route 53 が DNS プロバイダーの場合、レコード作成の詳細については、「CAA 形式」を参照してください。

AWS Certificate Manager パブリック証明書

パブリック証明書をリクエストしたら、「AWS Certificate Manager パブリック証明書のドメイン所有権を検証する」の説明に従ってドメインの所有権を検証する必要があります。

パブリック ACM 証明書は X.509 標準に準拠しており、次の制約事項が適用されます。

- 名前: DNS に準拠するサブジェクト名を使用する必要があります。詳しくは、ドメイン名 を参照してください。
- アルゴリズム: 暗号化の場合、証明書のプライベートキーのアルゴリズムは 2048 ビット RSA、256 ビット ECDSA、または 384 ビット ECDSA のいずれかである必要があります。
- 有効期限: 各証明書は 198 日間有効です。
- 更新: ACM は、有効期限の 45 日前にパブリック証明書を自動的に更新しようとします。

> **Note**
>
> パブリック ACM 証明書は、Nitro Enclave に接続されている Amazon EC2 インスタンスにインストールできます。任意の Amazon EC2 インスタンスで使用する[export a public certificate] (パブリック証明書をエクスポート) することもできます。Nitro Enclave に接続されていない Amazon EC2 インスタンスでのスタンドアロンウェブサーバーのセットアップについては、「チュートリアル: Amazon Linux 2 に LAMP ウェブサーバーをインストールする」または「チュートリアル: Amazon Linux AMI を使用した LAMP ウェブサーバーのインストール」を参照してください。

管理者は ACM 条件付きキーポリシーを使用して、エンドユーザーによる新しい証明書の発行方法を制御します。これらの条件付きキーにより、証明書リクエストに関連するドメイン、検証方法、およびその他の属性に制限を設けることができます。証明書をリクエストするときに問題が発生した場合は、証明書のリクエストのトラブルシューティング を参照してください。

を使用してプライベート PKI の証明書をリクエストするには AWS Private CA、「」を参照してくださいでプライベート証明書をリクエストする AWS Certificate Manager。

トピック

- AWS Certificate Manager パブリック証明書の特性と制限
- でパブリック証明書をリクエストする AWS Certificate Manager

- AWS Certificate Manager エクスポート可能なパブリック証明書
- AWS Certificate Manager パブリック証明書のドメイン所有権を検証する

AWS Certificate Manager パブリック証明書の特性と制限

ACM が提供するパブリック証明書には、次の特性と制限があります。これらは ACM によって提供される証明書にのみ適用されます。これらの特徴は、インポートした証明書には適用されない場合があります。

ブラウザとアプリケーションの信頼

ACM 証明書は、Google Chrome、Microsoft Edge、そして Mozilla Firefox および Apple Safari を含むすべての主要なブラウザから信頼されています。ACM 証明書を使用しているサイトに TLS で接続すると、ブラウザにロックアイコンが表示されます。Java は ACM 証明書も信頼します。

認証機関と階層

ACM を通じてリクエストしたパブリック証明書は、Amazon が管理するパブリック認証局 (CA) である Amazon Trust Services から取得されます。Amazon ルート CA 1 ~ 4 は、Starfield G2 Root Certificate Authority - G2 によってクロス署名されています。Starfield ルートは、Android (Gingerbread 以降のバージョン) および iOS (バージョン 4.1 以降) で信頼されています。Amazon ルートは iOS 11 以降で信頼されています。Amazon または Starfield ルートを含むブラウザ、アプリケーション、または OS は、ACM パブリック証明書を信頼します。

ACM は、証明書タイプ (RSA または ECDSA) に基づいてランダムに割り当てられた中間 CA を通じて、お客様にリーフ証明書またはエンドエンティティ証明書を発行します。ACM は、このランダム選択のため、中間 CA 情報を提供しません。

ドメイン検証 (DV)

ACM 証明書はドメイン検証され、ドメイン名のみを識別します。ACM 証明書をリクエストする場合、指定されたすべてのドメインの所有権またはコントロールを証明する必要があります。E メールまたは DNS を使用して所有権を検証できます。詳しくは、AWS Certificate Manager DNS 検証 または AWS Certificate Manager E メール検証 を参照してください。

HTTP 検証

ACM は、CloudFront で使用するパブリック TLS 証明書を発行する際に、ドメイン所有権の検証のための HTTP 検証をサポートします。この方法では、HTTP リダイレクトを使用してドメインの所有権を証明し、DNS 検証と同様の自動更新を提供します。HTTP 検証は現在、CloudFront Distribution Tenants 機能を通じてのみ使用できます。

HTTP リダイレクト

HTTP 検証の場合、ACM は RedirectFrom URL と RedirectTo URL を提供します。ドメイン制御を実証するには、RedirectTo から RedirectFrom へのリダイレクトを設定する必要があります。RedirectFrom URL には検証済みドメインが含まれ、RedirectTo は一意の検証トークンを含む CloudFront インフラストラクチャ内の ACM 制御の口ケーションを指します。

によって管理されます

別のサービスによって管理される ACM の証明書は、ManagedBy フィールドでサービスの ID を示します。CloudFront で HTTP 検証を使用する証明書の場合、このフィールドには「CLOUDFRONT」と表示されます。これらの証明書は CloudFront を通じてのみ使用できます。ManagedBy フィールドは、DescribeCertificate と ListCertificates API、および ACM コンソールの証明書インベントリと詳細ページに表示されます。

ManagedBy フィールドは、「使用できる」属性と相互に排他的です。CloudFront マネージド証明書の場合、他の AWS サービスを通じて新しい使用状況を追加することはできません。これらの証明書は、CloudFront API を通じて追加のリソースでのみ使用できます。

中間 CA とルート CA のローテーション

Amazon は、復元力のある証明書インフラストラクチャを維持するために、事前の通知なしに中間認証機関を廃止することがあります。これらの変更は、お客様には影響しません。詳細については、「Amazon introduces dynamic intermediate certificate authorities」(Amazon が動的中間認証機関を導入) を参照してください。

Amazon がルート CA を中止すると、変更は必要に応じて速やかに行われます。Amazon は、利用可能なすべての方法を使用して、、E メール Health Dashboard、テクニカルアカウントマネージャーへの連絡など、 AWS お客様に通知します。

失効時のファイアウォールアクセス

取り消されたエンドエンティティ証明書は、OCSP と CRL を使用して失効情報を検証し、発行します。一部のお客様のファイアウォールでは、これらのメカニズムを許可するために追加のルールが必要になる場合があります。

失効トラフィックを特定するには、次の URL ワイルドカードパターンを使用します。

- OCSP

  ```
  http://ocsp.?????.amazontrust.com
  ```

  ```
  http://ocsp.*.amazontrust.com
  ```

- CRL

  ```
  http://crl.?????.amazontrust.com/?????.crl

  http://crl.*.amazontrust.com/*.crl
  ```

  アスタリスク (*) は 1 つ以上の英数字、疑問符 (?) は単一の英数字を表し、ハッシュマーク (#) は数字を表します。

キーアルゴリズム

証明書では、アルゴリズムやキーサイズを指定する必要があります。ACM は、以下の RSA および ECDSA パブリックキーアルゴリズムをサポートしています。

- RSA 1024 ビット (RSA_1024)
- RSA 2048 ビット (RSA_2048)*
- RSA 3072 ビット (RSA_3072)
- RSA 4096 ビット (RSA_4096)
- ECDSA 256 ビット (EC_prime256v1)*
- ECDSA 384 ビット (EC_secp384r1)*
- ECDSA 521 ビット (EC_secp521r1)

ACM は、アスタリスク (*) でマークされたアルゴリズムを使用して新しい証明書をリクエストできます。他のアルゴリズムは、インポートされた証明書でのみサポートされます。

> **Note**
>
> AWS Private CA CA によって署名されたプライベート PKI 証明書の場合、署名アルゴリズムファミリー (RSA または ECDSA) は CA のシークレットキーアルゴリズムファミリーと一致する必要があります。

ECDSA キーは同等のセキュリティの RSA キーよりも小さく計算効率も優れていますが、すべてのネットワーククライアントが ECDSA をサポートしているわけではありません。この表は、NIST から適用され、同等のセキュリティ強度における RSA と ECDSA のキーサイズ (ビット単位) を比較しています。

アルゴリズムとキーのセキュリティ比較

セキュリティ強度	RSA キーサイズ	ECDSA キーサイズ
128	3072	256
192	7680	384
256	15360	521

セキュリティの強度は 2 の累乗で、暗号化をを解読するために必要な推測回数に関係します。例えば、3072 ビットの RSA キーと 256 ビットの ECDSA キーはどちらも、2^{128} 回以下の推測で取得できます。

アルゴリズムの選択については、 AWS ブログ記事「How to evaluate and use ECDSA certificates in AWS Certificate Manager」を参照してください。

> **Important**
>
> 統合されたサービス では、リソースへの関連付けがサポートされているアルゴリズムとキーサイズのみが許可されます。サポートは、証明書が IAM にインポートされているか ACM にインポートされているかによって異なります。詳細については、各サービスののドキュメントを参照してください。
>
> - Elastic Load Balancing については、「Application Load Balancer の HTTPS リスナー」を参照してください。
> - CloudFront の場合は、「サポートされる SSL/TLS プロトコルと暗号」を参照してください。

マネージド型更新とデプロイ

ACM は、ACM 証明書の更新とプロビジョニングを管理します。自動更新は、証明書の設定ミス、失効、期限切れによるダウンタイムを防止することができます。詳しくは、でのマネージド証明書の更新 AWS Certificate Manager を参照してください。

複数のドメイン名

各 ACM 証明書には、少なくとも 1 つの完全修飾ドメイン名 (FQDN) が含まれる必要があり、追加の名前を付けくわえることができます。たとえば、`www.example.com` の証明書には

www.example.net を含めることもできます。これは、ベアドメイン (Zone Apexまたは裸ドメイン) にも適用されます。つまり、www.example.com に証明書をリクエストし、example.com という名前を追加できます。詳しくは、AWS Certificate Manager パブリック証明書 を参照してください。

プーニーコード

国際化ドメイン名 については、次の Punycode 要件を満たす必要があります。

1. パターン「<character><character>—」で始まるドメイン名は「xn—」と一致する必要があります。
2. 「xn—」で始まるドメイン名も有効な国際化ドメイン名である必要があります。

Punycode の例

ドメイン名	フルフィル #1	フルフィル #2	許可されています	メモ
example.com	該当なし	該当なし	✓	「<character><character>—」で始まらない
a--example.com	該当なし	該当なし	✓	「<character><character>—」で始まらない
abc--example.com	該当なし	該当なし	✓	「<character><character>—」で始まらない
xn--xyz.com	はい	はい	✓	有効な国際化ドメイン名 (简.com に解決)
xn--example.com	はい	なし	✗	有効な国際化ドメイン名ではありません
ab--example.com	いいえ	いいえ	✗	「xn--」で始まる必要があります。

有効期間

ACM 証明書は 198 日間有効です。

ワイルドカード名

ACM は、ドメイン名にアスタリスク (*) を使うことで、同じドメイン内の複数のサイトを保護するワイルドカード証明書を作成できます。たとえば、*.example.com は、www.example.com と images.example.com を保護します。

ワイルドカード証明書では、アスタリスク (*) はドメイン名の左側に付ける必要があり、1 つのサブドメインレベルのみを保護できます。たとえば、*.example.com は login.example.com と test.example.com を保護しますが、test.login.example.com は保護しません。また、*.example.com は、サブドメインのみを保護し、ベアドメインまたは apex ドメイン (example.com) は保護しないことに注意してください。ベアドメインとそのサブドメインの両方の証明書をリクエストするには、example.com や *.example.com などの複数のドメイン名を指定します。

> **Important**
>
> CloudFront を使用する場合、HTTP 検証ではワイルドカード証明書がサポートされないことにご注意ください。ワイルドカード証明書の場合は、DNS 検証または E メール検証のいずれかを使用する必要があります。自動証明書更新をサポートしているため、DNS 検証をお勧めします。

でパブリック証明書をリクエストする AWS Certificate Manager

AWS Certificate Manager パブリック証明書は、ACM コンソール AWS CLI、または API からリクエストできます。これらの証明書は、統合して使用すること AWS のサービス も、外部で使用するためにエクスポートすることもできます AWS クラウド。

次のリストでは、パブリック証明書と エクスポート可能なパブリック証明書の違いを説明します。

パブリック証明書

Elastic Load Balancing、Amazon CloudFront、Amazon API Gateway AWS のサービス などの統合 で ACM パブリック証明書を使用します。詳細については、「サービスと ACM の統合」を参照してください。

> **Note**
>
> 2025 年 6 月 17 日より前に作成された ACM パブリック証明書は、エクスポートできません。

エクスポート可能なパブリック証明書

エクスポート可能なパブリック証明書は と統合 AWS のサービス されており、外部でも使用できます AWS クラウド。詳細については、「AWS Certificate Manager エクスポート可能なパブリック証明書」および「サービスと ACM の統合」を参照してください。パブリック証明書をエクスポートできるようにするには、新しい ACM パブリック証明書を作成し、エクスポートを有効にする必要があります。

以下のセクションでは、パブリック ACM 証明書をリクエスト、エクスポート、および取り消す方法を説明します。

トピック

- コンソールを使用してパブリック証明書をリクエストする
- CLI を使用してパブリック証明書をリクエストする

コンソールを使用してパブリック証明書をリクエストする

ACM パブリック証明書をリクエストするには (コンソール)

1. AWS マネジメントコンソールにサインインし、https://console.aws.amazon.com/acm/home で ACM コンソールを開きます。

   [Request a certificate] (証明書のリクエスト) を選択します。
2. [Domain names] (ドメイン名) セクションで、ドメイン名を入力します。

   www.example.com のような完全修飾ドメイン名 (FQDN) や **example.com** のようなネイキッドドメインあるいは apex ドメイン名を使用できます。また、同じドメインで複数のサイト名を保護するために、最左位置にアスタリスク (*) をワイルドカードとして使用できます。たとえば、***.example.com** は、**corp.example.com** と **images.example.com** を保護します。ワイルドカード名は、ACM 証明書の [Subject] (件名) フィールドと [Subject Alternative Name] (サブジェクト代替名) 拡張子に表示されます。

ワイルドカード証明書をリクエストする場合、アスタリスク (*) はドメイン名の左側に付ける必要があり、1 つのサブドメインレベルのみを保護できます。たとえば、***.example.com** は **login.example.com** および **test.example.com** を保護できますが、**test.login.example.com** を保護することはできません。また、***.example.com**は、**example.com** のサブドメインのみを保護し、ネイキッドドメインまたは apex ドメイン (**example.com**) は保護しないことに注意してください。両方を保護するには、次のステップを参照してください。

> **Note**
>
> RFC 5280 に準拠している場合、この手順で入力するドメイン名 (技術的には Common Name) の長さは、ピリオドを含む 64 オクテット (文字) を超えることはできません。ただし、後続の各サブジェクト代替名 (SAN) は、次の手順で、長さが最大 253 オクテットまで指定できます。

- 別の名前を追加するには、[Add another name to this certificate] (この証明書に別の名前を追加) を選択し、テキストボックスに名前を入力します。これは、ネイキッドドメインまたは apex ドメイン (**example.com** など) の両方とそのサブドメイン (***.example.com** など) を保護するために役立ちます。

3. ACM エクスポート可能なパブリック証明書を作成する場合は、[Enable export] (エクスポートを有効にする) オプションを選択します。証明書のプライベートキーにアクセスして、 AWS クラウドの外部で使用できます。詳細については、「AWS Certificate Manager エクスポート可能なパブリック証明書」を参照してください。
4. [Validation method] (検証方法) セクションで、必要に応じて [DNS validation – recommended] (DNS 検証 - 推奨) または [Email validation] (E メール検証) を選択します。

> **Note**
>
> DNS 設定を編集できる場合は、E メール検証ではなく DNS ドメイン検証を使用することをお勧めします。DNS 検証には E メール検証と比べていくつかの利点があります。「AWS Certificate Manager DNS 検証」を参照してください。

ACM は証明書を発行する前に、証明書リクエストのドメイン名の所有者または管理者を検証します。E メール検証または DNS 検証のいずれかを使用できます。

a. E メール検証を選択すると、ACM はドメイン名フィールドで指定したドメインに検証 E メールを送信します。検証ドメインを指定すると、ACM はその検証ドメインに E メールを送信します。E メール検証の詳細については、AWS Certificate Manager E メール検証 を参照してください。

b. DNS 検証を使用する場合は、ACM から提供される CNAME レコードを DNS 設定に追加するだけです。DNS 検証の詳細については、AWS Certificate Manager DNS 検証 を参照してください。

5. [Key algorithm] (キーアルゴリズム) セクションで、アルゴリズムを選択します。

6. [Tags] (タグ) ページで、オプションで証明書にタグを付けることができます。タグは、 AWS リソースを識別して整理するためのメタデータとして機能するキーと値のペアです。ACM タグパラメータのリスト、および証明書作成後にタグを追加する方法については、AWS Certificate Manager リソースにタグを付ける を参照してください。

   タグの追加が完了したら、[Request] (リクエスト) を選択します。

7. リクエストが処理されると、コンソールは証明書リストに戻り、リストには新しい証明書の情報が表示されます。

   トラブルシューティングのトピック「証明書のリクエストの失敗」に記載されているいずれかの理由で失敗しない限り、リクエストされると証明書のステータスが [Pending validation] (検証保留中) になります。ACM が証明書の検証を 72 時間繰り返し、タイムアウトします。証明書のステータスが [Failed] (失敗) または [Validation timed out] (検証タイムアウト) の場合、リクエストを削除し、DNS での検証 または E メール検証 で問題を修正してから、再度お試しください。検証が成功すると、証明書のステータスは [Issued] (発行済み) になります。

> **Note**
>
> リストの配列方法によっては、探している証明書がすぐには表示されない場合があります。右側の黒い三角形をクリックすると、配列を変更できます。また、右上のページ番号を使用して、証明書の複数のページを検索することもできます。

CLI を使用してパブリック証明書をリクエストする

request-certificate コマンドを使用して、コマンドラインに新しいパブリック ACM 証明書をリクエストします。検証方法のオプション値は DNS と EMAIL です。キーアルゴリズムのオプション値

は、RSA_2048 (パラメータが明示的に指定されていない場合のデフォルト)、EC_prime256v1、および EC_secp384r1 です。

```
aws acm request-certificate \
--domain-name www.example.com \
--key-algorithm EC_Prime256v1 \
--validation-method DNS \
--idempotency-token 1234 \
--options CertificateTransparencyLoggingPreference=DISABLED,Export=ENABLED
```

このコマンドは、新しいパブリック証明書の Amazon リソースネーム (ARN) を出力します。

```
{
    "CertificateArn": "arn:aws:acm:Region:444455556666:certificate/certificate_ID"
}
```

AWS Certificate Manager エクスポート可能なパブリック証明書

AWS Certificate Manager エクスポート可能なパブリック証明書を使用すると、Amazon EC2 インスタンス、コンテナ、オンプレミスホストなど、どこでも SSL/TLS 証明書をプロビジョニング、管理、デプロイできます。この機能は、ACM が発行したパブリック証明書を統合以上に拡張し AWS のサービス、インフラストラクチャ全体で証明書を一元的に制御できるようにします。

利点

ACM エクスポート可能なパブリック証明書の利点を以下に示します。

- 証明書管理の簡素化: ACM を使用してすべてのリソースの証明書を一元管理します。
- 証明書発行の高速化: 証明書に短時間でアクセスして使用します。
- 自動更新: ACM は証明書の更新を自動的に処理し、新しい証明書をデプロイする準備ができたら通知します。詳しくは、ACM の Amazon EventBridge サポート を参照してください。
- コスト効率: 作成したエクスポート可能なパブリック証明書に対してのみ支払います。
- 柔軟なデプロイ: 標準の SSL/TLS 証明書をサポートするサーバーまたはアプリケーションで証明書 を使用します。

ACM エクスポート可能なパブリック証明書の仕組み

ACM エクスポート可能なパブリック証明書の仕組みを以下に示します。

1. ドメインの ACM を使用して、エクスポート可能な証明書をリクエストします。
2. DNS または E メールの検証を使用してドメインの所有権を検証します。
3. 証明書、秘密キー、および証明書チェーンをエクスポートします。
4. 証明書をサーバー、またはアプリケーションにデプロイします。
5. ACM は更新を管理し、新しい証明書が利用可能になったときに通知を送信します。

セキュリティに関する考慮事項

以下は、ACM エクスポート可能なパブリック証明書を使用する際のセキュリティ上の考慮事項です。詳しくは、でのデータ保護 AWS Certificate Manager を参照してください。

- 安全なストレージとアクセスコントロールを使用して、エクスポートされたプライベートキーを保護します。
- キーの侵害が疑われる場合は、ACM の失効機能を使用します。
- 更新された証明書をデプロイするときに、適切なキーローテーション手順を実装します。

制限事項

ACM 証明書の制限事項を次に示します。

- 証明書の有効期間は 198 日間です。
- ACM は、有効期限の 45 日前に失効するように設定された証明書を更新します。
- エクスポートされた証明書のデプロイプロセスを管理する必要があります。

料金

で作成したエクスポート可能なパブリック SSL/TLS 証明書には追加料金がかかります AWS Certificate Manager。最新の ACM 料金情報については、 AWS ウェブサイトのAWS Certificate Manager 「サービス料金」ページを参照してください。

ベストプラクティス

ACM 証明書を使用する際のベストプラクティスを以下に示します。

- 証明書が更新されたら、すぐに使用を開始する必要があります。

- 更新された証明書の自動デプロイプロセスをテストして実装します。
- Amazon EventBridge メトリクスとアラーム を使用して証明書のデプロイをモニタリングします。

AWS Certificate Manager パブリック証明書をエクスポートする

次の手順では、ACM コンソールで ACM パブリック証明書をエクスポートする方法を説明します。または、 `export-certificate` AWS CLI または ExportCertificate API アクションを使用することもできます。

> **Note**
>
> 2025 年 6 月 17 日より前に作成された ACM パブリック証明書は、エクスポートできません。

パブリック証明書のエクスポート (コンソール)

1. にサインイン AWS マネジメントコンソール し、https://console.aws.amazon.com/acm/ で ACM コンソールを開きます。
2. [List certificates] (証明書を一覧表示する) を選択し、エクスポートする証明書のチェックボックスをオンにします。
   - あるいは、証明書を選択することもできます。証明書の詳細ページで、[Export] (エクスポート) を選択します。
3. [More actions] (その他のアクション) を選択し、 [Export] (エクスポート) を選択します。
4. プライベートキーのパスフレーズを入力して確定します。
5. 証明書ファイルは、ダウンロードまたはコピーできます。

> **Note**
>
> ACM コンソールでは、.pem 証明書ファイルをエクスポートできます。.pem ファイルを .ppk などの別のファイル形式に変換できます。詳細については、「re:Post の記事」を参照してください。

パブリック証明書のエクスポート (AWS CLI)

export-certificate AWS CLI コマンドまたは ExportCertificate API アクションを使用して、パブリック証明書とプライベートキーをエクスポートします。コマンドを実行するときにパスフレーズを割り当てる必要があります。セキュリティを強化するには、ファイルエディタを使用してパスフレーズをファイルに保存し、ファイルを指定することでパスフレーズを指定します。これにより、パスフレーズがコマンド履歴に格納されるのを防ぎ、入力時に他のユーザーがパスフレーズを見るのを防ぎます。

> **Note**
>
> パスフレーズを含むファイルは、ラインターミネータで終了してはなりません。パスワードファイルは、次のように確認できます。

```
$ file -k passphrase.txt
passphrase.txt: ASCII text, with no line terminators
```

次の例では、コマンド出力を jq にパイプして PEM 形式を適用しています。

```
[Windows/Linux]$ aws acm export-certificate \
    --certificate-arn arn:aws:acm:us-east-1:111122223333:certificate/certificate_ID \
    --passphrase fileb://path-to-passphrase-file  \
    | jq -r '"\(.Certificate)\(.CertificateChain)\(.PrivateKey)"'
```

これにより base64 でエンコードされた PEM 形式の証明書が出力されます。これには、次の省略された例のように、証明書チェーンと暗号化されたプライベートキーも含まれます。

```
-----BEGIN CERTIFICATE-----
MIIDTDCCAjSgAwIBAgIRANWuFpqA16g3IwStE3vVpTwwDQYJKoZIhvcNAQELBQAw
EzERMA8GA1UECgwIdHJvbG9sb2wwHhcNMTkwNzE5MTYxNTU1WhcNMjAwODE5MTcx
NTU1WjAXMRUwEwYDVQQDDAx3d3cuc3B1ZHMuaW8wggEiMA0GCSqGSIb3DQEBAQUA
...
8UNFQvNoo1VtICL4cwWOdLOkxpwkkKWtcEkQuHE1v5Vn6HpbfFmxkdPEasoDhthH
FFWIf4/+VOlbDLgjU4HgtmV4IJDtqM9rGOZ42eFYmmc3eQO0GmigBBwwXp3j6hoi
74YM+igvtILnbYkPYhY9qz8h7lHUmannS8j6YxmtpPY=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIC8zCCAdugAwIBAgIRAM/jQ/6h2/MI1NYWX3dDaZswDQYJKoZIhvcNAQELBQAw
```

```
EzERMA8GA1UECgwIdHJvbG9sb2wwHhcNMTkwNjE5MTk0NTE2WhcNMjkwNjE5MjA0
NTE2WjATMREwDwYDVQQKDAh0cm9sb2xvbDCCASIwDQYJKoZIhvcNAQEBBQADggEP
...
j2PAOviqIXjwr08Zo/rTy/8m6LAsmm3LVVYKLyPdl+KB6M/+H93Z1/Bs8ERqqga/
6lfM6iw2JHtkW+q4WexvQSoqRXFhCZWbWPZTUpBS0d4/Y5q92S3iJLRa/JQ0d4U1
tWZyqJ2rj2RL+h7CE71XIAM//oHGcDDPaQBFD2DTisB/+ppGeDuB
-----END CERTIFICATE-----
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFKzBVBgkqhkiG9w0BBQ0wSDAnBgkqhkiG9w0BBQwwGgQUMrZb7kZJ8nTZg7aB
1zmaQh4vwloCAggAMB0GCWCGSAFlAwQBKgQQDViroIHStQgNOjR6nTUnuwSCBNAN
JM4SG202YPUiddWeWmX/RKGg3lIdE+A0WLTPskNCdCAHqdhOSqBwt65qUTZe3gBt
...
ZGipF/DobHDMkpwiaRR5sz6nG4wcki0ryYjAQrdGsR6EVvUUXADkrnrrxuHTWjFl
wEuqyd8X/ApkQsYFX/nhepOEIGWf8Xu0nrjQo77/evhG0sHXborGzgCJwKuimPVy
Fs5kw5mvEoe5DAe3rSKsSUJ1tM4RagJj2WH+BC04SZWNH8kxfOC1E/GSLBCixv3v
+Lwq38CEJRQJLdpta8NcLKnFBwmmVs9OV/VXzNuHYg==
-----END ENCRYPTED PRIVATE KEY-----
```

すべてをファイルに出力するには、前出の例に > リダイレクトを追加し、次の結果を得ます。

```
$ aws acm export-certificate \
     --certificate-arn arn:aws:acm:us-east-1:111122223333:certificate/certificate_ID \
     --passphrase fileb://path-to-passphrase-file \
     | jq -r '"\(.Certificate)\(.CertificateChain)\(.PrivateKey)"' \
     > /tmp/export.txt
```

ACM 証明書を使用して Kubernetes ワークロードを保護する

AWS Controllers for Kubernetes (ACK) で AWS Certificate Manager エクスポート可能なパブリック証明書を使用して、ACM から Kubernetes ワークロードにパブリック TLS 証明書を発行およびエクスポートできます。この統合により、Amazon Elastic Kubernetes Service (Amazon EKS) ポッドを保護し、Kubernetes Ingress で TLS を終了できます。開始するには、GitHub の「ACM Controller for Kubernetes」を参照してください。

AWS Controllers for Kubernetes (ACK) は、Kubernetes API を拡張して、ネイティブ Kubernetes マニフェストを使用して AWS リソースを管理します。ACM 用 ACK サービスコントローラーは、Kubernetes ワークフロー内で証明書のライフサイクル管理を自動化します。Kubernetes で ACM 証明書リソースを作成すると、ACK コントローラーは次のアクションを実行します。

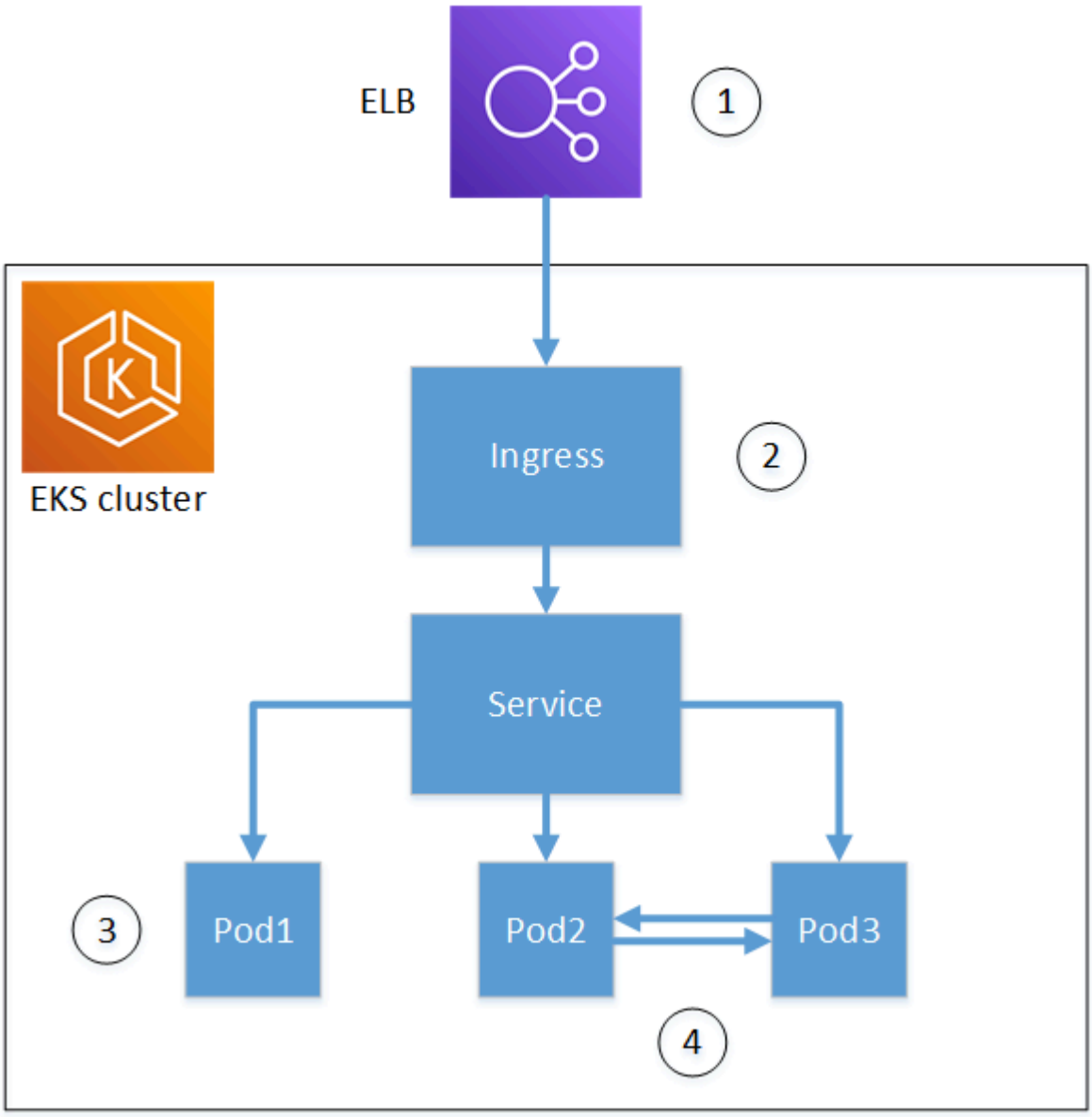
1. 証明書署名リクエスト (CSR) を生成する ACM から証明書をリクエストします。
2. ドメイン検証が完了し、ACM が証明書を発行するのを待ちます。

3. `exportTo` フィールドが指定されている場合、 は発行された証明書とプライベートキーをエクスポートし、指定した Kubernetes シークレットに保存します。
4. `exportTo` フィールドが指定され、証明書が更新対象である場合、 は有効期限が切れる前に更新された証明書で Kubernetes シークレットを更新します。

パブリックに発行された証明書は、ACM が発行する前にドメイン検証が必要です。Amazon Route 53 の ACK サービスコントローラーを使用して、ホストゾーンに必要な DNS 検証 CNAME レコードを自動的に作成できます。

証明書の使用オプション

Kubernetes では、いくつかの方法で ACM 証明書を使用できます。



1. ロードバランサーの終了 (エクスポートなし): ACK を介して証明書を発行し、それらを使用して AWS ロードバランサーで TLS を終了します。証明書は ACM に残り、AWS Load Balancerコントローラーによって自動的に検出されます。この方法では、証明書をエクスポートする必要はありません。
2. イングレス終了 (エクスポートあり): ACM から証明書をエクスポートし、イングレスレベルで TLS 終了のために Kubernetes シークレットに保存します。これにより、Kubernetes ワークロード内で証明書を直接使用できます。

> **Note**
>
> プライベート証明書を必要とするユースケースについては、「証明書マネージャープラグインである AWS Private CA Connector for Kubernetes」を参照してください。

前提条件

ACM 用の ACK サービスコントローラーをインストールする前に、以下があることを確認してください。

- Kubernetes クラスター。
- Helm がインストールされました。
- クラスターと通信できるように kubectl が設定されている。
- eksctl EKS でポッド ID の関連付けを設定するためにインストールされている。

ACM 用の ACK サービスコントローラーをインストールする

Helm を使用して、ACM 用の ACK サービスコントローラーを Amazon EKS クラスターにインストールします。

1. ACK コントローラーの名前空間を作成します。

```
$ kubectl create namespace ack-system --dry-run=client -o yaml | kubectl apply -f -
```

2. ACK コントローラーのポッド ID 関連付けを作成します。*CLUSTER_NAME* をクラスター名に、*REGION* を AWS リージョンに置き換えます。

```
$ eksctl create podidentityassociation --cluster CLUSTER_NAME --region REGION \
    --namespace ack-system \
    --create-service-account \
    --service-account-name ack-acm-controller \
    --permission-policy-arns arn:aws:iam::aws:policy/
AWSCertificateManagerFullAccess
```

3. Amazon ECR Public レジストリにログインします。

```
$ aws ecr-public get-login-password --region us-east-1 | helm registry login --
username AWS --password-stdin public.ecr.aws
```

4. ACM 用の ACK サービスコントローラーをインストールします。*REGION* を自分の AWS リージョンに置き換えます。

```
$ helm install -n ack-system ack-acm-controller oci://public.ecr.aws/
aws-controllers-k8s/acm-chart --set serviceAccount.create=false --set
 serviceAccount.name=ack-acm-controller --set aws.region=REGION
```

5. コントローラーが実行されていることを確認します。

```
$ kubectl get pods -n ack-system
```

ポッド ID の関連付けの詳細については、「Amazon EKS ユーザーガイド」の「EKS Pod Identity」を参照してください。

例: Ingress で TLS を終了する

次の例は、ACM 証明書をエクスポートし、それを使用して Kubernetes Ingress レベルで TLS を終了する方法を示しています。この設定では、ACM 証明書を作成し、Kubernetes シークレットにエクスポートし、TLS 終了に証明書を使用するように Ingress リソースを設定します。

この例では、以下のようになっています：

- エクスポートされた証明書を保存するためにシークレットが作成されます (`exported-cert-secret`)
- ACK Certificate リソースは、ドメインの ACM に証明書をリクエストし、シー`exported-cert-secret`クレットにエクスポートします。
- Ingress リソースは を参照して`exported-cert-secret`、受信トラフィックの TLS を終了します。

`${HOSTNAME}` を独自のドメイン名に置き換えます。

```
apiVersion: v1
kind: Secret
type: kubernetes.io/tls
metadata:
  name: exported-cert-secret
  namespace: demo-app
data:
  tls.crt: ""
```

```
  tls.key: ""
---
apiVersion: acm.services.k8s.aws/v1alpha1
kind: Certificate
metadata:
  name: exportable-public-cert
  namespace: demo-app
spec:
  domainName: ${HOSTNAME}
  options:
    certificateTransparencyLoggingPreference: ENABLED
  exportTo:
    namespace: demo-app
    name: exported-cert-secret
    key: tls.crt
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-traefik
  namespace: demo-app
spec:
  tls:
  - hosts:
    - ${HOSTNAME}
    secretName: exported-cert-secret
  ingressClassName: traefik
  rules:
  - host: ${HOSTNAME}
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: whoami
            port:
              number: 80
```

デプロイされると、ACM の ACK サービスコントローラーは更新を含む証明書のライフサイクルを自動的に管理します。ACM が証明書を更新すると、コントローラーはシーexported-cert-secretクレットを新しい証明書で更新し、Ingress が手動で介入することなく有効な証明書を使用し続けます。

AWS Certificate Manager パブリック証明書を取り消す

AWS Certificate Manager エクスポート可能なパブリック証明書は、ACM コンソール AWS CLI、または API アクションを使用して取り消すことができます。

> **Warning**
> 証明書が取り消された後は、その証明書を再利用することはできません。証明書の取り消しは永久に有効です。

組織のポリシーに準拠したり、キーの侵害を軽減するために、証明書を取り消す必要がある場合があります。証明書を取り消す場合は理由が必要です。次の理由を使用できます。

- 未指定
- 所属が変更されました
- 優先
- オペレーションの停止

詳細については、Amazon Trust Services Certificate Subscriber Agreement および Amazon Trust Service を参照してください。

AWS には、証明書失効をチェックするための 2 つのサービスがあります。オンライン証明書ステータスプロトコル (OCSP) と証明書失効リストです。OCSP では、クライアントは権限のある失効データベースをクエリして、リアルタイムでステータスを返します。OCSP は、証明書に埋め込まれた検証情報に依存します。

考慮事項

証明書を取り消す前に考慮すべき事項を次に示します。

- 取り消すことができるのは、以前にエクスポートされた証明書のみです。
- エクスポート不可能なパブリック証明書 を取り消すことはできません。これらの証明書が不要になった場合は、削除する必要があります。
- 証明書が不要になった場合は、証明書を取り消す代わりに証明書を削除する必要があります。
- 証明書の失効プロセスはグローバルです。取り消すことを選択したすべての有効な証明書は、関連付けられている ARN とともに取り消されます。

- 証明書の失効は永久に有効です。失効した証明書を取得して再利用することはできません。
- 証明書の失効が有効になるまでに、最大 24 時間かかる場合があります。

証明書の取り消し (コンソール)

次の手順では、ACM パブリックまたはプライベート証明書を取り消す方法について説明します。

1. にサインイン AWS マネジメントコンソール し、https://console.aws.amazon.com/acm/ で ACM コンソールを開きます。
2. [List certificates] (証明書を一覧表示する) を選択し、取り消す証明書のチェックボックスを選択します。
   - あるいは、証明書を選択することもできます。証明書の詳細ページで、取り消しを選択します。
3. [More actions] (その他のアクション) を選択し、[Revoke] (取り消し)を選択します。
4. 取り消し理由を入力し、**revoke** と入力して、取り消し を選択する必要があるダイアログボックスが表示されます。

証明書の取り消し (AWS CLI)

`revoke-certificate` AWS CLI コマンドまたは `RevokeCertificate` API アクションを使用して、ACM パブリック証明書またはプライベート証明書を取り消します。`list-certificates` コマンドを呼び出すことで証明書の ARN を取得できます。

```
$ aws acm revoke-certificate \
    --certificate-arn arn:aws:acm:us-
east-1:111122223333:certificate/12345678-1234-1234-1234 \
    --revocation-reason "UNSPECIFIED"
```

> **Warning**
>
> 証明書が取り消された後は、その証明書を再利用することはできません。証明書の取り消しは永久に有効です。

以下は `revoke-certificate` コマンドの出力です。

```
arn:aws:acm:us-east-1:111122223333:certificate/12345678-1234-1234-1234
```

自動更新イベントを設定する

AWS Certificate Manager エクスポート可能なパブリック証明書と Amazon EventBridge を使用すると、証明書の自動更新イベントを設定できます。

1. 証明書の更新を監視するために Amazon EventBridge イベントを設定します。詳細については、「ACM の Amazon EventBridge サポート」を参照してください。
2. 更新が発生したときに証明書のデプロイを処理する自動化を作成します。詳しくは、ACM での Amazon EventBridge を使用してアクションを開始 を参照してください。
3. 更新またはデプロイの失敗を警告するように、EventBridge イベントを設定します。

証明書の更新を強制する

ACM パブリック証明書とプライベート証明書は、ACM コンソール、更新証明書 AWS CLI、または `RenewCertificate` API アクションを使用して更新できます。更新できるのは、以前にエクスポートされた証明書のみです。

> **Important**
>
> ACM エクスポート可能なパブリック証明書を更新すると、追加料金が発生します。最新の ACM 料金情報については、 AWS ウェブサイトのAWS Certificate Manager 「サービス料金」ページを参照してください。

証明書を更新するには (コンソール)

次の手順では、ACM パブリックまたはプライベート証明書を強制的に更新する方法について説明します。

1. にサインイン AWS マネジメントコンソール し、https://console.aws.amazon.com/acm/ で ACM コンソールを開きます。
2. [List certificates] (証証明書を一覧表示する) を選択し、更新する証明書のチェックボックスをオンにします。

