[AWS Black Belt Online Seminar] AWS Certificate Manager

サービスカットシリーズ

Partner Solutions Architect 清水毅

2018/12/19

AWS 公式 Webinar
https://amzn.to/JPWebinar

過去資料
https://amzn.to/JPArchive

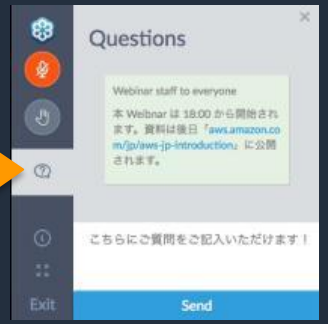
AWS Black Belt Online Seminar とは

「サービス別」「ソリューション別」「業種別」のそれぞれのテーマに分かれて、アマゾン ウェブ サービス ジャパン株式会社が主催するオンラインセミナーシリーズです。

質問を投げることができます！

- 書き込んだ質問は、主催者にしか見えません
- 今後のロードマップに関するご質問は
  お答えできませんのでご了承下さい

① 吹き出しをクリック
② 質問を入力
③ Sendをクリック



Twitter ハッシュタグは以下をご利用ください
#awsblackbelt

内容についての注意点

- 本資料では2018年12月19日時点のサービス内容および価格についてご説明しています。最新の情報はAWS公式ウェブサイト(http://aws.amazon.com)にてご確認ください。

- 資料作成には十分注意しておりますが、資料内の価格とAWS公式ウェブサイト記載の価格に相違があった場合、AWS公式ウェブサイトの価格を優先とさせていただきます。

- 価格は税抜表記となっています。日本居住者のお客様が東京リージョンを使用する場合、別途消費税をご請求させていただきます。

- AWS does not offer binding price quotes. AWS pricing is publicly available and is subject to change in accordance with the AWS Customer Agreement available at http://aws.amazon.com/agreement/. Any pricing information included in this document is provided only as an estimate of usage charges for AWS services based on certain information that you have provided. Monthly charges will be based on your actual use of AWS services, and may vary from the estimates provided.