- あるいは、証明書を選択することもできます。証明書の詳細ページで、[Renew] (更新) を選択します。

3. [More actions] (その他のアクション) を選択し、[Renew] (更新) を選択します。
4. ダイアログボックスが表示されるので、**renew** と入力して [Renew] (更新) を選択します。

証明書を更新するには (AWS CLI)

renew-certificate AWS CLI コマンドまたは RenewCertificate API アクションを使用して、ACM パブリック証明書またはプライベート証明書を更新します。list-certificates コマンドを呼び出すことで証明書の ARN を取得できます。renew-certificate コマンドはレスポンスを返しません。

```
$ aws acm renew-certificate \
    --certificate-arn arn:aws:acm:us-
east-1:111122223333:certificate/12345678-1234-1234-1234-123456789012
```

AWS Certificate Manager パブリック証明書のドメイン所有権を検証する

Amazon 認証局 (CA) がサイトの証明書を発行する前に、 AWS Certificate Manager (ACM) は、ユーザーがリクエストで指定したすべてのドメイン名の所有者または管理者であることを証明する必要があります。所有権を証明するには、ドメインネームシステム (DNS) 検証、E メール検証、または HTTP 検証を使用して所有権を証明することを選択できます。

> **Note**
>
> 検証は、ACM によって発行されたパブリックに信頼できる証明書にのみ適用されます。ACM は、インポートされた証明書の、またはプライベート CA によって署名された証明書のドメインの所有権を検証しません。ACM は Amazon VPC プライベートホストゾーンまたは他のプライベートドメインのリソースを検証できません。詳しくは、証明書の検証のトラブルシューティング を参照してください。

次の理由により、E メール検証よりも DNS 検証を使用することをお勧めします。

- Amazon Route 53 を使用してパブリック DNS レコードを管理する場合、ACM を使用してレコードを直接更新できます。
- 証明書は使用中で DNS レコードが残っている状態であれば、DNS で検証済みの証明書は ACM によって自動的に更新されます。
- E メール検証された証明書は、ドメイン所有者によるアクションが必要です。ACM は、有効期限切れの 45 日前に更新通知の送信を開始します。これらの通知は、ドメインの 5 つの一般的な管理者アドレスの 1 つまたは複数に送信されます。通知には、ドメイン所有者が簡単に更新するためにクリックできるリンクが含まれています。リストされているすべてのドメインが検証されると、ACM は同じ ARN で更新された証明書を発行します。

ドメインの DNS データベースを編集できない場合は、代わりに E メール検証を使用する必要があります。

HTTP 検証は、CloudFront で使用される証明書で使用できます。この方法では、HTTP リダイレクトを使用してドメインの所有権を証明し、DNS 検証と同様の自動更新を提供します。

> **Note**
>
> E メール検証を使用して証明書を作成した後は、DNS による検証に切り替えることはできません。DNS 検証を使用するには、証明書を削除し、DNS 検証を使用する新しい証明書を作成します。

トピック

- AWS Certificate Manager DNS 検証
- AWS Certificate Manager E メール検証
- AWS Certificate Manager HTTP 検証

AWS Certificate Manager DNS 検証

ドメインネームシステム (DNS) は、ネットワークに接続されているリソースのディレクトリサービスです。DNS プロバイダーは、ドメインを定義するレコードを含むデータベースを維持します。DNS 検証を選択すると、このデータベースに追加する必要がある 1 つ以上の CNAME レコードが ACM から提供されます。これらのレコードには、ドメインを制御する証拠となる一意のキーと値のペアが含まれています。

> **Note**
>
> E メール検証を使用して証明書を作成した後は、DNS による検証に切り替えることはできません。DNS 検証を使用するには、証明書を削除し、DNS 検証を使用する新しい証明書を作成します。

例えば、追加の名前として `example.com` を使用して `www.example.com` ドメインの証明書をリクエストする場合、ACM によって 2 つの CNAME レコードが作成されます。各レコードは、ユーザーのドメインおよびアカウントに固有のものとして作成され、名前と値が含まれます。値は、ACM が証明書を自動的に更新するために使用する AWS ドメインを指すエイリアスです。CNAME レコードを DNS データベースに追加できるのは 1 回のみです。証明書は使用中で CNAME レコードが残っている状態であれば、証明書は ACM によって自動的に更新されます。

> **Important**
>
> パブリック DNS レコードを管理するために Amazon Route 53 を使用しない場合は、レコードの追加方法について DNS プロバイダーに問い合わせてください。ドメインの DNS データベースを編集する権限がない場合は、代わりに E メール検証を使用する必要があります。

検証を繰り返さなくても、CNAME レコードが残っている限り、完全修飾ドメイン名 (FQDN) で追加の ACM 証明書をリクエストできます。つまり、同じドメイン名を持つ置換証明書、または異なるサブドメインを対象とする証明書を作成できます。CNAME 検証トークンはどの AWS リージョンでも機能するため、複数のリージョンで同じ証明書を再作成できます。また、削除された証明書を置き換えることもできます。

自動更新を停止するには、関連付けられている AWS サービスから証明書を削除するか、CNAME レコードを削除します。DNS プロバイダーが Route 53 ではない場合は、レコードを削除する方法をプロバイダーに問い合わせてください。Route 53 がプロバイダーである場合は、Route 53 開発者ガイドの「リソースレコードセットの削除」を参照してください。証明書のマネージド型更新の詳細については、でのマネージド証明書の更新 AWS Certificate Manager を参照してください。

> **Note**
>
> DNS 構成で 5 つを超える CNAME が連結されている場合、CNAME 解決は失敗します。より長いチェーンが必要な場合は、E メール検証を使用することをお勧めします。

ACM の CNAME レコードの 仕組み

> **Note**
>
> このセクションは、Route 53 を DNS プロバイダーとして使用していないユーザーを対象としています。

Route 53 を DNS プロバイダーとして使用していない場合は、ACM から提供された CNAME レコードを、プロバイダーのデータベースに (通常は Web サイトを介して) 手動で入力する必要があります。CNAME レコードは、リダイレクトメカニズムやベンダー固有のメタデータのコンテナーとしてなど、さまざまな目的で使用されます。ACM では、これらのレコードにより、初期ドメイン所有権の検証と継続的な自動証明書の更新が可能になります。

次の表に、6 つのドメイン名に対する CNAME レコードの例を示します。各レコードのレコード名-レコード値ペアは、ドメイン名の所有権を認証する役割を果たします。

表では、最初の 2 つのレコード名-レコード値のペアは同じです。これは、`*.example.com` など、ワイルドカードドメインの場合、ACM によって作成された文字列が、そのベースドメイン `example.com` で作成されたものと同じになることを示します。それ以外の場合は、ペアのレコード名およびレコード値は、ドメイン名ごとに異なります。

CNAME レコードの例

ドメイン名	レコード名	レコード値	Comment
*.example.com	*_x1*.example.com。	*_x2*.acm-validations.aws。	Identical
example.com	*_x1*.example.com。	*_x2*.acm-validations.aws。	
www.example.com	*_x3*.www.example.com。	*_x4*.acm-validations.aws。	Unique
host.example.com	*_x5*.host.example.com。	*_x6*.acm-validations.aws。	Unique
subdomain.example.com	*_x7*.subdomain.example.com。	*_x8*.acm-validations.aws。	Unique

ドメイン名	レコード名	レコード値	Comment
host.subdomain.example.com	_*x9*.host.subdomain.example.com。	_*x10*.acm-validations.aws。	Unique

アンダースコア (_) に続く *xN* の値は、ACM によって生成される長い文字列です。例:

```
_3639ac514e785e898d2646601fa951d5.example.com.
```

が生成される一般的なレコード名です。関連付けされたレコード値は

```
_98d2646601fa951d53639ac514e785e8.acm-validation.aws.
```

同じ DNS レコードで。

> **Note**
>
> DNS プロバイダーがアンダースコアで始まる CNAME 値をサポートしていない場合は、「DNS 検証の問題のトラブルシューティング」を参照してください。

証明書をリクエストし、DNS 検証を指定すると、ACM は次の形式で CNAME 情報を提供します。

ドメイン名	レコード名	レコードタイプ	レコード値
example.com	_a79865eb4cd1a6ab990a45779b4e0b96.example.com。	CNAME	_424c7224e9b0146f9a8808af955727d0.acm-validations.aws。

ドメイン名は、証明書に関連付けられた FQDN です。レコード名は、キーと値のペアのキーとして機能するレコードを一意に識別します。レコード値は、キーと値のペアの値として機能します。

これらの 3 つの値 (ドメイン名、レコード名、レコード値) はすべて、DNS レコードを追加するための DNS プロバイダーのウェブインターフェイスの該当するフィールドに入力する必要があります。

プロバイダーは、レコード名 (または単に「名前」) フィールドの処理に一貫性がありません。場合によっては、上記のように文字列全体を提供することが期待されます。他のプロバイダーは、入力したどの文字列にも自動的にドメイン名を付加します。つまり、(この例では)

```
_a79865eb4cd1a6ab990a45779b4e0b96
```

名前フィールドのみに入力することを意味します。これについて間違っていると思われる場合は、ドメイン名を含むレコード名 (.example.com など) を入力すると、次のようになります。

```
_a79865eb4cd1a6ab990a45779b4e0b96.example.com.example.com.
```

この場合、検証は失敗します。したがって、プロバイダーが期待する入力のタイプを事前に決定する必要があります。

DNS 検証のセットアップ

このセクションでは、DNS 検証を使用するためにパブリック証明書を設定する方法について説明します。

コンソールで DNS 検証を設定するには

> **Note**
>
> この手順では、少なくとも 1 つの証明書がすでに作成されており、それを作成した AWS リージョンで作業していることを前提としています。コンソールを開き、代わりに最初に使用する画面が表示する場合、またはコンソールを正常に開き、一覧に証明書が表示されない場合は、正しいリージョンが指定されていることを確認してください。

1. ACM コンソール (https://console.aws.amazon.com/acm/) を開きます。
2. 証明書のリストで、証明書の設定を行う [Pending validation] (検証保留中) ステータスが付いた証明書の [Certificate ID] (証明書 ID) を選択します。このように、証明書の詳細ページを開きます。
3. [Domains] (ドメイン) セクションで、次の 2 つの手順の 1 つを完了します。

   a. (オプション) Route 53 で検証します。

      次の条件が true である場合に、アクティブな [Create record in Route 53] (Route 53 でレコードを作成) ボタンが表示されます。

- Route 53 を DNS プロバイダーとして使用します。
- Route 53 がホストするゾーンに対する書き込み許可があります。
- FQDN がまだ検証されていません。

> **Note**
>
> 実際に Route 53 を使用しているが、[Create record in Route 53] (Route 53 でレコードを作成) が見つからないか無効になっている場合は、ACM コンソールで [Create record in Route 53] ボタンが表示されない を参照してください。

[Create record in Route 53] (Route 53 でレコードを作成) を選択し、[Create records] (レコードを作成) を選択します。[Certificate status] (証明書のステータス) ページで、[Successfully created DNS records] (DNS レコードが正常に作成された) ことを伝えるステータスバナーと共にページが開くでしょう。

新しい証明書は [Pending validation] (検証保留中) のステータスを最大 30 分間表示し続けます。

> **Tip**
>
> ACM が自動的に Route 53 にレコードを作成するようプログラムによってリクエストすることはできません。ただし、Route 53 を AWS CLI または API コールして、Route 53 DNS データベースにレコードを作成できます。Route 53 レコードセットの詳細については、「リソースレコードセットの使用」を参照してください。

b. (オプション) Route 53 を DNS プロバイダーとして使用していない場合は、CNAME 情報を取得し、それを DNS データベースに追加する必要があります。新しい証明書の詳細ページで次の 2 つのいずれかの方法を使用して、この処理を行うことができます。

- [Domains] (ドメイン) セクションに表示されている CNAME コンポーネントをコピーします。この情報は、DNS データベースに手動で追加する必要があります。
- 別の方法としては、[Export to CSV] (CSV へエクスポート) を選択します。結果ファイル内の情報は、DNS データベースに手動で追加する必要があります。

> **Important**
>
> 検証の問題を回避するには、DNS プロバイダーのデータベースに情報を追加する前に、ACM の CNAME レコードの 仕組み をレビューします。問題が発生した場合は、DNS 検証の問題のトラブルシューティング を参照してください。

CNAME の値を生成してから 72 時間以内に ACM でドメイン名が検証されない場合、ACM では証明書のステータスが [Validation timed out] に変更されます。この結果が生じる主な理由として、DNS 設定を ACM によって生成された値で正常に更新しなかったことが考えられます。この問題を修正するには、CNAME の手順を確認してから新しい証明書をリクエストする必要があります。

AWS Certificate Manager E メール検証

Amazon 認証局 (CA) がサイトの証明書を発行する前に、 AWS Certificate Manager (ACM) は、ユーザーがリクエストで指定したすべてのドメインの所有者または管理者であることを確認する必要があります。E メールまたは DNS のいずれかを使用して検証を実行できます。このトピックでは、E メール検証について説明します。

E メール検証を使用して問題が発生した場合は、E メール検証の問題のトラブルシューティング を参照してください。

E メール検証の仕組み

ACM は、ドメインごとに次の 5 つの一般的なシステム E メールに検証 E メールメッセージを送信します。これらの E メールをスーパードメインで受信する場合は、そのスーパードメインを検証ドメインとして指定することもできます。ベースとなるウェブサイトアドレスまでの任意のサブドメインは有効であり、@ の後に追加されて E メールアドレスのドメインとして使用されます。たとえば、subdomain.example.com の検証ドメインとして example.com を指定すると、admin@example.com に E メールが届きます。

- administrator@your_domain_name
- hostmaster@your_domain_name
- postmaster@your_domain_name
- webmaster@your_domain_name
- admin@your_domain_name

ドメインを所有していることを証明するには、これらの E メールに含まれている検証リンクを選択する必要があります。ACM は、証明書の有効期限の 45 日前に、証明書を更新するために、これらの同じアドレスに検証 E メールを送信します。

ACM API または CLI を使用したマルチドメイン証明書リクエストの E メール検証では、リクエストに他のドメインのサブドメインが含まれていても、リクエストした各ドメインごとに E メールメッセージが送信されます。ドメイン所有者は、ACM が証明書を発行する前に、これらの各ドメインの E メールメッセージを検証する必要があります。

このプロセスの例外

www またはワイルドカードアスタリスク (*) で始まるドメイン名に対して ACM 証明書をリクエストする場合、ACM は、先頭の **www** またはアスタリスクを削除し、管理アドレスに E メールを送信します。これらのアドレスはドメイン名の残りの部分に admin@、administrator@、hostmaster@、postmaster@、および webmaster@ を前置することによって形成されます。例えば、www.example.com に ACM 証明書をリクエストする場合、admin@www.example.com の代わりに admin@example.com に E メールが送信されます。同じように、*.test.example.com に ACM 証明書をリクエストする場合、admin@test.example.com に E メールが送信されます。残りの一般的な管理者アドレスも、同様に形成されます。

> **Important**
>
> ACM は、新しい証明書または更新の WHOIS E メール検証をサポートしなくなりました。一般的なシステムアドレスは引き続きサポートされます。詳細については、ブログ記事 を参照してください。

考慮事項

E メール検証については、次の考慮事項に従ってください。

- E メール検証を使用するには、ドメインに登録されている作業用 E メールアドレスが必要です。E メールアドレスの設定手順は、このガイドの対象外です。
- 検証は、ACM によって発行されたパブリックに信頼できる証明書にのみ適用されます。ACM は、インポートされた証明書の、またはプライベート CA によって署名された証明書のドメインの所有権を検証しません。ACM は Amazon VPC プライベートホストゾーンまたは他のプライベートドメインのリソースを検証できません。詳しくは、証明書の検証のトラブルシューティング を参照してください。

- E メール検証を使用して証明書を作成した後は、DNS による検証に切り替えることはできません。DNS 検証を使用するには、証明書を削除し、DNS 検証を使用する新しい証明書を作成します。

証明書の有効期限切れと更新

ACM 証明書は 198 日間有効です。証明書を更新するには、ドメイン所有者によるアクションが必要です。ACM は、有効期限切れの 45 日前にドメインに関連付けられた E メールアドレスに更新通知の送信を開始します。通知には、ドメイン所有者が更新するためにクリックできるリンクが含まれています。リストされているすべてのドメインが検証されると、ACM は同じ ARN で更新された証明書を発行します。

(オプション) 検証 E メールの再送信

各検証 E メールには、証明書リクエストの承認に使用できるトークンが含まれています。ただし、承認プロセスに必要な検証 E メールがスパムフィルターによってブロックされたり、転送中に紛失した場合、トークンは 72 時間後に自動的に有効期限切れになります。元の E メールを受信しなかった場合、またはトークンの期限が切れた場合は、E メールの再送信をリクエストできます。検証 E メールを再送信する方法については、「検証 E メールを再送信する」を参照してください

E メール検証に関する永続的な問題については、に関する問題のトラブルシューティング AWS Certificate Manager の E メール検証の問題のトラブルシューティング セクションを参照してください。

E AWS Certificate Manager メール検証を自動化する

E メール検証された ACM 証明書には、通常、ドメイン所有者による手動による操作が必要です。大量の E メール検証された証明書を扱う組織は、必要なレスポンスを自動化できるパーサーを作成することを優先する場合があります。E メール検証を使用するユーザーを支援するために、このセクションの情報は、ドメイン検証 E メールメッセージに使用されるテンプレートと、検証プロセスの完了に関連するワークフローについて説明します。

検証 E メールテンプレート

検証 E メールメッセージは、新しい証明書が要求されるか、既存の証明書が更新されるかに応じて、次の 2 つのいずれかの形式になります。強調表示された文字列の内容は、検証対象のドメインに固有の値に置き換える必要があります。

新しい証明書を検証する

E メールテンプレートのテキスト:

```
Greetings from Amazon Web Services,

We received a request to issue an SSL/TLS certificate for requested_domain.

Verify that the following domain, AWS account ID, and certificate identifier
 correspond
to a request from you or someone in your organization.

Domain: fqdn
AWS account ID: account_id
AWS Region name: region_name
Certificate Identifier: certificate_identifier

To approve this request, go to Amazon Certificate Approvals
(https://region_name.acm-certificates.amazon.com/approvals?
code=validation_code&context=validation_context)
and follow the instructions on the page.

This email is intended solely for authorized individuals for fqdn. To express any
 concerns
about this email or if this email has reached you in error, forward it along with a
 brief
explanation of your concern to validation-questions@amazon.com.

Sincerely,
Amazon Web Services
```

更新のための証明書の検証

E メールテンプレートのテキスト:

```
Greetings from Amazon Web Services,

We received a request to issue an SSL/TLS certificate for requested_domain.
This email is a request to validate ownership of the domain in order to renew
the existing, currently in use, certificate. Certificates have defined
validity periods and email validated certificates, like this one, require you
to re-validate for the certificate to renew.
```

```
Verify that the following domain, AWS account ID, and certificate identifier
correspond to a request from you or someone in your organization.

Domain: fqdn
AWS account ID: account_id
AWS Region name: region_name
Certificate Identifier: certificate_identifier

To approve this request, go to Amazon Certificate Approvals at
https://region_name.acm-certificates.amazon.com/approvals?code=
$validation_code&context=$validation_context
and follow the instructions on the page.

This email is intended solely for authorized individuals for fqdn. You can see
more about how AWS Certificate Manager validation works here -
https://docs.aws.amazon.com/acm/latest/userguide/email-validation.html.
To express any concerns about this email or if this email has reached you in
error, forward it along with a brief explanation of your concern to
validation-questions@amazon.com.

Sincerely,
Amazon Web Services

--
Amazon Web Services, Inc. is a subsidiary of Amazon.com, Inc. Amazon.com is a
registered trademark of Amazon.com, Inc.

This message produced and distributed by Amazon Web Services, Inc.,
410 Terry Ave. North, Seattle, WA 98109-5210.

(c)2015-2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Our privacy policy is posted at https://aws.amazon.com/privacy
```

から新しい検証メッセージを受け取ったら AWS、パーサーのup-to-date信頼できるテンプレートとして使用することをお勧めします。2020 年 11 月より前に設計されたメッセージパーサーを持っているユーザーは、テンプレートに対して行われた次の変更に注意する必要があります。

- E メール件名は、「Certificate request for *domain name*」ではなく、「"Certificate approval for *domain name*」のようになります。
- AWS account ID がダッシュやハイフンなしで表示されるようになりました。
- Certificate Identifier では、短縮形式の代わりに証明書 ARN 全体が表示されます。たとえば、*3b4d78e1-0882-4f51-954a-298ee44ff369* ではなく、*arn:aws:acm:us-*

east-1:000000000000:certificate/3b4d78e1-0882-4f51-954a-298ee44ff369 となります。

- 証明書の承認 URL に、`certificates.amazon.com` ではなく、`acm-certificates.amazon.com` が含まれるようになりました。
- 証明書の承認 URL をクリックして開いた承認フォームに、承認ボタンが含まれるようになりました。承認ボタン div の名前が、`approval_button` ではなく、`approve-button` となりました。
- 新しくリクエストされた証明書と更新証明書の両方の検証メッセージは、同じ E メール形式です。

検証ワークフロー

このセクションでは、E メール検証された証明書の更新ワークフローについて説明します。

- ACM コンソールがマルチドメイン証明書リクエストを処理すると、パブリック証明書をリクエストするときに指定したドメイン名または検証ドメインに検証 E メールメッセージを送信します。ドメイン所有者は、ACM が証明書を発行する前に、各ドメインの E メールメッセージを検証する必要があります。詳細については、「E メールを使用したドメインの所有権の検証」を参照してください。
- ACM API または CLI を使用したマルチドメイン証明書リクエストの E メール検証では、リクエストに他のドメインのサブドメインが含まれていても、リクエストした各ドメインごとに E メールメッセージが送信されます。ドメイン所有者は、ACM が証明書を発行する前に、これらの各ドメインの E メールメッセージを検証する必要があります。

  ACM コンソールを使用して既存の証明書の E メールを再送信すると、元の証明書リクエストで指定された検証ドメイン、または検証ドメインが指定されていない場合は対象のドメインに E メールが送信されます。別のドメインで検証 E メールを受信するには、検証に使用する検証ドメインを指定して、新しい証明書をリクエストできます。または、API、SDK、または CLI を使用して、`ValidationDomain` パラメータを使用して ResendValidationEmail を呼び出すこともできます。ただし、`ResendValidationEmail` リクエストで指定された検証ドメインは、その呼び出しにのみ使用され、今後の検証 E メール用に証明書 Amazon リソースネーム (ARN) には保存されません。元の証明書リクエストで指定されていないドメイン名で検証 E メールを受信するたびに、`ResendValidationEmail` を呼び出す必要があります。

> **Note**
>
> 2020 年 11 月以前は、apex ドメインのみを検証する必要がありました。ACM は、サブドメインもカバーする証明書を発行していました。それ以前に設計されたメッセージパー

サーを使用しているユーザーは、E メール検証ワークフローの変更に注意する必要があります。

- ACM API または CLI を使用すると、マルチドメイン証明書リクエストに関するすべての検証 E メールメッセージを apex ドメインに送信できます。API では、RequestCertificate アクションの DomainValidationOptions パラメータを使用して ValidationDomain の値を指定します。これは、DomainValidationOption タイプのメンバーです。CLI では、request-certificate コマンドの --domain-validation-options パラメータを使用して、ValidationDomain の値を指定します。

AWS Certificate Manager HTTP 検証

ハイパーテキスト転送プロトコル (HTTP) は、World Wide Web 上のデータ通信の基本プロトコルです。CloudFront で使用される証明書の HTTP 検証を選択すると、ACM はこのプロトコルを活用してドメインの所有権を検証します。ACM は CloudFront と連携して、特定の URL と、ドメイン上のその URL でアクセスできるようにする必要がある一意のトークンを提供します。このトークンは、ドメインを制御する証拠として機能します。ドメインから CloudFront インフラストラクチャ内の ACM 制御の口ケーションへのリダイレクトを設定することで、ドメインのコンテンツを変更し、所有権を検証する能力を示します。ACM と CloudFront のシームレスな統合により、特に CloudFront ディストリビューションの証明書発行プロセスが簡素化されます。

Important

HTTP 検証は、ワイルドカードドメイン証明書 (*.example.com など) をサポートしていません。ワイルドカード証明書の場合は、代わりに DNS 検証または E メール検証を使用する必要があります。

例えば、CloudFront を使用して追加名として www.example.com を持つ example.com ドメインの証明書をリクエストすると、ACM は HTTP 検証用の 2 セットの URLs を提供します。各セットには、ドメインと AWS アカウント専用に作成された redirectFrom URL と redirectTo URL が含まれています。redirectFrom URL は、設定する必要があるドメイン (http://example.com/.well-known/pki-validation/example.txt など) のパスです。redirectTo URL は、一意の検証トークンが保存されている CloudFront インフラストラクチャ内の ACM 制御の口ケーションを指します。これらのリダイレクトは 1 回だけ設定する必要があります。認証機関がドメインの所有権を検証しようとすると、CloudFront が redirectFrom URL にリダイレクトする redirectTo URL からファイルをリクエストし、検証トークンへのアクセスを許可します。証明書

が CloudFront で使用されており、リダイレクトが維持されている限り、ACM は証明書を自動的に更新します。

CloudFront で完全修飾ドメイン名 (FQDN) の HTTP 検証を設定したら、HTTP リダイレクトが設定されている限り、検証プロセスを繰り返すことなく、その FQDN の追加の ACM 証明書をリクエストできます。つまり、同じドメイン名で代替証明書を作成できます。リダイレクトがまだアクティブであれば、検証プロセスを再度実行せずに、削除された証明書を置き換えることもできます。

HTTP 検証済みの証明書の自動更新を停止するには、2 つのオプションがあります。証明書は、関連付けられている CloudFront ディストリビューションから削除するか、検証用に設定した HTTP リダイレクトを削除できます。CloudFront 以外のコンテンツ配信ネットワーク (CDN) またはウェブサーバーを使用してリダイレクトを管理している場合は、そのドキュメントを参照してリダイレクトを削除する方法を確認してください。CloudFront を使用してリダイレクトを管理している場合は、ディストリビューションの設定を更新することでリダイレクトを削除できます。証明書のマネージド型更新の詳細については、でのマネージド証明書の更新 AWS Certificate Manager を参照してください。自動更新を停止すると証明書の有効期限が切れ、HTTPS トラフィックが中断される可能性があることにご注意ください。

ACM の HTTP リダイレクトの仕組み

> **Note**
>
> このセクションは、コンテンツ配信に CloudFront を使用し、SSL/TLS 証明書管理に ACM を使用しているお客様を対象としています。

ACM と CloudFront で HTTP 検証を使用する場合は、HTTP リダイレクトを設定する必要があります。これらのリダイレクトにより、ACM は最初の証明書発行と継続的な自動更新のためにドメインの所有権を検証できます。リダイレクトメカニズムは、ドメイン上の特定の URL を、一意の検証トークンが保存されている CloudFront インフラストラクチャ内の ACM 制御の口ケーションを指すことによって機能します。

次の表は、ドメイン名のリダイレクト設定の例を示しています。HTTP 検証はワイルドカードドメイン (*.example.com など) をサポートしていないことにご注意してください。各設定の Redirect From-Redirect To ペアは、ドメイン名の所有権を認証する役割を果たします。

HTTP リダイレクト設定の例

ドメイン名	Redirect From	Redirect To	Comment
example.com	`http://example.com/.well-known/pki-validation/ x2.txt`	`https://validation. region.acm-validations.aws/ y2/.well-known/pki-validation/ x2.txt`	Unique
www.example.com	`http://www.example.com/.well-known/pki-validation/ x3.txt`	`https://validation. region.acm-validations.aws/ y3/.well-known/pki-validation/ x3.txt`	Unique
host.example.com	`http://host.example.com/.well-known/pki-validation/ x4.txt`	`https://validation. region.acm-validations.aws/ y4/.well-known/pki-validation/ x4.txt`	Unique
subdomain.example.com	`http://subdomain.example.com/.well-known/pki-validation/ x5.txt`	`https://validation. region.acm-validations.aws/ y5/.well-known/pki-validation/ x5.txt`	Unique
host.subdomain.example.com	`http://host.subdomain.example.com/.well-known/pki-validation/ x6.txt`	`https://validation. region.acm-validations.aws/ y6/.well-known/pki-validation/ x6.txt`	Unique

ファイル名の *xN* 値と ACM 制御ドメインの *yN* 値は、ACM によって生成される一意の識別子です。例えば、

```
http://example.com/.well-known/pki-validation/3639ac514e785e898d2646601fa951d5.txt
```

は生成されたRedirect From URL の代表例です。関連する Redirect To URL は

```
https://validation.region.acm-validations.aws/98d2646601fa/.well-known/pki-
validation/3639ac514e785e898d2646601fa951d5.txt
```

同じ検証レコードである可能性があります。

> **Note**
>
> ウェブサーバーまたはコンテンツ配信ネットワークが指定されたパスでのリダイレクトの設定をサポートしていない場合は、HTTP 検証問題のトラブルシューティング を参照してください。

証明書をリクエストし、HTTP 検証を指定すると、ACM は次の形式でリダイレクト情報を提供します。

ドメイン名	Redirect To
example.com	https://validation.*region*.acm-validations.aws/*a424c7224e9b* /.well-known/pki-validation/*a79865eb4cd1a6ab990a45779b4e0b96* .txt

ドメイン名	Redirect To

ドメイン名は、証明書に関連付けられた FQDN です。Redirect From は、ACM が検証ファイルを検索するドメイン上の URL です。Redirect To は、実際の検証ファイルがホストされている ACM 制御の URL です。

Redirect From URL から Redirect To URL にリクエストをリダイレクトするようにウェブサーバーまたは CloudFront ディストリビューションを設定する必要があります。このリダイレクトをセットアップする正確な方法は、ウェブサーバーソフトウェアまたは CloudFront 設定によって異なります。ACM がドメインの所有権を検証し、証明書を発行または更新できるように、リダイレクトが正しく設定されていることを確認します。

HTTP 検証の設定

ACM は、CloudFront で使用するパブリック SSL/TLS 証明書を発行するときに、HTTP 検証を使用してドメインの所有権を検証します。このセクションでは、HTTP 検証を使用するためにパブリック証明書を設定する方法について説明します。

コンソールで HTTP 検証を設定するには

> **Note**
>
> この手順では、CloudFront を通じて証明書をリクエスト済みであり、それを作成した AWS リージョンで作業していることを前提としています。HTTP 検証は、CloudFront Distribution Tenants 機能を通じてのみ使用できます。

1. ACM コンソール (https://console.aws.amazon.com/acm/) を開きます。
2. 証明書のリストで、証明書の設定を行う [Pending validation] (検証保留中) ステータスが付いた証明書の [Certificate ID] (証明書 ID) を選択します。このように、証明書の詳細ページを開きます。
3. ドメインセクションには、証明書リクエストの各ドメインのRedirect FromとRedirect Toの値が表示されます。

4. ドメインごとに、Redirect From URL からRedirect To URL への HTTP リダイレクトを設定します。これを行うには、CloudFront ディストリビューション設定を使用します。
5. Redirect From URL から Redirect To URL にリクエストをリダイレクトするように CloudFront ディストリビューションを設定します。このリダイレクトを設定する方法は、CloudFront 設定によって異なります。
6. リダイレクトを設定すると、ACM は自動的にドメインの所有権の検証を試みます。このプロセスには最長 30 分かかることがあります。

ACM がリダイレクト値を生成してから 72 時間以内にドメイン名を検証できない場合、ACM では証明書のステータスが [Validation timed out] に変更されます。この結果の最も可能性の高い理由は、HTTP リダイレクトが正常に設定されなかったことです。この問題を解決するには、リダイレクトの手順を確認した後、新しい証明書をリクエストする必要があります。

Important

検証の問題を回避するには、Redirect From のロケーションのコンテンツが Redirect To のロケーションのコンテンツと一致することを確認してください。問題が発生した場合は、HTTP 検証に関する問題のトラブルシューティング を参照してください。

Note

DNS 検証とは異なり、ACM が HTTP リダイレクトを自動的に作成するようにプログラムでリクエストすることはできません。これらのリダイレクトは、CloudFront ディストリビューション設定を通じて設定する必要があります。

HTTP 検証の仕組みの詳細については、ACM の HTTP リダイレクトの仕組み を参照してください。

のプライベート証明書 AWS Certificate Manager

によって作成された既存のプライベート CA にアクセスできる場合 AWS Private CA、 AWS Certificate Manager (ACM) はプライベートキーインフラストラクチャ (PKI) での使用に適した証明書をリクエストできます。CA は、お客様のアカウントに存在するか、別のアカウントによってお客様と共有される場合があります。プライベート CA 作成の詳細については、プライベート証明書機関の作成を参照してください。

デフォルトでは、プライベート CA によって署名された証明書は信頼されないため、ACM はそれらに対する検証を一切サポートしていません。そのため、管理者は、組織のクライアントの信頼ストアにインストールするためのアクションを実行する必要があります。

プライベート ACM 証明書は X.509 標準に準拠しており、次の制約事項が適用されます。

- 名前: DNS に準拠するサブジェクト名を使用する必要があります。詳しくは、ドメイン名 を参照してください。
- アルゴリズム: 暗号化の場合、証明書のプライベートキーのアルゴリズムは 2048 ビット RSA、256 ビット ECDSA、または 384 ビット ECDSA のいずれかである必要があります。

  > **Note**
  >
  > 指定された署名アルゴリズムファミリー (RSA または ECDSA) は、CA のシークレットキーのアルゴリズムファミリーと一致する必要があります。

- 有効期限: 各プライベート証明書は 13 か月間 (395 日間) 有効です。署名した CA 証明書の終了日は、リクエストした証明書の終了日より後になっている必要があります。以前になっていると、証明書リクエストは失敗します。

  > **Note**
  >
  > プライベート証明書の有効期間は、パブリック証明書よりも長くなります。パブリック ACM 証明書は 198 日間有効です。パブリック証明書の詳細については、「」を参照してくださいでパブリック証明書をリクエストする AWS Certificate Manager。

- 更新: ACM は 11 か月後にプライベート証明書を自動的に更新しようとします。

エンドエンティティ証明書の署名に使用されるプライベート CA には、次に示す独自の制限が適用されます。

- CA はステータスがアクティブである必要があります。

> **Note**
>
> パブリックに信頼された証明書とは異なり、プライベート CA によって署名された証明書は、検証の必要がありません。

トピック

- AWS Private CA を使用して ACM プライベート証明書に署名するための条件
- でプライベート証明書をリクエストする AWS Certificate Manager
- AWS Certificate Manager プライベート証明書をエクスポートする

AWS Private CA を使用して ACM プライベート証明書に署名するための条件

AWS Private CA を使用して、次の 2 つのケースのいずれかで ACM 証明書に署名できます。

- 単一アカウント: 同じ AWS アカウントにある署名 CA と発行された AWS Certificate Manager (ACM) 証明書。

  単一アカウントの発行と更新を有効にするには、 AWS Private CA 管理者が ACM サービスプリンシパルに証明書を作成、取得、および一覧表示するためのアクセス許可を付与する必要があります。これは、API アクション CreatePermission または AWS CLI コマンド create-permission を使用して AWS Private CA 行われます。アカウント所有者は、証明書の発行を担当する IAM ユーザー、グループ、またはロールに、これらのアクセス許可を割り当てます。
- クロスアカウント: 署名 CA と発行された ACM 証明書は異なる AWS アカウントに存在し、CA へのアクセスは証明書が存在するアカウントに付与されています。

  クロスアカウント発行と更新を有効にするには、管理者は AWS Private CA AWS Private CA API アクション PutPolicy または AWS CLI コマンド put-policy を使用してリソースベースのポリシーを CA にアタッチする必要があります。このポリシーは、CA への制限付きアクセスを許可する

他のアカウントのプリンシパルを指定します。詳細については、「ACM Private CA でのリソースベースのポリシーの使用」を参照してください。

クロスアカウントシナリオでは、ACM がプリンシパルとして PCA ポリシーとやり取りするサービスリンクロール (SLR) をセットアップする必要もあります。ACM は、最初の証明書の発行時に SLR を自動的に作成します。

ACM では、アカウントに SLR が存在するかどうかを判断できないという警告が表示されることがあります。必要な `iam:GetRole` アクセス許可がすでにアカウントの ACM SLR に付与されている場合、SLR の作成後にアラートは再発しません。再発する場合は、ユーザーまたはアカウント管理者が `iam:GetRole` アクセス許可を ACM に付与するか、アカウントを ACM 管理ポリシー `AWSCertificateManagerFullAccess` に関連付けます。

詳細については、「ACM でのサービスにリンクされたロールの使用」を参照してください。

> **Important**
>
> ACM 証明書は、自動的に更新する前に、サポートされている AWS サービスにアクティブに関連付ける必要があります。ACM がサポートするリソースについては、サービスと ACM の統合 を参照してください。

でプライベート証明書をリクエストする AWS Certificate Manager

プライベート証明書のリクエスト (コンソール)

1. AWS マネジメントコンソールにサインインし、https://console.aws.amazon.com/acm/home で ACM コンソールを開きます。

   [Request a certificate] (証明書のリクエスト) を選択します。

2. [Request certificate] (証明書のリクエスト) ページで、[Request a private certificate] (プライベート証明書のリクエスト) と [Next] (次へ) を選択して続行します。

3. [Certificate authority details] (認証権限の詳細) セクションで、[Certificate authority] (認証権限) メニューを選択し、使用可能なプライベート CA の 1 つを選択します。CA が別のアカウントから共有されている場合、ARN には所有権情報が付加されます。

   CA に関する詳細が表示され、正しい CA を選択したことについての確認に役立ちます。

- [所有者]
- タイプ
- 共通名 (CN)
- 組織 (O)
- 組織単位 (OU)
- 国名 (C)
- 州または県
- 市区町村

4. [Domain names] (ドメイン名) セクションで、ドメイン名を入力します。**www.example.com** のような完全修飾ドメイン名 (FQDN) や **example.com** のようなネイキッドドメインあるいは apex ドメイン名を使用できます。また、同じドメインで複数のサイト名を保護するために、最左位置にアスタリスク (*) をワイルドカードとして使用できます。たとえば、***.example.com** は、**corp.example.com** と **images.example.com** を保護します。ワイルドカード名は、ACM 証明書の [Subject] (件名) フィールドと [Subject Alternative Name] (サブジェクト代替名) 拡張子に表示されます。

   > **Note**
   >
   > ワイルドカード証明書をリクエストする場合、アスタリスク (*) はドメイン名の左側に付ける必要があり、1 つのサブドメインレベルのみを保護できます。たとえば、***.example.com** は **login.example.com** および **test.example.com** を保護できますが、**test.login.example.com** を保護することはできません。また、***.example.com**は、**example.com** のサブドメインのみを保護し、ネイキッドドメインまたは apex ドメイン (**example.com**) は保護しないことに注意してください。両方を保護するには、次のステップを参照してください。

   オプションで、[この証明書に別の名前を追加] を選択し、テキストボックスに名前を入力します。これは、ネイキッドドメインまたは apex ドメイン (**example.com** など) の両方とそのサブドメイン (***.example.com** など) の認証のために役立ちます。

5. [Key algorithm] (キーアルゴリズム) セクションで、アルゴリズムを選択します。

   アルゴリズムの選択に役立つ情報については、 AWS ブログ記事「 で ECDSA 証明書を評価して使用する方法 AWS Certificate Manager」を参照してください。

6. [Tags] (タグ) セクションで、オプションで証明書にタグを付けることができます。タグは、AWS リソースを識別して整理するためのメタデータとして機能するキーと値のペアです。ACM タグパラメータのリスト、および証明書作成後にタグを追加する方法については、AWS Certificate Manager リソースにタグを付ける を参照してください。
7. [Certificate renewal permissions] (証明書の更新許可) セクションで、証明書の更新許可に関する通知を確認します。これらの許可により、選択した CA で署名するプライベート PKI 証明書を自動的に更新できます。詳細については、「ACM でのサービスにリンクされたロールの使用」を参照してください。
8. 必要な情報をすべて提供して、[Request] (リクエスト) を選択します。コンソールによって証明書リストに戻り、新しい証明書を表示できます。

> **Note**
>
> リストの配列方法によっては、探している証明書がすぐには表示されない場合があります。右側の黒い三角形をクリックすると、配列を変更できます。また、右上のページ番号を使用して、証明書の複数のページを検索することもできます。

プライベート証明書のリクエスト (CLI)

ACM で request-certificate コマンドを使用してプライベート証明書をリクエストします。

> **Note**
>
> CA によって署名されたプライベート PKI 証明書をリクエストする場合 AWS Private CA、指定された署名アルゴリズムファミリー (RSA または ECDSA) は CA のシークレットキーのアルゴリズムファミリーと一致する必要があります。

```
aws acm request-certificate \
--domain-name www.example.com \
--idempotency-token 12563 \
--certificate-authority-arn arn:aws:acm-pca:Region:444455556666:\
certificate-authority/CA_ID
```

このコマンドは、新しいプライベート証明書の Amazon リソースネーム (ARN) を出力します。

```
{
```

```
    "CertificateArn": "arn:aws:acm:Region:444455556666:certificate/certificate_ID"
}
```

ほとんどの場合、ACM は、共有 CA を初めて使用するときに、サービスにリンクされたロール (SLR) をアカウントに自動的にアタッチします。SLR によって、発行するエンドエンティティ証明書の自動更新が可能になります。SLR が存在するかどうかを確認するには、以下のコマンドを使用して IAM にクエリを実行することができます。

```
aws iam get-role --role-name AWSServiceRoleForCertificateManager
```

SLR が存在する場合、コマンドの出力は次のようになります。

```
{
   "Role":{
      "Path":"/aws-service-role/acm.amazonaws.com/",
      "RoleName":"AWSServiceRoleForCertificateManager",
      "RoleId":"AAAAAAA0000000BBBBBBB",
      "Arn":"arn:aws:iam::{account_no}:role/aws-service-role/acm.amazonaws.com/
AWSServiceRoleForCertificateManager",
      "CreateDate":"2020-08-01T23:10:41Z",
      "AssumeRolePolicyDocument":{
         "Version":"2012-10-17",
         "Statement":[
            {
               "Effect":"Allow",
               "Principal":{
                  "Service":"acm.amazonaws.com"
               },
               "Action":"sts:AssumeRole"
            }
         ]
      },
      "Description":"SLR for ACM Service for accessing cross-account Private CA",
      "MaxSessionDuration":3600,
      "RoleLastUsed":{
         "LastUsedDate":"2020-08-01T23:11:04Z",
         "Region":"ap-southeast-1"
      }
   }
}
```

SLR がない場合は、「ACM でのサービスにリンクされたロールの使用」を参照してください。

AWS Certificate Manager プライベート証明書をエクスポートする

プライベート PKI 環境の任意の場所で使用するために AWS Private CA 、 によって発行された証明書をエクスポートできます。エクスポートされたファイルには、証明書、証明書チェーン、暗号化されたプライベートキーが含まれます。このファイルは安全に保存する必要があります。詳細については AWS Private CA、AWS Private Certificate Authority 「 ユーザーガイド」を参照してください。

> **Note**
>
> ACM を通じて発行されたパブリック証明書をエクスポートする場合は、「ACM エクスポート可能なパブリック証明書」を参照してください。

トピック

- プライベート証明書のエクスポート (コンソール)
- プライベート証明書 (CLI) をエクスポートします。

プライベート証明書のエクスポート (コンソール)

1. AWS マネジメントコンソールにサインインし、https://console.aws.amazon.com/acm/home で ACM コンソールを開きます。
2. [Certificate Manager] を選択します。
3. エクスポートする証明書のリンクを選択します。
4. [エクスポート] を選択します。
5. プライベートキーのパスフレーズを入力して確定します。

   > **Note**
   >
   > パスフレーズを作成する際は、#、$、% 以外の任意の ASCII 文字を使用できます。

6. [PEM エンコードの生成] を選択します。
7. 証明書、証明書チェーン、および暗号化されたキーをメモリにコピーするか、それぞれの [Export to a file] (ファイルにエクスポート) を選択します。
8. [Done] (完了) をクリックします。

プライベート証明書 (CLI) をエクスポートします。

export-certificate コマンドを使用して、プライベート証明書とプライベートキーをエクスポートします。コマンドを実行するときにパスフレーズを割り当てる必要があります。セキュリティを強化するには、ファイルエディタを使用してパスフレーズをファイルに保存し、ファイルを指定することでパスフレーズを指定します。これにより、パスフレーズがコマンド履歴に格納されるのを防ぎ、入力時に他のユーザーがパスフレーズを見るのを防ぎます。

> **Note**
>
> パスフレーズを含むファイルは、ラインターミネータで終了してはなりません。パスワードファイルは、次のように確認できます。
>
> ```
> $ file -k passphrase.txt
> passphrase.txt: ASCII text, with no line terminators
> ```

次の例では、コマンド出力を jq にパイプして PEM 形式を適用しています。

```
[Windows/Linux]
$ aws acm export-certificate \
     --certificate-arn arn:aws:acm:Region:444455556666:certificate/certificate_ID \
     --passphrase fileb://path-to-passphrase-file  \
     | jq -r '"\(.Certificate)\(.CertificateChain)\(.PrivateKey)"'
```

これにより base64 でエンコードされた PEM 形式の証明書が出力されます。これには、次の省略された例のように、証明書チェーンと暗号化されたプライベートキーも含まれます。

```
-----BEGIN CERTIFICATE-----
MIIDTDCCAjSgAwIBAgIRANWuFpqA16g3IwStE3vVpTwwDQYJKoZIhvcNAQELBQAw
EzERMA8GA1UECgwIdHJvbG9sb2wwHhcNMTkwNzE5MTYxNTU1WhcNMjAwODE5MTcx
NTU1WjAXMRUwEwYDVQQDDAx3d3cuc3B1ZHMuaW8wggEiMA0GCSqGSIb3DQEBAQUA
...
8UNFQvNoo1VtICL4cwWOdLOkxpwkkKWtcEkQuHE1v5Vn6HpbfFmxkdPEasoDhthH
FFWIf4/+VOlbDLgjU4HgtmV4IJDtqM9rGOZ42eFYmmc3eQO0GmigBBwwXp3j6hoi
74YM+igvtILnbYkPYhY9qz8h7lHUmannS8j6YxmtpPY=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIC8zCCAdugAwIBAgIRAM/jQ/6h2/MI1NYWX3dDaZswDQYJKoZIhvcNAQELBQAw
EzERMA8GA1UECgwIdHJvbG9sb2wwHhcNMTkwNjE5MTk0NTE2WhcNMjkwNjE5MjA0
```

```
NTE2WjATMREwDwYDVQQKDAh0cm9sb2xvbDCCASIwDQYJKoZIhvcNAQEBBQADggEP
...
j2PAOviqIXjwr08Zo/rTy/8m6LAsmm3LVVYKLyPdl+KB6M/+H93Z1/Bs8ERqqga/
6lfM6iw2JHtkW+q4WexvQSoqRXFhCZWbWPZTUpBS0d4/Y5q92S3iJLRa/JQ0d4U1
tWZyqJ2rj2RL+h7CE71XIAM//oHGcDDPaQBFD2DTisB/+ppGeDuB
-----END CERTIFICATE-----
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFKzBVBgkqhkiG9w0BBQ0wSDAnBgkqhkiG9w0BBQwwGgQUMrZb7kZJ8nTZg7aB
1zmaQh4vwloCAggAMB0GCWCGSAFlAwQBKgQQDViroIHStQgNOjR6nTUnuwSCBNAN
JM4SG202YPUiddWeWmX/RKGg3lIdE+A0WLTPskNCdCAHqdhOSqBwt65qUTZe3gBt
...
ZGipF/DobHDMkpwiaRR5sz6nG4wcki0ryYjAQrdGsR6EVvUUXADkrnrrxuHTWjFl
wEuqyd8X/ApkQsYFX/nhepOEIGWf8Xu0nrjQo77/evhG0sHXborGzgCJwKuimPVy
Fs5kw5mvEoe5DAe3rSKsSUJ1tM4RagJj2WH+BC04SZWNH8kxfOC1E/GSLBCixv3v
+Lwq38CEJRQJLdpta8NcLKnFBwmmVs9OV/VXzNuHYg==
-----END ENCRYPTED PRIVATE KEY-----
```

すべてをファイルに出力するには、前出の例に > リダイレクトを追加し、次の結果を得ます。

```
$ aws acm export-certificate \
     --certificate-arn arn:aws:acm:Region:444455556666:certificate/certificate_ID \
     --passphrase fileb://path-to-passphrase-file \
     | jq -r '"\(.Certificate)\(.CertificateChain)\(.PrivateKey)"' \
     > /tmp/export.txt
```

証明書を にインポートする AWS Certificate Manager

AWS Certificate Manager (ACM) が提供する SSL/TLS 証明書のリクエストに加えて、外部で取得した証明書をインポートできます AWS。これは、サードパーティーの認証権限 (CA) をすでに取得している場合、または ACM 発行の証明書によって満たされないアプリケーション固有の要件がある場合に行います。

インポートした証明書は、AWS ACM と統合されているサービスで使用できます。インポートした証明書は、ACM が提供する証明書と同じように動作しますが、重要な例外が 1 つあります。&ACM; はインポートされた証明書にマネージド型更新を提供しません。

インポートした証明書を更新するには、発行者から新しい証明書を取得するか、または手動で ACM に新しい証明書を [reimport] (再インポート) することもできます。このアクションによって、証明書の関連付けと Amazon リソースネーム (ARN) が維持されます。別の方法として、まったく新しい証明書をインポートすることもできます。同じドメイン名の複数の証明書をインポートできますが、一度に 1 つずつインポートする必要があります。

> **Important**
>
> インポートした証明書の有効期限をモニタリングし、失効する前に更新する責任はお客様に帰します。Amazon CloudWatch Events を使用して、インポートした証明書の有効期限切れが近づいたときに通知を送信することで、このタスクを簡素化できます。詳細については、Amazon EventBridge の使用 を参照してください。

ACM のすべての証明書は、インポートした証明書を含め、リージョナルリソースとなります。異なる AWS リージョン内で同じ証明書を Elastic Load Balancing ロードバランサーで使用するには、使用する各リージョン内に証明書をインポートする必要があります。Amazon CloudFront で証明書を使用するには、米国東部 (バージニア北部) リージョンに証明書をインポートする必要があります。詳細については、サポート対象のリージョン を参照してください。

ACM に証明書をインポートする方法については、以下のトピックを参照してください。証明書のインポートで問題が発生した場合は、証明書のインポートの問題 を参照してください。

トピック

- ACM 証明書をインポートする前提条件
- インポートのための証明書とキー形式

- 証明書のインポート
- 証明書の再インポート

ACM 証明書をインポートする前提条件

自己署名 SSL/TLS 証明書を ACM にインポートするには、証明書とそのプライベートキーを両方とも提供する必要があります。AWS 認定権限 (CA) のないものによって署名されている証明書をインポートするには、証明書のプライベートキーおよびパブリックキーも含める必要があります。証明書は、このトピックで説明されているすべての基準を満たす必要があります。

インポートした証明書についてはすべて、暗号化アルゴリズムとキーサイズを指定する必要があります。ACM では、次のアルゴリズムがサポートされています (括弧内の API 名)。

- RSA 1024 ビット (`RSA_1024`)
- RSA 2048 ビット (`RSA_2048`)
- RSA 3072 ビット (`RSA_3072`)
- RSA 4096 ビット (`RSA_4096`)
- ECDSA 256 ビット (`EC_prime256v1`)
- ECDSA 384 ビット (`EC_secp384r1`)
- ECDSA 521 ビット (`EC_secp521r1`)

また、以下の追加要件に注意してください。

- ACM 統合サービスでは、リソースへの関連付けがサポートされているアルゴリズムとキーサイズのみが許可されます。たとえば、CloudFront は 1024 ビット RSA、2048 ビット RSA、3072 ビット RSA、4096 ビット RSA、楕円プライムカーブ 256 ビットのキーのみをサポートし、Application Load Balancer は ACM から利用可能なすべてのアルゴリズムをサポートします。詳細については、使用しているサービスのドキュメントを参照してください。
- 証明書は、SSL/TLS X.509 バージョン 3 の証明書である必要があります。証明書には、パブリックキー、ウェブサイトの完全修飾ドメイン名 (FQDN) または IP アドレス、発行者に関する情報が含まれている必要があります。
- 証明書は、お客様が所有するプライベートキーで自己署名することも、発行元 CA のプライベートキーで署名することも可能です。プライベートキーは、5KB (5,120 バイト) 以下の大きさで、暗号化されていないものを提供する必要があります。

- 証明書が CA によって署名されており、証明書チェーンを提供することを選択する場合、チェーンは PEM エンコードされている必要があります。
- 証明書はインポート時に有効である必要があります。有効になる前、または有効期限が切れた後で証明書をインポートすることはできません。`NotBefore` 証明書のフィールドには有効開始日が、`NotAfter` フィールドには終了日が含まれています。
- 証明書の資料 (証明書、プライベートキー、および証明書チェーン) はすべて PEM エンコードされる必要があります。DER エンコードされた資料をアップロードすると、エラーが発生します。詳細な説明と例については、インポートのための証明書とキー形式 を参照してください。
- 証明書を更新 (再インポート) する場合に、以前にインポートされた証明書に拡張子が存在しなければ、`KeyUsage` または `ExtendedKeyUsage` の拡張子を追加できません。

  例外: 以前の証明書と比較して、クライアント認証 ExtendedKeyUsage が欠落している証明書を再インポートできます。これは、証明機関がChrome のルートプログラム要件に準拠するために、ClientAuth EKU で証明書を発行しなくなるという業界の変化に対応できます。

  > **Important**
  >
  > クライアント認証機能が必要な場合は、ACM は以前にインポートした証明書へのロールバックをサポートしていないため、ユーザー側で追加の検証を実装する必要があります。

- AWS CloudFormation は、ACM への証明書のインポートをサポートしていません。

インポートのための証明書とキー形式

ACM では、証明書、証明書チェーン、およびプライベートキー (存在する場合) を個別にインポートし、PEM 形式で各コンポーネントをエンコードする必要があります。PEM は Privacy Enhanced Mail の略です。PEM 形式は、証明書、証明書リクエスト、証明書チェーン、およびキーを表すために頻繁に使用されます。PEM 形式ファイルの一般的な拡張子は `.pem` ですが、このとおりである必要はありません。

> **Note**
>
> AWS は、PEM ファイルやその他の証明書形式を操作するためのユーティリティを提供しません。以下の例は、単純な操作のために汎用テキストエディタに依存しています。より複雑なタスク (ファイル形式の変換やキーの抽出など) を実行する必要がある場合は、OpenSSL などの無料のオープンソースツールが容易に入手できます。

次の例では、インポートするファイルの形式を示します。コンポーネントが 1 つのファイルで表示される場合は、テキストエディタを使用して (慎重に) 3 つのファイルに分割します。PEM ファイルで文字を誤って編集した場合、またはいずれかの行の末尾に 1 つ以上のスペースを追加した場合、証明書、証明書チェーン、またはプライベートキーは無効になることに注意してください。

Example 1. PEM エンコードされた証明書

```
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
```

Example 2. PEM エンコードされた証明書チェーン

証明書チェーンには 1 つまたは複数の証明書が含まれます。テキストエディタ、Windows の copy コマンド、または Linux の cat コマンドを使用して、ファイルをチェーンに連結します。証明書は順に連結し、各ディレクトリが 1 つ前のディレクトリを認定するようにする必要があります。プライベート証明書をインポートする場合は、最後にルート証明書をコピーします。次の例には 3 つの証明書が含まれていますが、証明書チェーンに含まれている証明書はそれ以上またはそれ以下である可能性があります。

> **Important**
>
> 証明書チェーンに証明書をコピーしないでください。

```
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
```

Example 3. PEM エンコードされたプライベートキー

X.509 バージョン 3 証明書は、パブリックキーアルゴリズムを使用します。X.509 証明書または証明書リクエストを作成するときは、プライベートキーとパブリックキーのキーペアを作成するために

使用する必要があるアルゴリズムとキービットサイズを指定します。パブリックキーは証明書またはリクエストに配置されます。関連付けられたプライベートキーシークレットを保持する必要があります。証明書をインポートするときに、プライベートキーを指定します。キーは非暗号化される必要があります。次の例に、RSA プライベートキーを示します。

```
-----BEGIN PRIVATE KEY-----
Base64-encoded private key
-----END PRIVATE KEY-----
```

次の例は、PEM エンコード楕円曲線プライベートキーを示しています。キーの作成方法によっては、パラメータブロックが含まれない場合があります。パラメータブロックが含まれる場合、ACM により、インポート処理中にキーを使用する前に削除されます。

```
-----BEGIN EC PARAMETERS-----
Base64-encoded parameters
-----END EC PARAMETERS-----
-----BEGIN EC PRIVATE KEY-----
Base64-encoded private key
-----END EC PRIVATE KEY-----
```

証明書のインポート

、、 AWS CLIまたは ACM API を使用して AWS マネジメントコンソール、外部で取得した証明書 (サードパーティーの信頼サービスプロバイダーが提供する証明書) を ACM にインポートできます。以下のトピックでは、 AWS マネジメントコンソール と の使用方法を示します AWS CLI。AWS 発行者以外から証明書を取得する手順は、このガイドの範囲外です。

> **Important**
>
> 選択した署名アルゴリズムは ACM 証明書をインポートする前提条件 を満たす必要があります。

トピック

- インポート (コンソール)
- インポート (AWS CLI)

インポート (コンソール)

次の例では、 AWS マネジメントコンソールを使用して証明書をインポートする方法を示しています。

1. ACM コンソール (https://console.aws.amazon.com/acm/home) を開きます。ACM を初めて使用する場合は、AWS Certificate Manager 見出しを探し、その下にある [Get Started (使用を開始)] ボタンを選択します。
2. [証明書をインポートする] を選択します。
3. 以下の操作を実行します。

   a. 証明書本文の場合、インポートする PEM エンコードされた証明書を貼り付けます。これは `-----BEGIN CERTIFICATE-----` から始まり `-----END CERTIFICATE-----` で終わる必要があります。

   b. 証明書のプライベートキーの場合、PEM エンコードされ、暗号化されていないプライベートキーを貼り付けます。これは `-----BEGIN PRIVATE KEY-----` から始まり `-----END PRIVATE KEY-----` で終わる必要があります。

   c. (オプション) [証明書チェーン] では、PEM エンコードされた証明書チェーンを貼り付けます。
4. (オプション) インポートした証明書にタグを追加するには、[タグ] を選択します。タグとは AWS リソースに割り当てるラベルです。タグはそれぞれ、1 つのキーとオプションの 1 つの値で設定されており、どちらもお客様側が定義します。タグを使用して、リソースを整理したり、AWS コストを追跡したりできます。
5. [インポート] を選択します。

インポート (AWS CLI)

次の例では、AWS Command Line Interface (AWS CLI) を使用して証明書をインポートする方法を示しています。例では、次のように想定しています。

- PEM エンコードされた証明書は、`Certificate.pem` というファイルに保存されます。
- PEM エンコードされた証明書チェーンは、`CertificateChain.pem` というファイルに保存されます。
- PEM エンコードされ、非暗号化されたプライベートキーは、`PrivateKey.pem` というファイルに保存されます。

次の例を使用するには、ファイル名を独自の名前に置き換えて、ひと続きの行にコマンドを入力します。次の例では、読みやすくするために改行とスペースを追加しています。

```
$ aws acm import-certificate --certificate fileb://Certificate.pem \
      --certificate-chain fileb://CertificateChain.pem \
      --private-key fileb://PrivateKey.pem
```

`import-certificate` コマンドが成功した場合、インポートされた証明書の Amazon リソースネーム (ARN) が返されます。

証明書の再インポート

証明書をインポートして他の AWS サービスに関連付けると、元の証明書 AWS のサービス関連付けを維持しながら、証明書の有効期限が切れる前にその証明書を再インポートできます。ACM と統合された AWS サービスの詳細については、「」を参照してくださいサービスと ACM の統合。

証明書を再インポートする場合は次の条件が適用されます。

- ドメイン名を追加または削除できます。
- 証明書からすべてのドメイン名を削除することはできません。
- 最初にインポートされた証明書に キーの使用 拡張機能が存在する場合、新しい拡張機能値を追加できますが、既存の値を削除することはできません。
- 最初にインポートされた証明書に 拡張キーの使用 拡張機能が存在する場合、新しい拡張機能値を追加できますが、既存の値を削除することはできません。

  例外: クライアント認証の拡張キー使用法は削除できます。これは、証明機関がChrome のルートプログラム要件に準拠するために、ClientAuth EKU で証明書を発行しなくなるという業界の変化に対応できます。

  > **Important**
  >
  > クライアント認証機能が必要な場合は、ACM は以前にインポートした証明書へのロールバックをサポートしていないため、ユーザー側で追加の検証を実装する必要があります。

- キーのタイプとサイズを変更することはできません。
- 証明書を再インポートするときは、リソースタグを適用できません。

トピック

- 再インポート (コンソール)
- 再インポート(AWS CLI)

再インポート (コンソール)

次の例では、 AWS マネジメントコンソールを使用して証明書を再インポートする方法を示しています。

1. ACM コンソール (https://console.aws.amazon.com/acm/home) を開きます。
2. 再インポートする証明書を選択または展開します。
3. 証明書の詳細ペインを開き、[Reimport certificate] ボタンを選択します。名前の横にあるチェックボックスをオンにして証明書を選択した場合は、[Actions] メニューの [Reimport certificate] を選択します。
4. 証明書の本文には、PEM エンコードされたエンドエンティティ証明書を貼り付けます。
5. 証明書のプライベートキーの場合、証明書のパブリックキーに関連付けられた、暗号化されていない PEM エンコード形式のプライベートキーを貼り付けます。
6. (オプション) [証明書チェーン] では、PEM エンコードされた証明書チェーンを貼り付けます。証明書チェーンには、すべての中間発行証明機関のための 1 個以上の証明書、およびルート証明書が含まれます。インポートする証明書が自己割り当ての場合、証明書チェーンは不要です。
7. 証明書に関する情報を確認します。エラーがない場合は、[Reimport] を選択します。

再インポート(AWS CLI)

次の例では、AWS Command Line Interface (AWS CLI) を使用して証明書を再インポートする方法を示しています。例では、次のように想定しています。

- PEM エンコードされた証明書は、`Certificate.pem` というファイルに保存されます。
- PEM エンコードされた証明書チェーンは、`CertificateChain.pem` というファイルに保存されます。
- (プライベート証明書のみ) PEM エンコードされた、暗号化されていないプライベートキーは、`PrivateKey.pem` という名前のファイルに保存されます。
- 再インポートする証明書の ARN があります。

次の例を使用するには、ファイル名と ARN を自分のものに置き換えて、ひと続きの行にコマンドを入力します。次の例では、読みやすくするために改行とスペースを追加しています。

> **Note**
>
> 証明書を再インポートするには、証明書 ARN を指定する必要があります。

```
$ aws acm import-certificate --certificate fileb://Certificate.pem \
      --certificate-chain fileb://CertificateChain.pem \
      --private-key fileb://PrivateKey.pem \
      --certificate-
arn arn:aws:acm:region:123456789012:certificate/12345678-1234-1234-1234-12345678901
```

`import-certificate` コマンドが成功した場合、証明書の Amazon リソースネーム (ARN) が返されます。

証明書管理

ACM コンソールまたは を使用して AWS CLI 、アカウントの証明書を管理できます。

- 証明書の一覧表示 は、ACM によって管理される証明書を表示します。リストには、各証明書に関する概要情報が表示されます。
- 証明書の詳細を表示する は、個々の証明書の詳細を表示します。
- 証明書を削除する は、アカウントから削除します。削除された証明書は、削除後しばらくの間、リストに表示される場合があります。

によって管理される証明書を一覧表示する AWS Certificate Manager

ACM コンソールまたは を使用して AWS CLI 、ACM によって管理される証明書を一覧表示できます。コンソールでは、1 ページに最大 500 の証明書、CLI では最大 1,000 の証明書を一覧表示できます。

次のコンソールを使用して証明書を一覧表示するには

1. ACM コンソール (https://console.aws.amazon.com/acm/) を開きます。
2. 証明書リストの情報を確認します。右上のページ番号を使用して、証明書の複数のページを検索できます。各証明書は 1 つの行を占有し、デフォルトで各証明書の次の列が表示されます。

- ドメイン名 — 証明書の完全修飾ドメイン名 (FQDN)。
- Type (種類)— 証明書の種類。想定される値は次のとおりです。Amazon issued (Amazon が発行済み) | Private (プライベート) | Imported (インポート済み)
- ステータス – 証明書のステータス。想定される値は次のとおりです。Pending validation (検証保留中) | Issued (発行済み) | Inactive (非アクティブ) | Expired (有効期限切れ) | Revoked (失効) | Failed (失敗) | Validation Timed out (検証タイムアウト)
- 使用中? – ACM 証明書が Elastic Load Balancing や CloudFront などの AWS サービスにアクティブに関連付けられているかどうか。値は、[No] または [Yes] となります。
- Renewal eligibility (更新の適格性) — 有効期限が近づいたときに ACM が自動的に証明書を更新できるかどうか。設定可能な値は、「Eligible] または [Ineligible] (不適格) です。資格規則については、でのマネージド証明書の更新 AWS Certificate Manager を参照してください。

コンソールの右上の設定アイコンを選択して、ページに表示される証明書の数のカスタマイズ、セルの内容の改行動作の指定、および追加の情報フィールドの表示を行うことができます。以下のオプションフィールドが利用できます。

- Additional domain names (追加のドメイン名) – 証明書に含まれる 1 つまたは複数のドメイン名 (サブジェクトの別名)。
- Requested at (リクエスト時刻) — ACM が証明書をリクエストした時刻。
- Issued at (発行時刻) – 証明書が発行された時刻。この情報は、Amazon が発行した証明書でのみ利用可能で、エクスポートには使用できません。
- Not before (有効期限前)– 証明書が無効になるまで時間。
- Not after (有効期限後)– 証明書が無効になった後の時間。
- Revoked at (失効時刻) — 失効した証明書の失効日時。
- Name tag (名前タグ) — この証明書の Name というタグの値 (そのようなタグが存在する場合)。
- 更新ステータス — リクエストされた証明書の更新ステータス。このフィールドは、更新がリクエストされた場合にのみ表示され、値が表示されます。設定可能な値は、自動更新を保留中、検証待ち、成功、失敗です。

> **Note**
>
> 証明書ステータスの変更が有効になるまでに数時間ほどかかる場合があります。問題が発生した場合、証明書のリクエストは 72 時間後にタイムアウトになるため、発行または更新プロセスを最初から繰り返す必要があります。

ページサイズ設定では、各コンソールページで返される証明書の数を指定します。

使用可能な証明書の詳細については、AWS Certificate Manager 証明書の詳細を表示する を参照してください。

を使用して証明書を一覧表示するには AWS CLI

リスト証明書コマンドを使用して、次の例に示すように ACM で管理される証明書を一覧表示します。

```
$ aws acm list-certificates --max-items 10
```

コマンドは以下のような情報を返します。

```
{
    "CertificateSummaryList": [
        {
            "CertificateArn":
 "arn:aws:acm:Region:444455556666:certificate/certificate_ID",
            "DomainName": "example.com"
  "SubjectAlternativeNameSummaries": [
                "example.com",
                "other.example.com"
            ],
            "HasAdditionalSubjectAlternativeNames": false,
            "Status": "ISSUED",
            "Type": "IMPORTED",
            "KeyAlgorithm": "RSA-2048",
            "KeyUsages": [
                "DIGITAL_SIGNATURE",
                "KEY_ENCIPHERMENT"
            ],
            "ExtendedKeyUsages": [
                "NONE"
            ],
            "InUse": false,
            "RenewalEligibility": "INELIGIBLE",
            "NotBefore": "2022-06-14T23:42:49+00:00",
            "NotAfter": "2032-06-11T23:42:49+00:00",
            "CreatedAt": "2022-08-25T19:28:05.531000+00:00",
            "ImportedAt": "2022-08-25T19:28:05.544000+00:00"
        },...
    ]
}
```

デフォルトで、keyTypes RSA_1024 または RSA_2048 と少なくとも 1 つのドメインが指定されている証明書のみが返されます。ドメインレス証明書や、異なるアルゴリズムやビットサイズを使用する証明書など、制御する他の証明書を表示するには、次の例に示すように --includes パラメータを指定します。パラメータでは、フィルタ構造のメンバーを指定できます。

```
$ aws acm list-certificates --max-items 10 --includes keyTypes=RSA_4096
```

AWS Certificate Manager 証明書の詳細を表示する

ACM コンソールまたは を使用して AWS CLI 、証明書に関する詳細メタデータを一覧表示できます。

次のコンソールで証明書の詳細を表示するには

1. ACM コンソール (https://console.aws.amazon.com/acm/) を開くと、証明書が表示されます。右上のページ番号を使用して、証明書の複数のページを検索できます。
2. リスト化された証明書についての詳細なメタデータを表示するには、証明書 ID を選択してください。ページが開いて、次の情報が表示されます。

   - 証明書のステータス
     - Identifier (識別子) — 一意の 32 バイトかつ 16 進数の証明書の識別子
     - ARN — フォーム `arn:aws:acm:`*`Region`*`:`*`444455556666`*`:certificate/`*`certificate_ID` の Amazon リソースネーム (ARN)
     - タイプ — ACM 証明書の管理カテゴリを識別します。想定される値は次のとおりです。Amazon が発行済み|プライベート|インポート済み。詳しくは、AWS Certificate Manager パブリック証明書、でプライベート証明書をリクエストする AWS Certificate Manager または 証明書を にインポートする AWS Certificate Manager を参照してください。
     - Status (ステータス) — 証明書のステータス。想定される値は次のとおりです。Pending validation (検証保留中) | Issued (発行済み) | Inactive (非アクティブ) | Expired (有効期限切れ) | Revoked (失効) | Failed (失敗) | Validation Timed out (検証タイムアウト)
     - Detailed status (詳細ステータス) — 証明書が発行またはインポートされた日付と時刻
   - ドメイン
     - Domain (ドメイン) — 証明書の完全修飾ドメイン名(FQDN)。
     - Status (ステータス) — ドメイン検証のステータス。想定される値は次のとおりです。Pending validation (検証保留中) | Revoked (失効) | Failed (失敗) | Validation Timed out (検証タイムアウト) | Success (成功)
   - 詳細
     - 使用中? — 証明書がAWS 統合サービスに関連付けられているかどうか、想定される値は次のとおりです。はい|いいえ
     - Domain name (ドメイン名) — 証明書の最初の完全修飾ドメイン名 (FQDN)。

- Managed by (管理者) – ACM で証明書を管理する AWS サービスを識別します。
- Number of additional names (追加の名前の数) — 有効な証明書のドメイン名の数
- Serial number (シリアル番号) — 16 バイトかつ 16 進数の証明書のシリアル番号
- Public key info (公開キー情報) — キーペアを生成した暗号化アルゴリズム
- 署名アルゴリズム — 証明書に署名するために使用される暗号化アルゴリズム。
- Can be used with (併用が可能) — ACM [integrated services] (統合サービス) のリストで、これらのパラメータを持つ証明書をサポートします
- Requested at (リクエスト対象) — 発行リクエストの日付と時刻
- Issued at (発行対象) — 発行の日時 (該当する場合)
- Imported at (インポート対象) — インポートの日時 (該当する場合)
- Not before (以前は不可) — 証明書の有効期間の開始
- Not after (以後は不可) — 証明書の有効期限の日時
- Renewal eligibility (更新の適格性) — 想定される値は次のとおりです。Eligible (適格) | Ineligible (不適格) 資格規則については、でのマネージド証明書の更新 AWS Certificate Manager を参照してください。
- 更新ステータス — リクエストされた証明書の更新ステータス。このフィールドは、更新がリクエストされた場合にのみ表示され、値が表示されます。設定可能な値は、自動更新を保留中、検証待ち、成功、失敗です。

  > **Note**
  >
  > 証明書ステータスの変更が有効になるまでに数時間ほどかかる場合があります。問題が発生した場合、証明書のリクエストは 72 時間後にタイムアウトになるため、発行または更新プロセスを最初から繰り返す必要があります。

- CA (認証権限) — 署名 CA の ARN

- タグ
  - キー
  - 値
- 検証の状態 — 該当する場合、指定できる値は以下のとおりです。
  - 保留中 — 検証が要求され、完了していません。
  - 検証がタイムアウトしました — リクエストされた検証がタイムアウトしましたが、リクエストを繰り返すことができます。

- なし — 証明書はプライベート PKI 用の証明書か、自己署名であり、検証は必要ありません。

を使用して証明書の詳細を表示するには AWS CLI

次のコマンドに示すように、 の describe-certificate AWS CLI を使用して証明書の詳細を表示します。

```
$ aws acm describe-certificate --certificate-arn
 arn:aws:acm:Region:444455556666:certificate/certificate_ID
```

コマンドは以下のような情報を返します。

```
{
    "Certificate": {
        "CertificateArn": "arn:aws:acm:Region:444455556666:certificate/certificate_ID",
        "Status": "EXPIRED",
        "Options": {
            "CertificateTransparencyLoggingPreference": "ENABLED"
        },
        "SubjectAlternativeNames": [
            "example.com",
            "www.example.com"
        ],
        "DomainName": "gregpe.com",
        "NotBefore": 1450137600.0,
        "RenewalEligibility": "INELIGIBLE",
        "NotAfter": 1484481600.0,
        "KeyAlgorithm": "RSA-2048",
        "InUseBy": [
            "arn:aws:cloudfront::account:distribution/E12KXPQHVLSYVC"
        ],
        "SignatureAlgorithm": "SHA256WITHRSA",
        "CreatedAt": 1450212224.0,
        "IssuedAt": 1450212292.0,
        "KeyUsages": [
            {
                "Name": "DIGITAL_SIGNATURE"
            },
            {
                "Name": "KEY_ENCIPHERMENT"
            }
```

```
        ],
        "Serial": "07:71:71:f4:6b:e7:bf:63:87:e6:ad:3c:b2:0f:d0:5b",
        "Issuer": "Amazon",
        "Type": "AMAZON_ISSUED",
        "ExtendedKeyUsages": [
            {
                "OID": "1.3.6.1.5.5.7.3.1",
                "Name": "TLS_WEB_SERVER_AUTHENTICATION"
            },
            {
                "OID": "1.3.6.1.5.5.7.3.2",
                "Name": "TLS_WEB_CLIENT_AUTHENTICATION"
            }
        ],
        "DomainValidationOptions": [
            {
                "ValidationEmails": [
                    "hostmaster@example.com",
                    "admin@example.com",
                    "postmaster@example.com",
                    "webmaster@example.com",
                    "administrator@example.com"
                ],
                "ValidationDomain": "example.com",
                "DomainName": "example.com"
            },
            {
                "ValidationEmails": [
                    "hostmaster@example.com",
                    "admin@example.com",
                    "postmaster@example.com",
                    "webmaster@example.com",
                    "administrator@example.com"
                ],
                "ValidationDomain": "www.example.com",
                "DomainName": "www.example.com"
            }
        ],
        "Subject": "CN=example.com"
    }
}
```

によって管理される証明書を削除する AWS Certificate Manager

ACM コンソールまたは を使用して AWS CLI 証明書を削除できます。チケットの削除は最終的に一貫性が保たれます。証明書は、削除された後もしばらくリストに表示される場合があります。

> **Important**
>
> - 別の AWS サービスで使用されている ACM 証明書は削除できません。使用中の証明書を削除するには、まず証明書の関連付けを削除する必要があります。これは、関連付けられたサービスのコンソールまたは CLI を使用して行われます。
> - プライベート認証局 (CA)によって発行された証明書を削除しても、CA には影響しません。CA が削除されるまで課金され続けます。詳細については、「AWS Private Certificate Authority ユーザーガイド」の「Private CAの削除」を参照してください。

次のコンソールを使用して証明書を削除するには

1. ACM コンソール (https://console.aws.amazon.com/acm/) を開きます。
2. 証明書のリストで、ACM 証明書のチェックボックスをオンにして [Delete] (削除) を選択します。

> **Note**
>
> リストの配列方法によっては、探している証明書がすぐには表示されない場合があります。右側の黒い三角形をクリックすると、配列を変更できます。また、右上のページ番号を使用して、証明書の複数のページを検索することもできます。

を使用して証明書を削除するには AWS CLI

次のコマンドに示すように、証明書の削除コマンドを使用して証明書を削除します。

```
$ aws acm delete-certificate --certificate-arn
 arn:aws:acm:Region:444455556666:certificate/certificate_ID
```

でのマネージド証明書の更新 AWS Certificate Manager

ACM は、Amazon 発行の SSL/TLS 証明書の更新の管理方法について説明します。つまり、ACM は証明書を自動的に更新するか (DNS 検証を使用している場合)、有効期限切れが近づくと E メール通知を送信します。これらのサービスは、パブリック ACM 証明書とプライベート ACM 証明書の両方に対して提供されます。

証明書は、次の考慮事項を条件として、自動更新の対象となります。

- Elastic Load Balancing や CloudFront などの別の AWS サービスに関連付けられている場合は対象です。
- 発行後や最終更新後にエクスポートしている場合、対象となります。
- それが ACM RequestCertificate API を呼び出して発行したプライベート証明書であり、次にエクスポートされるか、別の AWS サービスに関連付けられた場合、対象となります。
- それがマネジメントコンソールを通して発行したプライベート証明書であり、次にエクスポートされるか、別の AWS サービスに関連付けられた場合、対象となります
- AWS Private CA IssueCertificate API を呼び出して発行したプライベート証明書である場合は、対象外です。
- インポートしている場合は、対象外です。
- すでに有効期限が切れている場合、対象外です。

加えて、国際化ドメイン名に関連する次の Punycode 要件を満たす必要があります:

1. パターン「<character><character>—」で始まるドメイン名は「xn—」と一致する必要があります。
2. 「xn—」で始まるドメイン名も有効な国際化ドメイン名である必要があります。

Punycode の例

ドメイン名	フルフィル #1	フルフィル #2	許可されています	メモ
example.com	該当なし	該当なし	✓	「<character><character>—」で始まらない
a--example.com	該当なし	該当なし	✓	「<character><character>—」で始まらない
abc--example.com	該当なし	該当なし	✓	「<character><character>—」で始まらない
xn--xyz.com	はい	はい	✓	有効な国際化ドメイン名 (簡.com に解決)
xn--example.com	はい	なし	✗	有効な国際化ドメイン名ではありません
ab--example.com	いいえ	いいえ	✗	「xn--」で始まる必要があります。

ACM が証明書を更新するとき、証明書の Amazon リソースネーム (ARN) は変更されません。また、ACM 証明書はリージョナルリソースです。複数の AWS リージョンに同じドメイン名の証明書がある場合は、それぞれの証明書を個別に更新する必要があります。

トピック

- ACM パブリック証明書の更新
- でのプライベート証明書の更新 AWS Certificate Manager
- 証明書の更新ステータスの確認

ACM パブリック証明書の更新

パブリックに信頼されたマネージド証明書を発行する場合は、ドメイン所有者であることを証明 AWS Certificate Manager する必要があります。これは、DNS での検証 または E メールでの検証 のどちらかの方法で行います。証明書が更新のために作成されると、ACM は以前に選択した方法と同じ方法で所有権を再検証します。以下のトピックでは、各ケースでの更新プロセスの仕組みについて説明します。

トピック

- DNS によって検証されたドメインの更新
- E メール検証済みドメインの更新
- HTTP によって検証されたドメインの更新

DNS によって検証されたドメインの更新

管理された更新は、最初に DNS 検証 を使用して発行された ACM 証明書に対して完全に自動化されています。。

有効期限の 45 日前に、ACM は次の更新基準をチェックします。

> **Note**
>
> 以前に発行された 395 日間の有効期間の証明書は、有効期限の 60 日前に更新され、198 日間の更新された有効期間を受け取ります。198 日間の有効期間を持つ証明書は、有効期限の 45 日前に更新されます。

- 証明書は現在、 AWS サービスで使用されています。
- 必要な ACM 提供の DNS CNAME レコード (一意のサブジェクト代替名ごとに 1 つ) はすべて存在し、パブリック DNS 経由でアクセスできます。

これらの条件が満たされると、ACM はドメイン名を検証済みと見なし、証明書を更新します。

更新中にドメインを自動的に検証できない場合、ACM は AWS Health イベントと Amazon EventBridge イベントを送信します。これらのイベントは、有効期限の 30 日前、15 日前、7 日前、3 日前、1 日前に送信されます。詳細については、「ACM の Amazon EventBridge サポート」を参照してください。

E メール検証済みドメインの更新

ACM 証明書は 198 日間有効です。証明書を更新するには、ドメイン所有者によるアクションが必要です。ACM は、有効期限切れの 45 日前にドメインに関連付けられた E メールアドレスに更新通知の送信を開始します。通知には、ドメイン所有者が更新するためにクリックできるリンクが含まれています。リストされているすべてのドメインが検証されると、ACM は同じ ARN で更新された証明書を発行します。

更新中にドメインを自動的に検証できない場合、ACM は AWS Health イベントと Amazon EventBridge イベントを送信します。これらのイベントは、有効期限の 30 日前、15 日前、7 日前、3 日前、1 日前に送信されます。詳細については、「ACM の Amazon EventBridge サポート」を参照してください。

検証 E メールメッセージの詳細については、AWS Certificate Manager E メール検証 を参照してください。

検証 E メールにプログラムで応答する方法については、E AWS Certificate Manager メール検証を自動化する を参照してください。

検証 E メールを再送信する

証明書をリクエストするときにドメインの E メール検証を設定した後 (「」を参照AWS Certificate Manager E メール検証)、 AWS Certificate Manager API を使用して、ACM が証明書の更新用にドメイン検証 E メールを送信するようにリクエストできます。このためには、次の条件が満たされる必要があります。

- ACM 証明書を最初にリクエストするときに E メール検証を使用した。
- 証明書の更新ステータスが [検証保留中] である。証明書の更新ステータスの閲覧に関する詳細は、証明書の更新ステータスの確認 を参照してください。
- ACM が証明書の更新用に送信した元のドメイン検証 E メールメッセージが受信されない、あるいは見つかりません。

証明書リクエストで最初に設定したドメインとは異なるドメインに検証 E メールを送信するには、ACM API の ResendValidationEmail オペレーション AWS CLI、または AWS SDKsを使用できます。ACM は、指定された検証ドメインに E メールを送信します。ブラウザ AWS CLI で にアクセスするには、サポートされているリージョン AWS CloudShell で を使用します。

ACM がドメイン検証 E メールメッセージ (コンソール) を再送信するようにリクエストするには

1. https://console.aws.amazon.com/acm/home で AWS Certificate Manager コンソールを開きます。
2. 検証が必要な証明書の [Certificate ID] (証明書 ID) を選択します。
3. [Resend validation email] (検証 E メールを再送信する) を選択します。

ACM がドメイン検証 E メールを再送信するようにリクエストするには (ACM API)

ACM API で ResendValidationEmail オペレーションを使用します。これにより、証明書の ARN、手動による検証が必要となるドメイン、ドメイン検証 E メールを受信するドメインが渡されます。次の例では、 AWS CLIを使用してこのオペレーションを行う方法を示します この例では読みやすいように改行が含まれています。

```
$ aws acm resend-validation-email \
 --certificate-arn arn:aws:acm:region:account:certificate/certificate_ID \
 --domain subdomain.example.com \
 --validation-domain example.com
```