自己紹介

清水 毅（しみず つよし）

所属：

SaaS Partner Solutions Architect

好きなAWSサービス：

AWS Certificate Manager、Amazon GuardDuty

趣味：

Aerial Photography by a Drone（国交省航空局日本全国包括許可）

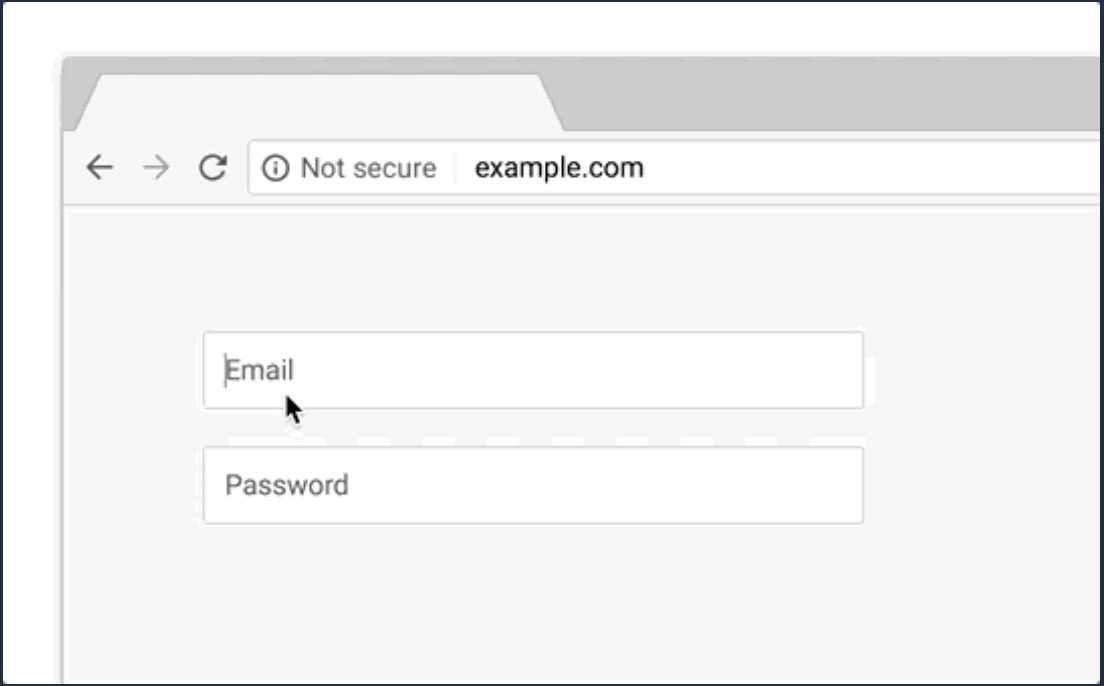
Photographed at Okinawa Oodomari Beach

Agenda

- はじめに
  - TLS
  - TLSサーバー証明書
- AWS Certificate Manager （ACM）
  - 特徴
  - デモ
- AWS Certificate Manager （ACM） Private CA
  - 特徴
  - ユースケース
- まとめ

はじめに

2018年10月 ChromeにてHTTPサイトが警告表示



“HTTP pages will be marked as affirmatively "Not Secure" using red color and the non-secure icon in the URL bar if the user **interacts with any input field**.”

（HTTPページでの入力フィールドを利用するとURLバーを赤色に表示）

https://www.chromium.org/Home/chromium-security/marking-http-as-non-secure

Takes effect: October 2018 (Chrome 70)
Announcement: Evolving Chrome's security indicators (May 17, 2018)

TLS（SSL）

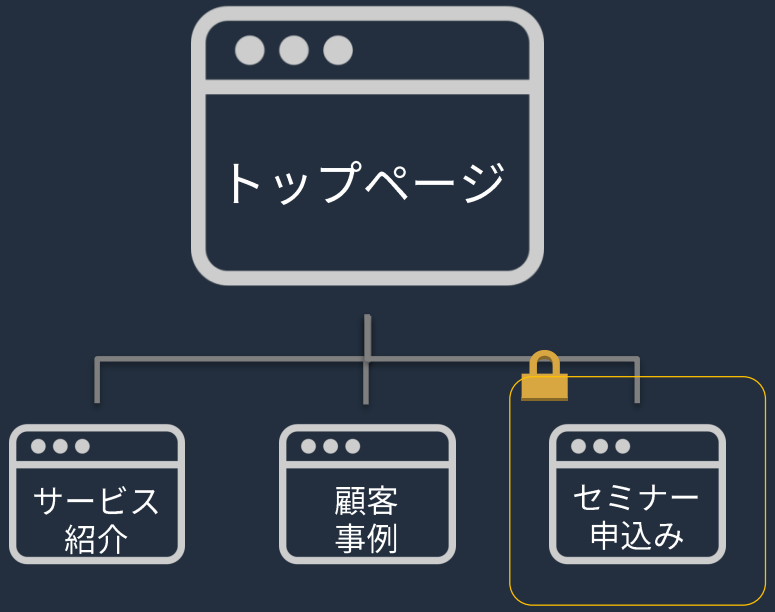
- TLS（SSL）とは
  - SSL（Secure Sockets Layer）/ TLS（Transport Layer Security）トランスポート層のプロトコルであり、データを暗号化して送受信するためのもの（HTTPやFTPの通信をセキュアに）

- TLSの成り立ち
  - SSLのバージョンアップにより、SSL3.0をもとにTLS1.0が制定（1999年）
  - 実際にはTLSを利用していても、SSLの名称が普及しているため、一般にSSLや、SSL/TLSと記述することがある
  - TLS1.3が公開（2018年）