HTTP によって検証されたドメインの更新

ACM は、CloudFront 経由の HTTP 検証を使用して最初に発行された証明書の自動マネージド更新を提供します。

有効期限の 45 日前に、ACM は次の更新基準をチェックします。

> **Note**
>
> 以前に発行された 395 日間の有効期間の証明書は、有効期限の 60 日前に更新され、198 日間の更新された有効期間を受け取ります。198 日間の有効期間を持つ証明書は、有効期限の 45 日前に更新されます。

- 証明書は現在、CloudFront によって使用されています。
- 必要な HTTP 検証レコードはすべてアクセス可能で、期待されるコンテンツが含まれています。

これらの条件が満たされると、ACM はドメイン名を検証済みと見なし、証明書を更新します。

更新中にドメインを自動的に検証できない場合、ACM は AWS Health イベントと Amazon EventBridge イベントを送信します。これらのイベントは、有効期限の 30 日前、15 日前、7 日前、3 日前、1 日前に送信されます。詳細については、「ACM の Amazon EventBridge サポート」を参照してください。

更新を成功させるには、 `RedirectFrom` ロケーションにあるコンテンツが証明書内の各ドメインの `RedirectTo` ロケーションにあるコンテンツと一致することを確認する必要があります。

でのプライベート証明書の更新 AWS Certificate Manager

からプライベート CA によって署名された ACM 証明書 AWS Private CA は、マネージド更新の対象となります。パブリックに信頼できる ACM 証明書とは異なり、プライベート PKI の証明書には検証は必要ありません。信頼は、管理者が適切なルート CA 証明書をクライアントの信頼ストアにインストールしたときに確立されます。

> **Note**
>
> マネージド更新の対象になるのは、ACM コンソールまたは ACM API の RequestCertificate アクションを使用して取得した証明書のみです。 AWS Private CA API の IssueCertificate アクション AWS Private CA を使用して から直接発行された証明書は、ACM によって管理されません。

マネージド証明書の有効期限切れの 60 日前になると、ACM によって、自動的に更新が試みられます。これには、手動でエクスポートおよびインストールされた証明書 (例えば、オンプレミスのデータセンターで) が含まれます。また、ユーザーは ACM API の RenewCertificate アクションを使用して、いつでも更新を強制できます。強制された更新の Java 実装のサンプルについては、証明書の更新 を参照してください。

更新後、証明書のサービスへのデプロイは、次のいずれかの方法で実行されます。

- 証明書が、ACM 統合サービスに関連付けられている場合、追加のユーザーのアクションなしに、新しい証明書が古い証明書を置き換えます。
- 証明書が、ACM 統合サービスに関連付けられていない場合、更新された証明書をエクスポートしてインストールするには、ユーザーのアクションが必要です。これらのアクションは、手動で実行するか、以下のように 、AWS Health、Amazon EventBridge、および AWS Lambda からの支援を得て実行することができます。詳しくは、更新された証明書のエクスポートの自動化 を参照してください。

更新された証明書のエクスポートの自動化

次の手順は、ACM が更新されたときにプライベート PKI 証明書のエクスポートを自動化するためのソリューションの例を示します。この例では、ACM から証明書とそのプライベートキーのみをエクスポートします。エクスポート後は、証明書をターゲットデバイスにインストールする必要があります。

コンソールを使用して証明書のエクスポートを自動化するには

1. AWS Lambda デベロッパーガイドの手順に従って、ACM エクスポート API を呼び出す Lambda 関数を作成して設定します。

   a. Lambda 関数の作成。

   b. 関数の Lambda 実行ロールを作成し、次の信頼ポリシーを追加します。このポリシーは、ACM APIの ExportCertificate アクションを呼び出すことにより、更新された証明書とプライベート鍵を取得するための関数内のコードに許可を付与します。

      JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"acm:ExportCertificate",
         "Resource":"*"
      }
   ]
}
```

2. Amazon EventBridge でルールを作成し、ACM ヘルスイベントをリッスンし、検出した場合は Lambda 関数を呼び出します。ACM は、証明書の更新を試みるたびに AWS Health イベントに書き込みます。これらの通知の詳細については、Personal Health Dashboard (PHD) を使用してステータスを確認する を参照してください。

   次のイベントパターンを追加して、ルールを設定します。

```
{
   "source":[
```

```
      "aws.health"
   ],
   "detail-type":[
      "AWS Health Event"
   ],
   "detail":{
      "service":[
         "ACM"
      ],
      "eventTypeCategory":[
         "scheduledChange"
      ],
      "eventTypeCode":[
         "AWS_ACM_RENEWAL_STATE_CHANGE"
      ]
   },
   "resources":[
      "arn:aws:acm:region:account:certificate/certificate_ID"
   ]
}
```

3. ターゲットシステムに証明書を手動でインストールして、更新プロセスを完了します。

プライベート PKI 証明書のマネージド更新のテスト

ACM API または を使用して AWS CLI 、ACM マネージド更新ワークフローの設定を手動でテストできます。そうすることで、有効期限切れの前に ACM によって証明書が自動的に更新されることを確認できます。

> **Note**
>
> AWS Private CAによって発行され、エクスポートされた証明書の更新のみをテストできます。

以下で説明する API アクションまたは CLI コマンドを使用すると、ACM は証明書の更新を試みます。更新が成功すると、ACM はマネジメントコンソールまたは API 出力に表示される証明書のメタデータを更新します。証明書が ACM 統合サービスに関連付けられている場合、新しい証明書がデプロイされ、Amazon CloudWatch Events で更新イベントが生成されます。更新に失敗すると、ACM はエラーを返し、修復アクションを提案します。(この情報は、記述証明書コマンドを使用して閲覧

できます。) 証明書が統合サービスを通じてデプロイされていない場合、証明書をエクスポートし、リソースに手動でインストールする必要があります。

> **Important**
>
> ACM で AWS Private CA 証明書を更新するには、まず ACM サービスプリンシパルにそのアクセス許可を付与する必要があります。詳細については、「ACM への証明書更新アクセス権限の割り当て」を参照してください。

証明書の更新を手動でテストするには (AWS CLI)

1. renew-certificate コマンドを使用して、プライベートにエクスポートされた証明書を更新します。

```
aws acm renew-certificate \
 --certificate-arn arn:aws:acm:region:account:certificate/certificate_ID
```

2. 次に、describe-certificate コマンドを使用して、証明書の更新詳細が更新されたことを確認します。

```
aws acm describe-certificate \
 --certificate-arn arn:aws:acm:region:account:certificate/certificate_ID
```

証明書の更新を手動でテストするには (ACM API)

- RenewCertificate リクエストを送信し、更新するプライベート証明書の ARN を指定します。次に、DescribeCertificate オペレーションを使用して、証明書の更新詳細が更新されたことを確認します。

証明書の更新ステータスの確認

証明書を更新しようとすると、ACM は証明書の詳細に 更新ステータス情報 フィールドを表示します。 AWS Certificate Manager コンソール、ACM API、 AWS CLI、または を使用して、ACM 証明書の更新ステータス Health Dashboard を確認できます。コンソール AWS CLI、または ACM API を使用する場合、更新ステータスには、以下に示す 4 つの可能なステータス値のいずれかを含めることができます。 Health Dashboardを使用する場合も同様の値が表示されます。

自動更新を保留中

ACM は、証明書のドメイン名を自動的に検証しようとしています。詳しくは、DNS によって検証されたドメインの更新 を参照してください。これ以上、何もする必要はありません。

検証保留中

ACM は、証明書の 1 つまたは複数のドメイン名を自動的に検証できませんでした。これらのドメイン名を検証するためにアクションを実行する必要があり、実行しない場合には証明書は更新されません。証明書の E メール検証を使用した場合は、ACM からの E メールを探し、その E メールのリンクに従って検証を実行します。DNS 検証を使用した場合は、DNS レコードが存在しており、証明書が引き続き使用されることを確認します。

Success

証明書にあるすべてのドメイン名が検証され、ACM は証明書を更新しました。これ以上、何もする必要はありません。

失敗

証明書が有効期限切れになる前に、1 つまたは複数のドメイン名が検証されませんでした。そして、ACM は証明書を更新しませんでした。新しい証明書をリクエスト できます。

証明書が Elastic Load Balancing や CloudFront などの別の AWS サービスに関連付けられている場合、または発行または最終更新以降にエクスポートされている場合、証明書は更新の対象となります。

> **Note**
>
> 証明書ステータスの変更が有効になるまでに数時間ほどかかる場合があります。問題が発生した場合、更新リクエストは 72 時間後にタイムアウトになるため、更新プロセスを最初から繰り返す必要があります。トラブルシューティングのヘルプについては、証明書のリクエストのトラブルシューティング を参照してください。

トピック

- ステータスの確認 (コンソール)
- ステータスの確認 (API)
- ステータスの確認 (CLI)

- Personal Health Dashboard (PHD) を使用してステータスを確認する

ステータスの確認 (コンソール)

次の手順では、ACM コンソールを使用して ACM 証明書の更新ステータスを確認する方法について説明します。

1. https://console.aws.amazon.com/acm/home で AWS Certificate Manager コンソールを開きます。
2. 証明書を展開して詳細を表示します。
3. [詳細] セクションで [更新ステータス] を見つけます。ステータスが表示されない場合、ACM でこの証明書のマネージド更新プロセスが開始されていません。

ステータスの確認 (API)

DescribeCertificate アクションを使用してステータスを確認する方法を示す Java のサンプルについては、証明書についての説明 を参照してください。

ステータスの確認 (CLI)

以下のサンプルは、AWS Command Line Interface (AWS CLI) を使用して 証明書の更新ステータスを確認する方法を示しています。

```
aws acm describe-certificate \
 --certificate-arn arn:aws:acm:region:account:certificate/certificate_ID
```

レスポンスの RenewalStatus フィールドの値を書き留めます。[RenewalStatus] フィールドが表示されない場合、ACM で証明書のマネージド更新プロセスが開始されていません。

Personal Health Dashboard (PHD) を使用してステータスを確認する

ACM は、パブリック証明書の場合は有効期限の 45 日前、プライベート証明書の場合は 60 日前に自動的に ACM 証明書を更新しようとします。ACM が証明書を自動的に更新できない場合は、有効期限から 45 Health Dashboard 日 (プライベートのみ) 、30 日、15 日、7 日、3 日、1 日の間隔で証明書更新イベント通知を に送信し、アクションを実行する必要があることを知らせます。 Health Dashboard は AWS Health サービスの一部です。セットアップを行う必要はなく、アカウントで認

証されているすべてのユーザーが表示できます。詳細については、AWS Health ユーザーガイドを参照してください。

> **Note**
>
> ACM は、PHD タイムラインの 1 つのイベントに、連続した更新イベント通知を書き込みます。更新が成功するまで、各通知は前の通知を上書きします。

Health Dashboardを使用するには:

1. https://phd.aws.amazon.com/phd/home#/ Health Dashboard で にログインします。
2. [Event log] を選択します。
3. [Filter by tags or attributes] の場合、[Service] を選択します。
4. [Certificate Manager] を選択します。
5. [Apply] (適用) を選択します。
6. [Event category] の場合、[Scheduled Change] を選択します。
7. [Apply] (適用) を選択します。

AWS Certificate Manager リソースにタグを付ける

タグとは、ACM 証明書に割り当てることができるラベルです。各タグは、キー と 値 から構成されます。 AWS Certificate Manager コンソール、 AWS Command Line Interface (AWS CLI)、または ACM API を使用して、ACM 証明書のタグを追加、表示、または削除できます。ACM コンソールに表示するタグを選択することができます。

必要に応じたカスタムタグを作成できます。たとえば、各 ACM 証明書が目的とする環境を識別するために、複数の ACM 証明書に `Environment = Prod` や `Environment = Beta` をタグ付けすることができます。次のリストでは、その他のカスタムタグの例をいくつか追加しています。

- `Admin = Alice`
- `Purpose = Website`
- `Protocol = TLS`
- `Registrar = Route53`

他の AWS リソースもタグ付けをサポートしています。こうして同じタグを別々のリソースに割り当てることができ、これらのリソースが関連付けられることを示します。たとえば、ACM 証明書、ロードバランサー、そして example.com ウェブサイトで使用されるその他のリソースに `Website = example.com` のようなタグを割り当てることができます。

トピック

- タグの制限
- タグの管理

タグの制限

ACM の証明書タグには以下のような基本制限があります。

- ACM の証明書あたりのタグの最大数は 50 です。
- タグキーの最大長は 127 文字です。
- タグ値の最大長は 255 文字です。
- タグのキーと値では、大文字と小文字が区別されます。

- aws: プレフィックスは AWS 使用のために予約されています。キーが で始まるタグを追加、編集、または削除することはできませんaws:。aws: で始まるタグは、リソースあたりのタグのクォータにカウントされません。
- 複数のサービス間およびリソース間でタグ付けスキーマを使用することを予定する場合、他のサービスでも許可される文字に制限が適用されることがあるのでご注意ください。該当するサービスのドキュメントを参照してください。
- ACM 証明書タグは、 AWS マネジメントコンソールのリソースグループとタグエディタでは使用できません。

AWS タグ付け規則の一般的な情報については、AWS 「リソースのタグ付け」を参照してください。

タグの管理

AWS マネジメントコンソール、、または AWS Certificate Manager API を使用して、タグを追加、編集 AWS Command Line Interface、削除できます。

タグの管理 (コンソール)

を使用して、タグ AWS マネジメントコンソール を追加、削除、編集できます。タグを列で表示することもできます。

タグを追加する

次の手順で、ACM コンソールを使用してタグを追加します。

証明書にタグを追加するには (コンソール)

1. にサインイン AWS マネジメントコンソール し、https://console.aws.amazon.com/acm/home で AWS Certificate Manager コンソールを開きます。
2. タグを付加する証明書の横にある矢印を選択します。
3. 詳細ペインで、[Tags] まで下へスクロールします。
4. [Edit] と [Add Tag] を選択します。
5. タグのキーと値を入力します。
6. [保存] を選択します。

タグの削除

次の手順で、ACM コンソールを使用してタグを削除します。

タグを削除するには (コンソール)

1. にサインイン AWS マネジメントコンソール し、https://console.aws.amazon.com/acm/home で AWS Certificate Manager コンソールを開きます。
2. 削除するタグの証明書の横にある矢印を選択します。
3. 詳細ペインで、[Tags] まで下へスクロールします。
4. [編集] を選択します。
5. 削除するタグの横にある [X] を選択します。
6. [保存] を選択します。

タグの編集

次の手順で、ACM コンソールを使用してタグを編集します。

タグを編集するには (コンソール)

1. にサインイン AWS マネジメントコンソール し、https://console.aws.amazon.com/acm/home で AWS Certificate Manager コンソールを開きます。
2. 編集する証明書の横にある矢印を選択します。
3. 詳細ペインで、[Tags] まで下へスクロールします。
4. [編集] を選択します。
5. 編集するタグのキーまたは値を変更します。
6. [保存] を選択します。

タグを列に表示する

次の手順を使用して ACM コンソールでタグを列で表示します。

タグを列で表示するには (コンソール)

1. にサインイン AWS マネジメントコンソール し、https://console.aws.amazon.com/acm/home で AWS Certificate Manager コンソールを開きます。

2. コンソールの右上の歯車アイコン

を選び、列に表示するタグを選択します。

3. 列として表示するタグの横にあるチェックボックスをオンにします。

タグを管理する (CLI)

AWS CLIを使用してタグの追加、一覧表示、および削除する方法については、以下のトピックを参照してください。

- add-tags-to-certificate
- list-tags-for-certificate
- remove-tags-from-certificate

タグを管理する (ACM API)

API を使用してタグの追加、一覧表示、および削除する方法については、以下のトピックを参照してください

- AddTagsToCertificate
- ListTagsForCertificate
- RemoveTagsFromCertificate

サービスと ACM の統合

AWS Certificate Manager は、増え続ける AWS サービスをサポートしています。ACM 証明書または プライベート AWS Private CA 証明書を、 AWS ベースのウェブサイトまたはアプリケーションに直接インストールすることはできません。

> **Note**
>
> パブリック ACM 証明書は、Nitro Enclave に接続されている Amazon EC2 インスタンスにインストールできます。任意の Amazon EC2 インスタンスで使用する[export a public certificate] (パブリック証明書をエクスポート) することもできます。Nitro Enclave に接続されていない Amazon EC2 インスタンスでのスタンドアロンウェブサーバーのセットアップについては、「チュートリアル: Amazon Linux 2 に LAMP ウェブサーバーをインストールする」または「チュートリアル: Amazon Linux AMI を使用した LAMP ウェブサーバーのインストール」を参照してください。

ACM 証明書は、次のサービスでサポートされています。

エラスティックロードバランシング

Elastic Load Balancing は、受信したアプリケーショントラフィックを複数の Amazon EC2 インスタンスに自動的に分散します。Elastic Load Balancing は問題のあるインスタンスを検出し、その問題のあるインスタンスが復旧するまで、自動的にトラフィック経路を正常なインスタンスに変更します。Elastic Load Balancing は、着信トラフィックに応じて、自動的にそのリクエスト処理能力を拡張します。ロードバランシングの詳細については、「Elastic Load Balancing ユーザーガイド」を参照してください。

通常の場合、SSL/TLS 経由で安全なコンテンツを供給するために、ロードバランサーまたはバックエンド Amazon EC2 インスタンスのどちらかに SSL/TLS 証明書がインストールされていることがロードバランサーより要求されます。ACM は Elastic Load Balancing と統合して、ロードバランサーに証明書をデプロイします。詳細については、「Application Load Balancer の作成」を参照してください。

Amazon CloudFront

Amazon CloudFront は、エッジロケーションのワールドワイドネットワークからコンテンツを配信することで、動的および静的なウェブコンテンツをエンドユーザーに高速で配信できるウェブ

サービスです。エンドユーザーが CloudFront を通じて提供しているコンテンツを要求すると、ユーザーは最も遅延が少ないエッジロケーションにルーティングされます。これにより、可能な限り最良のパフォーマンスでコンテンツが配信されます。コンテンツがこのエッジロケーションに現在存在する場合、CloudFront はコンテンツを直ちに配信します。コンテンツがこのエッジロケーションに現在存在しない場合、CloudFront は最終的なコンテンツソースとして識別する Amazon S3 バケットまたはウェブサーバーからそのコンテンツを取得します。CloudFront の詳細については、Amazon CloudFront デベロッパーガイドを参照してください。

SSL/TLS 経由で安全なコンテンツを供給するために、CloudFront ディストリビューションまたはバックエンドコンテンツソースのどちらかに SSL/TLS 証明書がインストールされていることが CloudFront より要求されます。ACM は CloudFront と統合され、CloudFront ディストリビューションに ACM 証明書をデプロイします。詳細については、「SSL/TLS 証明書を取得する」を参照してください。

> **Note**
>
> CloudFront で ACM 証明書を使用するには、米国東部 (バージニア北部) リージョンの証明書をリクエスト (またはインポート) している必要があります。

アマゾン エラスティックKubernetesサービス

Amazon Elastic Kubernetes Service は、独自の Kubernetes コントロールプレーンをインストール、運用、保守 AWS することなく、 で Kubernetes を簡単に実行できるようにするマネージド Kubernetes サービスです。Amazon EKS の詳細については、「Amazon Elastic Kubernetes Service ユーザーガイド」を参照してください。

AWS Controllers for Kubernetes (ACK) で ACM を使用して、Kubernetes ワークロードに TLS 証明書を発行およびエクスポートできます。この統合により、Amazon EKS ポッドを保護し、Kubernetes Ingress または AWS ロードバランサーで TLS を終了できます。ACM は証明書を自動的に更新し、ACK コントローラーは更新された証明書で Kubernetes シークレットを更新します。詳細については、「ACM 証明書を使用して Kubernetes ワークロードを保護する」を参照してください。

Amazon Cognito

Amazon Cognito は、ウェブおよびモバイルアプリの認証、認可、およびユーザー管理を提供します。ユーザーは、 AWS アカウント 認証情報を使用して直接サインインすることも、Facebook、Amazon、Google、Apple などのサードパーティーを通じてサインインすること

もできます。Amazon Cognito の詳細については、「Amazon Cognito デベロッパーガイド」を参照してください。

Cognito ユーザープールを設定して Amazon CloudFront プロキシを使用する場合、CloudFront により、カスタムドメインを保護するために ACM 証明書が配置される可能性があります。この場合には、証明書を削除する前に、この証明書の CloudFront との関連を削除する必要があることに注意してください。

AWS Elastic Beanstalk

Elastic Beanstalk は、これらのアプリケーションを実行するインフラストラクチャを気にすることなく、 AWS クラウドでのアプリケーションのデプロイと管理に役立ちます。 は、管理の複雑さ AWS Elastic Beanstalk を軽減します。アプリケーションをアップロードするだけで、Elastic Beanstalk は容量のプロビジョニング、ロードバランシング、スケーリング、および正常性モニタリングといった詳細を自動的に処理します。Elastic Beanstalk は、Elastic Load Balancing サービスを使用してロードバランサーを作成します Elastic Beanstalk の詳細については、「AWS Elastic Beanstalk Elastic Beanstalk 開発者ガイド」を参照してください。

証明書を選択するには、Elastic Beanstalk コンソールでアプリケーションのロードバランサーを設定する必要があります。詳細については、「HTTPS を復号するために Elastic Beanstalk 環境のロードバランサーを設定する」を参照してください。

AWS App Runner

App Runner は、ソースコードまたはコンテナイメージから AWS クラウド内のスケーラブルで安全なウェブアプリケーションに直接デプロイするための、高速でシンプルで費用対効果の高い方法を提供する AWS サービスです。新しいテクノロジーを学習したり、使用するコンピューティングサービスを決定したり、 AWS リソースをプロビジョニングして設定する方法を知っている必要はありません。App Runner の詳細については、「AWS App Runner デベロッパーガイド」を参照してください。

カスタムドメイン名を App Runner サービスに関連付けると、App Runner はドメインの有効性を追跡する証明書を内部で作成します。それらは ACM に保存されます。App Runner は、ドメインがサービスから関連付け解除された後、またはサービスが削除された後 7 日間これらの証明書を削除しません。このプロセス全体が自動化されているため、証明書を自分で追加または管理する必要はありません。詳細については、「AWS App Runner デベロッパーガイドの App Runner サービスのカスタムドメイン名の管理」を参照してください。

Amazon API Gateway

モバイルデバイスの増加と IoT (モノのインターネット) の成長とともに、データにアクセスしたり、 AWS上のバックエンドシステムとやり取りしたりするために API を作成するケースがま

すます増えてきています。API Gateway を利用すると、API を発行、管理、監視、保護できます。API を API Gateway にデプロイした後、カスタムドメイン名を設定すると、API に簡単にアクセスできます。カスタムドメイン名を設定するには、SSL/TLS 証明書を提供する必要があります。証明書を生成またはインポートするには、ACM を使用できます。Amazon API Gateway の詳細については、「Amazon API Gateway デベロッパーガイド」を参照してください。

AWS Nitro Enclaves

AWS Nitro Enclaves は、Amazon EC2 インスタンスから enclaves と呼ばれる分離された実行環境を作成できる Amazon EC2 機能です。Enclaves は、分離された、強化された、制約の厳しい仮想マシンです。親インスタンスとのセキュアなローカルソケット接続のみを提供します。永続的ストレージ、対話型アクセス、外部ネットワークはありません。ユーザーはエンクレーブに SSH 接続できません。また、エンクレーブ内のデータとアプリケーションには、親インスタンスのプロセス、アプリケーション、またはユーザー (ルートまたは管理者を含む) からアクセスできません。

Nitro Enclaves に接続されている EC2 インスタンスは、ACM 証明書をサポートします。詳細については、「Nitro Enclaves AWS Certificate Manager」を参照してください。

> **Note**
>
> Nitro Enclave に接続されていない EC2 インスタンスに ACM 証明書を関連付けることはできません。

AWS CloudFormation

CloudFormation は、Amazon Web Services リソースのモデル化とセットアップに役立ちます。Elastic Load Balancing や API Gateway など、使用する AWS リソースを記述するテンプレートを作成します。次に、 CloudFormation はプロビジョニングとそのためのリソースの設定を行います。 AWS リソースを個別に作成して設定し、何が何に依存しているかを把握する必要はありません。 はこれらすべて CloudFormation を処理します。ACM 証明書はテンプレートリソースとして含まれます。つまり、 CloudFormation は サービスで使用できる AWS ACM 証明書をリクエストして、安全な接続を有効にできます。さらに、ACM 証明書は、設定できる多くの AWS リソースに含まれています CloudFormation。

CloudFormation に関する一般的な情報については、CloudFormation ユーザーガイド を参照してください。CloudFormation でサポートされている ACM リソースの詳細については、「AWS::CertificateManager::Certificate」を参照してください。

が提供する強力な自動化により CloudFormation、特に新しい AWS アカウントでは、証明書のクォータを簡単に超えることができます。ACM のベストプラクティスに従うことをお勧めします CloudFormation。

> **Note**
>
> を使用して ACM 証明書を作成すると CloudFormation、 CloudFormation スタックは CREATE_IN_PROGRESS 状態のままになります。それ以上のスタック操作は、証明書の検証 E メールの指示に従うまで延期されます。詳細については、「スタックの作成、更新、または削除オペレーションの際、リソースが安定しない」を参照してください。

AWS Amplify

Amplify は、フロントエンドのウェブおよびモバイル開発者がフルスタックアプリケーションを迅速かつ簡単に構築できるようにする、専用のツールと機能のセットです AWS。Amplify は、Amplify Hosting と Amplify Studio の 2 つのサービスを提供します。Amplify Hosting は、継続的なデプロイメントでフルスタックのサーバーレスウェブアプリケーションをホストするための git ベースのワークフローを提供します。Amplify Studio は、スケーラブルなフルスタックのウェブおよびモバイルアプリの作成を簡素化するビジュアル開発環境です。Studio を使用して、すぐに使用できる一連の UI コンポーネントを使用してフロントエンド UI を構築し、アプリのバックエンドを作成し、2 つを接続します。Amplify の詳しい情報については、AWS Amplify ユーザーガイドを参照してください。

カスタムドメインをアプリケーションに接続すると、Amplify コンソールは ACM 証明書を発行してセキュリティを確保します。

Amazon OpenSearch Service

Amazon OpenSearch Service はログ分析、リアルタイムのアプリケーションモニタリング、クリックストリーム分析などのユースケース向けの検索および分析エンジンです。詳細については、「Amazon OpenSearch Service 開発者ガイド」を参照してください。

カスタムドメインとエンドポイントを含む OpenSearch Service クラスターを作成する場合、ACM を使用して、関連付けられた Application Load Balancer に証明書をプロビジョニングできます。

AWS Network Firewall

AWS Network Firewall は、すべての Amazon Virtual Private Cloud (VPCs。Network Firewall の詳細については、AWS Network Firewall デベロッパーガイドをご参照ください。

Network Firewall ファイアウォールは、TLS 検査のために ACM と統合されます。Network Firewall で TLS 検査を使用する場合は、ファイアウォールを通過する SSL/TLS トラフィックの復号化および再暗号化に ACM 証明書を設定する必要があります。Network Firewall と ACM for TLS インスペクションの動作については、「AWS Network Firewall デベロッパーガイド」の「TLS インスペクション構成で SSL/TLS 証明書を使用するための条件」を参照してください。

のセキュリティ AWS Certificate Manager

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からメリットを得られます。

セキュリティは、 AWS 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 – クラウドで AWS 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 AWS 責任があります。 AWS また、 では、安全に使用できるサービスも提供しています。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。が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 AWS Certificate Manager、「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、 AWS Certificate Manager (ACM) を使用する際の責任共有モデルの適用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 ACM を設定する方法を示します。また、ACM 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。

トピック

- でのデータ保護 AWS Certificate Manager
- の Identity and Access Management AWS Certificate Manager
- の耐障害性 AWS Certificate Manager
- のインフラストラクチャセキュリティ AWS Certificate Manager
- ベストプラクティス

でのデータ保護 AWS Certificate Manager

責任 AWS 共有モデル、 でのデータ保護に適用されます AWS Certificate Manager。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、データプライバシーに関するよくある質問を参照してください。欧州でのデータ保護の詳細については、AWS セキュリティブログに投稿された「AWS 責任共有モデルおよび GDPR」のブログ記事を参照してください。

データ保護の目的で、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、 AWS のサービス API、 AWS CLIまたは 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。外部サーバーに URL を提供する場合、そのサーバーへのリクエストを検証できるように、認証情報を URL に含めないことを強くお勧めします。

証明書の秘密鍵のセキュリティ

パブリック証明書をリクエストすると、 AWS Certificate Manager (ACM) はパブリック/プライベートキーペアを生成します。インポートされた証明書の場合、キーペアが生成されます。パブリック

キーは証明書の一部となります。ACM は証明書とそれに対応するプライベートキーを保存し、プライベートキーを保護するために AWS Key Management Service (AWS KMS) を使用します。このプロセスの動作は次のようになります。

1. AWS リージョンで証明書を初めてリクエストまたはインポートすると、ACM はエイリアス aws/acm AWS KMS key でマネージド を作成します。この KMS キーは、各 AWS アカウントと各 AWS リージョンで一意です。
2. ACM は、この KMS キーを使用して証明書のプライベートキーを暗号化します。ACM は暗号化されたバージョンのプライベートキーのみを保存します (ACM はプレーンテキスト形式でプライベートキーを保存しません)。ACM は同じ KMS キーを使用して、特定の AWS アカウントと特定の AWS リージョンのすべての証明書のプライベートキーを暗号化します。
3. 証明書を AWS Certificate Managerと統合されたサービスに関連付けると、ACM は証明書と暗号化されたプライベートキーをサービスに送信します。サービスが KMS キーを使用して証明書のプライベートキーを復号 AWS KMS できるようにするグラントも に作成されます。許可の詳細については、「AWS Key Management Service デベロッパーガイドの許可の使用」を参照してください。ACM でサポートされているサービスの詳細については、「サービスと ACM の統合」を参照してください。

   > **Note**
   >
   > 自動的に作成された AWS KMS 権限はユーザーが制御できます。何らかの理由でこの許可を削除すると、統合サービスの ACM 機能は失われます。

4. 統合サービスは、KMS キーを使用してプライベートキーを復号化します。続いて、サービスは、証明書と復号された (プレーンテキスト) プライベートキーを使用してクライアントと安全な通信チャネル (SSL/TLS セッション) を確立します。
5. 証明書と統合サービスとの関連付けが解除されると、ステップ 3 で作成された許可は廃止されます。つまり、サービスは KMS キーを使用して証明書のプライベートキーを復号化できなくなります。

の Identity and Access Management AWS Certificate Manager

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、ACM リソースを使用するための 認証 (サインイン) および 許可 (アクセス許可を持たせる) を行うことができる人を制御します。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
- アイデンティティを使用した認証
- ポリシーを使用したアクセスの管理
- が IAM と AWS Certificate Manager 連携する方法
- のアイデンティティベースのポリシーの例 AWS Certificate Manager
- ACM API のアクセス許可: アクションとリソースのリファレンス
- AWS の マネージドポリシー AWS Certificate Manager
- ACM での条件キーの使用
- ACM でのサービスにリンクされたロール (SLR) の使用
- AWS Certificate Manager ID とアクセスのトラブルシューティング

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 (「AWS Certificate Manager ID とアクセスの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 (「が IAM と AWS Certificate Manager 連携する方法」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 (「のアイデンティティベースのポリシーの例 AWS Certificate Manager」を参照)

アイデンティティを使用した認証

認証とは、ID 認証情報 AWS を使用して にサインインする方法です。、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 ルート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。

フェデレーション 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 にアクセスする必要がある 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。

が IAM と AWS Certificate Manager 連携する方法

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

で使用できる IAM 機能 AWS Certificate Manager

IAM 機能	ACM 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	はい
ポリシー条件キー (サービス固有)	はい
ACL	なし
ABAC (ポリシー内のタグ)	あり
一時的な認証情報	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

ACM 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。

ACM 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。

ACM のアイデンティティベースのポリシーの例

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

ACM 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。

ACM 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。

ACM 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「AWS Certificate Managerで定義されるアクション」を参照してください。

ACM 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。

```
acm
```

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": [
      "acm:action1",
      "acm:action2"
         ]
```

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

ACM 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": "*"
```

ACM リソースのタイプとその ARN のリストを確認するには、「サービス認可リファレンス」の「AWS Certificate Manager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、「AWS Certificate Managerで定義されるアクション」を参照してください。

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

ACM のポリシー条件キー

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。

ACM 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「AWS Certificate Managerの条件キー」を参照してください。条件キーを使用できるアクションとリソースについては、「 で定義されるアクション AWS Certificate Manager」を参照してください。

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

ACM 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。

ACM での ABAC

ABAC (ポリシー内のタグ) のサポート: あり

属性ベースのアクセス制御 (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「属属性ベースのアクセス制御 (ABAC) を使用する」を参照してください。

ACM での一時的な認証情報の使用

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。

ACM のクロスサービアクセス権限

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。

ACM 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。

> **Warning**
>
> サービスロールのアクセス許可を変更すると、ACM の機能が阻害される可能性があります。ACM が指示する場合以外は、サービスロールを編集しないでください。

ACM のサービスリンクロール

サービスリンクロールのサポート: あり

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

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、「IAM と提携するAWS のサービス」を参照してください。表の「サービスリンクロール」列に Yes と記載されたサービスを見つけます。サービスにリンクされたロールに関するドキュメントをサービスで表示するには、[はい] リンクを選択します。

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

デフォルトでは、ユーザーおよびロールには、ACM 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。

ACM が定義するアクションとリソースタイプ (リソースタイプごとの ARN の形式を含む) の詳細については、「サービス認可リファレンス」の「AWS Certificate Manager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
- ACM コンソールの使用
- 自分の権限の表示をユーザーに許可する
- 証明書の一覧
- 証明書のリクエスト
- 証明書の取得
- 証明書のインポート

- 証明書の削除

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 ACM 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 – 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。

ACM コンソールの使用

AWS Certificate Manager コンソールにアクセスするには、最小限のアクセス許可のセットが必要です。これらのアクセス許可により、 の ACM リソースの詳細を一覧表示および表示できます AWS アカウント。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリシーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません。代わりに、実行しようとしている API オペレーションに一致するアクションのみへのアクセスが許可されます。

ユーザーとロールが引き続き ACM コンソールを使用できるようにするには、エンティティに ACM *AWSCertificateManagerReadOnly* AWS 管理ポリシーもアタッチします。詳細については、「IAM ユーザーガイド」の「ユーザーへのアクセス許可の追加」を参照してください。

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": "*"
        }
    ]
}
```

証明書の一覧

次のポリシーを使用すると、ユーザーはユーザーアカウントのすべての ACM 証明書を一覧表示できます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "acm:ListCertificates",
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> ACM 証明書が Elastic Load Balancing と CloudFront コンソールに表示されるためには、このアクセス許可が必要です。

証明書のリクエスト

次のポリシーは、ユーザーが ACM エクスポート可能なパブリック証明書をリクエストすることを拒否します。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DenyACMCertificateRequest",
      "Effect": "Deny",
      "Action": [
        "acm:RequestCertificate"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "acm:Export": "ENABLED"
        }
      }
    }
  ]
}
```

証明書の取得

次のポリシーを使用すると、ユーザーは特定の ACM 証明書を取得できます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "acm:GetCertificate",
```

```
        "Resource": "arn:aws:acm:us-
east-1:123456789012:certificate/certificate_ID"
    }
}
```

証明書のインポート

次のポリシーを使用すると、ユーザーは証明書をインポートできます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "acm:ImportCertificate",
        "Resource": "arn:aws:acm:us-
east-1:123456789012:certificate/certificate_ID"
    }
}
```

証明書の削除

次のポリシーを使用すると、ユーザーは特定の ACM 証明書を削除できます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "acm:DeleteCertificate",
        "Resource": "arn:aws:acm:us-
east-1:123456789012:certificate/certificate_ID"
    }
}
```

ACM API のアクセス許可: アクションとリソースのリファレンス

アクセスコントロールをセットアップし、IAM ユーザーまたはロールにアタッチできるアクセス権限ポリシーを作成するときは、以下の表をリファレンスとして使用できます。テーブルの最初の列には、各 AWS Certificate Manager API オペレーションが一覧表示されます。ポリシーの `Action` 要素でアクションを指定します。残りの列では、追加情報が示されます。

ACM ポリシーの IAM ポリシー要素を使用して、条件を表現できます。完全なリストについては、IAM ユーザーガイドの「利用可能なキー」を参照してください。

> **Note**
>
> アクションを指定するには、API オペレーション名 (`acm:RequestCertificate` など) の前に `acm:` プレフィックスを使用します。

ACM API オペレーションとアクセス許可

ACM API オペレーション	必要なアクセス許可 (API オペレーション)	リソース
AddTagsToCertificate	`acm:AddTagsToCertificate`	`arn:aws:acm:` *region*`:`*account*`:certificate/` *certificate_ID*
DeleteCertificate	`acm:DeleteCertificate`	`arn:aws:acm:` *region*`:`*account*`:certificate/` *certificate_ID*
DescribeCertificate	`acm:DescribeCertificate`	`arn:aws:acm:` *region*`:`*account*`:certificate/` *certificate_ID*
ExportCertificate	`acm:ExportCertificate`	`arn:aws:acm:` *region*`:`*account*`:certificate/` *certificate_ID*
GetAccountConfiguration	`acm:GetAccountConfiguration`	*

ACM API オペレーション	必要なアクセス許可 (API オペレーション)	リソース
GetCertificate	acm:GetCertificate	arn:aws:acm: *region*:*account*:certificate/ *certificate_ID*
ImportCertificate	acm:ImportCertificate	arn:aws:acm: *region*:*account*:certificate/* または *
ListCertificates	acm:ListCertificates	*
ListTagsForCertificate	acm:ListTagsForCertificate	arn:aws:acm: *region*:*account*:certificate/ *certificate_ID*
PutAccountConfiguration	acm:PutAccountConfiguration	*
RemoveTagsFromCertificate	acm:RemoveTagsFromCertificate	arn:aws:acm: *region*:*account*:certificate/ *certificate_ID*
RequestCertificate	acm:RequestCertificate	arn:aws:acm: *region*:*account*:certificate/* または *
ResendValidationEmail	acm:ResendValidationEmail	arn:aws:acm: *region*:*account*:certificate/ *certificate_ID*

ACM API オペレーション	必要なアクセス許可 (API オペレーション)	リソース
UpdateCertificateOptions	acm:UpdateCertificateOptions	arn:aws:acm: *region*:*account*:certificate/ *certificate_ID*

AWS の マネージドポリシー AWS Certificate Manager

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、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るため、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。

AWSCertificateManagerReadOnly

このポリシーは、ACM 証明書への読み取り専用アクセスを提供します。これにより、ユーザーは、概要表示、一覧表示、そして ACM 証明書の取得ができます。

コンソールでこの AWS 管理ポリシーを表示するには、https://console.aws.amazon.com/iam/home#policies/arn:aws:iam::aws:policy/AWSCertificateManagerReadOnly にアクセスします。

ポリシーの詳細の JSON リストについては、AWSCertificateManagerReadOnly を参照してください。

AWSCertificateManagerFullAccess

このポリシーは、すべての ACM アクションとリソースへのフルアクセスを提供します。

コンソールでこの AWS 管理ポリシーを表示するには、https://console.aws.amazon.com/iam/home#policies/arn:aws:iam::aws:policy/AWSCertificateManagerFullAccess にアクセスします。

ポリシーの詳細の JSON リストについては、AWSCertificateManagerFullAccess を参照してください。

AWS 管理ポリシーへの ACM 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 ACM 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、ACM ドキュメント履歴ページの RSS フィードを購読してください。

変更	説明	日付
`GetAccountConfiguration` AWSCertificateManagerReadOnly ポリシーのサポートを追加。	`AWSCertificateManagerReadOnly` ポリシーに `GetAccountConfiguration` API アクションを呼び出すアクセス許可が含まれるようになりました。	2021 年 3 月 3 日
ACM が変更の追跡を開始	ACM は、 AWS 管理ポリシーの変更の追跡を開始します。	2021 年 3 月 3 日

ACM での条件キーの使用

AWS Certificate Manager は AWS Identity and Access Management (IAM) 条件キーを使用して、証明書リクエストへのアクセスを制限します。IAM ポリシーまたはサービスコントロールポリシー (SCP) の条件キーを使用して、組織のガイドラインに準拠した証明書リクエストを作成できます。

> **Note**
>
> ACM 条件キーを などの AWS グローバル条件キーと組み合わせてaws:PrincipalArn、アクションを特定のユーザーまたはロールにさらに制限します。

ACM のサポートされている条件

ACM API オペレーションとサポートされている条件

条件キー	サポートされている ACM API オペレーション	型	説明
acm:ValidationMethod	RequestCertificate	文字列 (DNS、EMAIL、HTTP)	ACM 検証方法 に基づいてリクエストをフィルタリング
acm:DomainNames	RequestCertificate	ArrayOfString	ACM リクエストのドメイン名に基づいてフィルタリング
acm:KeyAlgorithm	RequestCertificate	String	ACM キーアルゴリズムとサイズに基づいてリクエストをフィルタリング
acm:CertificateTransparencyLogging	RequestCertificate	文字列 (ENABLED、DISABLED)	ACM 証明書の透明性ログ記録設定に基づいてリクエストをフィルタリング
acm:CertificateAuthority	RequestCertificate	ARN	ACM リクエストの認証機関に基づいてリクエストをフィルタリング