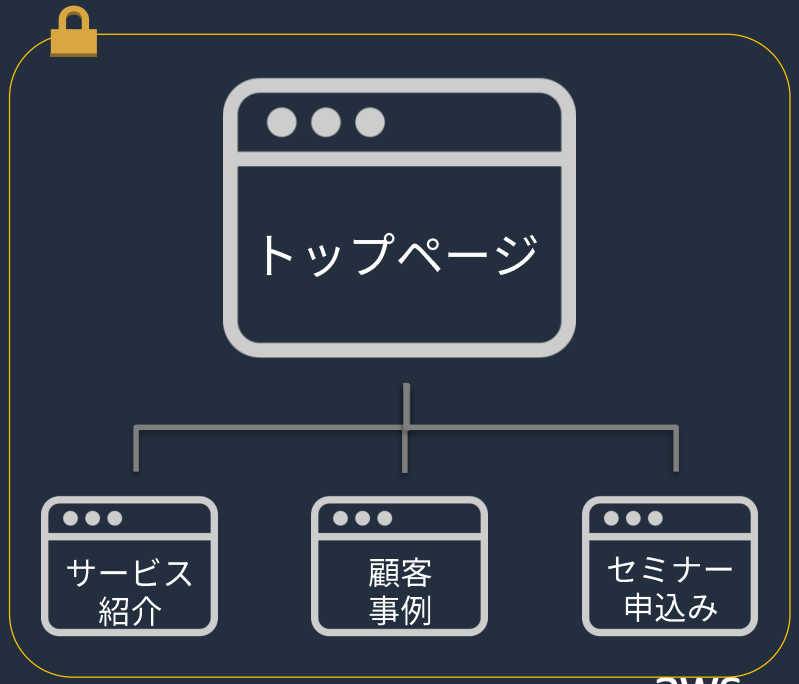
- 推奨バージョン
  - 現在、SSL3.0や初期のTLSバージョンは脆弱なため非推奨となり、より安全なバージョンの利用が推奨される