例 1: 検証方法の制限

次のポリシーは、arn:aws:iam::123456789012:role/AllowedEmailValidation ロールを使用して行われたリクエストを除き、E メール検証方式を使用する新しい証明書リクエストを拒否します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement":{
        "Effect":"Deny",
        "Action":"acm:RequestCertificate",
        "Resource":"*",
        "Condition":{
            "StringLike" : {
                "acm:ValidationMethod":"EMAIL"
            },
            "ArnNotLike": {
                "aws:PrincipalArn": [ "arn:aws:iam::123456789012:role/
AllowedEmailValidation"]
            }
        }
    }
}
```

例 2: ワイルドカードドメインの防止

次のポリシーは、ワイルドカードドメインを使用する新しい ACM 証明書リクエストをすべて拒否します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement":{
        "Effect":"Deny",
```

```
        "Action":"acm:RequestCertificate",
        "Resource":"*",
        "Condition": {
            "ForAnyValue:StringLike": {
                "acm:DomainNames": [
                    "${*}.*"
                ]
            }
        }
    }
}
```

例 3: 証明書ドメインの制限

次のポリシーは、末尾が *.amazonaws.com ではないドメインの新しい ACM 証明書リクエストをすべて拒否します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement":{
        "Effect":"Deny",
        "Action":"acm:RequestCertificate",
        "Resource":"*",
        "Condition": {
            "ForAnyValue:StringNotLike": {
                "acm:DomainNames": ["*.amazonaws.com"]
            }
        }
    }
}
```

ポリシーを特定のサブドメインにさらに制限することができます。このポリシーは、すべてのドメインが少なくとも 1 つの条件付きドメイン名と一致するリクエストのみを許可します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement":{
        "Effect":"Deny",
        "Action":"acm:RequestCertificate",
        "Resource":"*",
        "Condition": {
            "ForAllValues:StringNotLike": {
                "acm:DomainNames": ["support.amazonaws.com",
 "developer.amazonaws.com"]
            }
        }
    }
}
```

例 4: キーアルゴリズムの制限

次のポリシーは、条件キー StringNotLike を使用して、ECDSA 384 ビット (EC_secp384r1) キーアルゴリズムで要求された証明書のみを許可します。

JSON

```
{
    "Version":"2012-10-17",
        "Statement":{
        "Effect":"Deny",
        "Action":"acm:RequestCertificate",
        "Resource":"*",
        "Condition":{
            "StringNotLike" : {
                "acm:KeyAlgorithm":"EC_secp384r1"
            }
        }
    }
}
```

次のポリシーは、条件キー StringLike とワイルドカード * のマッチングを使用して、いずれかの RSA キーアルゴリズムを使用する ACM での新しい証明書のリクエストを防ぎます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement":{
        "Effect":"Deny",
        "Action":"acm:RequestCertificate",
        "Resource":"*",
        "Condition":{
            "StringLike" : {
                "acm:KeyAlgorithm":"RSA*"
            }
        }
    }
}
```

例 5: 認証機関の制限

以下のポリシーは、提供された Private Certificate Authority (PCA) ARN を使用するプライベート証明書のリクエストのみを許可します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement":{
        "Effect":"Deny",
        "Action":"acm:RequestCertificate",
        "Resource":"*",
        "Condition":{
            "StringNotLike": {
```

```
                "acm:CertificateAuthority":" arn:aws:acm-
pca:region:account:certificate-authority/CA_ID"
            }
        }
    }
}
```

このポリシーは acm:CertificateAuthority 条件を使用して、Amazon Trust Services が発行した公的に信頼できる証明書のリクエストのみを許可します。認証機関 ARN を false に設定すると、PCA からのプライベート証明書のリクエストが防止されます。

JSON

```
{
"Version":"2012-10-17",
    "Statement":{
        "Effect":"Deny",
        "Action":"acm:RequestCertificate",
        "Resource":"*",
        "Condition":{
            "Null" : {
                "acm:CertificateAuthority":"false"
            }
        }
    }
}
```

ACM でのサービスにリンクされたロール (SLR) の使用

AWS Certificate Manager は、 AWS Identity and Access Management (IAM) サービスにリンクされたロールを使用して、プライベート CA から発行されたプライベート証明書の自動更新を、 が共有している別のアカウントの有効にします AWS Resource Access Manager。サービスにリンクされたロール (SLR) は、ACM のサービスに直接リンクされた一意のタイプの IAM ロールです。SLR は、ACM によって事前定義されており、ユーザーの代わりにサービスから他の AWS のサービスを呼び出す必要のあるアクセス許可がすべて含まれています。

SLR を使用すると、無人証明書の署名に必要なアクセス許可を手動で追加する必要がなくなるため、ACM の設定が簡単になります。ACM は、この SLR 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、ACM のみがそのロールを引き受けることができます。定義されるアクセス許可は、信頼ポリシーとアクセス許可ポリシーに含まれており、そのアクセス許可ポリシーを他の IAM エンティティにアタッチすることはできません。

SLR をサポートする他のサービスの情報ついては、[AWS Services That Work with IAM] (IAM と連携するサービス) を参照して、[Service-Linked Role] (サービスリンクロール) 列が[Yes] (はい) となっているサービスを探してください。SLR に関するドキュメントをサービスで表示するには、[はい] リンクを選択します。

ACM の SLR アクセス許可

ACM は、Amazon Certificate Manager サービスロールポリシーという名前の SLR を使用します。

AWSServiceRoleForCertificateManager SLR では、以下のサービスを信頼してロールを引き受けます。

- `acm.amazonaws.com`

ロールのアクセス許可ポリシーは、指定したリソースに対して以下のアクションを実行することを ACM に許可します。

- アクション:"*" での`acm-pca:IssueCertificate`、`acm-pca:GetCertificate`

SLR の作成、編集、削除を IAM エンティティ (ユーザー、グループ、ロールなど) に許可するには、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 ユーザーガイドの「サービスリンクロールのアクセス許可」を参照してください。

> **Important**
>
> ACM では、アカウントに SLR が存在するかどうかを判断できないという警告が表示されることがあります。必要な `iam:GetRole` アクセス許可がすでにアカウントの ACM SLR に付与されている場合、SLR の作成後にアラートは再発しません。再発する場合は、管理者またはアカウント管理者が `iam:GetRole` アクセス許可を ACM に付与するか、アカウントを ACM 管理ポリシー `AWSCertificateManagerFullAccess` に関連付けます。

ACM の SLR の作成

ACM で使用する SLR 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 AWS API を使用して ACM 証明書を発行すると、ACM は、 が共有する別のアカウントのプライベート CA が証明書に署名 AWS RAM するのを初めて行うときに SLR を作成します。

アカウントに SLR が存在するかどうかを ACM が判断できないというメッセージが表示された場合は、 が AWS Private CA 必要とする読み取りアクセス許可がアカウントに付与されていない可能性があります。これにより、SLR のインストールが妨げられることはなく、証明書を発行することはできますが、問題を解決するまで ACM は証明書を自動的に更新できません。(詳細については、ACM のサービスにリンクされたロール (SLR) に関する問題 を参照してください)。

> **Important**
>
> この SLR がアカウントに表示されるのは、このロールでサポートされている機能を使用する別のサービスでアクションが完了した場合です。また、2017 年 1 月 1 日より前に ACM サービスを使用していた場合、SLR のサポートが開始された時点で、ACM が AWSServiceRoleForCertificateManager ロールをアカウントに作成済みです。詳細については、「IAM アカウントに新しいロールが表示される」を参照してください。

この SLR を削除した後で再度作成する必要がある場合は、次のいずれかの方法を使用できます。

- IAM コンソールで、[Role] (ロール)、[Create role] (ロールの作成)、[Certificate Manager] (証明書管理) を選択して、[CertificateManagerServiceRolePolicy] (証明書管理のサービスロールポリシー) ユースケースで新しいロールを作成します。
- IAM API CreateServiceLinkedRole または対応する AWS CLI コマンド create-service-linked-role を使用して、`acm.amazonaws.com`サービス名で SLR を作成します。

詳細については、IAM ユーザーガイドの「サービスリンクロールの作成」を参照してください。

ACM の SLR の編集

ACM では、AWSServiceRoleForCertificateManager のサービスにリンクされたロールを編集することはできません。ロールは多くのエンティティにより参照されるため、SLR を作成した後、ロールの名前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「サービスリンクロールの編集」を参照してください。

ACM の SLR の削除

通常、AWSServiceRoleForCertificateManager SLR を手動で削除する必要はありません。ただし、IAM コンソール、、 AWS CLI または AWS API を使用して手動でロールを削除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削除」を参照してください。

ACM SLR でサポートされるリージョン

ACM AWS Private CA は、ACM と の両方が利用可能なすべてのリージョンで SLRs 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。

リージョン名	リージョン識別子	ACM のサポート
米国東部 (バージニア北部)	us-east-1	はい
米国東部 (オハイオ)	us-east-2	はい
米国西部 (北カリフォルニア)	us-west-1	はい
米国西部 (オレゴン)	us-west-2	はい
アジアパシフィック (ムンバイ)	ap-south-1	はい
アジアパシフィック (大阪)	ap-northeast-3	はい
アジアパシフィック (ソウル)	ap-northeast-2	はい
アジアパシフィック (シンガポール)	ap-southeast-1	はい
アジアパシフィック (シドニー)	ap-southeast-2	はい
アジアパシフィック (東京)	ap-northeast-1	はい
カナダ (中部)	ca-central-1	はい
欧州 (フランクフルト)	eu-central-1	はい
欧州 (チューリッヒ)	eu-central-2	はい
欧州 (アイルランド)	eu-west-1	はい

リージョン名	リージョン識別子	ACM のサポート
欧州 (ロンドン)	eu-west-2	はい
欧州 (パリ)	eu-west-3	はい
南米 (サンパウロ)	sa-east-1	はい
AWS GovCloud (米国西部)	us-gov-west-1	はい
AWS GovCloud (米国東部) 東部	us-gov-east-1	はい

AWS Certificate Manager ID とアクセスのトラブルシューティング

次の情報は、ACM 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。

トピック

- ACM でアクションを実行する権限がない
- ACM で証明書をリクエストする権限がない
- 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 ACM リソース AWS アカウント へのアクセスを許可したい

ACM 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 acm:*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:
 acm:GetWidget on resource: my-example-widget
```

この場合、acm:*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。サインイン資格情報を提供した担当者が管理者です。

ACM で証明書をリクエストする権限がない

このエラーが表示された場合は、ACM または PKI 管理者が、現在の状態で証明書をリクエストできないようにするルールを設定しています。

以下の例のエラーは、IAM ユーザーがコンソールを使用して、組織の管理者により DENY で設定されたオプションを使用する証明書をリクエストしようとする場合に発生します。

```
User: arn:aws:sts::account::ID: is not authorized to perform: acm:RequestCertificate
on resource: arn:aws:acm:region:account:certificate/*
with an explicit deny in a service control policy
```

この場合は、管理者が設定したポリシーに従って再度リクエストする必要があります。または、ポリシーを更新して証明書のリクエストを許可する必要があります。

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 ACM にロールを渡すことができるようにする必要があります。

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

- ACM 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、「が IAM と AWS Certificate Manager 連携する方法」を参照してください。
- 所有 AWS アカウント している のリソースへのアクセスを提供する方法については、「IAM ユーザーガイド」の「所有 AWS アカウント している別の の IAM ユーザーへのアクセスを提供する」を参照してください。
- リソースへのアクセスをサードパーティーに提供する方法については AWS アカウント、IAM ユーザーガイドの「サードパーティー AWS アカウント が所有する へのアクセスを提供する」を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、IAM ユーザーガイド の 外部で認証されたユーザー (ID フェデレーション) へのアクセスの許可 を参照してください。
- クロスアカウントアクセス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。

の耐障害性 AWS Certificate Manager

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、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ー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、フォールトトレランス、および拡張性が優れています。

AWS リージョンとアベイラビリティ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。

のインフラストラクチャセキュリティ AWS Certificate Manager

マネージドサービスである AWS Certificate Manager は、 AWS グローバルネットワークセキュリティで保護されています。 AWS セキュリティサービスと がインフラストラクチャ AWS を保護する方法については、AWS 「 クラウドセキュリティ」を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して環境を AWS 設計するには、「Security Pillar AWS Well-Architected Framework」の「Infrastructure Protection」を参照してください。

AWS 公開された API コールを使用して、ネットワーク経由で ACM にアクセスします。クライアントは次をサポートする必要があります。

- Transport Layer Security (TLS)。TLS 1.2 が必須で、TLS 1.3 をお勧めします。
- DHE (楕円ディフィー・ヘルマン鍵共有) や ECDHE (楕円曲線ディフィー・ヘルマン鍵共有) などの完全前方秘匿性 (PFS) による暗号スイート。これらのモードは Java 7 以降など、ほとんどの最新システムでサポートされています。

ACM へのプログラムによるアクセス権の管理

ユーザーが の AWS 外部で を操作する場合は、プログラムによるアクセスが必要です AWS マネジメントコンソール。プログラムによるアクセスを許可する方法は、 がアクセスするユーザーのタイプによって異なります AWS。

ユーザーにプログラムによるアクセス権を付与するには、以下のいずれかのオプションを選択します。

プログラムによるアクセス権を必要とするユーザー	目的	方法
IAM	(推奨) コンソール認証情報を一時的な認証情報として使用して AWS CLI、、 AWS SDKs、または AWS APIs。	使用するインターフェイスの指示に従ってください。 • については AWS CLI、AWS Command Line Interface 「 ユーザーガイド」のAWS 「ローカル開発のための ログイン」を参照してください。

プログラムによるアクセス権を必要とするユーザー	目的	方法
		• AWS SDKs「 SDK およびツールリファレンスガイド」の「Login for AWS local development」を参照してください。 AWS SDKs
ワークフォースアイデンティティ (IAM アイデンティティセンターで管理されているユーザー)	一時的な認証情報を使用して AWS CLI、、 AWS SDKs、または AWS APIs。	使用するインターフェイスの指示に従ってください。 • については AWS CLI、「 AWS Command Line Interface ユーザーガイド」の「 を使用する AWS CLI ように を設定する AWS IAM アイデンティティセンター」を参照してください。 • AWS SDKs、ツール、API については、 AWS APIs「 SDK およびツールリファレンスガイド」の「IAM アイデンティティセンター認証」を参照してください。 AWS SDKs
IAM	一時的な認証情報を使用して AWS CLI、、 AWS SDKs、または AWS APIs。	「IAM ユーザーガイド」の「 AWS リソースでの一時的な認証情報の使用」の手順に従います。

プログラムによるアクセス権を必要とするユーザー	目的	方法
IAM	(非推奨) 長期認証情報を使用して、AWS CLI、 AWS SDKs、または AWS APIs。	使用するインターフェイスの指示に従ってください。 • については AWS CLI、「 AWS Command Line Interface ユーザーガイド」の「IAM ユーザー認証情報を使用した認証」を参照してください。 • AWS SDKs「 SDK とツールリファレンスガイド」の「長期認証情報を使用した認証」を参照してください。 AWS SDKs • API AWS APIs「IAM ユーザーのアクセスキーの管理」を参照してください。

ベストプラクティス

ベストプラクティスは、 AWS Certificate Manager (AWS Certificate Manager) をより効果的に使用するのに役立つ推奨事項です。次のベストプラクティスは、現在の ACM クライアントの実際の経験に基づいています。

トピック

- アカウントレベルの分離
- AWS CloudFormation
- カスタム信頼ストア
- 証明書のピンニング
- ドメイン検証
- ドメイン名の追加または削除
- 証明書の透明性ログ記録のオプトアウト

- オンにする AWS CloudTrail

アカウントレベルの分離

ポリシーでアカウントレベルの分離を使用して、アカウントレベルで証明書にアクセスできるユーザーを制御します。本稼働用の証明書は、テスト用証明書や開発証明書とは別のアカウントに保管してください。アカウントレベルの分離を使用できない場合は、ポリシーで kms:CreateGrant アクションを拒否することで、特定のロールへのアクセスを制限できます。これにより、アカウント内で証明書に署名できるロールを高レベルで制限できます。許可に関する情報、特に用語については、AWS Key Management Service デベロッパーガイドの「AWS KMSでの許可」を参照してください。

アカウント単位での kms:CreateGrant の使用制限以上の詳細な制御が必要な場合は、kms:EncryptionContext 条件キーを使用して特定の証明書に kms:CreateGrant を制限できます。キーとして arn:aws:acm を指定し、制限する ARN の値を指定します。次のポリシー例は、特定の証明書の使用を防ぎ、他の証明書の使用を許可します。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
       {
           "Sid": "VisualEditor0",
           "Effect": "Deny",
           "Action": "kms:CreateGrant",
           "Resource": "*",
           "Condition": {
               "StringEquals": {
                   "kms:EncryptionContext:aws:acm:arn": "arn:aws:acm:us-
east-1:111122223333:certificate/b26def74-1234-4321-9876-951d4c07b197"
               }
           }
       }
   ]
}
```

AWS CloudFormation

AWS CloudFormation を使用すると、使用する AWS リソースを記述するテンプレートを作成できます。 CloudFormation その後、 はそれらのリソースをプロビジョニングして設定します。CloudFormation は、Elastic Load Balancing、Amazon CloudFront、Amazon API Gateway などの ACM でサポートされているリソースをプロビジョニングできます。詳細については、「サービスと ACM の統合」を参照してください。

CloudFormation を使用して複数のテスト環境をすばやく作成および削除する場合は、環境ごとに個別の ACM 証明書を作成しないことをお勧めします。これを行うと、証明書のクォータをすぐに使い切ってしまいます。(詳細については、クォータ を参照してください)。代わりに、テストに使用しているすべてのドメイン名をカバーするワイルドカード証明書を作成します。例えば、`<version>.service.example.com` などの、バージョン番号だけ異なるドメイン名に対して ACM 証明書を繰り返し作成する場合は、代わりに `<*>.service.example.com` のワイルドカード証明書を 1 つ作成します。

> **Important**
>
> Amazon CloudFront ディストリビューションを使用している場合、HTTP 検証はワイルドカード証明書をサポートしていないことにご注意ください。Amazon CloudFront で使用するワイルドカード証明書を CloudFormation テンプレートに含める場合は、DNS 検証または E メール検証を使用する必要があります。自動更新機能には DNS 検証をお勧めします。

がテスト環境の作成 CloudFormation に使用するテンプレートにワイルドカード証明書を含めます。

カスタム信頼ストア

ACM 証明書で保護されたエンドポイントへの接続を確保するために、Amazon ルート をカスタム信頼ストアに含めることをお勧めします。Amazon ルート認証機関は、さまざまなキータイプとアルゴリズムを表すことができます。Starfield Services ルート認証局 - G2 は、更新できない他の古い信頼ストアやクライアントと互換性がある古いルートです。すべてのルート CA を含めることで、アプリケーションの最大の互換性を確保できます。

証明書のピンニング

証明書ピニング (SSL ピンニングとも呼ばれる) は、アプリケーションで、ホストに証明書階層ではなく X.509 証明書またはパブリックキーを直接関連付けることによって、リモートホストを検証するのに使用できるプロセスです。したがって、アプリケーションでは、ピニングを使用して SSL/

TLS 証明書チェーンの検証をバイパスします。一般的な SSL 検証プロセスでは、ルート認証局 (CA) 証明書から下位 CA 証明書 (存在する場合) まで、証明書チェーン全体の署名をチェックします。また、階層の最下位にあるリモートホストの証明書もチェックします。その代わりに、アプリケーションは、証明書をピニングすることにより、リモートホストに、ルート証明書またはチェーン内の他のものではなく、その証明書のみが信頼できるということを伝えられます。リモートホストの証明書またはパブリックキーを開発中にアプリケーションに追加できます。または、最初にホストに接続する際にアプリケーションが証明書またはキーを追加することができます。

> **Warning**
>
> アプリケーションでは、ACM 証明書をピニングしないことをお勧めします。&CM は、でのマネージド証明書の更新 AWS Certificate Manager を実行し、Amazon 発行の SSL/TLS 証明書を有効期限が切れる前に自動的に更新します。証明書を更新するために ACM は新しいパブリックキーとプライベートキーのペアを生成します。アプリケーションが ACM 証明書をピン留めし、証明書が新しいパブリックキーで正常に更新された場合、アプリケーションはドメインに接続できない可能性があります。

証明書をピンすることを決定した場合は、次のオプションを選択しても、アプリケーションのドメインへの接続が妨げられることはありません。

- 保持する証明書を ACM にインポートし、アプリケーションをインポートした証明書に固定化します。ACM はインポートした証明書を自動的に更新しようとはしません。
- パブリック証明書を使用している場合は、アプリケーションを利用可能なすべての Amazon ルート証明書に固定化します。プライベート証明書を使用している場合は、アプリケーションを CA のルート証明書にピンニングします。

ドメイン検証

Amazon 認証機関 (CA) がサイトの証明書を発行する前に、 AWS Certificate Manager (ACM) は、リクエストで指定したすべてのドメインを所有または管理していることを確認する必要があります。E メールまたは DNS のいずれかを使用して検証を実行できます。詳しくは、AWS Certificate Manager DNS 検証 または AWS Certificate Manager E メール検証 を参照してください。

ドメイン名の追加または削除

既存の ACM 証明書からドメイン名を追加または削除することはできません。代わりに、修正済みのドメイン名のリストから新しい証明書をリクエストする必要があります。たとえば、証明書に 5 つ

のドメイン名があり、さらに 4 つを追加する場合は、9 つのドメイン名すべてで新しい証明書をリクエストする必要があります。新しい証明書と同様に、元の証明書に対して事前に検証済みの名前を含むリクエスト内のすべてのドメイン名の所有権を検証する必要があります。

E メール検証を使用する場合、ドメインごとに最大で 8 件の検証 E メールメッセージが送信され、そのうち少なくとも 1 件を 72 時間以内に処理する必要があります。たとえば、5 つのドメイン名で証明書をリクエストすると、最大で 40 件の検証メッセージが送信され、そのうち少なくとも 5 件を 72 時間以内に処理する必要があります。証明書リクエストのドメイン名の数が増えると、E メールを使用してドメインの所有権を検証するために必要な作業も増えます。

代わりに DNS 検証を使用する場合は、検証する FQDN のデータベースに対して新しい DNS レコードを 1 つ書き込む必要があります。ACM はレコードを送信してデータベースを作成し、後でそのデータベースに対してクエリを実行してレコードが追加されたかどうかを判断します。レコードの追加によって、お客様がドメインの所有者または管理者であることがアサートされます。前述の例では、5 つのドメイン名で証明書をリクエストした場合、5 つの DNS レコードを作成する必要があります。可能な場合は、DNS 検証を使用することをお勧めします。

証明書の透明性ログ記録のオプトアウト

> **Important**
>
> 証明書の透明性のログ記録を無効にするために実行するアクションにかかわらず、証明書のバインド先のパブリックエンドポイントまたはプライベートエンドポイントにアクセスできるクライアントまたは個人によって、証明書が引き続きログに記録される場合があります。ただし、証明書には署名付き証明書タイムスタンプ (SCT) は含まれません。発行元の CA のみが、証明書に SCT を埋め込むことができます。

2018 年 4 月 30 日に、Google Chrome は証明書の透明性ログに記録されていないパブリック SSL/TLS 証明書の信頼を停止しました。したがって、2018 年 4 月 24 日から Amazon CA は、すべての新しい証明書と更新を少なくとも 2 つの公開ログに公開するようになりました。証明書がログに記録されると、削除することはできません。(詳細については、証明書の透明性ログ記録 を参照してください)。

ログ記録は、証明書をリクエストするとき、または証明書が更新されたときに自動的に実行されますが、オプトアウトすることもできます。その一般的な理由には、セキュリティとプライバシーに関する懸念があります。たとえば、内部ホストドメイン名のログ記録により、それ以外の場合には公開されない内部ネットワークについての情報が潜在的な攻撃者に提供されます。さらに、ログ記録により、新規または未リリース製品やウェブサイトの名前が漏洩する可能性があります。

証明書をリクエストするときに、透明性ログを出力しないようにするには、request-certificate AWS CLI コマンド、または RequestCertificate API オペレーションの options パラメータを使用します。証明書が 2018 年 4 月 24 日より前に発行され、更新中にログに記録されないようにする場合は、 コマンド、または UpdateCertificateOptions API オペレーションを使用してオプトアウトすることができます。

制限事項

- コンソールを使用して、透明性ログを有効または無効にすることはできません。
- 証明書が更新期間に入った後、通常はパブリック証明書の証明書の有効期限が切れる 45 日前にログ記録ステータスを変更することはできません。ステータスの変更が失敗しても、エラーメッセージは生成されません。

証明書がログに記録されると、ログから削除することはできません。その時点でオプトアウトしても効果はありません。証明書をリクエストしたときにログ記録を停止して、後でオプトインするように選択すると、証明書は更新されるまでログに記録されません。証明書をすぐにログに記録する場合は、新しい証明書を発行することをお勧めします。

次の例では、新しい証明書をリクエストするときに request-certificate コマンドを使用して証明書の透明性を無効にする方法を示しています。

```
aws acm request-certificate \
--domain-name www.example.com \
--validation-method DNS \
--options CertificateTransparencyLoggingPreference=DISABLED \
```

上記のコマンドは、新しい証明書の ARN を出力します。

```
{
    "CertificateArn": "arn:aws:acm:region:account:certificate/certificate_ID"
}
```

証明書がすでにある場合、証明書が更新されたときに記録されないようにするには、update-certificate-options コマンドを使用します。このコマンドは値を返しません。

```
aws acm update-certificate-options \
--certificate-arn arn:aws:acm:region:account:\
certificate/certificate_ID \
--options CertificateTransparencyLoggingPreference=DISABLED
```

オンにする AWS CloudTrail

ACM の使用を開始する前に、CloudTrail のログ記録を有効にします。CloudTrail を使用すると、AWS マネジメントコンソール、 AWS SDKs、、および上位レベルの Amazon Web Services を介して行われた AWS API コールなど、アカウントの API コールの履歴を取得して AWS Command Line Interface、 AWS デプロイをモニタリングできます。また、履歴では、ACM API を呼び出したユーザーとアカウント、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。API を使用して CloudTrail をアプリケーションに統合したり、組織用の証跡作成を自動化したり、証跡の状態を確認したり、CloudTrail のログ記録のオン/オフを管理者が切り替える方法を制御したりすることもできます。詳細については、「証跡の作成」を参照してください。ACM アクションの証跡の例については、「での CloudTrail の使用 AWS Certificate Manager」を参照してください。

モニタリングとログ記録 AWS Certificate Manager

モニタリングは、 および AWS Certificate Manager AWS ソリューションの信頼性、可用性、パフォーマンスを維持する上で重要な部分です。マルチポイント障害が発生した場合は、その障害をより簡単にデバッグできるように、 AWS ソリューションのすべての部分からモニタリングデータを収集する必要があります。

以下のトピックでは、ACM で使用できる AWS クラウドモニタリングツールについて説明します。

トピック

- Amazon EventBridge の使用
- での CloudTrail の使用 AWS Certificate Manager
- サポートされている CloudWatch メトリクス

Amazon EventBridge の使用

Amazon EventBridge (旧 CloudWatch Events) を使用して AWS サービスを自動化し、アプリケーションの可用性の問題やリソースの変更などのシステムイベントに自動的に対応できます。ACM を含む AWS のサービスからのイベントは、ほぼリアルタイムで Amazon EventBridge に配信されます。イベントを使用して、 AWS Lambda 関数、 AWS Batch ジョブ、Amazon SNS トピックなどのターゲットをトリガーできます。詳細については、「Amazon EventBridge とは?」を参照してください。

トピック

- ACM の Amazon EventBridge サポート
- ACM での Amazon EventBridge を使用してアクションを開始

ACM の Amazon EventBridge サポート

このトピックでは、Amazon EventBridge でサポートされる ACM 関連のイベントを一覧表示して説明します。

ACM 証明書の有効期限が近づいているイベント

ACM は、すべてのアクティブな証明書 (パブリック、プライベート、インポート) の有効期限イベントを、プライベート/インポートされた証明書の有効期限の 45 日前から、パブリック証明書の有効

期限の 30 日前から毎日送信します。このタイミングは、ACM API の PutAccountConfiguration アクションを使用して変更できます。

ACM によって生成された証明書は自動的に更新されますが、停止を回避するために、有効期限切れ前に、インポートされた証明書が再発行され、再インポートされる必要があります。詳細については、「証明書の再インポート」を参照してください。有効期限イベントを使用して、証明書を ACM に再インポートする自動化を設定できます。を使用したオートメーションの例については AWS Lambda、「」を参照してくださいACM での Amazon EventBridge を使用してアクションを開始。

ACM 証明書の有効期限が近づいている イベントには次の構造があります。

```
{
  "version": "0",
  "id": "id",
  "detail-type": "ACM Certificate Approaching Expiration",
  "source": "aws.acm",
  "account": "account",
  "time": "2020-09-30T06:51:08Z",
  "region": "region",
  "resources": [
    "arn:aws:acm:region:account:certificate/certificate_ID"
  ],
  "detail": {
    "DaysToExpiry": 31,
    "CommonName": "example.com"
  }
}
```

ACM 証明書期限切れイベント

> **Note**
>
> 証明書期限切れイベントは、インポートした証明書には使用できません。

お客様はこのイベントを聞いて、アカウントの ACM 発行のパブリック証明書またはプライベート証明書の有効期限が切れた場合にアラートを受け取ることができます。

ACM 証明書期限切れ イベントには次の構造があります。

```
{
    "version": "0",
    "id": "id",
    "detail-type": "ACM Certificate Expired",
    "source": "aws.acm",
    "account": "account",
    "time": "2019-12-22T18:43:48Z",
    "region": "region",
    "resources": [
        "arn:aws:acm:region:account:certificate/certificate_ID"
     ],
     "detail": {
        "CertificateType" : "AMAZON_ISSUED" | "PRIVATE",
        "CommonName": "example.com",
        "DomainValidationMethod" : "EMAIL" | "DNS",
        "CertificateCreatedDate" : "2018-12-22T18:43:48Z",
        "CertificateExpirationDate" : "2019-12-22T18:43:48Z",
        "InUse" : TRUE | FALSE,
        "Exported" : TRUE | FALSE
    }
 }
```

ACM 証明書利用可能イベント

お客様はこのイベントを聞いて、マネージドパブリック証明書またはプライベート証明書が使用可能になったときに通知を受け取ることができます。イベントは、発行、更新、インポート時に公開されます。プライベート証明書については、いったん利用可能になった後も、ホストにデプロイするにはお客様のアクションが必要です。

ACM 証明書利用可能 イベントには次の構造があります。

```
{
    "version": "0",
    "id": "id",
    "detail-type": "ACM Certificate Available",
    "source": "aws.acm",
    "account": "account",
    "time": "2019-12-22T18:43:48Z",
    "region": "region",
    "resources": [
        "arn:aws:acm:region:account:certificate/certificate_ID"
    ],
```

```
    "detail": {
       “Action” : "ISSUANCE" | "RENEWAL" | "IMPORT" | "REIMPORT",
       "CertificateType" : "AMAZON_ISSUED" | "PRIVATE" | "IMPORTED",
       "CommonName": "example.com",
       "DomainValidationMethod" : "EMAIL" | "DNS",
       "CertificateCreatedDate" : "2019-12-22T18:43:48Z",
       "CertificateExpirationDate" : "2019-12-22T18:43:48Z",
       "DaysToExpiry" : 198,
       "InUse" : TRUE | FALSE,
       "Exported" : TRUE | FALSE
     }
}
```

ACM 証明書更新アクション必要イベント

> **Note**
>
> 証明書更新アクション必要イベントは、インポートした証明書には使用できません。

お客様はこのイベントを聞いて、証明書を更新する前にお客様がアクションを起こさなければならない場合にアラートを受け取ることができます。たとえば、ACM が証明書を更新できない CAA レコードを顧客が追加した場合、ACM は、プライベート証明書の有効期限の 45 日前とパブリック証明書の有効期限の 30 日前に自動的に更新が失敗すると、このイベントを発行します。カスタマーアクションが実行されない場合、ACM は 30 日 (プライベートのみ) 、15 日、3 日、1 日、またはカスタマーアクションが実行されるか、証明書の有効期限が切れるか、証明書が更新対象外になるまで、さらに更新を試みます。これらの更新を試みるたびに、イベントが公開されます。

ACM 証明書更新アクション必要 イベントには次の構造があります。

```
{
   "version": "0",
   "id": "id",
   "detail-type": "ACM Certificate Renewal Action Required",
   "source": "aws.acm",
   "account": "account",
   "time": "2019-12-22T18:43:48Z",
   "region": "region",
   "resources": [
       "arn:aws:acm:region:account:certificate/certificate_ID"
    ],
```

```
    "detail": {
       "CertificateType" : "AMAZON_ISSUED" | "PRIVATE",
       "CommonName": "example.com",
       "DomainValidationMethod" : "EMAIL" | "DNS",
       "RenewalStatusReason" : "CAA_ERROR" | "PENDING_DOMAIN_VALIDATION" |
 "NO_AVAILABLE_CONTACTS" | "ADDITIONAL_VERIFICATION_REQUIRED" | "DOMAIN_NOT_ALLOWED"
 | "INVALID_PUBLIC_DOMAIN" | "DOMAIN_VALIDATION_DENIED" | "PCA_LIMIT_EXCEEDED"
 | "PCA_INVALID_ARN" | "PCA_INVALID_STATE" | "PCA_REQUEST_FAILED" |
 "PCA_NAME_CONSTRAINTS_VALIDATION" | "PCA_RESOURCE_NOT_FOUND" | "PCA_INVALID_ARGS" |
 "PCA_INVALID_DURATION" | "PCA_ACCESS_DENIED" | "SLR_NOT_FOUND" | "OTHER",
       "DaysToExpiry": 30,
       "CertificateExpirationDate" : "2019-12-22T18:43:48Z",
       "InUse" : TRUE | FALSE,
       "Exported" : TRUE | FALSE
   }
}
```

ACM 証明書の失効イベント

お客様はこのイベントを聞いて、アカウントの ACM 発行のパブリック証明書またはプライベート証明書が失効した場合にアラートを受け取ることができます。

> **Note**
>
> エクスポートされた証明書のみ失効できます。インポートされた証明書を revoke-certificate で取り消すことはできません。

ACM 証明書失効 イベントには次の構造があります。

```
{
  "version": "0",
  "id": "id",
  "detail-type": "ACM Certificate Revoked",
  "source": "aws.acm",
  "account": "account",
  "time": "2019-12-22T18:43:48Z",
  "region": "region",
 "resources": [
       "arn:aws:acm:region:account:certificate/certificate_ID"
    ],
  "detail": {
```

```
    "CertificateType" : "AMAZON_ISSUED" | "PRIVATE",
    "CommonName": "example.com",
    "CertificateExpirationDate" : "2019-12-22T18:43:48Z",
    "Exportable": TRUE | FALSE
  }
}
```

AWS ヘルスイベント

AWS ヘルスイベントは、更新の対象となる ACM 証明書に対して生成されます。更新資格の詳細については、「でのマネージド証明書の更新 AWS Certificate Manager」を参照してください。

ヘルスイベントは、次の 2 つのシナリオで生成されます。

- パブリック証明書またはプライベート証明書を正常に更新した場合。
- お客様が更新を行うためのアクションを実行する必要があるとき。これは、E メールメッセージ内のリンクをクリックするか (E メールで検証された証明書の場合)、またはエラーの解決を意味することがあります。各イベントには、次のイベントタイプコードの 1 つが含まれています。コードは、フィルタリングに使用できる変数として公開されます。
  - AWS_ACM_RENEWAL_STATE_CHANGE(証明書が更新された、有効期限が切れている、または有効期限が切れる予定)
  - CAA_CHECK_FAILURE(CAAチェックに失敗しました)
  - AWS_ACM_RENEWAL_FAILURE(プライベート CA によって署名された証明書について)

正常性イベントは、次の構造を持っています。この例では、AWS_ACM_RENEWAL_STATE_CHANGE イベントが生成されました。

```
{
   "source":[
      "aws.health"
   ],
   "detail-type":[
      "AWS Health Event"
   ],
   "detail":{
      "service":[
         "ACM"
      ],
      "eventTypeCategory":[
```

```
            "scheduledChange"
         ],
         "eventTypeCode":[
            "AWS_ACM_RENEWAL_STATE_CHANGE"
         ]
      }
}
```

ACM での Amazon EventBridge を使用してアクションを開始

これらのイベントに基づいて Amazon EventBridge ルールを作成し、Amazon EventBridge コンソールを使用して、イベントが検出されたときに実行されるアクションを設定できます。このセクションでは、Amazon EventBridge ルールを設定するためのサンプル手順と、その結果生じるアクションについて説明します。

トピック

- Amazon SNS でのイベントへの応答
- Lambda 関数を使用したイベントへの応答

Amazon SNS でのイベントへの応答

このセクションでは、ACM が正常性イベントを生成するときはいつでもテキスト通知を送信するように Amazon SNS を設定する方法について説明します。

レスポンスを設定するには、次の手順を実行します。

Amazon EventBridge ルールを作成してアクションをトリガーするには

1. Amazon EventBridge ルールを作成します。詳細については、「イベントに反応する Amazon EventBridge ルールの作成」を参照してください。

   a. Amazon EventBridge コンソール (https://console.aws.amazon.com/cloudwatch/) で、[イベント] > [ルール] ページに移動し、[ルールの作成] を選択します。

   b. [ルールの作成] ページで、[イベントパターン] を選択します。

   c. [サービス名] では、メニューから [Health] を選択します。

   d. [イベントタイプ] では、[特定の正常性イベント] を選択します。

   e. [特定のサービス] を選択し、メニューから [ACM] を選択します。

   f. [特定のイベントタイプカテゴリ] を選択し、[accountNotification] を選択します。

g. [任意のイベントタイプコード] を選択します。

h. [任意のリソース] を選択します。

i. [イベントパターンのプレビュー] エディタで、イベントによって放出される JSON パターンを貼り付けます。この例では、AWS ヘルスイベント セクションからパターンを使用します。

```
{
   "source":[
      "aws.health"
   ],
   "detail-type":[
      "AWS Health Event"
   ],
   "detail":{
      "service":[
         "ACM"
      ],
      "eventTypeCategory":[
         "scheduledChange"
      ],
      "eventTypeCode":[
         "AWS_ACM_RENEWAL_STATE_CHANGE"
      ]
   }
}
```

2. アクションを設定します。

ターゲットセクションでは、Amazon Simple Notification Service (SNS) など、イベントをすぐに消費できる多くのサービスから選択するか、カスタマイズした実行可能コードにイベントを渡すために、Lambda 関数を選択することができます。 AWS Lambda 実装の例については、「Lambda 関数を使用したイベントへの応答」を参照してください。

Lambda 関数を使用したイベントへの応答

この手順では、 AWS Lambda を使用して Amazon EventBridge をリッスンし、Amazon Simple Notification Service (SNS) で通知を作成し、検出結果を に発行して AWS Security Hub CSPM、管理者とセキュリティチームに可視性を提供する方法を示します。

Lambda 関数と IAM ロールを設定するには

1. まず AWS Identity and Access Management (IAM) ロールを設定し、Lambda 関数に必要なアクセス許可を定義します。このセキュリティのベストプラクティスにより、関数を呼び出す権限を持つユーザーを指定したり、そのユーザーに付与される権限を制限したりする柔軟性が得られます。ほとんどの AWS オペレーションをユーザーアカウントで直接実行することはお勧めしません。特に管理者アカウントで実行することはお勧めしません。

   IAM コンソール (https://console.aws.amazon.com/iam/) を開きます。

2. JSON ポリシーエディタを使用して、以下のテンプレートで定義したポリシーを作成します。独自のリージョンと AWS アカウントの詳細を入力します。詳細については、「JSON タブでのポリシーの作成」 を参照してください。

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "LambdaCertificateExpiryPolicy1",
            "Effect": "Allow",
            "Action": "logs:CreateLogGroup",
            "Resource": "arn:aws:logs:us-east-1:123456789012:*"
        },
        {
            "Sid": "LambdaCertificateExpiryPolicy2",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:123456789012:log-group:/aws/lambda/
handle-expiring-certificates:*"
            ]
        },
        {
            "Sid": "LambdaCertificateExpiryPolicy3",
            "Effect": "Allow",
            "Action": [
                "acm:DescribeCertificate",
```

```
                "acm:GetCertificate",
                "acm:ListCertificates",
                "acm:ListTagsForCertificate"
            ],
            "Resource": "*"
        },
        {
            "Sid": "LambdaCertificateExpiryPolicy4",
            "Effect": "Allow",
            "Action": "SNS:Publish",
            "Resource": "*"
        },
        {
            "Sid": "LambdaCertificateExpiryPolicy5",
            "Effect": "Allow",
            "Action": [
                "SecurityHub:BatchImportFindings",
                "SecurityHub:BatchUpdateFindings",
                "SecurityHub:DescribeHub"
            ],
            "Resource": "*"
        },
        {
            "Sid": "LambdaCertificateExpiryPolicy6",
            "Effect": "Allow",
            "Action": "cloudwatch:ListMetrics",
            "Resource": "*"
        }
    ]
}
```

3. IAM ロールを作成して、それに新しいポリシーをアタッチします。IAM ロールの作成とポリシーのアタッチについては、「 AWS サービスのロールの作成 (コンソール)」を参照してください。
4. https://console.aws.amazon.com/lambda/ で AWS Lambda コンソールを開きます。
5. Lambda 関数を作成する 詳細については、「コンソールで Lambda 関数を作成する」を参照してください。以下のステップを実行します。

   a. [Create function] ページで、[Author from scratch] を選択します。

   b. 「ハンドル有効期限証明書」などの名前を [関数名] フィールドで指定します。

   c. [Runtime (ランタイム)] リストで、[Python 3.8] を選択します。

d. [デフォルト実行ロールの変更] を拡張し、[既存のロールを使用する] を選択します。

e. [既存のロール] リストで、以前作成したロールを選択します。

f. [関数の作成] を選択してください。

g. [関数コード] に次のコードを挿入します。