“常時TLS（SSL）”が当たり前の時代に

一部TLS

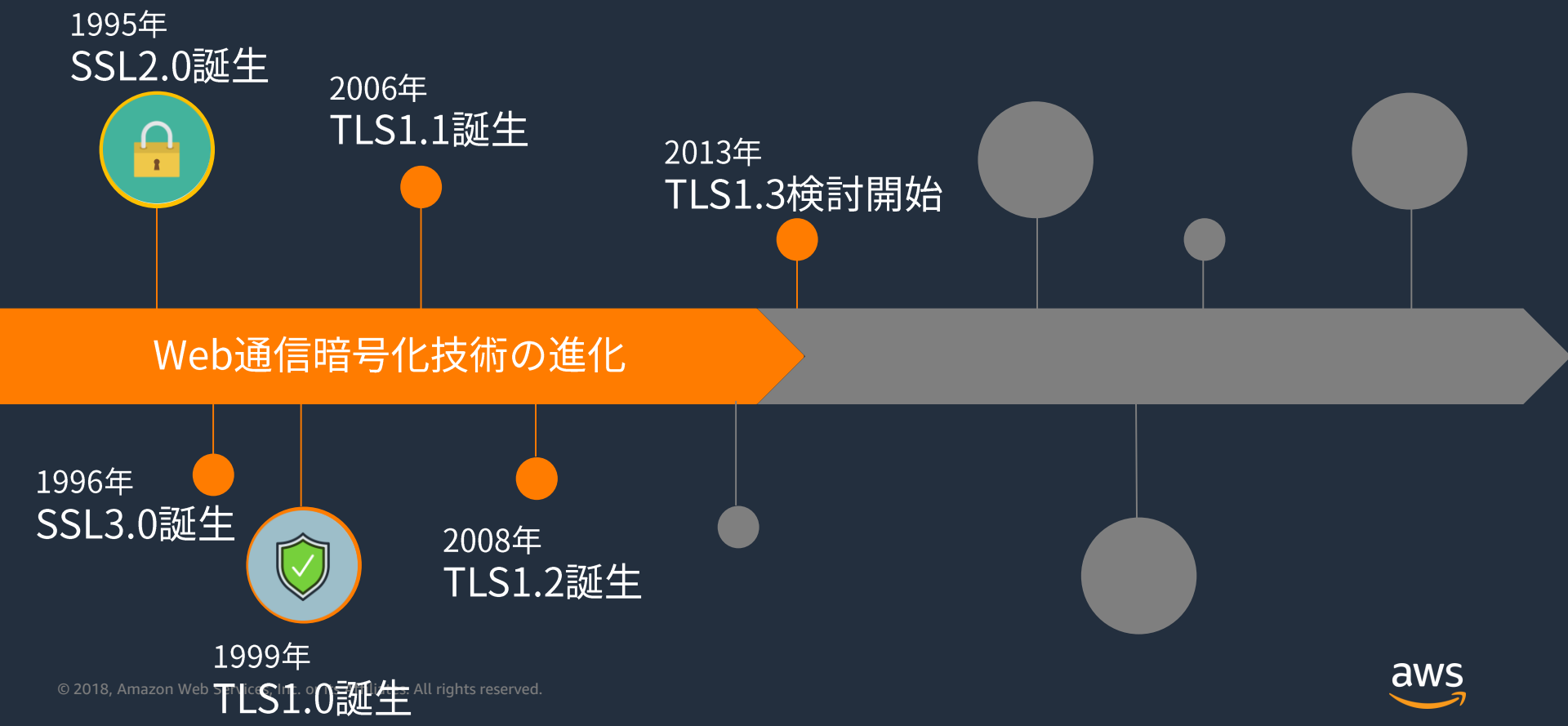


常時TLS



TLS（SSL）の歴史

引用：Amazon CloudFront TLS/SSL セミナー
TLS/SSLが加速している背景 （2016/8/4）
https://www.slideshare.net/HayatoKiriyama/amazon-cloudfront-tlsssl-seminar-20160804



脆弱性の歴史

1995年
SSL2.0誕生

2006年
TLS1.1誕生

2013年
TLS1.3検討開始

2014年4月
Heartbleed脆弱性

2015年
FREAK脆弱性

2016年3月
DROWN脆弱性

Web通信暗号化技術の進化

脆弱性との戦い

1996年
SSL3.0誕生

1999年
TLS1.0誕生

2008年
TLS1.2誕生

2011年
BEAST脆弱性

2014年9月
POODLE脆弱性

2018年
TLS1.3誕生

業界・ベンダーによる強制力の高まり

2014年4月
Heartbleed脆弱性

2015年
FREAK脆弱性

2016年3月
DROWN脆弱性

2016年7月
ATS必須化

2017年6月30日
TLS1.2必須化

脆弱性との戦い

業界による強制

2011年
BEAST脆弱性

2014年9月
POODLE脆弱性

2014年9月
Google
HTTPS優先
インデックス

2016年4月
PCI DSS v3.2公開

2016年8月
google.com
HTTPS強制

2018年6月
PCI DSS
TLS1.2移行期限

TLS（SSL）の進化

		SSL2.0	SSL3.0	TLS1.0	TLS1.1	TLS1.2
攻撃方法に対する耐性	ダウングレード攻撃 (最弱暗号アルゴリズムを強制)	脆弱	安全	安全	安全	安全
	バージョンロールバック攻撃 (SSL2.0を強制)	脆弱	安全	安全	安全	安全
	CBCモード時の脆弱性攻撃 (BEAST/POODLE攻撃など)	脆弱	脆弱	パッチ適用要	安全	安全
利用可能な暗号アルゴリズム	128ビットブロック暗号(AES, Camellia)	不可	不可	可	可	可
	認証付暗号利用モード(GCM, CCM)	不可	不可	不可	不可	可
	楕円曲線暗号	不可	不可	可	可	可
	SHA-2ハッシュ関数(SHA-256, SHA-384)	不可	不可	不可	不可	可

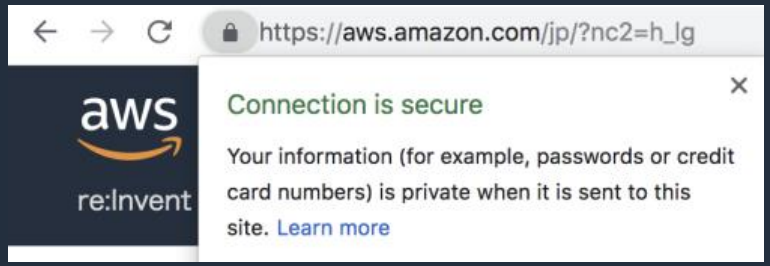
SSL/TLS暗号設定ガイドライン v2.0, IPA https://www.ipa.go.jp/security/ipg/documents/ipa-cryptrec-gl-3001-2.0.pdf

aws

TLSサーバー証明書

なぜTLSサーバー証明書を使うのか？

- 脅威
  - なりすまし、盗聴、改ざん、否認
- 対策
  - 特定：TLS通信を行うウェブサイト、アプリケーション、その他リソースの特定
  - 暗号化：安全なネットワーク通信
  - 視認：ブラウザユーザーに鍵アイコンを見せる



PKI（Public Key Infrastructure）

PKI（ Public Key Infrastructure・公開鍵暗号基盤）

- 公開鍵暗号技術と電子署名を使ってインターネット通信を安全にする基盤。（TLSサーバー証明書にも利用されている）

公開鍵暗号技術

- ペア（キーペア）になった公開鍵と秘密鍵を用いて暗号化と復号化を行えるようにする

電子署名

- 秘密鍵で暗号化されたデータを公開鍵で復号化できることが秘密鍵を保有している本人であると特定できる

電子証明書とルートCA

電子証明書
- PKIにおける鍵を持っている人を証明するもの　例）TLSサーバー証明書

CA
- インターネットの世界では、電子証明書を発行する組織を認証局（CA、Certificate AuthorityもしくはCertification Authority）と呼ぶ

ルートCA
- 証明書を発行するCAは、上位のCAに認証をしてもらうことで、その正当性を表明する。階層構造の最上位に位置するCAをルートCAと呼ぶ

ルートCA自身の証明
- ルートCAは、自分自身に対して電子証明書を発行する（厳格な審査に基づき信頼されたルートCAの証明書はブラウザ等から確認ができる）

証明書発行の流れ

証明書の発行

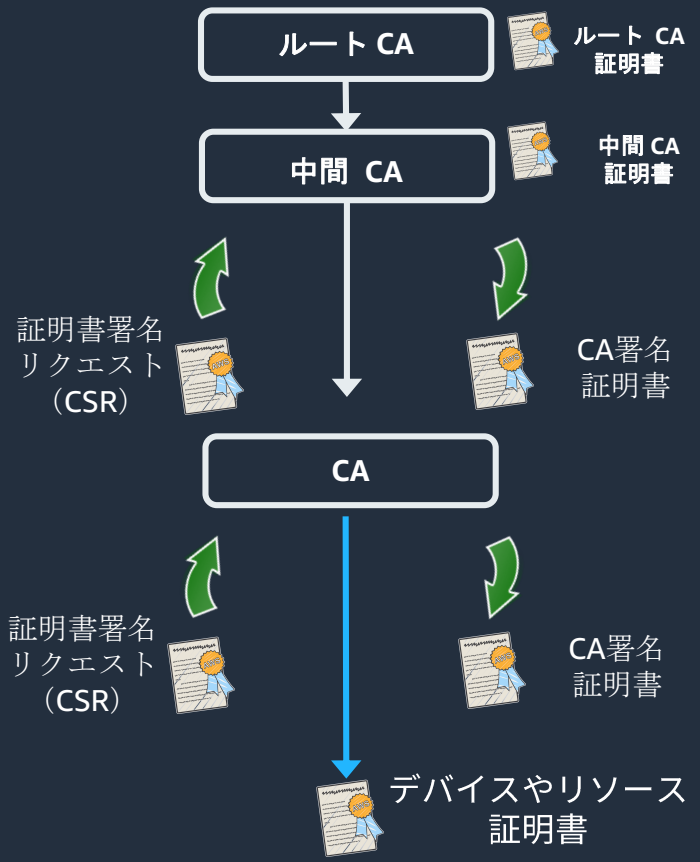
- 単一の機関がすべての証明を行うことは現実的ではない。実際には必要な相手にあわせて様々な組織が証明書を発行している