```
# Copyright 2021 Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
#
# Permission is hereby granted, free of charge, to any person obtaining a copy
 of this
# software and associated documentation files (the "Software"), to deal in the
 Software
# without restriction, including without limitation the rights to use, copy,
 modify,
# merge, publish, distribute, sublicense, and/or sell copies of the Software,
 and to
# permit persons to whom the Software is furnished to do so.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 IMPLIED,
# INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
# PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
 COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
 ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH
 THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

import json
import boto3
import os
from datetime import datetime, timedelta, timezone
# ----------------------------------------
# setup global data
# ----------------------------------------
utc = timezone.utc
# make today timezone aware
today = datetime.now().replace(tzinfo=utc)
# set up time window for alert - default to 45 if its missing
if os.environ.get('EXPIRY_DAYS') is None:
    expiry_days = 45
```

```
else:
    expiry_days = int(os.environ['EXPIRY_DAYS'])
expiry_window = today + timedelta(days = expiry_days)
def lambda_handler(event, context):
    # if this is coming from the ACM event, its for a single certificate
    if (event['detail-type'] == "ACM Certificate Approaching Expiration"):
        response = handle_single_cert(event, context.invoked_function_arn)
    return {
        'statusCode': 200,
        'body': response
    }
def handle_single_cert(event, context_arn):
    cert_client = boto3.client('acm')
    cert_details =
 cert_client.describe_certificate(CertificateArn=event['resources'][0])
    result = 'The following certificate is expiring within ' + str(expiry_days)
 + ' days: ' + cert_details['Certificate']['DomainName']
    # check the expiry window before logging to Security Hub and sending an SNS
    if cert_details['Certificate']['NotAfter'] < expiry_window:
        # This call is the text going into the SNS notification
        result = result + ' (' + cert_details['Certificate']['CertificateArn']
 + ') '
        # this call is publishing to SH
        result = result + ' - ' + log_finding_to_sh(event, cert_details,
 context_arn)
        # if there's an SNS topic, publish a notification to it
        if os.environ.get('SNS_TOPIC_ARN') is None:
            response = result
        else:
            sns_client = boto3.client('sns')
            response = sns_client.publish(TopicArn=os.environ['SNS_TOPIC_ARN'],
 Message=result, Subject='Certificate Expiration Notification')
    return result
def log_finding_to_sh(event, cert_details, context_arn):
    # setup for security hub
    sh_region = get_sh_region(event['region'])
    sh_hub_arn = "arn:aws:securityhub:{0}:{1}:hub/default".format(sh_region,
 event['account'])
    sh_product_arn = "arn:aws:securityhub:{0}:{1}:product/{1}/
default".format(sh_region, event['account'])
    # check if security hub is enabled, and if the hub arn exists
    sh_client = boto3.client('securityhub', region_name = sh_region)
    try:
        sh_enabled = sh_client.describe_hub(HubArn = sh_hub_arn)
```

```
   # the previous command throws an error indicating the hub doesn't exist or
lambda doesn't have rights to it so we'll stop attempting to use it
   except Exception as error:
       sh_enabled = None
       print ('Default Security Hub product doesn\'t exist')
       response = 'Security Hub disabled'
   # This is used to generate the URL to the cert in the Security Hub Findings
to link directly to it
   cert_id = right(cert_details['Certificate']['CertificateArn'], 36)
   if sh_enabled:
       # set up a new findings list
       new_findings = []
           # add expiring certificate to the new findings list
       new_findings.append({
           "SchemaVersion": "2018-10-08",
           "Id": cert_id,
           "ProductArn": sh_product_arn,
           "GeneratorId": context_arn,
           "AwsAccountId": event['account'],
           "Types": [
               "Software and Configuration Checks/AWS Config Analysis"
           ],
           "CreatedAt": event['time'],
           "UpdatedAt": event['time'],
           "Severity": {
               "Original": '89.0',
               "Label": 'HIGH'
           },
           "Title": 'Certificate expiration',
           "Description": 'cert expiry',
           'Remediation': {
               'Recommendation': {
                   'Text': 'A new certificate for ' +
cert_details['Certificate']['DomainName'] + ' should be imported to replace
the existing imported certificate before expiration',
                   'Url': "https://console.aws.amazon.com/acm/home?region=" +
event['region'] + "#/?id=" + cert_id
               }
           },
           'Resources': [
               {
                   'Id': event['id'],
                   'Type': 'ACM Certificate',
                   'Partition': 'aws',
```

```
                    'Region': event['region']
                }
            ],
            'Compliance': {'Status': 'WARNING'}
        })
        # push any new findings to security hub
        if new_findings:
            try:
                response =
 sh_client.batch_import_findings(Findings=new_findings)
                if response['FailedCount'] > 0:
                    print("Failed to import {}
 findings".format(response['FailedCount']))
            except Exception as error:
                print("Error: ", error)
                raise
    return json.dumps(response)
# function to setup the sh region
def get_sh_region(event_region):
    # security hub findings may need to go to a different region so set that
 here
    if os.environ.get('SECURITY_HUB_REGION') is None:
        sh_region_local = event_region
    else:
        sh_region_local = os.environ['SECURITY_HUB_REGION']
    return sh_region_local
# quick function to trim off right side of a string
def right(value, count):
    # To get right part of string, use negative first index in slice.
    return value[-count:]
```

h. [環境変数] で、[編集] を選択し、オプションで次の変数を追加します。

- (オプション) EXPIRY_DAYS

  証明書の有効期限切れの通知が送信されるまでのリードタイムを日数で指定します。この関数のデフォルトは 45 日ですが、カスタム値を指定できます。

- (オプション) SNS_TOPIC_ARN

  Amazon SNS の ARN を指定します。arn:aws:sns:*<region>*:*<account-number>*:*<topic-name>* の形式で完全な ARN を指定します。

- (オプション) SECURITY_HUB_REGION

別のリージョン AWS Security Hub CSPM の を指定します。これを指定しない場合、実行中の Lambda 関数のリージョンが使用されます。関数が複数のリージョンで実行されている場合、すべての証明書メッセージを 1 つのリージョンの Security Hub CSPM に送信することをお勧めします。

i. [基本設定] で、 [タイムアウト] を 30 秒に設定します。

j. ページの上部で、[デプロイ] を選択します。

このソリューションの使用を開始するには、以下の手順のタスクを実行します。

有効期限切れの E メール通知を自動化するには

この例では、Amazon EventBridge を通じてイベントが発生した時点で、有効期限が切れる証明書ごとに 1 通の E メールが送信されます。デフォルトでは、ACM は有効期限切れより 45 日以下の証明書に対してイベントを毎日発生させます。(この期間は、ACM API の PutAccountConfiguration オペレーションを使用してカスタマイズできます。) これらの各イベントは、次の自動アクションのカスケードをトリガーします。

```
ACM raises Amazon EventBridge event #
>>>>>>> events

          Event matches Amazon EventBridge rule #

                    Rule calls Lambda function #

                              Function sends SNS email and logs a Finding in Security
 Hub CSPM
```

1. Lambda 関数を作成し、アクセス許可を設定します。(すでに完了しています。「Lambda 関数と IAM ロールを設定するには」を参照してください)。
2. Lambda 関数の標準 SNS トピックを作成して、通知の送信に使用します。詳細については、「Amazon SNS トピックの作成」を参照してください。
3. 新しい SNS トピックに利害関係者を登録します。詳細については、「Amazon SNS トピックにサブスクライブする」を参照してください。
4. Lambda 関数をトリガーする Amazon EventBridge ルールの作成 詳細については、「イベントに反応する Amazon EventBridge ルールの作成」を参照してください。

Amazon EventBridge コンソール (https://console.aws.amazon.com/cloudwatch/) で、[イベント] > [ルール] ページに移動し、[ルールの作成] を選択します。[サービス名]、[イベントタイプ]、および [Lambda 関数] を指定します。[イベントパターンのプレビュー] エディタで、次のコードを貼り付けます。

```
{
  "source": [
    "aws.acm"
  ],
  "detail-type": [
    "ACM Certificate Approaching Expiration"
  ]
}
```

Lambda が受け取るようなイベントが [サンプルイベントを表示] の下に表示されます。

```
{
  "version": "0",
  "id": "9c95e8e4-96a4-ef3f-b739-b6aa5b193afb",
  "detail-type": "ACM Certificate Approaching Expiration",
  "source": "aws.acm",
  "account": "123456789012",
  "time": "2020-09-30T06:51:08Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:acm:us-east-1:123456789012:certificate/61f50cd4-45b9-4259-b049-
d0a53682fa4b"
  ],
  "detail": {
    "DaysToExpiry": 31,
    "CommonName": "My Awesome Service"
  }
}
```

次をクリーンアップするには：

設定例や設定が不要になったら、セキュリティ上の問題や将来の予想外の料金を避けるために、すべてのトレースを削除することがベストプラクティスです。

- IAM ポリシーおよびロール

- Lambda function
- CloudWatch Events ルール
- Lambda に関連付けられた CloudWatch Logs
- SNS トピック

での CloudTrail の使用 AWS Certificate Manager

AWS Certificate Manager は、ACM のユーザー AWS CloudTrail、ロール、または のサービスによって実行されたアクションを記録する AWS サービスである と統合されています。CloudTrail は、デフォルトで AWS アカウントによって有効にされます。CloudTrail は、ACM コンソールからの呼び出しや ACM API オペレーションへのコード呼び出しを含む、ACM の API コールをイベントとしてキャプチャします。[追跡] を設定する場合は、ACM のイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的な配信を有効にすることができます 追跡を設定しない場合でも、CloudTrail コンソールの [Event history (イベント履歴)] で最新のイベントを表示できます。

CloudTrail で収集された情報を使用して、ACM に対するリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト日時などの詳細を確認できます。詳細については、CloudTrail イベント履歴でのイベントの表示を参照してください。ACM でサポートされているイベントアクティビティが発生すると、そのアクティビティはイベント履歴の他の AWS サービスイベントとともに CloudTrail イベントに記録されます。最近のイベントは、 AWS アカウントで表示、検索、ダウンロードできます。

さらに、CloudTrail ログで収集されたイベントデータをさらに分析して処理するように他の AWS サービスを設定できます。

CloudTrail の詳細については、次のドキュメントを参照してください。

- AWS CloudTrail ユーザーガイド。
- 証跡の作成のための概要
- CloudTrail がサポートするサービスと統合
- CloudTrail 用 Amazon SNS 通知の構成
- 複数のリージョンから CloudTrail ログファイルを受け取る および 複数のアカウントから CloudTrail ログファイルを受け取る

トピック

- CloudTrail ロギングでサポートされている ACM API アクション

- 統合サービスに対する API 呼び出しのログギング

CloudTrail ロギングでサポートされている ACM API アクション

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

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。

- リクエストが AWS アカウントのルートユーザー または AWS Identity and Access Management (IAM) ユーザー認証情報を使用し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 サービスによって行われたかどうか

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

以下のセクションでは、サポートされている API オペレーションのログの例を示します。

- 証明書へのタグの追加 (AddTagsToCertificate)
- 証明書の削除 (DeleteCertificate)
- 証明書についての説明 (DescribeCertificate)
- 証明書のエクスポート (ExportCertificate)
- 証明書のインポート (ImportCertificate)
- 証明書の一覧 (ListCertificates)
- 証明書のタグを一覧表示する (ListTagsForCertificate)
- 証明書からタグを削除する (RemoveTagsFromCertificate)
- 証明書のリクエスト (RequestCertificate)
- 検証 E メールの再送信 (ResendValidationEmail)
- 証明書の取得 (GetCertificate)

証明書へのタグの追加 (AddTagsToCertificate)

次の CloudTrail の例は、AddTagsToCertificate API への呼び出しの結果を示しています。

```
{

   "Records":[
      {
         "eventVersion":"1.04",
         "userIdentity":{
            "type":"IAMUser",
            "principalId":"AIDACKCEVSQ6C2EXAMPLE",
            "arn":"arn:aws:iam::123456789012:user/Alice",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "userName":"Alice"
         },
         "eventTime":"2016-04-06T13:53:53Z",
         "eventSource":"acm.amazonaws.com",
         "eventName":"AddTagsToCertificate",
         "awsRegion":"us-east-1",
         "sourceIPAddress":"192.0.2.0",
         "userAgent":"aws-cli/1.10.16",
         "requestParameters":{
            "tags":[
               {
                  "value":"Alice",
                  "key":"Admin"
               }
            ],
            "certificateArn":"arn:aws:acm:us-east-1:123456789012:certificate/
fedcba98-7654-3210-fedc-ba9876543210"
         },
         "responseElements":null,
         "requestID":"fedcba98-7654-3210-fedc-ba9876543210",
         "eventID":"fedcba98-7654-3210-fedc-ba9876543210",
         "eventType":"AwsApiCall",
         "recipientAccountId":"123456789012"
      }
   ]
}
```


証明書の削除 (DeleteCertificate)

次の CloudTrail の例は、DeleteCertificate API への呼び出しの結果を示しています。


```
{
```

```
   "Records":[
      {
         "eventVersion":"1.04",
         "userIdentity":{
            "type":"IAMUser",
            "principalId":"AIDACKCEVSQ6C2EXAMPLE",
            "arn":"arn:aws:iam::123456789012:user/Alice",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "userName":"Alice"
         },
         "eventTime":"2016-03-18T00:00:26Z",
         "eventSource":"acm.amazonaws.com",
         "eventName":"DeleteCertificate",
         "awsRegion":"us-east-1",
         "sourceIPAddress":"192.0.2.0",
         "userAgent":"aws-cli/1.9.15",
         "requestParameters":{
            "certificateArn":"arn:aws:acm:us-east-1:123456789012:certificate/
fedcba98-7654-3210-fedc-ba9876543210"
         },
         "responseElements":null,
         "requestID":"01234567-89ab-cdef-0123-456789abcdef",
         "eventID":"01234567-89ab-cdef-0123-456789abcdef",
         "eventType":"AwsApiCall",
         "recipientAccountId":"123456789012"
      }
   ]
}
```


証明書についての説明 (DescribeCertificate)

次の CloudTrail の例は、DescribeCertificate API への呼び出しの結果を示しています。

> **Note**
>
> DescribeCertificate オペレーションの CloudTrail ログは、指定する ACM 証明書についての情報を表示しません。証明書に関する情報は、 コンソール、 AWS Command Line Interface、または DescribeCertificate API を使用して表示できます。

```
{
   "Records":[
      {
         "eventVersion":"1.04",
         "userIdentity":{
            "type":"IAMUser",
            "principalId":"AIDACKCEVSQ6C2EXAMPLE",
            "arn":"arn:aws:iam::123456789012:user/Alice",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "userName":"Alice"
         },
         "eventTime":"2016-03-18T00:00:42Z",
         "eventSource":"acm.amazonaws.com",
         "eventName":"DescribeCertificate",
         "awsRegion":"us-east-1",
         "sourceIPAddress":"192.0.2.0",
         "userAgent":"aws-cli/1.9.15",
         "requestParameters":{
            "certificateArn":"arn:aws:acm:us-east-1:123456789012:certificate/
fedcba98-7654-3210-fedc-ba9876543210"
         },
         "responseElements":null,
         "requestID":"fedcba98-7654-3210-fedc-ba9876543210",
         "eventID":"fedcba98-7654-3210-fedc-ba9876543210",
         "eventType":"AwsApiCall",
         "recipientAccountId":"123456789012"
      }
   ]
}
```

証明書のエクスポート (ExportCertificate)

次の CloudTrail の例は、ExportCertificate API への呼び出しの結果を示しています。

```
{
   "Records":[
      {
         "version":"0",
         "id":"01234567-89ab-cdef-0123-456789abcdef",
         "detail-type":"AWS API Call via CloudTrail",
         "source":"aws.acm",
         "account":"123456789012",
```

```
         "time":"2018-05-24T15:28:11Z",
         "region":"us-east-1",
         "resources":[

         ],
         "detail":{
            "eventVersion":"1.04",
            "userIdentity":{
               "type":"Root",
               "principalId":"123456789012",
               "arn":"arn:aws:iam::123456789012:user/Alice",
               "accountId":"123456789012",
               "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
               "userName":"Alice"
            },
            "eventTime":"2018-05-24T15:28:11Z",
            "eventSource":"acm.amazonaws.com",
            "eventName":"ExportCertificate",
            "awsRegion":"us-east-1",
            "sourceIPAddress":"192.0.2.0",
            "userAgent":"aws-cli/1.15.4 Python/2.7.9 Windows/8 botocore/1.10.4",
            "requestParameters":{
              "certificateArn":"arn:aws:acm:us-
east-1:123456789012:certificate/12345678-1234-1234-1234-123456789012",
               "passphrase": "HIDDEN_DUE_TO_SECURITY_REASONS"
            },
            "responseElements":{
                "certificateChain":
                "-----BEGIN CERTIFICATE-----
                base64 certificate
                -----END CERTIFICATE-----
                -----BEGIN CERTIFICATE-----
                base64 certificate
                -----END CERTIFICATE-----",
                "privateKey":"**********",
                "certificate":
                "-----BEGIN CERTIFICATE-----
                base64 certificate
                -----END CERTIFICATE-----",
                "privateKey": "HIDDEN_DUE_TO_SECURITY_REASONS"
            },
            "requestID":"01234567-89ab-cdef-0123-456789abcdef",
            "eventID":"fedcba98-7654-3210-fedc-ba9876543210",
            "readOnly": false,
```

```
            "eventType":"AwsApiCall"
                "managementEvent": true,
                "recipientAccountId": "123456789012",
                "eventCategory": "Management",
                "tlsDetails": {
                     "tlsVersion": "TLSv1.3",
                     "cipherSuite": "TLS_AES_128_GCM_SHA256",
                     "clientProvidedHostHeader": "acm.us-east-1.amazonaws.com"
                 },
                 "sessionCredentialFromConsole": "true"
}
```

証明書のインポート (ImportCertificate)

ACM の ImportCertificate API オペレーションへの呼び出しを記録する CloudTrail のログエントリの例を次に示します。

```
{
   "eventVersion":"1.04",
   "userIdentity":{
      "type":"IAMUser",
      "principalId":"AIDACKCEVSQ6C2EXAMPLE",
      "arn":"arn:aws:iam::111122223333:user/Alice",
      "accountId":"111122223333",
      "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
      "userName":"Alice"
   },
   "eventTime":"2016-10-04T16:01:30Z",
   "eventSource":"acm.amazonaws.com",
   "eventName":"ImportCertificate",
   "awsRegion":"ap-southeast-2",
   "sourceIPAddress":"54.240.193.129",
   "userAgent":"Coral/Netty",
   "requestParameters":{
      "privateKey":{
         "hb":[
            "byte",
            "byte",
            "byte",
            "..."
         ],
         "offset":0,
```

```
         "isReadOnly":false,
         "bigEndian":true,
         "nativeByteOrder":false,
         "mark":-1,
         "position":0,
         "limit":1674,
         "capacity":1674,
         "address":0
      },
      "certificateChain":{
         "hb":[
            "byte",
            "byte",
            "byte",
            "..."
         ],
         "offset":0,
         "isReadOnly":false,
         "bigEndian":true,
         "nativeByteOrder":false,
         "mark":-1,
         "position":0,
         "limit":2105,
         "capacity":2105,
         "address":0
      },
      "certificate":{
         "hb":[
            "byte",
            "byte",
            "byte",
            "..."
         ],
         "offset":0,
         "isReadOnly":false,
         "bigEndian":true,
         "nativeByteOrder":false,
         "mark":-1,
         "position":0,
         "limit":2503,
         "capacity":2503,
         "address":0
      }
   },
```

```
   "responseElements":{
      "certificateArn":"arn:aws:acm:ap-
southeast-2:111122223333:certificate/01234567-89ab-cdef-0123-456789abcdef"
   },
   "requestID":"01234567-89ab-cdef-0123-456789abcdef",
   "eventID":"01234567-89ab-cdef-0123-456789abcdef",
   "eventType":"AwsApiCall",
   "recipientAccountId":"111122223333"
}
```

証明書の一覧 (ListCertificates)

次の CloudTrail の例は、ListCertificates API への呼び出しの結果を示しています。

> **Note**
>
> ListCertificates オペレーションの CloudTrail ログに ACM 証明書は表示されません。証明書リストは、コンソール、 AWS Command Line Interface、または ListCertificates API を使用して表示できます。

```
{
   "Records":[
      {
         "eventVersion":"1.04",
         "userIdentity":{
            "type":"IAMUser",
            "principalId":"AIDACKCEVSQ6C2EXAMPLE",
            "arn":"arn:aws:iam::123456789012:user/Alice",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "userName":"Alice"
         },
         "eventTime":"2016-03-18T00:00:43Z",
         "eventSource":"acm.amazonaws.com",
         "eventName":"ListCertificates",
         "awsRegion":"us-east-1",
         "sourceIPAddress":"192.0.2.0",
         "userAgent":"aws-cli/1.9.15",
         "requestParameters":{
            "maxItems":1000,
            "certificateStatuses":[
```

```
                "ISSUED"
            ]
         },
         "responseElements":null,
         "requestID":"74c99844-ec9c-11e5-ac34-d1e4dfe1a11b",
         "eventID":"cdfe1051-88aa-4aa3-8c33-a325270bff21",
         "eventType":"AwsApiCall",
         "recipientAccountId":"123456789012"
      }
   ]
}
```


証明書のタグを一覧表示する (ListTagsForCertificate)

次の CloudTrail の例は、ListTagsForCertificate API への呼び出しの結果を示しています。

> **Note**
>
> ListTagsForCertificate オペレーションの CloudTrail ログにタグは表示されません。コンソール、 AWS Command Line Interface、または ListTagsForCertificate API を使用して、タグのリストを表示できます。


```
{
   "Records":[
      {
         "eventVersion":"1.04",
         "userIdentity":{
            "type":"IAMUser",
            "principalId":"AIDACKCEVSQ6C2EXAMPLE",
            "arn":"arn:aws:iam::123456789012:user/Alice",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "userName":"Alice"
         },
         "eventTime":"2016-04-06T13:30:11Z",
         "eventSource":"acm.amazonaws.com",
         "eventName":"ListTagsForCertificate",
         "awsRegion":"us-east-1",
         "sourceIPAddress":"192.0.2.0",
         "userAgent":"aws-cli/1.10.16",
         "requestParameters":{
```

```
            "certificateArn":"arn:aws:acm:us-
east-1:123456789012:certificate/12345678-1234-1234-1234-123456789012"
         },
         "responseElements":null,
         "requestID":"b010767f-fbfb-11e5-b596-79e9a97a2544",
         "eventID":"32181be6-a4a0-48d3-8014-c0d972b5163b",
         "eventType":"AwsApiCall",
         "recipientAccountId":"123456789012"
      }
   ]
}
```

証明書からタグを削除する (RemoveTagsFromCertificate)

次の CloudTrail の例は、RemoveTagsFromCertificate API への呼び出しの結果を示しています。

```
{
   "Records":[
      {
         "eventVersion":"1.04",
         "userIdentity":{
            "type":"IAMUser",
            "principalId":"AIDACKCEVSQ6C2EXAMPLE",
            "arn":"arn:aws:iam::123456789012:user/Alice",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "userName":"Alice"
         },
         "eventTime":"2016-04-06T14:10:01Z",
         "eventSource":"acm.amazonaws.com",
         "eventName":"RemoveTagsFromCertificate",
         "awsRegion":"us-east-1",
         "sourceIPAddress":"192.0.2.0",
         "userAgent":"aws-cli/1.10.16",
         "requestParameters":{
            "certificateArn":"arn:aws:acm:us-
east-1:123456789012:certificate/12345678-1234-1234-1234-123456789012",
            "tags":[
               {
                  "value":"Bob",
                  "key":"Admin"
               }
            ]
```

```
         },
         "responseElements":null,
         "requestID":"40ded461-fc01-11e5-a747-85804766d6c9",
         "eventID":"0cfa142e-ef74-4b21-9515-47197780c424",
         "eventType":"AwsApiCall",
         "recipientAccountId":"123456789012"
      }
   ]
}
```

証明書のリクエスト (RequestCertificate)

次の CloudTrail の例は、RequestCertificate API への呼び出しの結果を示しています。

```
{
   "Records":[
      {
         "eventVersion":"1.04",
         "userIdentity":{
            "type":"IAMUser",
            "principalId":"AIDACKCEVSQ6C2EXAMPLE",
            "arn":"arn:aws:iam::123456789012:user/Alice",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "userName":"Alice"
         },
         "eventTime":"2016-03-18T00:00:49Z",
         "eventSource":"acm.amazonaws.com",
         "eventName":"RequestCertificate",
         "awsRegion":"us-east-1",
         "sourceIPAddress":"192.0.2.0",
         "userAgent":"aws-cli/1.9.15",
         "requestParameters":{
            "domainName":"example.com",
            "validationMethod": "DNS",
            "idempotencyToken":"8186023d89681c3ad5",
            "options": {
            "export": "ENABLED"
        },
        "keyAlgorithm": "RSA_2048"
         },
         "responseElements":{
```

```
            "certificateArn":"arn:aws:acm:us-
east-1:123456789012:certificate/12345678-1234-1234-1234-123456789012"
         },
         "requestID":"77dacef3-ec9c-11e5-ac34-d1e4dfe1a11b",
         "eventID":"a4954cdb-8f38-44c7-8927-a38ad4be3ac8",
         "eventType":"AwsApiCall",
         "tlsDetails": {
           "tlsVersion": "TLSv1.3",
           "cipherSuite": "TLS_AES_128_GCM_SHA256",
           "clientProvidedHostHeader": "acm.us-east-1.amazonaws.com"
          },
         "recipientAccountId":"123456789012"
      }
   ]
}
```

証明書の取り消し (RevokeCertificate)

次の CloudTrail のサンプルは、RevokeCertificate API への呼び出しの結果を示す例です。

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:Role-Session-Name",
        "arn": arn:aws:sts::111122223333:assumed-role/Role-Name/Role-Session-Name",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/Admin",
                "accountId": "123456789012",
                "userName": "Admin"
            },
            "attributes": {
                "creationDate": "2016-01-01T19:35:52Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime":"2016-01-01T21:11:45Z",
```

```
    "eventSource": "acm.amazonaws.com",
    "eventName": "RevokeCertificate",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:128.0) Gecko/20100101
 Firefox/128.0",
    "requestParameters": {
        "certificateArn": "arn:aws:acm:us-
east-1:123456789012:certificate/12345678-1234-1234-1234-123456789012",
        "revocationReason": "UNSPECIFIED"
    },
    "responseElements": {
        "certificateArn": "arn:aws:acm:us-
east-1:123456789012:certificate/12345678-1234-1234-1234-123456789012"
    },
    "requestID": "01234567-89ab-cdef-0123-456789abcdef",
    "eventID": "01234567-89ab-cdef-0123-456789abcdef",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "acm.us-east-1.amazonaws.com"
    },
    "sessionCredentialFromConsole": "true"
}
```

検証 E メールの再送信 (ResendValidationEmail)

次の CloudTrail の例は、ResendValidationEmail API への呼び出しの結果を示しています。

```
{
   "Records":[
      {
         "eventVersion":"1.04",
         "userIdentity":{
            "type":"IAMUser",
            "principalId":"AIDACKCEVSQ6C2EXAMPLE",
            "arn":"arn:aws:iam::123456789012:user/Alice",
            "accountId":"123456789012",
```

```
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "userName":"Alice"
         },
         "eventTime":"2016-03-17T23:58:25Z",
         "eventSource":"acm.amazonaws.com",
         "eventName":"ResendValidationEmail",
         "awsRegion":"us-east-1",
         "sourceIPAddress":"192.0.2.0",
         "userAgent":"aws-cli/1.9.15",
         "requestParameters":{
            "domain":"example.com",
            "certificateArn":"arn:aws:acm:us-
east-1:123456789012:certificate/12345678-1234-1234-1234-123456789012",
            "validationDomain":"example.com"
         },
         "responseElements":null,
         "requestID":"23760b88-ec9c-11e5-b6f4-cb861a6f0a28",
         "eventID":"41c11b06-ca91-4c1c-8c61-af349ea8bab8",
         "eventType":"AwsApiCall",
         "recipientAccountId":"123456789012"
      }
   ]
}
```

証明書の取得 (GetCertificate)

次の CloudTrail の例は、GetCertificate API への呼び出しの結果を示しています。

```
{

   "Records":[
      {
         "eventVersion":"1.04",
         "userIdentity":{
            "type":"IAMUser",
            "principalId":"AIDACKCEVSQ6C2EXAMPLE",
            "arn":"arn:aws:iam::123456789012:user/Alice",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "userName":"Alice"
         },
         "eventTime":"2016-03-18T00:00:41Z",
         "eventSource":"acm.amazonaws.com",
```

```
         "eventName":"GetCertificate",
         "awsRegion":"us-east-1",
         "sourceIPAddress":"192.0.2.0",
         "userAgent":"aws-cli/1.9.15",
         "requestParameters":{
            "certificateArn":"arn:aws:acm:us-
east-1:123456789012:certificate/12345678-1234-1234-1234-123456789012"
         },
         "responseElements":{
            "certificateChain":

            "-----BEGIN CERTIFICATE-----
            Base64-encoded certificate chain
            -----END CERTIFICATE-----",
            "certificate":
            "-----BEGIN CERTIFICATE-----
            Base64-encoded certificate
            -----END CERTIFICATE-----"

         },
         "requestID":"744dd891-ec9c-11e5-ac34-d1e4dfe1a11b",
         "eventID":"7aa4f909-00dd-478a-9a00-b2709bcad2bb",
         "eventType":"AwsApiCall",
         "recipientAccountId":"123456789012"
      }
   ]
}
```

統合サービスに対する API 呼び出しのログギング

CloudTrail を使用して、ACM に統合されるサービスによって行われた API コールを監査できます。CloudTrail の使用方法の詳細については、AWS CloudTrail ユーザーガイドを参照してください。以下の例は、ACM 証明書をプロビジョンする AWS リソースに応じて生成される可能性のあるログの種類を示します。

トピック

- ロードバランサーの作成

ロードバランサーの作成

CloudTrail を使用して、ACM に統合されるサービスによって行われた API コールを監査できます。CloudTrail の使用方法の詳細については、AWS CloudTrail ユーザーガイドを参照してください。次の例は、ACM 証明書をプロビジョニングする AWS リソースに応じて生成できるログのタイプを示しています。

トピック

- ロードバランサーの作成
- ロードバランサーを使用して Amazon EC2 インスタンスを登録する
- プライベートキーの暗号化
- プライベートキーの復号

ロードバランサーの作成

以下の例では、Alice という IAM ユーザーによって `CreateLoadBalancer` 関数を呼び出す例を示しています。ロードバランサーの名前は `TestLinuxDefault`、リスナーは ACM 証明書を使用して作成されます。


```
{

   "eventVersion":"1.03",
   "userIdentity":{
      "type":"IAMUser",
      "principalId":"AIDACKCEVSQ6C2EXAMPLE",
      "arn":"arn:aws:iam::111122223333:user/Alice",
      "accountId":"111122223333",
      "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
      "userName":"Alice"
   },
   "eventTime":"2016-01-01T21:10:36Z",
   "eventSource":"elasticloadbalancing.amazonaws.com",
   "eventName":"CreateLoadBalancer",
   "awsRegion":"us-east-1",
   "sourceIPAddress":"192.0.2.0/24",
   "userAgent":"aws-cli/1.9.15",
   "requestParameters":{
      "availabilityZones":[
         "us-east-1b"
      ],
```

```
      "loadBalancerName":"LinuxTest",
      "listeners":[
         {
            "sSLCertificateId":"arn:aws:acm:us-
east-1:111122223333:certificate/12345678-1234-1234-1234-123456789012",
            "protocol":"HTTPS",
            "loadBalancerPort":443,
            "instanceProtocol":"HTTP",
            "instancePort":80
         }
      ]
   },
   "responseElements":{
      "dNSName":"LinuxTest-1234567890.us-east-1.elb.amazonaws.com"
   },
   "requestID":"19669c3b-b0cc-11e5-85b2-57397210a2e5",
   "eventID":"5d6c00c9-a9b8-46ef-9f3b-4589f5be63f7",
   "eventType":"AwsApiCall",
   "recipientAccountId":"111122223333"
}
```

ロードバランサーを使用して Amazon EC2 インスタンスを登録する

Amazon Elastic Compute Cloud (Amazon EC2) インスタンスでウェブサイトまたはアプリケーションをプロビジョニングするとき、ロードバランサーによってそのインスタンスが認識される必要があります。Elastic Load Balancing コンソールまたは AWS Command Line Interfaceによって、これが可能になります。次の例は、 AWS アカウント 123456789012 の LinuxTest という名前のロードバランサー`RegisterInstancesWithLoadBalancer`の への呼び出しを示しています。

```
{
   "eventVersion":"1.03",
   "userIdentity":{
      "type":"IAMUser",
      "principalId":"AIDACKCEVSQ6C2EXAMPLE",
      "arn":"arn:aws:iam::123456789012:user/ALice",
      "accountId":"123456789012",
      "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
      "userName":"Alice",
      "sessionContext":{
         "attributes":{
            "mfaAuthenticated":"false",
            "creationDate":"2016-01-01T19:35:52Z"
         }
```

```
      },
      "invokedBy":"signin.amazonaws.com"
   },
   "eventTime":"2016-01-01T21:11:45Z",
   "eventSource":"elasticloadbalancing.amazonaws.com",
   "eventName":"RegisterInstancesWithLoadBalancer",
   "awsRegion":"us-east-1",
   "sourceIPAddress":"192.0.2.0/24",
   "userAgent":"signin.amazonaws.com",
   "requestParameters":{
      "loadBalancerName":"LinuxTest",
      "instances":[
         {
            "instanceId":"i-c67f4e78"
         }
      ]
   },
   "responseElements":{
      "instances":[
         {
            "instanceId":"i-c67f4e78"
         }
      ]
   },
   "requestID":"438b07dc-b0cc-11e5-8afb-cda7ba020551",
   "eventID":"9f284ca6-cbe5-42a1-8251-4f0e6b5739d6",
   "eventType":"AwsApiCall",
   "recipientAccountId":"123456789012"
}
```

プライベートキーの暗号化

以下に、ACM 証明書に関連付けられたプライベートキーを暗号化する Encrypt 呼び出しの例を示します。暗号化は AWS内で実行されます。

```
{
   "Records":[
      {
         "eventVersion":"1.03",
         "userIdentity":{
            "type":"IAMUser",
            "principalId":"AIDACKCEVSQ6C2EXAMPLE",
            "arn":"arn:aws:iam::111122223333:user/acm",
```

```
            "accountId":"111122223333",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "userName":"acm"
         },
         "eventTime":"2016-01-05T18:36:29Z",
         "eventSource":"kms.amazonaws.com",
         "eventName":"Encrypt",
         "awsRegion":"us-east-1",
         "sourceIPAddress":"AWS Internal",
         "userAgent":"aws-internal",
         "requestParameters":{
            "keyId":"arn:aws:kms:us-east-1:123456789012:alias/aws/acm",
            "encryptionContext":{
               "aws:acm:arn":"arn:aws:acm:us-
east-1:123456789012:certificate/12345678-1234-1234-1234-123456789012"
            }
         },
         "responseElements":null,
         "requestID":"3c417351-b3db-11e5-9a24-7d9457362fcc",
         "eventID":"1794fe70-796a-45f5-811b-6584948f24ac",
         "readOnly":true,
         "resources":[
            {
               "ARN":"arn:aws:kms:us-
east-1:123456789012:key/87654321-4321-4321-4321-210987654321",
               "accountId":"123456789012"
            }
         ],
         "eventType":"AwsServiceEvent",
         "recipientAccountId":"123456789012"
      }
   ]
}
```

プライベートキーの復号

以下に、ACM 証明書に関連付けられたプライベートキーを復号化する Decrypt 呼び出しの例を示します。復号は 内で実行され AWS、復号されたキーが離れることはありません AWS。

```
{
   "eventVersion":"1.03",
   "userIdentity":{
      "type":"AssumedRole",
```

```
      "principalId":"AIDACKCEVSQ6C2EXAMPLE:1aba0dc8b3a728d6998c234a99178eff",
      "arn":"arn:aws:sts::111122223333:assumed-role/
DecryptACMCertificate/1aba0dc8b3a728d6998c234a99178eff",
      "accountId":"111122223333",
      "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
      "sessionContext":{
         "attributes":{
            "mfaAuthenticated":"false",
            "creationDate":"2016-01-01T21:13:28Z"
         },
         "sessionIssuer":{
            "type":"Role",
            "principalId":"APKAEIBAERJR2EXAMPLE",
            "arn":"arn:aws:iam::111122223333:role/DecryptACMCertificate",
            "accountId":"111122223333",
            "userName":"DecryptACMCertificate"
         }
      }
   },
   "eventTime":"2016-01-01T21:13:28Z",
   "eventSource":"kms.amazonaws.com",
   "eventName":"Decrypt",
   "awsRegion":"us-east-1",
   "sourceIPAddress":"AWS Internal",
   "userAgent":"aws-internal/3",
   "requestParameters":{
      "encryptionContext":{
         "aws:elasticloadbalancing:arn":"arn:aws:elasticloadbalancing:us-
east-1:123456789012:loadbalancer/LinuxTest",
         "aws:acm:arn":"arn:aws:acm:us-
east-1:123456789012:certificate/87654321-4321-4321-4321-210987654321"
      }
   },
   "responseElements":null,
   "requestID":"809a70ff-b0cc-11e5-8f42-c7fdf1cb6e6a",
   "eventID":"7f89f7a7-baff-4802-8a88-851488607fb9",
   "readOnly":true,
   "resources":[
      {
         "ARN":"arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012",
         "accountId":"123456789012"
      }
   ],
```

```
   "eventType":"AwsServiceEvent",
   "recipientAccountId":"123456789012"
}
```

サポートされている CloudWatch メトリクス

Amazon CloudWatch は、 AWS リソースのモニタリングサービスです。CloudWatch を使用して、メトリクスの収集と追跡、アラームの設定、 AWS リソースの変更への自動対応を行うことができます。ACM は、有効期限が切れるまで、アカウント内の証明書ごとに 1 日に 2 回メトリクスを公開します。

`AWS/CertificateManager` 名前空間には、次のメトリクスが含まれます。

メトリクス	説明	Unit	ディメンション
`DaysToExpiry`	証明書の有効期限が切れるまでの日数。ACMは、証明書の有効期限が切れると、このメトリクスの公開を停止します。	整数	CertificateArn • 値：CA 証明書の ARN

CloudWatch メトリクスの詳細については、次のトピックを参照してください。

- Amazon CloudWatch メトリクスの使用
- Amazon CloudWatch アラームの作成

SDK for Java AWS Certificate Manager で を使用する

AWS Certificate Manager API を使用して、HTTP リクエストを送信してプログラムでサービスとやり取りできます。詳細については、「APIリファレンスAWS Certificate Manager」を参照してください。

ウェブ API (または HTTP API) に加えて、 AWS SDKs とコマンドラインツールを使用して ACM やその他のサービスとやり取りできます。詳細については、「Amazon ウェブ サービスのツール」を参照してください。

以下のトピックでは、 AWS SDKs の 1 つである を使用して AWS SDK for JavaAPI で使用可能なオペレーションの一部を実行する方法を示します AWS Certificate Manager 。

トピック

- 証明書にタグを追加する
- 証明書の削除
- 証明書についての説明
- 証明書のエクスポート
- 証明書と証明書チェーンの取得
- 証明書のインポート
- 証明書の一覧
- 証明書の更新
- 証明書タグの一覧表示
- 証明書からタグを削除する
- 証明書のリクエスト
- 検証 E メールの再送信

証明書にタグを追加する

次の例は、AddTagsToCertificate 機能を使用する方法を示しています。

```
package com.amazonaws.samples;

import java.io.IOException;
import java.nio.ByteBuffer;
```

```
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;

import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.certificatemanager.AWSCertificateManager;
import com.amazonaws.services.certificatemanager.AWSCertificateManagerClientBuilder;
import com.amazonaws.services.certificatemanager.model.ImportCertificateRequest;
import com.amazonaws.services.certificatemanager.model.ImportCertificateResult;
/**
 * This sample demonstrates how to use the ImportCertificate function in the AWS
 Certificate Manager
 * service.
 *
 * Input parameters:
 *   Accesskey - AWS access key
 *   SecretKey - AWS secret key
 *   CertificateArn - Use to reimport a certificate (not included in this example).
 *   region - AWS region
 *   Certificate - PEM file that contains the certificate to import. Ex: /data/certs/
servercert.pem
 *   CertificateChain - The certificate chain, not including the end-entity
 certificate.
 *   PrivateKey - The private key that matches the public key in the certificate.
 *
 * Output parameter:
 *   CertificcateArn - The ARN of the imported certificate.
 *
 */
public class AWSCertificateManagerSample {

    public static void main(String[] args) throws IOException {
     String accessKey = "";
     String secretKey = "";
     String certificateArn = null;
     Regions region = Regions.DEFAULT_REGION;
     String serverCertFilePath = "";
     String privateKeyFilePath = "";
     String caCertFilePath = "";

     ImportCertificateRequest req = new ImportCertificateRequest()
       .withCertificate(getCertContent(serverCertFilePath))
```

```
        .withPrivateKey(getCertContent(privateKeyFilePath))

  .withCertificateChain(getCertContent(caCertFilePath)).withCertificateArn(certificateArn);

      AWSCertificateManager client =
  AWSCertificateManagerClientBuilder.standard().withRegion(region)
        .withCredentials(new AWSStaticCredentialsProvider(new
  BasicAWSCredentials(accessKey, secretKey)))
        .build();
      ImportCertificateResult result = client.importCertificate(req);

      System.out.println(result.getCertificateArn());

      List<Tag> expectedTags =
  ImmutableList.of(Tag.builder().withKey("key").withValue("value").build());

      AddTagsToCertificateRequest addTagsToCertificateRequest =
  AddTagsToCertificateRequest.builder()
              .withCertificateArn(result.getCertificateArn())
              .withTags(tags)
              .build();

      client.addTagsToCertificate(addTagsToCertificateRequest);
     }

     private static ByteBuffer getCertContent(String filePath) throws IOException {
      String fileContent = new String(Files.readAllBytes(Paths.get(filePath)));
      return StandardCharsets.UTF_8.encode(fileContent);
     }
}
```

証明書の削除

次の例は、DeleteCertificate 機能を使用する方法を示しています。成功した場合、関数は空のセット {} を返します。

```
package com.amazonaws.samples;

import com.amazonaws.services.certificatemanager.AWSCertificateManagerClientBuilder;
import com.amazonaws.services.certificatemanager.AWSCertificateManager;
import com.amazonaws.services.certificatemanager.model.DeleteCertificateRequest;
import com.amazonaws.services.certificatemanager.model.DeleteCertificateResult;
```

```
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.regions.Regions;

import com.amazonaws.services.certificatemanager.model.InvalidArnException;
import com.amazonaws.services.certificatemanager.model.ResourceInUseException;
import com.amazonaws.services.certificatemanager.model.ResourceNotFoundException;
import com.amazonaws.AmazonClientException;

/**
 * This sample demonstrates how to use the DeleteCertificate function in the AWS
 Certificate
 * Manager service.
 *
 * Input parameter:
 *   CertificateArn - The ARN of the certificate to delete.
 *
 */

public class AWSCertificateManagerExample {

   public static void main(String[] args) throws Exception{

      // Retrieve your credentials from the C:\Users\name\.aws\credentials file in
 Windows
      // or the ~/.aws/credentials file in Linux.
      AWSCredentials credentials = null;
      try {
          credentials = new ProfileCredentialsProvider().getCredentials();
      }
      catch (Exception ex) {
          throw new AmazonClientException("Cannot load the credentials from file.",
 ex);
      }

      // Create a client.
      AWSCertificateManager client = AWSCertificateManagerClientBuilder.standard()
              .withRegion(Regions.US_EAST_1)
              .withCredentials(new AWSStaticCredentialsProvider(credentials))
              .build();

      // Create a request object and specify the ARN of the certificate to delete.
```

```
      DeleteCertificateRequest req = new DeleteCertificateRequest();

 req.setCertificateArn("arn:aws:acm:region:account:certificate/
12345678-1234-1234-1234-123456789012");

      // Delete the specified certificate.
      DeleteCertificateResult result = null;
      try {
         result = client.deleteCertificate(req);
      }
      catch (InvalidArnException ex)
      {
         throw ex;
      }
      catch (ResourceInUseException ex)
      {
         throw ex;
      }
      catch (ResourceNotFoundException ex)
      {
         throw ex;
      }

      // Display the result.
      System.out.println(result);

   }
}
```

証明書についての説明

次の例は、DescribeCertificate 関数を使用する方法を示しています。

```
package com.amazonaws.samples;

import com.amazonaws.services.certificatemanager.AWSCertificateManagerClientBuilder;
import com.amazonaws.services.certificatemanager.AWSCertificateManager;
import com.amazonaws.services.certificatemanager.model.DescribeCertificateRequest;
import com.amazonaws.services.certificatemanager.model.DescribeCertificateResult;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.AWSCredentials;
```

```
import com.amazonaws.regions.Regions;

import com.amazonaws.services.certificatemanager.model.InvalidArnException;
import com.amazonaws.services.certificatemanager.model.ResourceNotFoundException;
import com.amazonaws.AmazonClientException;

/**
 * This sample demonstrates how to use the DescribeCertificate function in the AWS
 Certificate
 * Manager service.
 *
 * Input parameter:
 *   CertificateArn - The ARN of the certificate to be described.
 *
 * Output parameter:
 *   Certificate information
 *
 */

public class AWSCertificateManagerExample {

   public static void main(String[] args) throws Exception{

      // Retrieve your credentials from the C:\Users\name\.aws\credentials file in
 Windows
      // or the ~/.aws/credentials file in Linux.
      AWSCredentials credentials = null;
      try {
          credentials = new ProfileCredentialsProvider().getCredentials();
      }
      catch (Exception ex) {
          throw new AmazonClientException("Cannot load the credentials from file.",
 ex);
      }

      // Create a client.
      AWSCertificateManager client = AWSCertificateManagerClientBuilder.standard()
              .withRegion(Regions.US_EAST_1)
              .withCredentials(new AWSStaticCredentialsProvider(credentials))
              .build();

      // Create a request object and set the ARN of the certificate to be described.
      DescribeCertificateRequest req = new DescribeCertificateRequest();
```

```
 req.setCertificateArn("arn:aws:acm:region:account:certificate/
12345678-1234-1234-1234-123456789012");

      DescribeCertificateResult result = null;
      try{
         result = client.describeCertificate(req);
      }
      catch (InvalidArnException ex)
      {
         throw ex;
      }
      catch (ResourceNotFoundException ex)
      {
         throw ex;
      }

      // Display the certificate information.
      System.out.println(result);

   }
}
```

成功した場合、上記の例は、次のような情報を表示します。

```
{
    Certificate: {
        CertificateArn:
 arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012,
        DomainName: www.example.com,
        SubjectAlternativeNames: [www.example.com],
        DomainValidationOptions: [{
            DomainName: www.example.com,
        }],
        Serial: 10: 0a,
        Subject: C=US,
        ST=WA,
        L=Seattle,
        O=ExampleCompany,
        OU=sales,
        CN=www.example.com,
        Issuer: ExampleCompany,
        ImportedAt: FriOct0608: 17: 39PDT2017,
```

```
        Status: ISSUED,
        NotBefore: ThuOct0510: 14: 32PDT2017,
        NotAfter: SunOct0310: 14: 32PDT2027,
        KeyAlgorithm: RSA-2048,
        SignatureAlgorithm: SHA256WITHRSA,
        InUseBy: [],
        Type: IMPORTED,
    }
}
```

証明書のエクスポート

次の例は、ExportCertificate 関数を使用する方法を示しています。この関数はプライベート認証機関 (CA) によって発行されたプライベート証明書を PKCS #8 形式でエクスポートします。(パブリック証明書は、ACM で発行されたものでもインポートされたものでもエクスポートできません。) また、証明書チェーンおよびプライベートキーもエクスポートします。この例では、キーのパスフレーズがローカルファイルに保存されます。

```
package com.amazonaws.samples;

import com.amazonaws.AmazonClientException;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.regions.Regions;

import com.amazonaws.services.certificatemanager.AWSCertificateManagerClientBuilder;
import com.amazonaws.services.certificatemanager.AWSCertificateManager;


import com.amazonaws.services.certificatemanager.model.ExportCertificateRequest;
import com.amazonaws.services.certificatemanager.model.ExportCertificateResult;

import com.amazonaws.services.certificatemanager.model.InvalidArnException;
import com.amazonaws.services.certificatemanager.model.InvalidTagException;
import com.amazonaws.services.certificatemanager.model.ResourceNotFoundException;

import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.RandomAccessFile;
import java.nio.ByteBuffer;
```

```
import java.nio.channels.FileChannel;

public class ExportCertificate {

   public static void main(String[] args) throws Exception {

      // Retrieve your credentials from the C:\Users\name\.aws\credentials file in
 Windows
      // or the ~/.aws/credentials in Linux.
      AWSCredentials credentials = null;
      try {
          credentials = new ProfileCredentialsProvider().getCredentials();
      }
      catch (Exception ex) {
          throw new AmazonClientException("Cannot load your credentials from file.",
 ex);
      }

      // Create a client.
      AWSCertificateManager client = AWSCertificateManagerClientBuilder.standard()
              .withRegion(Regions.your_region)
              .withCredentials(new AWSStaticCredentialsProvider(credentials))
              .build();

      // Initialize a file descriptor for the passphrase file.
      RandomAccessFile file_passphrase = null;

      // Initialize a buffer for the passphrase.
      ByteBuffer buf_passphrase = null;

      // Create a file stream for reading the private key passphrase.
      try {
         file_passphrase = new RandomAccessFile("C:\\Temp\\password.txt", "r");
      }
      catch (IllegalArgumentException ex) {
         throw ex;
      }
      catch (SecurityException ex) {
         throw ex;
      }
      catch (FileNotFoundException ex) {
         throw ex;
      }
```

```
      // Create a channel to map the file.
      FileChannel channel_passphrase = file_passphrase.getChannel();

      // Map the file to the buffer.
      try {
         buf_passphrase = channel_passphrase.map(FileChannel.MapMode.READ_ONLY, 0,
channel_passphrase.size());

         // Clean up after the file is mapped.
         channel_passphrase.close();
         file_passphrase.close();
      }
      catch (IOException ex)
      {
         throw ex;
      }

      // Create a request object.
      ExportCertificateRequest req = new ExportCertificateRequest();

      // Set the certificate ARN.
      req.withCertificateArn("arn:aws:acm:region:account:"
            +"certificate/M12345678-1234-1234-1234-123456789012");

      // Set the passphrase.
      req.withPassphrase(buf_passphrase);

      // Export the certificate.
      ExportCertificateResult result = null;

      try {
         result = client.exportCertificate(req);
      }
      catch(InvalidArnException ex)
      {
         throw ex;
      }
      catch (InvalidTagException ex)
      {
         throw ex;
      }
      catch (ResourceNotFoundException ex)
      {
         throw ex;
```

```
        }

        // Clear the buffer.
        buf_passphrase.clear();

        // Display the certificate and certificate chain.
        String certificate = result.getCertificate();
        System.out.println(certificate);

        String certificate_chain = result.getCertificateChain();
        System.out.println(certificate_chain);

        // This example retrieves but does not display the private key.
        String private_key = result.getPrivateKey();
    }
}
```

証明書と証明書チェーンの取得

次の例は、GetCertificate 機能を使用する方法を示しています。

```
package com.amazonaws.samples;

import com.amazonaws.regions.Regions;
import com.amazonaws.services.certificatemanager.AWSCertificateManagerClientBuilder;
import com.amazonaws.services.certificatemanager.AWSCertificateManager;
import com.amazonaws.services.certificatemanager.model.GetCertificateRequest;
import com.amazonaws.services.certificatemanager.model.GetCertificateResult;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.AWSCredentials;

import com.amazonaws.services.certificatemanager.model.InvalidArnException;
import com.amazonaws.services.certificatemanager.model.ResourceNotFoundException;
import com.amazonaws.services.certificatemanager.model.RequestInProgressException;
import com.amazonaws.AmazonClientException;

/**
 * This sample demonstrates how to use the GetCertificate function in the AWS
 Certificate
 * Manager service.
 *
```

```
 * Input parameter:
 *   CertificateArn - The ARN of the certificate to retrieve.
 *
 * Output parameters:
 *   Certificate - A base64-encoded certificate in PEM format.
 *   CertificateChain - The base64-encoded certificate chain in PEM format.
 *
 */

public class AWSCertificateManagerExample {

   public static void main(String[] args) throws Exception{

      // Retrieve your credentials from the C:\Users\name\.aws\credentials file in
 Windows
      // or the ~/.aws/credentials file in Linux.
      AWSCredentials credentials = null;
      try {
          credentials = new ProfileCredentialsProvider().getCredentials();
      }
      catch (Exception ex) {
          throw new AmazonClientException("Cannot load the credentials from the
 credential profiles file.", ex);
      }

      // Create a client.
      AWSCertificateManager client = AWSCertificateManagerClientBuilder.standard()
              .withRegion(Regions.US_EAST_1)
              .withCredentials(new AWSStaticCredentialsProvider(credentials))
              .build();

      // Create a request object and set the ARN of the certificate to be described.
      GetCertificateRequest req = new GetCertificateRequest();

 req.setCertificateArn("arn:aws:acm:region:account:certificate/
12345678-1234-1234-1234-123456789012");

      // Retrieve the certificate and certificate chain.
      // If you recently requested the certificate, loop until it has been created.
      GetCertificateResult result = null;
      long totalTimeout = 120000l;
      long timeSlept = 0l;
      long sleepInterval = 10000l;
      while (result == null && timeSlept < totalTimeout) {
```

```
         try {
            result = client.getCertificate(req);
         }
         catch (RequestInProgressException ex) {
            Thread.sleep(sleepInterval);
         }
         catch (ResourceNotFoundException ex)
         {
            throw ex;
         }
         catch (InvalidArnException ex)
         {
            throw ex;
         }

         timeSlept += sleepInterval;
      }

      // Display the certificate information.
      System.out.println(result);
   }
}
```

前述の例は、以下のような出力を作成します。

```
{Certificate: -----BEGIN CERTIFICATE-----
    base64-encoded certificate
-----END CERTIFICATE-----,
CertificateChain: -----BEGIN CERTIFICATE-----
    base64-encoded certificate chain
-----END CERTIFICATE-----
}
```

証明書のインポート

次の例は、ImportCertificate 機能を使用する方法を示しています。

```
package com.amazonaws.samples;

import com.amazonaws.services.certificatemanager.AWSCertificateManagerClientBuilder;
import com.amazonaws.services.certificatemanager.AWSCertificateManager;
```

```
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.regions.Regions;

import com.amazonaws.services.certificatemanager.model.ImportCertificateRequest;
import com.amazonaws.services.certificatemanager.model.ImportCertificateResult;
import com.amazonaws.services.certificatemanager.model.LimitExceededException;
import com.amazonaws.services.certificatemanager.model.ResourceNotFoundException;
import com.amazonaws.AmazonClientException;
import java.io.FileNotFoundException;
import java.io.IOException;

import java.io.RandomAccessFile;
import java.nio.ByteBuffer;
import java.nio.channels.FileChannel;

/**
 * This sample demonstrates how to use the ImportCertificate function in the AWS
 Certificate Manager
 * service.
 *
 * Input parameters:
 *   Certificate - PEM file that contains the certificate to import.
 *   CertificateArn - Use to reimport a certificate (not included in this example).
 *   CertificateChain - The certificate chain, not including the end-entity
 certificate.
 *   PrivateKey - The private key that matches the public key in the certificate.
 *
 * Output parameter:
 *   CertificcateArn - The ARN of the imported certificate.
 *
 */
public class AWSCertificateManagerSample {

   public static void main(String[] args) throws Exception {

      // Retrieve your credentials from the C:\Users\name\.aws\credentials file in
 Windows
      // or the ~/.aws/credentials file in Linux.
      AWSCredentials credentials = null;
      try {
          credentials = new ProfileCredentialsProvider().getCredentials();
      }
```

```
        catch (Exception ex) {
            throw new AmazonClientException(
                "Cannot load the credentials from file.", ex);
        }

        // Create a client.
        AWSCertificateManager client = AWSCertificateManagerClientBuilder.standard()
                .withRegion(Regions.US_EAST_1)
                .withCredentials(new AWSStaticCredentialsProvider(credentials))
                .build();

        // Initialize the file descriptors.
        RandomAccessFile file_certificate = null;
        RandomAccessFile file_chain = null;
        RandomAccessFile file_key = null;

        // Initialize the buffers.
        ByteBuffer buf_certificate = null;
        ByteBuffer buf_chain = null;
        ByteBuffer buf_key = null;

        // Create the file streams for reading.
        try {
           file_certificate = new RandomAccessFile("C:\\Temp\\certificate.pem", "r");
           file_chain = new RandomAccessFile("C:\\Temp\\chain.pem", "r");
           file_key = new RandomAccessFile("C:\\Temp\\private_key.pem", "r");
        }
        catch (IllegalArgumentException ex) {
           throw ex;
        }
        catch (SecurityException ex) {
           throw ex;
        }
        catch (FileNotFoundException ex) {
           throw ex;
        }

        // Create channels for mapping the files.
        FileChannel channel_certificate = file_certificate.getChannel();
        FileChannel channel_chain = file_chain.getChannel();
        FileChannel channel_key = file_key.getChannel();

        // Map the files to buffers.
        try {
```

```
        buf_certificate = channel_certificate.map(FileChannel.MapMode.READ_ONLY, 0,
channel_certificate.size());
        buf_chain = channel_chain.map(FileChannel.MapMode.READ_ONLY, 0,
channel_chain.size());
        buf_key = channel_key.map(FileChannel.MapMode.READ_ONLY, 0,
channel_key.size());

        // The files have been mapped, so clean up.
        channel_certificate.close();
        channel_chain.close();
        channel_key.close();
        file_certificate.close();
        file_chain.close();
        file_key.close();
     }
     catch (IOException ex)
     {
        throw ex;
     }

     // Create a request object and set the parameters.
     ImportCertificateRequest req = new ImportCertificateRequest();
     req.setCertificate(buf_certificate);
     req.setCertificateChain(buf_chain);
     req.setPrivateKey(buf_key);

     // Import the certificate.
     ImportCertificateResult result = null;
     try {
        result = client.importCertificate(req);
     }
     catch(LimitExceededException ex)
     {
        throw ex;
     }
     catch (ResourceNotFoundException ex)
     {
        throw ex;
     }

     // Clear the buffers.
     buf_certificate.clear();
     buf_chain.clear();
     buf_key.clear();
```

```
      // Retrieve and display the certificate ARN.
      String arn = result.getCertificateArn();
      System.out.println(arn);
    }
}
```

証明書の一覧

以下の例は、ListCertificates 関数を使用する方法を示しています。

```
package com.amazonaws.samples;

import com.amazonaws.services.certificatemanager.AWSCertificateManagerClientBuilder;
import com.amazonaws.services.certificatemanager.AWSCertificateManager;
import com.amazonaws.services.certificatemanager.model.ListCertificatesRequest;
import com.amazonaws.services.certificatemanager.model.ListCertificatesResult;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.regions.Regions;

import com.amazonaws.AmazonClientException;

import java.util.Arrays;
import java.util.List;

/**
 * This sample demonstrates how to use the ListCertificates function in the AWS
 Certificate
 * Manager service.
 *
 * Input parameters:
 *   CertificateStatuses - An array of strings that contains the statuses to use for
 filtering.
 *   MaxItems - The maximum number of certificates to return in the response.
 *   NextToken - Use when paginating results.
 *
 * Output parameters:
 *   CertificateSummaryList - A list of certificates.
 *   NextToken - Use to show additional results when paginating a truncated list.
 *
```

```
 */

public class AWSCertificateManagerExample {

   public static void main(String[] args) throws Exception{

      // Retrieve your credentials from the C:\Users\name\.aws\credentials file in
 Windows
      // or the ~/.aws/credentials file in Linux.
      AWSCredentials credentials = null;
      try {
          credentials = new ProfileCredentialsProvider().getCredentials();
      }
      catch (Exception ex) {
          throw new AmazonClientException("Cannot load the credentials from file.",
 ex);
      }

      // Create a client.
      AWSCertificateManager client = AWSCertificateManagerClientBuilder.standard()
              .withRegion(Regions.US_EAST_1)
              .withCredentials(new AWSStaticCredentialsProvider(credentials))
              .build();

      // Create a request object and set the parameters.
      ListCertificatesRequest req = new ListCertificatesRequest();
      List<String> Statuses = Arrays.asList("ISSUED", "EXPIRED", "PENDING_VALIDATION",
 "FAILED");
      req.setCertificateStatuses(Statuses);
      req.setMaxItems(10);

      // Retrieve the list of certificates.
      ListCertificatesResult result = null;
      try {
         result = client.listCertificates(req);
      }
      catch (Exception ex)
      {
         throw ex;
      }

      // Display the certificate list.
      System.out.println(result);
   }
```

```
}
```

前述のサンプルは、以下のような出力を作成します。

```
{
    CertificateSummaryList: [{
        CertificateArn:
 arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012,
        DomainName: www.example1.com
    },
    {
        CertificateArn:
 arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012,
        DomainName: www.example2.com
    },
    {
        CertificateArn:
 arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012,
        DomainName: www.example3.com
    }]
}
```

証明書の更新

次の例は、RenewCertificate 関数を使用する方法を示しています。この関数は、プライベート認証機関 (CA) によって発行され、ExportCertificate 関数でエクスポートされたプライベート証明書を更新します。この時点では、エクスポートされたプライベート証明書のみをこの関数で更新できます。ACM で AWS Private CA 証明書を更新するには、まず ACM サービスプリンシパルにそのアクセス許可を付与する必要があります。詳細については、「ACM への証明書更新アクセス許可の割り当て」を参照してください。

```
package com.amazonaws.samples;

import com.amazonaws.AmazonClientException;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.regions.Regions;

import com.amazonaws.services.certificatemanager.AWSCertificateManagerClientBuilder;
```

```
import com.amazonaws.services.certificatemanager.AWSCertificateManager;

import com.amazonaws.services.certificatemanager.model.RenewCertificateRequest;
import com.amazonaws.services.certificatemanager.model.RenewCertificateResult;


import com.amazonaws.services.certificatemanager.model.InvalidArnException;
import com.amazonaws.services.certificatemanager.model.ResourceNotFoundException;
import com.amazonaws.services.certificatemanager.model.ValidationException;

import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.RandomAccessFile;
import java.nio.ByteBuffer;
import java.nio.channels.FileChannel;

public class RenewCertificate {

   public static void main(String[] args) throws Exception {

      // Retrieve your credentials from the C:\Users\name\.aws\credentials file in
 Windows
      // or the ~/.aws/credentials in Linux.
      AWSCredentials credentials = null;
      try {
          credentials = new ProfileCredentialsProvider().getCredentials();
      }
      catch (Exception ex) {
          throw new AmazonClientException("Cannot load your credentials from file.",
 ex);
      }

      // Create a client.
      AWSCertificateManager client = AWSCertificateManagerClientBuilder.standard()
              .withRegion(Regions.your_region)
              .withCredentials(new AWSStaticCredentialsProvider(credentials))
              .build();


      // Create a request object and specify the ARN of the certificate to renew.
      RenewCertificateRequest req = new RenewCertificateRequest();
      req.withCertificateArn("arn:aws:acm:region:account:"
            +"certificate/M12345678-1234-1234-1234-123456789012");
```

```
      // Renew the certificate.
      RenewCertificateResult result = null;
      try {
         result = client.renewCertificate(req);
      }
      catch(InvalidArnException ex)
      {
         throw ex;
      }
      catch (ResourceNotFoundException ex)
      {
         throw ex;
      }
      catch (ValidationException ex)
      {
         throw ex;
      }

      // Display the result.
     System.out.println(result);
   }
}
```

証明書タグの一覧表示

以下の例は、ListTagsForCertificate 関数を使用する方法を示しています。

```
package com.amazonaws.samples;

import com.amazonaws.services.certificatemanager.AWSCertificateManagerClientBuilder;
import com.amazonaws.services.certificatemanager.AWSCertificateManager;
import com.amazonaws.services.certificatemanager.model.ListTagsForCertificateRequest;
import com.amazonaws.services.certificatemanager.model.ListTagsForCertificateResult;

import com.amazonaws.services.certificatemanager.model.InvalidArnException;
import com.amazonaws.services.certificatemanager.model.ResourceNotFoundException;
import com.amazonaws.AmazonClientException;

import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.regions.Regions;
```

```
/**
 * This sample demonstrates how to use the ListTagsForCertificate function in the AWS
 Certificate
 * Manager service.
 *
 * Input parameter:
 *   CertificateArn - The ARN of the certificate whose tags you want to list.
 *
*/

public class AWSCertificateManagerExample {

   public static void main(String[] args) throws Exception{

      // Retrieve your credentials from the C:\Users\name\.aws\credentials file in
 Windows
      // or the ~/.aws/credentials file in Linux.
      AWSCredentials credentials = null;
      try {
          credentials = new ProfileCredentialsProvider().getCredentials();
      }
      catch (Exception ex) {
          throw new AmazonClientException("Cannot load your credentials from file.",
 ex);
      }

      // Create a client.
      AWSCertificateManager client = AWSCertificateManagerClientBuilder.standard()
              .withRegion(Regions.US_EAST_1)
              .withCredentials(new AWSStaticCredentialsProvider(credentials))
              .build();

      // Create a request object and specify the ARN of the certificate.
      ListTagsForCertificateRequest req = new ListTagsForCertificateRequest();

 req.setCertificateArn("arn:aws:acm:region:account:certificate/
12345678-1234-1234-1234-123456789012");

      // Create a result object.
      ListTagsForCertificateResult result = null;
      try {
         result = client.listTagsForCertificate(req);
```

```
      }
      catch(InvalidArnException ex) {
         throw ex;
      }
      catch(ResourceNotFoundException ex) {
         throw ex;
      }

      // Display the result.
      System.out.println(result);

   }
}
```

前述のサンプルは、以下のような出力を作成します。

```
{Tags: [{Key: Purpose,Value: Test}, {Key: Short_Name,Value: My_Cert}]}
```

証明書からタグを削除する

以下の例は、RemoveTagsFromCertificate 関数を使用する方法を示しています。

```
package com.amazonaws.samples;

import com.amazonaws.services.certificatemanager.AWSCertificateManagerClientBuilder;
import com.amazonaws.services.certificatemanager.AWSCertificateManager;
import
 com.amazonaws.services.certificatemanager.model.RemoveTagsFromCertificateRequest;
import com.amazonaws.services.certificatemanager.model.RemoveTagsFromCertificateResult;
import com.amazonaws.services.certificatemanager.model.Tag;

import com.amazonaws.services.certificatemanager.model.InvalidArnException;
import com.amazonaws.services.certificatemanager.model.InvalidTagException;
import com.amazonaws.services.certificatemanager.model.ResourceNotFoundException;
import com.amazonaws.AmazonClientException;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.regions.Regions;

import java.util.ArrayList;
```

```
/**
 * This sample demonstrates how to use the RemoveTagsFromCertificate function in the
 AWS Certificate
 * Manager service.
 *
 * Input parameters:
 *   CertificateArn - The ARN of the certificate from which you want to remove one or
 more tags.
 *   Tags - A collection of key-value pairs that specify which tags to remove.
 *
*/

public class AWSCertificateManagerExample {

   public static void main(String[] args) throws Exception {

      // Retrieve your credentials from the C:\Users\name\.aws\credentials file in
 Windows
      // or the ~/.aws/credentials file in Linux.
      AWSCredentials credentials = null;
      try {
          credentials = new ProfileCredentialsProvider().getCredentials();
      }
      catch (Exception ex) {
          throw new AmazonClientException("Cannot load your credentials from file.",
 ex);
      }

      // Create a client.
      AWSCertificateManager client = AWSCertificateManagerClientBuilder.standard()
              .withRegion(Regions.US_EAST_1)
              .withCredentials(new AWSStaticCredentialsProvider(credentials))
              .build();

      // Specify the tags to remove.
      Tag tag1 = new Tag();
      tag1.setKey("Short_Name");
      tag1.setValue("My_Cert");

      Tag tag2 = new Tag()
            .withKey("Purpose")
            .withValue("Test");
```

```
      // Add the tags to a collection.
      ArrayList<Tag> tags = new ArrayList<Tag>();
      tags.add(tag1);
      tags.add(tag2);

      // Create a request object.
      RemoveTagsFromCertificateRequest req = new RemoveTagsFromCertificateRequest();

 req.setCertificateArn("arn:aws:acm:region:account:certificate/
12345678-1234-1234-1234-123456789012");
      req.setTags(tags);

      // Create a result object.
      RemoveTagsFromCertificateResult result = null;
      try {
         result = client.removeTagsFromCertificate(req);
      }
      catch(InvalidArnException ex)
      {
         throw ex;
      }
      catch(InvalidTagException ex)
      {
         throw ex;
      }
      catch(ResourceNotFoundException ex)
      {
         throw ex;
      }

      // Display the result.
      System.out.println(result);
   }
}
```

証明書のリクエスト

次の例は、RequestCertificate 機能を使用する方法を示しています。

```
package com.amazonaws.samples;

import com.amazonaws.services.certificatemanager.AWSCertificateManagerClientBuilder;
import com.amazonaws.services.certificatemanager.AWSCertificateManager;
```

```
import com.amazonaws.services.certificatemanager.model.RequestCertificateRequest;
import com.amazonaws.services.certificatemanager.model.RequestCertificateResult;

import
 com.amazonaws.services.certificatemanager.model.InvalidDomainValidationOptionsException;
import com.amazonaws.services.certificatemanager.model.LimitExceededException;
import com.amazonaws.AmazonClientException;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.regions.Regions;

import java.util.ArrayList;

/**
 * This sample demonstrates how to use the RequestCertificate function in the AWS
 Certificate
 * Manager service.
 *
 * Input parameters:
 *   DomainName - FQDN of your site.
 *   DomainValidationOptions - Domain name for email validation.
 *   IdempotencyToken - Distinguishes between calls to RequestCertificate.
 *   SubjectAlternativeNames - Additional FQDNs for the subject alternative names
 extension.
 *
 * Output parameter:
 *   Certificate ARN - The Amazon Resource Name (ARN) of the certificate you requested.
 *
*/

public class AWSCertificateManagerExample {

   public static void main(String[] args) {

      // Retrieve your credentials from the C:\Users\name\.aws\credentials file in
 Windows
      // or the ~/.aws/credentials file in Linux.
      AWSCredentials credentials = null;
      try {
          credentials = new ProfileCredentialsProvider().getCredentials();
      }
      catch (Exception ex) {
```

```
            throw new AmazonClientException("Cannot load your credentials from file.",
 ex);
        }

        // Create a client.
        AWSCertificateManager client = AWSCertificateManagerClientBuilder.standard()
                .withRegion(Regions.US_EAST_1)
                .withCredentials(new AWSStaticCredentialsProvider(credentials))
                .build();

        // Specify a SAN.
        ArrayList<String> san = new ArrayList<String>();
        san.add("www.example.com");

        // Create a request object and set the input parameters.
        RequestCertificateRequest req = new RequestCertificateRequest();
        req.setDomainName("example.com");
        req.setIdempotencyToken("1Aq25pTy");
        req.setSubjectAlternativeNames(san);

        // Create a result object and display the certificate ARN.
        RequestCertificateResult result = null;
        try {
           result = client.requestCertificate(req);
        }
        catch(InvalidDomainValidationOptionsException ex)
        {
           throw ex;
        }
        catch(LimitExceededException ex)
        {
           throw ex;
        }

        // Display the ARN.
        System.out.println(result);

    }

}
```

前述のサンプルは、以下のような出力を作成します。

```
{CertificateArn:
 arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012}
```

検証 E メールの再送信

以下の例では、ResendValidationEmail 機能を使用する方法について説明します。

```
package com.amazonaws.samples;

import com.amazonaws.services.certificatemanager.AWSCertificateManagerClientBuilder;
import com.amazonaws.services.certificatemanager.AWSCertificateManager;
import com.amazonaws.services.certificatemanager.model.ResendValidationEmailRequest;
import com.amazonaws.services.certificatemanager.model.ResendValidationEmailResult;

import
 com.amazonaws.services.certificatemanager.model.InvalidDomainValidationOptionsException;
import com.amazonaws.services.certificatemanager.model.ResourceNotFoundException;
import com.amazonaws.services.certificatemanager.model.InvalidStateException;
import com.amazonaws.services.certificatemanager.model.InvalidArnException;
import com.amazonaws.AmazonClientException;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.regions.Regions;

/**
 * This sample demonstrates how to use the ResendValidationEmail function in the AWS
 Certificate
 * Manager service.
 *
 * Input parameters:
 *   CertificateArn - Amazon Resource Name (ARN) of the certificate request.
 *   Domain - FQDN in the certificate request.
 *   ValidationDomain - The base validation domain that is used to send email.
 *
*/

public class AWSCertificateManagerExample {

   public static void main(String[] args) {
```

```
      // Retrieve your credentials from the C:\Users\name\.aws\credentials file in
 Windows
      // or the ~/.aws/credentials file in Linux.
      AWSCredentials credentials = null;
      try {
          credentials = new ProfileCredentialsProvider().getCredentials();
      }
      catch (Exception ex) {
          throw new AmazonClientException("Cannot load your credentials from file.",
 ex);
      }

      // Create a client.
      AWSCertificateManager client = AWSCertificateManagerClientBuilder.standard()
              .withRegion(Regions.US_EAST_1)
              .withCredentials(new AWSStaticCredentialsProvider(credentials))
              .build();

      // Create a request object and set the input parameters.
      ResendValidationEmailRequest req = new ResendValidationEmailRequest();

 req.setCertificateArn("arn:aws:acm:region:account:certificate/
12345678-1234-1234-1234-123456789012");
      req.setDomain("gregpe.io");
      req.setValidationDomain("gregpe.io");

      // Create a result object.
      ResendValidationEmailResult result = null;
      try {
         result = client.resendValidationEmail(req);
      }
      catch(ResourceNotFoundException ex)
      {
         throw ex;
      }
      catch (InvalidStateException ex)
      {
         throw ex;
      }
      catch (InvalidArnException ex)
      {
         throw ex;
      }
      catch (InvalidDomainValidationOptionsException ex)
```

```
        {
            throw ex;
        }

        // Display the result.
        System.out.println(result.toString());

    }
}
```

前述のサンプルは、検証 E メールに再送信し、空のセットを表示します。

に関する問題のトラブルシューティング AWS Certificate Manager

AWS Certificate Managerを使用する際に問題が発生した場合は、以下のトピックを参照してください。

> **Note**
>
> このセクションで問題が対処されていない場合は、AWS ナレッジセンターにアクセスすることをおすすめします。

トピック

- 証明書のリクエストのトラブルシューティング
- 証明書の検証のトラブルシューティング
- マネージド証明書の更新のトラブルシューティング
- その他の問題のトラブルシューティング
- 例外処理

証明書のリクエストのトラブルシューティング

ACM 証明書のリクエスト時に問題に直面した場合は、以下のトピックを参考にしてください。

トピック

- 証明書リクエストのタイムアウト
- 証明書のリクエストの失敗

証明書リクエストのタイムアウト

ACM 証明書のリクエストは、72 時間以内に検証されない場合にタイムアウトとなります。この状態を修正するには、コンソールを開き、証明書のレコードを見つけ、そのチェックボックスをクリックし、[アクション] を選択し、[削除] を選択します。次に [アクション] および [証明書のリクエスト] を選択して、再度開始します。詳しくは、AWS Certificate Manager DNS 検証 または AWS Certificate

Manager E メール検証 を参照してください。可能な場合は、DNS 検証を使用することをお勧めします。

証明書のリクエストの失敗

リクエストが ACM に失敗し、次のいずれかのエラーメッセージが表示された場合は、推奨されるステップに従って問題を解決してください。失敗した証明書のリクエストを再送信することはできません。問題を解決した後で、新しいリクエストを送信します。

トピック

- エラーメッセージ: 利用可能な連絡先がない
- エラーメッセージ: 追加の検証が必要
- エラーメッセージ: 無効なパブリックドメイン
- エラーメッセージ: その他

エラーメッセージ: 利用可能な連絡先がない

証明書のリクエスト時に E メール検証を選択しましたが、ACM は、リクエストに記載された 1 つまたは複数のドメイン名を検証するために使用する E メールアドレスを検索できませんでした。この問題を解決するには、次のオプションがあります。

- ドメインが E メールを受信するように設定されていることを確認します。ACM のメールサーバーが ドメイン検証 E メールの送信先を特定できるように、ドメインのネームサーバーにメールエクスチェンジャレコード (MX) を保持していることが必要です。

前述した作業のうち 1 つだけを完了すれば、この問題は解決します。両方を行う必要はありません。この問題を解決したら、新しい証明書をリクエストします。

ACM からドメイン検証 E メールを受信できることを確認する方法の詳細については、「AWS Certificate Manager E メール検証」 または「検証 E メールが受信されない」を参照してください。以上の手順に従っても、引き続き「No Available Contacts (利用できる連絡先がありません)」メッセージが表示される場合には、調査を実施できるようにこの案件を AWSに報告してください。

エラーメッセージ: 追加の検証が必要

ACM は、この証明書のリクエストを処理するために追加の情報を必要とします。この状況は不正保護対策として生じることがあります。例えば、「Alexa の上位 1,000 のウェブサイト」内にランク付

けされているドメインを使用する場合です。要求された情報を提供するには、サポートセンターからサポートにお問い合わせください。サポートプランを契約していない場合は、ACM ディスカッションフォーラム に新しいスレッドを投稿してください。

> **Note**
>
> 末尾が amazonaws.com、cloudfront.net、または elasticbeanstalk.com などの Amazon が所有するドメイン名に証明書をリクエストすることはできません。

エラーメッセージ: 無効なパブリックドメイン

証明書リクエストの 1 つまたは複数のドメイン名が有効ではありません。通常の場合、リクエストするドメイン名が有効な最上位レベルのドメインでないことがその理由です。失敗したリクエスト内の誤字や脱字を修正し、リクエスト内のすべてのドメイン名が有効な最上位レベルのドメインであることを確認して、再度証明書をリクエストしてください。例えば、`example.invalidpublicdomain` に ACM 証明書をリクエストすることはできません。これは、「invalidpublicdomain」が有効な最上位レベルのドメインではないからです。この理由による失敗を引き続き受信した場合は、サポートセンターにお問い合わせください。サポートプランを契約していない場合は、ACM ディスカッションフォーラム に新しいスレッドを投稿してください。

エラーメッセージ: その他

通常、証明書リクエストの 1 つまたは複数のドメイン名に誤字や脱字がある場合にこの失敗が発生します。失敗したリクエスト内のすべての脱字や誤字を修正してから、再度証明書をリクエストしてください。この失敗メッセージが引き続き表示される場合は、サポートセンターから サポートにお問い合わせください。サポートプランを契約していない場合は、ACM ディスカッションフォーラム に新しいスレッドを投稿してください。

証明書の検証のトラブルシューティング

ACM の証明書リクエストのステータスが [Pending validation] の場合、リクエストはユーザーによるアクションを待っている状態です。リクエストを実行したときに E メール検証を選択した場合、本人または承認された担当者は検証用 E メールメッセージに対応する必要があります。これらのメッセージは、リクエストされたドメイン用の共通 E メールアドレスに送信されています。詳しくは、AWS Certificate Manager E メール検証 を参照してください。DNS 検証を選択した場合は、ACM によって作成された固有の CNAME レコードを DNS データベースに記述する必要があります。詳しくは、AWS Certificate Manager DNS 検証 を参照してください。

> **Important**
>
> ユーザーは、証明書リクエストに含まれる各ドメイン名の所有者または管理者であるという検証を受ける必要があります。E メール検証を選択した場合は、各ドメインの検証 E メールメッセージが送信されます。E メールが届かない場合は、検証 E メールが受信されない を参照してください。DNS 検証を選択した場合は、各ドメインに対して CNAME レコードを作成する必要があります。

> **Note**
>
> パブリック ACM 証明書は、Nitro Enclave に接続されている Amazon EC2 インスタンスにインストールできます。任意の Amazon EC2 インスタンスで使用する[export a public certificate] (パブリック証明書をエクスポート) することもできます。Nitro Enclave に接続されていない Amazon EC2 インスタンスでのスタンドアロンウェブサーバーのセットアップについては、「チュートリアル: Amazon Linux 2 に LAMP ウェブサーバーをインストールする」または「チュートリアル: Amazon Linux AMI を使用した LAMP ウェブサーバーのインストール」を参照してください。

E メール検証の代わりに DNS 検証を使用することをお勧めします。

検証の問題が発生した場合は、次のトピックを参照してください。

トピック

- DNS 検証の問題のトラブルシューティング
- E メール検証の問題のトラブルシューティング
- HTTP 検証に関する問題のトラブルシューティング

DNS 検証の問題のトラブルシューティング

証明書の DNS 検証で問題が発生した場合は、次のガイダンスを参照してください。

DNS トラブルシューティングの最初のステップは、以下のようなツールを使用してドメインの現在のステータスを確認することです。

- dig — Linux、Windows

- nslookup — Linux、Windows

トピック

- DNS プロバイダーがアンダースコアを禁止している
- DNS プロバイダーによって追加されたデフォルトの末尾のピリオド
- GoDaddy での DNS 検証に失敗する
- ACM コンソールで [Create record in Route 53] ボタンが表示されない
- プライベート (信頼されていない) ドメインで Route 53 検証が失敗する
- 検証は成功したが、発行または更新に失敗する
- VPN で DNS サーバーの検証が失敗する

DNS プロバイダーがアンダースコアを禁止している

DNS プロバイダーがアンダースコアで始まる CNAME 値を禁止している場合は、ACM によって提供された値からアンダースコアを削除し、ドメインを検証してください。たとえば、CNAME 値 `_x2.acm-validations.aws` を検証目的で `x2.acm-validations.aws` に変更できます。ただし、CNAME name パラメータは常にアンダースコアで始まる必要があります。

下の表の右側のいずれかの値を使用して、ドメインを検証できます。

名前	タイプ	値
`_<random value>.example.com.`	CNAME	`_<random value>.acm-validations.aws.`
`_<random value>.example.com.`	CNAME	`<random value>.acm-validations.aws.`

DNS プロバイダーによって追加されたデフォルトの末尾のピリオド

一部の DNS プロバイダーは、デフォルトで、指定した CNAME 値に末尾のピリオドを追加します。その結果、自分でピリオドを追加するとエラーが発生します。たとえば、「`<random_value>.acm-validations.aws.`」は拒否されますが、「`<random_value>.acm-validations.aws`」は受け入れられます。

GoDaddy での DNS 検証に失敗する

Godaddy およびその他のレジストリに登録されているドメインの DNS 検証は、ACM が提供している CNAME 値を変更しない限り、失敗することがあります。たとえば example.com をドメイン名として使用している場合、発行される CNAME レコードは次の形式になります。

```
NAME: _ho9hv39800vb3examplew3vnewoib3u.example.com. VALUE:
 _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws.
```

GoDaddy と互換性のある CNAME レコードを作成するには、次に示すように、[NAME] フィールドの末尾で apex ドメイン (ピリオドを含む) を切り捨てます。

```
NAME: _ho9hv39800vb3examplew3vnewoib3u VALUE:
 _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws.
```

ACM コンソールで [Create record in Route 53] ボタンが表示されない

DNS プロバイダーとして Amazon Route 53 を選択した場合、 AWS Certificate Manager は直接操作してドメインの所有権を検証できます。状況によっては、予想に反してコンソールの [Create records in Route 53] ボタンが利用できない場合があります。このような場合には以下の原因が考えられますので、確認してください。

- DNS プロバイダーとして Route 53 を使用していない。
- さまざまなアカウントを通じて ACM および Route 53 にログインしている。
- Route 53 によりホストされるゾーンでレコードを作成する IAM アクセス許可を持っていない。
- ユーザーまたは第三者によりすでにドメインが検証されている。
- ドメインがパブリックにアドレス解決できない。

プライベート (信頼されていない) ドメインで Route 53 検証が失敗する

DNS 検証中に、ACM はパブリックホストゾーンで CNAME を検索します。見つからなかった場合、72 時間後にタイムアウトし、[検証タイムアウト] ステータスになります。プライベート PKI 内の信頼されていないドメインなどの Amazon VPC プライベートホストゾーンまたは自己署名証明書では、それを使用して DNS レコードをホストすることはできません。

AWS は、 AWS Private CAサービスを通じてパブリックに信頼されていないドメインをサポートします。

検証は成功したが、発行または更新に失敗する

DNS が正しいにもかかわらず、証明書の発行が「検証保留中」で失敗した場合は、CAA レコードによって発行がブロックされていないことを確認します。詳しくは、(オプション) CAA レコードの設定 を参照してください。

VPN で DNS サーバーの検証が失敗する

VPN で DNS サーバーを検索し、ACM がそのサーバーに対して証明書の検証に失敗した場合は、サーバーがパブリックにアクセスできるかどうかを確認します。ACM DNS 検証を使用したパブリック証明書の発行では、ドメインレコードがパブリックインターネット経由で解決可能であることが必要です。

E メール検証の問題のトラブルシューティング

証明書ドメインの E メール検証で問題が発生した場合は、次のガイダンスを参照してください。

トピック

- 検証 E メールが受信されない
- E メール検証のための永続的な初期タイムスタンプ
- DNS 検証に切り替えられません

検証 E メールが受信されない

ACM に証明書をリクエストして E メール検証を選択すると、5 つの一般的な管理者アドレスにドメイン検証 E メールが送信されます。詳しくは、AWS Certificate Manager E メール検証 を参照してください。検証 E メールの受信に問題が発生した場合には、次の推奨事項を参照してください。

E メールを探す場所

ACM は、リクエストされたドメイン名に検証 E メールメッセージを送信します。これらの E メールをスーパードメインで受信する場合は、そのスーパードメインを検証ドメインとして指定することもできます。ベースとなるウェブサイトアドレスまでの任意のサブドメインは有効であり、@ の後に追加されて E メールアドレスのドメインとして使用されます。たとえば、subdomain.example.com の検証ドメインとして example.com を指定すると、admin@example.com に E メールが届きます。ACM コンソールに表示される E メールアドレスのリスト (または、CLI あるいは API から返されるリスト) にアクセスして、検証 E メールが送信されるべきアドレスを確認してください。リストを表示するには、[Validation not complete] とラベル付けされたボックスからドメイン名の横にあるアイコンをクリックします。

E メールがスパムと判断されている

スパムフォルダで検証 E メールを探します。

GMail では E メールが自動的にソートされます。

GMail を使用している場合、検証 E メールが自動的に [Updates] または [Promotions] タブにソートされている場合があります。

サポートセンターに問い合わせする

上記のガイダンスを参照した後でもドメイン検証 E メールを受信できない場合、サポート センターにアクセスしてサポートケースを作成してください。サポート契約を保持していない場合は、ACM ディスカッションフォーラムにメッセージを投稿できます。

E メール検証のための永続的な初期タイムスタンプ

証明書の最初の E メール検証リクエストのタイムスタンプは、その後の検証更新のリクエスト後も保持されます。これは、ACM 操作におけるエラーの証拠ではありません。

DNS 検証に切り替えられません

E メール検証を使用して証明書を作成した後は、DNS による検証に切り替えることはできません。DNS 検証を使用するには、証明書を削除し、DNS 検証を使用する新しい証明書を作成します。

HTTP 検証に関する問題のトラブルシューティング

証明書の HTTP 検証で問題が発生した場合は、次のガイダンスを参照してください。

HTTP トラブルシューティングの最初のステップは、以下のようなツールを使用してドメインの現在のステータスを確認することです。

- curl — Linux と Windows
- wget — Linux と Windows

トピック

- RedirectFrom と RedirectTo ロケーションのコンテンツの不一致
- CloudFront 設定の誤り
- HTTP リダイレクトの問題
- 検証タイムアウト

RedirectFrom と RedirectTo ロケーションのコンテンツの不一致

RedirectFrom ロケーションのコンテンツが RedirectTo ロケーションのコンテンツと一致しない場合、検証は失敗します。証明書内の各ドメインのコンテンツが同一であることを確認します。

CloudFront 設定の誤り

CloudFront ディストリビューションが検証コンテンツを提供するように正しく設定されていることを確認します。オリジンと動作の設定が正しく、ディストリビューションがデプロイされていることを確認します。

HTTP リダイレクトの問題

コンテンツを直接提供する代わりにリダイレクトを使用している場合は、以下の手順に従って設定を確認します。

リダイレクト設定を確認するには

1. RedirectFrom URL の値をコピーしてブラウザのアドレスバーに貼り付けます。
2. 新しいブラウザタブに RedirectTo URL を貼り付けます。
3. 両方の URL のコンテンツを比較して、正確に一致することを確認します。
4. リダイレクトが 302 ステータスコードを返すことを確認します。

検証タイムアウト

コンテンツが想定期間内に利用できないと、HTTP 検証がタイムアウトすることがあります。検証問題をトラブルシューティングするには、以下の手順に従います。

検証タイムアウトをトラブルシューティングするには

1. 次のいずれかを実行して、検証保留中のドメインを確認します。
    a. ACM コンソールを開き、証明書の詳細ページを表示します。検証保留中 としてマークされたドメインを探します。
    b. DescribeCertificate API オペレーションを呼び出して、各ドメインの検証ステータスを表示します。
2. 保留中のドメインごとに、検証コンテンツがインターネットからアクセスできることを確認します。

マネージド証明書の更新のトラブルシューティング

ACM は、ACM 証明書の有効期限が切れる前に自動的に更新を試み、ユーザーによるアクションを不要にします。でのマネージド証明書の更新 AWS Certificate Manager で問題が発生した場合は、次のトピックを参照してください。

自動ドメイン検証の準備

ACM によって証明書を自動的に更新するには、以下の条件が満たされている必要があります。

- 証明書は、ACM と統合された AWS サービスに関連付ける必要があります。ACM がサポートするリソースについては、サービスと ACM の統合 を参照してください。
- E メールで検証された証明書の場合、ACM は、証明書に記載されている各ドメインの管理者のE メールアドレスで連絡できる必要があります。試行されるE メールアドレスは、AWS Certificate Manager E メール検証 に一覧表示されます。
- DNS 検証済み証明書の場合、DNS 設定に適切な CNAME レコードが、AWS Certificate Manager DNS 検証 の説明とおりに含まれていることを確認します。
- HTTP 検証済みの証明書の場合は、リダイレクトが AWS Certificate Manager HTTP 検証 の説明に従って設定されていることを確認します。

マネージド証明書更新のエラーを処理する

証明書の有効期限 (DNS の場合は 45 日、EMAIL の場合は 45 日、プライベートの場合は 60 日) が近づくと、ACM は資格基準を満たしていれば証明書の更新を試みます。更新を正常に行うには、何らかのアクションが必要となる場合があります。詳しくは、でのマネージド証明書の更新 AWS Certificate Manager を参照してください。

E メール検証済みの証明書のマネージド証明書更新

ACM 証明書は 198 日間有効です。証明書を更新するには、ドメイン所有者によるアクションが必要です。ACM は、有効期限切れの 45 日前にドメインに関連付けられた E メールアドレスに更新通知の送信を開始します。通知には、ドメイン所有者が更新するためにクリックできるリンクが含まれています。リストされているすべてのドメインが検証されると、ACM は同じ ARN で更新された証明書を発行します。

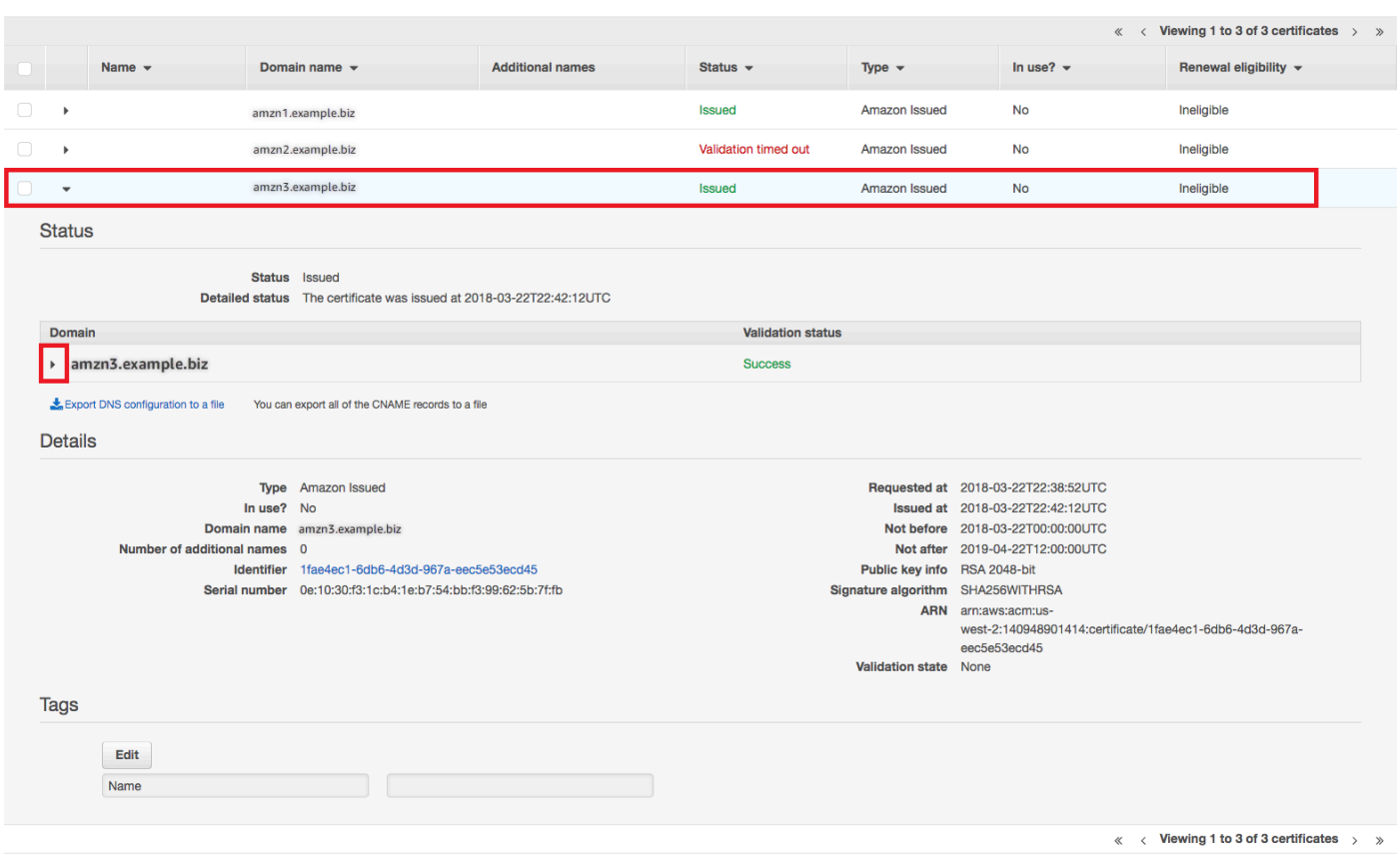
どのドメインが `PENDING_VALIDATION` 状態にあり、それらのドメインの検証プロセスを繰り返しているかを特定する手順については「E メールによる検証」を参照してください。

DNS で検証済みの証明書のマネージド型の証明書更新

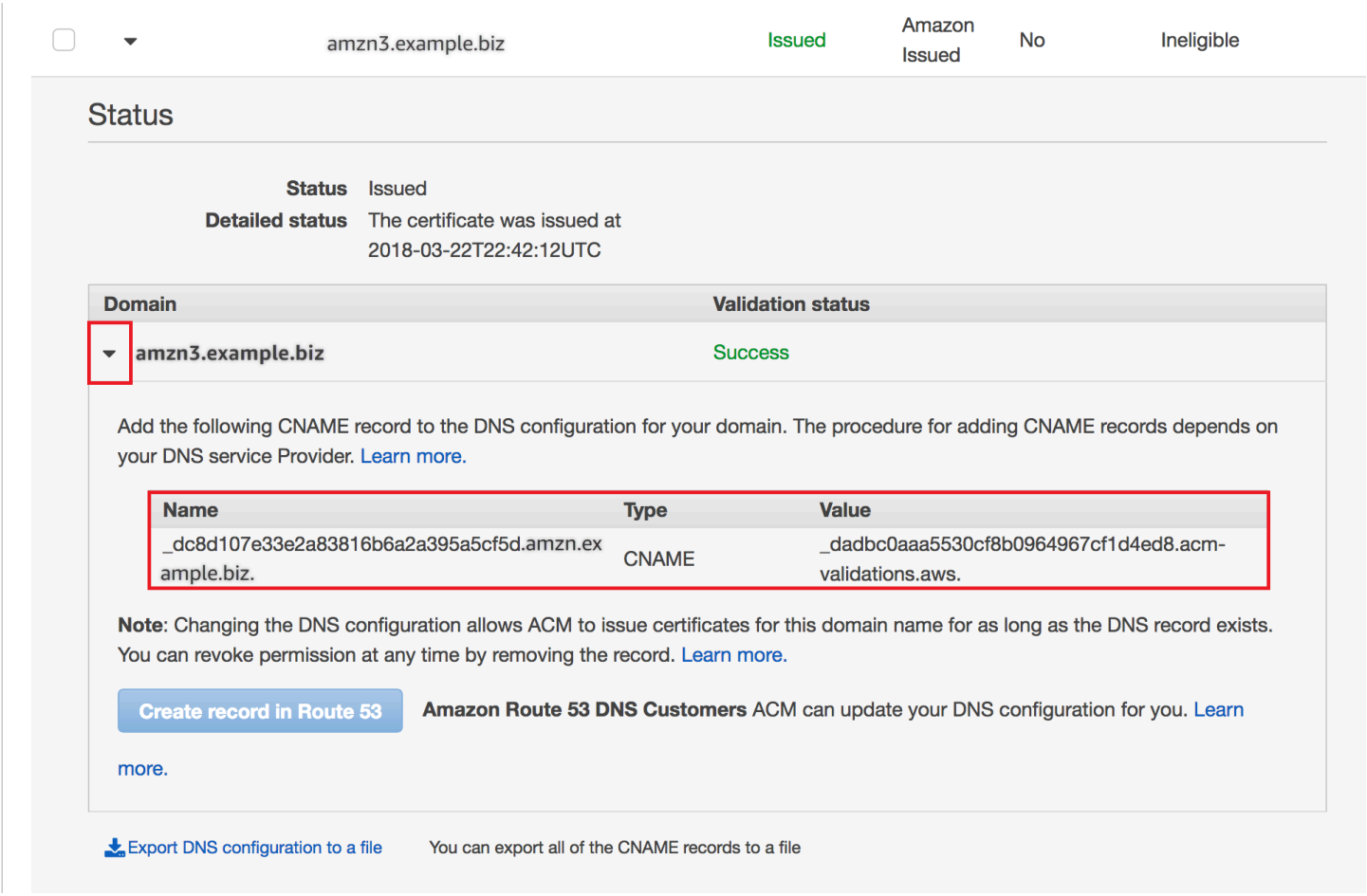
ACM は、DNS 検証済みの証明書の TLS 検証を試みません。ACM で、DNS 検証で検証した証明書の更新に失敗した場合は、DNS 設定の CNAME レコードが欠落しているか、内容が不正確であることが考えられます。失敗した場合は、ACM より、証明書が自動的に更新されない旨の通知が送信されます。

> **Important**
>
> 正しい CNAME レコードを DNS データベースに挿入する必要があります。これを行う方法については、ドメインレジストラにお問い合わせください。

ドメインの CNAME レコードを検索するには、証明書とそのドメインエントリを ACM コンソールで展開します。詳細については、以下の図を参照してください。また、ACM API の DescribeCertificate オペレーションまたはACM CLIの describe-certificate コマンドを使用して、CNAME レコードを取得することができます。詳しくは、AWS Certificate Manager DNS 検証を参照してください。



コンソールからターゲット証明書を選択します。



証明書ウィンドウを展開して、証明書の CNAME 情報を検索します。

問題が解決しない場合は、サポートセンターまでお問い合わせください。

HTTP 検証済みの証明書のマネージド型の証明書更新

ACM は、HTTP 検証済み証明書を自動的に更新しようとします。更新が失敗した場合、HTTP 検証レコードの問題が原因である可能性があります。失敗した場合、ACM は証明書を自動的に更新できなかったことを通知します。

> **Important**
>
> `RedirectFrom` ロケーションのコンテンツが証明書内の各ドメインの `RedirectTo` ロケーションのコンテンツと一致することを確認する必要があります。

ドメインの HTTP 検証情報を検索するには、証明書とそのドメインエントリを ACM コンソールで展開します。この情報は、ACM API の DescribeCertificate オペレーション、または ACM CLI の describe-certificate コマンドを使用して取得することもできます。詳しくは、AWS Certificate Manager HTTP 検証 を参照してください。

問題が解決しない場合は、サポートセンターまでお問い合わせください。

更新のタイミングについて

でのマネージド証明書の更新 AWS Certificate Manager は非同期プロセスです。これは、ステップがすぐに連続して発生しないことを意味します。ACM 証明書のすべてのドメイン名が検証されてから、ACM が新しい証明書を取得するまでに時間がかかることがあります。ACM が更新された証明書を取得した時間からその証明書が使用される AWS リソースにデプロイされる時間まで、さらなる遅延が生じる場合もあります。これによって、証明書ステータスの変更がコンソールに表示されるまでに数時間ほどかかる場合があります。

その他の問題のトラブルシューティング

このセクションでは、ACM 証明書の発行や検証に関連しない問題についてガイダンスを示します。

トピック

- CAA (Certification Authority Authorization) の問題
- 証明書のインポートの問題
- 証明書のピンニングの問題
- API Gateway の問題
- 作業証明書が予期せず失敗した場合の対処方法
- ACM のサービスにリンクされたロール (SLR) に関する問題

CAA (Certification Authority Authorization) の問題

CAA DNS レコードを使用して、Amazon 認証局 (CA) によるドメインまたはサブドメイン用の ACM 証明書の発行を指定できます。証明書の発行中に、「CAA (Certification Authority Authentication) エラーにより、1 つまたは複数のドメイン名の検証に失敗しました。」というメッセージが表示された場合は、CAA DNS レコードを調べてください。ACM 証明書リクエストが正常に検証された後でこのエラーが表示された場合は、CAA レコードを更新して、証明書を再度リクエストする必要があり

ます。CAA レコードの [value] フィールドに、次のいずれかのドメイン名が含まれている必要があります。

- amazon.com
- amazontrust.com
- awstrust.com
- amazonaws.com

CAA レコード作成についての詳細は、(オプション) CAA レコードの設定 を参照してください。

> **Note**
>
> CAA チェックを実行しない場合は、ドメインの CAA レコードを設定しないことができます。

証明書のインポートの問題

サードパーティーの証明書を ACM にインポートし、統合されたサービスと関連付けることができます。問題が発生した場合は、前提条件と証明書形式に関するトピックを確認してください。特に、以下の点に注意してください。

- X.509 バージョン 3 の SSL/TLS 証明書のみをインポートすることができます。
- 証明書は自己署名するか、認証局 (CA) によって署名できます。
- 証明書が CA によって署名されている場合は、認証機関のルートへのパスを提供する中間証明書チェーンを含める必要があります。
- 証明書が自己署名証明書である場合は、プライベートキーをプレーンテキストに含める必要があります。
- チェーンの各証明書は、先行する 1 つの証明書を直接認定する必要があります。
- 中間証明書チェーンにエンドエンティティ証明書を含めないでください。
- 証明書、証明書チェーン、およびプライベートキー (存在する場合) は PEMエンコードされている必要があります。一般に、PEMエンコーディングは、プレーンテキストのヘッダー行とフッター行で始まって終わる Base64 でエンコードされた ASCII テキストのブロックで構成されています。PEM ファイルをコピーまたはアップロードするときに、行やスペースを追加したり、その他の変更を加たりすることはできません。OpenSSL 検証ユーティリティを使用して、証明書チェーンを確認することができます。

- プライベートキー (存在する場合) は暗号化されていない必要があります。(ヒント：パスフレーズがある場合は、暗号化されます)。
- ACM と統合されたサービスは、ACM がサポートするアルゴリズムとキーサイズを使用する必要があります。証明書が機能することを確認するには、 AWS Certificate Manager ユーザーガイドと各サービスのドキュメントを参照してください。
- 統合されたサービスによる証明書のサポートは、証明書のインポート先が IAM であるか ACM であるかによって異なる場合があります。
- 証明書は、インポート時に有効である必要があります。
- すべての証明書の詳細情報が、コンソールに表示されます。ただし、デフォルトでは、keyTypesフィルターを指定せずに ListCertificates API または list-certificates AWS CLI コマンドを呼び出すと、 RSA_1024 または RSA_2048証明書のみが表示されます。

証明書のピンニングの問題

証明書を更新するために、ACM は新しいパブリックキーとプライベートキーのペアを生成します。アプリケーションが SSL ピン留めとも呼ばれる を使用して ACM 証明書を固定証明書のピンニングする場合、 が証明書 AWS を更新した後、アプリケーションはドメインに接続できない場合があります。このため、ACM 証明書をピンニングしないことをお勧めします。アプリケーションで証明書をピンニングする必要がある場合は、次の操作を実行できます。

- 保持する証明書を ACM にインポートし、アプリケーションをインポートした証明書にピンニングします。ACM はインポートした証明書にマネージド更新を提供しません。
- パブリック証明書を使用している場合は、アプリケーションを利用可能なすべての Amazon ルート証明書に固定化します。プライベート証明書を使用している場合は、アプリケーションを CA のルート証明書にピンニングします。

API Gateway の問題

エッジ最適化された API エンドポイントをデプロイすると、API Gateway は CloudFront ディストリビューションを設定します。CloudFront ディストリビューションは、アカウントではなく、API Gateway によって所有されています。ディストリビューションは、API のデプロイ時に使用した ACM 証明書にバインドされます。バインドを削除して、ACM が証明書を削除できるようにするには、証明書に関連付けられている API Gateway カスタムドメインを削除する必要があります。

リージョン API エンドポイントをデプロイすると、ユーザーに代わって API Gateway により Application Load Balancer (ALB) が作成されます。API Gateway によって所有されているロードバラ

ンサーは、ユーザーからは見えません。ALB は、API のデプロイ時に使用した ACM 証明書にバインドされます。バインドを削除して、ACM が証明書を削除できるようにするには、証明書に関連付けられている API Gateway カスタムドメインを削除する必要があります。

作業証明書が予期せず失敗した場合の対処方法

ACM 証明書を統合サービスに関連付けたのに、証明書が機能しなくなり、統合サービスがエラーを返し始めた場合は、ACM 証明書を使用するためにサービスが必要とするアクセス許可の変更が原因である可能性があります。

たとえば、Elastic Load Balancing (ELB) には、証明書のプライベートキーを復号 AWS KMS key する を復号するアクセス許可が必要です。このアクセス許可は、証明書を ELB に関連付けるときに ACM が適用するリソースベースのポリシーによって付与されます。ELB がそのアクセス許可の付与を失った場合、ELB は次に証明書キーの復号を試みるときに失敗します。

問題を調査するには、 の AWS KMS コンソールを使用して許可のステータスを確認しますhttps://console.aws.amazon.com/kms。次に、次のいずれかのアクションを実行します。

- 統合サービスに対して付与されたアクセス許可が失効したと思われる場合は、統合サービスのコンソールにアクセスし、サービスから証明書の関連付けを解除してから、再関連付けします。これにより、リソースベースのポリシーが再適用され、新しい許可が設定されます。
- ACM に付与されたアクセス許可が取り消されたと思われる場合は、https://console.aws.amazon.com/support/home#/ サポート の にお問い合わせください。

ACM のサービスにリンクされたロール (SLR) に関する問題

別のアカウントによって共有されているプライベート CA によって署名された証明書を発行すると、ACM は最初に を使用してサービスにリンクされたロール (SLR) を設定し、プリンシパルとして AWS Private CA リソースベースのアクセスポリシーとやり取りしようとします。共有 CA からプライベート証明書を発行し、SLR が設定されていない場合、ACM はその証明書を自動的に更新できません。

ACM では、アカウントに SLR が存在するかどうかを判断できないという警告が表示されることがあります。必要な `iam:GetRole` アクセス許可がすでにアカウントの ACM SLR に付与されている場合、SLR の作成後にアラートは再発しません。再発する場合は、ユーザーまたはアカウント管理者が `iam:GetRole` アクセス許可を ACM に付与するか、アカウントを ACM 管理ポリシー `AWSCertificateManagerFullAccess` に関連付けます。

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許可」を参照してください。

例外処理

AWS Certificate Manager コマンドは、いくつかの理由で失敗することがあります。各例外の詳細については、次の表を参照してください。

プライベート証明書の例外処理

によって発行されたプライベート PKI 証明書を更新しようとすると、次の例外が発生する可能性があります AWS Private CA。

> **Note**
>
> AWS Private CA は、中国 (北京) リージョンおよび中国 (寧夏) リージョンではサポートされていません。

ACM エラーコード	Comment
`PCA_ACCESS_DENIED`	プライベート CA が ACM にアクセス許可を付与していません。これにより、 AWS Private CA `AccessDeniedException` 障害コードがトリガーされます。 この問題を解決するには、 AWS Private CA CreatePermission オペレーションを使用して ACM サービスプリンシパルに必要なアクセス許可を付与します。
`PCA_INVALID_DURATION`	リクエストした証明書の有効期間が、発行元のプライベート CA の有効期間を超えています。これにより、`ValidationException` 障害コードが AWS Private CA トリガーされます。 この問題を解決するには、有効期間が適切な新しい CA 証明書をインストールします。

ACM エラーコード	Comment
PCA_INVALID_STATE	呼び出し先のプライベート CA は、リクエストした ACM オペレーションを実行するための正しい状態ではありません。これにより、InvalidStateException 障害コードが AWS Private CA トリガーされます。 問題の解決方法は次のとおりです。 • CA のステータスが CREATING である場合は、作成が完了するまで待機してから、CA 証明書をインストールします。 • CA のステータスが PENDING_CERTIFICATE である場合は、CA 証明書をインストールします。 • CA のステータスが DISABLED である場合は、ACTIVE ステータスに更新します。 • CA のステータスが DELETED である場合は、復元します。 • CA のステータスが EXPIRED である場合は、新しい証明書をインストールします。 • CA のステータスが FAILED であり、問題を解決できない場合は、サポート にお問い合わせください。
PCA_LIMIT_EXCEEDED	プライベート CA が発行クォータに達しています。これにより、LimitExceededException 障害コードが AWS Private CA トリガーされます。このヘルプに進む前に、リクエストを繰り返してください。 エラーが引き続き発生する場合は、サポート に連絡してクォータの引き上げをリクエストしてください。

ACM エラーコード	Comment
PCA_REQUEST_FAILED	ネットワークエラーまたはシステムエラーが発生しました。これにより、RequestFailedException 障害コードが AWS Private CA トリガーされます。このヘルプに進む前に、リクエストを繰り返してください。 エラーが引き続き発生する場合は、サポートまでお問い合わせください。
PCA_RESOURCE_NOT_FOUND	プライベート CA が完全に削除されています。これにより、ResourceNotFoundException 障害コードが AWS Private CA トリガーされます。正しい ARN を使用していることを確認してください。これが失敗すると、この CA は使用できません。 この問題を解決するには、新しい CA を作成します。
SLR_NOT_FOUND	別のアカウントに存在するプライベート CA によって署名された証明書を更新するには、ACM では、証明書が存在するアカウントに、サービスにリンクされたロール (SLR) が必要です。削除した SLR を再作成する必要がある場合は、「ACM の SLR の作成」を参照してください。

クォータ

以下の AWS Certificate Manager (ACM) サービスクォータは、各アカウントごとに各 AWS AWS リージョンに適用されます。

調整できるクォータを確認するには、AWS 全般のリファレンスガイドの「ACM クォータテーブル」を参照してください。クォータの引き上げをリクエストするには、サポート センターにケースを作成します。

一般的なクォータ

Item	デフォルトのクォータ
ACM 証明書の数 期限切れの証明書と失効した証明書は、引き続きこの合計にカウントされます。 から CA によって署名された証明書は、この合計にはカウント AWS Private CA されません。	2500
1 年間の ACM 証明書の数 (過去 365 日間) 年間でリージョンおよびアカウントごとに、ACM 証明書のクォータを最大 2 倍に増やすことをリクエストできます。たとえば、クォータが 2,500 の場合は、年間でリージョンおよびアカウントごとに、最大 5,000 の ACM 証明書をリクエストできます。一度に所有できる証明書は 2,500 のみです。年間で 5,000 の証明書をリクエストするには、同年中に 2,500 の証明書を削除してクォータ内に収まるようにする必要があります。2,500 を超える証明書が必要な場合は、その都度、サポート センターに連絡する必要があります。	5,000

Item	デフォルトのクォータ
から CA によって署名された証明書は、この合計にはカウント AWS Private CA されません。	
インポートされた証明書の数	2,500
1 年間にインポートされた証明書の数 (過去 365 日間)	5,000
ACM 証明書ごとのドメイン名の数 ACM 証明書ごとのドメイン名数のデフォルトのクォータは 10 です。クォータはこれより大きい場合もあります。 送信する最初のドメイン名は、証明書のサブジェクト共通名 (CN) として含まれます。すべての名前は、サブジェクト代替名拡張子に含まれています。 最大 100 のドメイン名をリクエストすることができます。クォータの引き上げをリクエストするには、ACM サービスのために Service Quotas コンソールでリクエストを作成します。ただし、E メール検証を使用する場合は、ケースを作成する前に、ドメイン名を追加するほど多くの管理作業が必要になる状況を理解してください。詳しくは、ドメイン検証 を参照してください。 ACM 証明書ごとのドメイン名のクォータは、ACM が提供する証明書にのみ適用されます。このクォータは、ACM にインポートする証明書には適用されません。以下のセクションは、ACM 証明書にのみ適用されます。	10

Item	デフォルトのクォータ
プライベート CA の数 ACM は AWS Private Certificate Authority () と統合されていますAWS Private CA。ACM コンソール AWS CLI、または ACM API を使用して、 がホストする既存のプライベート認証機関 (CA) からプライベート証明書をリクエストできます AWS Private CA。これらの証明書は ACM 環境で管理され、ACM によって発行されたパブリック証明書と同じ制限が適用されます。詳細については、「でプライベート証明書をリクエストする AWS Certificate Manager」を参照してください。スタンドアロン AWS Private CA サービスを使用してプライベート証明書を発行することもできます。詳細については「プライベート証明書の発行」を参照してください。 削除されたプライベート CA は、復元期間の終了時までクォータにカウントされます。詳細については、「プライベート CA の削除」を参照してください。	200
CA あたりのプライベート証明書の数 (有効期間)	1,000,000

API レートクォータ

次のクォータは、各リージョンと各アカウントの ACM API に適用されます。ACM は、API オペレーションに合わせて、さまざまなクォータで API リクエストをスロットリングします。スロットリングとは、1 秒あたりのリクエスト数に関するオペレーションのクォータを超えたことにより、ACM が有効なリクエストを拒否することを意味します。リクエストがスロットリングされると、ACM は `ThrottlingException` エラーを返します。以下の表に示しているのは、各 API オペレーションと、ACM がそのオペレーションに合わせてリクエストをスロットリングするクォータです。

> **Note**
>
> 次の表に示す API アクションに加えて、ACM は `IssueCertificate` から外部 AWS Private CAアクションを呼び出すこともできます。`IssueCertificate` に関する最新のレートクォータ情報については、 AWS Private CAのエンドポイントとクォータを参照してください。

各 ACM API オペレーションの 1 秒あたりのクォータリクエスト数

API コール	1 秒あたりのリクエスト
`AddTagsToCertificate`	5
`DeleteCertificate`	10
`DescribeCertificate`	10
`ExportCertificate`	10
`GetAccountConfiguration`	1
`GetCertificate`	10
`ImportCertificate`	1
`ListCertificates`	8
`ListTagsForCertificate`	10
`PutAccountConfiguration`	1
`RemoveTagsFromCertificate`	5
`RenewCertificate`	5
`RequestCertificate`	5
`ResendValidationEmail`	1
`UpdateCertificateOptions`	5

詳細については、「AWS Certificate Manager API リファレンス」を参照してください。

ドキュメント履歴

次の表は、2018 AWS Certificate Manager 年以降の のドキュメントリリース履歴を示しています。

変更	説明	日付
パブリック証明書の有効期間の更新	パブリック ACM 証明書は 198 日間有効になり、13 か月 (395 日) から削減されました。この更新により、ACM が発行したパブリック証明書は、CA/B フォーラムの今後の証明書の有効期間要件に準拠します。新しい要件に従い、2026 年 3 月 15 日以降に発行されたパブリック証明書の最大有効期間は 200 日である必要があります。パブリック証明書の更新期間は、有効期限の 45 日前に更新されました。プライベート証明書は 13 か月間 (395 日間) 有効です。	2026 年 2 月 18 日
証証明書の再インポートの変更	ACM では、以前の証明書に ClientAuth EKU がない場合にのみ、証明書を同じ ARN に再インポートできます。これは、証明機関がChrome のルートプログラム要件に準拠するために、ClientAuth EKU で証明書を発行しなくなるという業界の変化に対応できます。	2025 年 10 月 22 日
証明書発行に関する注意事項を追加	ACM 証明書の概念トピックに、TLS ウェブクライアント	2025 年 7 月 23 日

	認証拡張機能を使用した ACM 証明書発行への変更を詳述するメモを追加しました。	
認証拡張機能への参照を削除	サンプル証明書から TLS ウェブクライアント認証拡張機能への参照を削除しました。	2025 年 7 月 3 日
AWS Certificate Manager エクスポート可能なパブリック証明書	ACM パブリック証明書をエクスポートできます。	2025 年 6 月 17 日
ACM が CloudFront による HTTP 検証をサポート	ACM は、CloudFront ディストリビューションの証明書を発行する際に、ドメイン所有権の検証の HTTP 検証をサポートするようになりました。	2025 年 4 月 24 日
E メールエクスチェンジャ (MX) の E メール検証の廃止	ACM コンソールは E メールエクスチェンジャ (MX) をサポートしなくなりました。	2024 年 7 月 11 日
アカウントレベルの分離に関するベストプラクティスの追加	可能な限り、ポリシーでアカウントレベルの分離を使用します。不可能な場合は、アカウントレベルまたはポリシーの暗号化コンテキスト条件キーを使用してアクセス許可を制限できます。	2024 年 6 月 11 日
WHOIS E メール検証の今後の廃止	2024 年 6 月以降の WHOIS E メール検証の廃止に関する注意を追加しました。	2024 年 2 月 5 日

条件キーサポートの追加	ACM 証明書をリクエストする際に、IAM 条件キーをサポートするようにしました。サポートされている条件のリストについては、「https://docs.aws.amazon.com/acm/latest/userguide/acm-conditions.html#acm-conditions-supported」を参照してください。	2023 年 8 月 24 日
ECDSA サポートの追加	パブリック ACM 証明書を要求する際に、楕円曲線デジタル署名アルゴリズム (ECDSA) をサポートするようにしました。サポートされているキーアルゴリズムのリストについては、https://docs.aws.amazon.com/acm/latest/userguide/acm-certificate.html#algorithms を参照してください。	2022 年 11 月 8 日
新しい CloudWatch イベント	ACM 証明書の有効期限切れ、ACM 証明書が利用可能、ACM 証明書の更新アクションが必要、というイベントを追加しました。サポートされる CloudWatch イベントのリストについては、https://docs.aws.amazon.com/acm/latest/userguide/cloudwatch-events.html を参照してください。	2022 年 10 月 27 日

インポート用のキーアルゴリズムタイプの更新	ACM にインポートされた証明書には、追加の RSA アルゴリズムと楕円曲線アルゴリズムを使用するキーがある場合があります。現在サポートされているキーアルゴリズムのリストについては、https://docs.aws.amazon.com/acm/latest/userguide/import-certificate-prerequisites.html を参照してください。	2021 年 7 月 14 日
別のチャプターとして「モニタリングとログ」を推進する	モニタリングとロギングに関するドキュメントを、独自のチャプターに移動しました。この変更は、CloudWatch メトリクス、CloudWatch Events/EventBridge、および CloudTrail を対象としています。詳しくは、https://docs.aws.amazon.com/acm/latest/userguide/monitoring-and-logging.html を参照してください。	2021 年 3 月 23 日
CloudWatch メトリクスおよびイベントのサポートを追加	DaysToExpiry メトリクス、イベント、サポート API が追加されました。詳しくは、https://docs.aws.amazon.com/acm/latest/userguide/cloudwatch-metrics.html または https://docs.aws.amazon.com/acm/latest/userguide/cloudwatch-events.html を参照してください。	2021 年 3 月 3 日

クロスアカウントサポートの追加	からプライベート CAs を使用するためのクロスアカウントサポートを追加しました AWS Private CA。詳細については、「https://docs.aws.amazon.com/acm/latest/userguide/ca-access.html」を参照してください。	2020 年 8 月 17 日
リージョンサポートの追加	AWS 中国 (北京および寧夏) リージョンのリージョンサポートを追加しました。サポートされているリージョンの詳細なリストについては、https://docs.aws.amazon.com/general/latest/gr/rande.html#acm-pca_region を参照してください。	2020 年 3 月 4 日
更新ワークフローテストの追加	ACM マネージド更新ワークフローの設定を手動でテストできるようになりました。詳細については、「ACM のマネージド更新設定のテスト」を参照してください。	2019 年 3 月 14 日
証明書の透明性ログ記録は現在デフォルト	ACM パブリック証明書を証明書の透明性ログにデフォルトで発行する機能が追加されました。	2018 年 4 月 24 日

の起動 AWS Private CA	ACM Private Certificate Manager (CM) と の拡張機能を起動しました。 AWS Certificate Manager これにより、ユーザーはプライベートデジタル証明書の発行と取り消しのための安全なマネージドインフラストラクチャを確立できます。詳細については「AWS プライベート証明書機関」を参照してください。	2018 年 4 月 4 日
証明書の透明性ログ記録	証明書の透明性ログ記録がベストプラクティスに追加されました。	2018 年 3 月 27 日

次の表は、2018 AWS Certificate Manager 年以前の のドキュメントリリース履歴を示しています。

変更	説明	リリース日
新しいコンテンツ	DNS 検証を AWS Certificate Manager DNS 検証 に追加しました。	2017 年 11 月 21 日
新しいコンテンツ	新しい Java コード例を SDK for Java AWS Certificate Manager で を使用する に追加しました。	2017 年 10 月 12 日
新しいコンテンツ	CAA レコードに関する情報を (オプション) CAA レコードの設定 に追加しました。	2017 年 9 月 21 日
新しいコンテンツ	.IO ドメインについての情報を に関する問題のトラブルシューティング AWS Certificate Manager に追加しました。	2017 年 7 月 07 日

変更	説明	リリース日
新しいコンテンツ	証明書の再インポートについての情報を 証明書の再インポート に追加しました。	2017 年 7 月 07 日
新しいコンテンツ	証明書のピニングについての情報を ベストプラクティス と に関する問題のトラブルシューティング AWS Certificate Manager に追加しました。	2017 年 7 月 07 日
新しいコンテンツ	CloudFormation を に追加しましたサービスと ACM の統合。	2017 年 5 月 27 日
更新	詳細情報を クォータ に追加しました。	2017 年 5 月 27 日
新しいコンテンツ	の Identity and Access Management AWS Certificate Manager に関するドキュメントを追加しました。	2017 年 4 月 28 日
更新	検証 E メールの送信先を示すグラフィックを追加しました。「AWS Certificate Manager E メール検証」を参照してください。	2017 年 4 月 21 日
更新	ドメインの E メールの設定についての情報を追加しました。AWS Certificate Manager E メール検証 を参照してください。	2017 年 4 月 6 日

変更	説明	リリース日
更新	証明書更新ステータスをコンソールで確認するための情報が追加されました。「証明書の更新ステータスの確認」を参照してください。	2017 年 3 月 28 日
更新	Elastic Load Balancing の使用についてドキュメントを更新しました。	2017 年 3 月 21 日
新しいコンテンツ	AWS Elastic Beanstalk と Amazon API Gateway のサポートが追加されました。「サービスと ACM の統合」を参照してください。	2017 年 3 月 21 日
更新	マネージド証明書の更新 に関するドキュメントが更新されました。	2017 年 2 月 20 日
新しいコンテンツ	インポートされた証明書 に関するドキュメントを追加しました。	2016 年 10 月 13 日
新しいコンテンツ	ACM アクション AWS CloudTrail のサポートが追加されました。「での CloudTrail の使用 AWS Certificate Manager」を参照してください。	2016 年 3 月 25 日
新規ガイド	このリリースでは AWS Certificate Managerを導入しています。	2016 年 1 月 21 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。