トラストチェーン

- 証明書を発行した組織の適切性を階層的に確認していくことで、最終的に正しさを確認できる（トラストチェーン、証明書チェーンと呼ぶ）

証明書の正当性確認

- 相手の証明書の正当性を確認するには、適切な組織によって証明されているかを確認することになる



認証の種類

自己署名証明

- 自分で署名し、TLSサーバー証明書を発行

企業内証明

- 社内用にプライベートCAを作成し、TLSプライベート証明書を発行

ドメイン認証（DV）

- ドメインの所有・管理していることを確認、TLSサーバー証明書の発行
  （発行されたTLS証明書の属性には組織情報が記載されない）

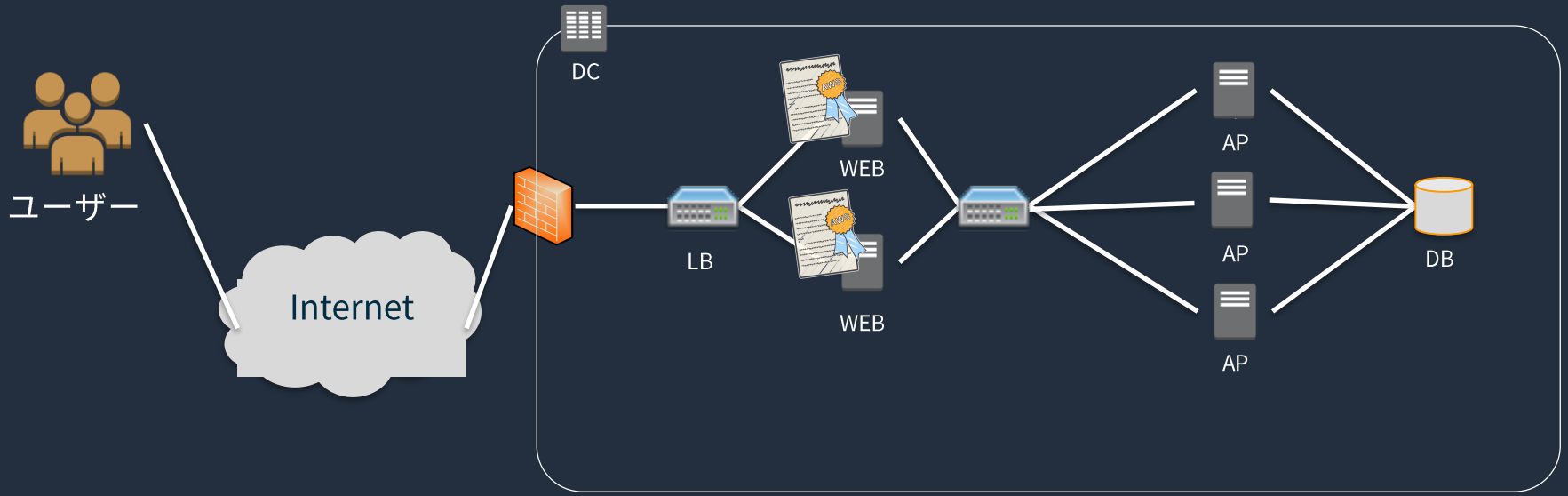
組織認証（OV）

- 組織情報の審査を経てから発行
  （サイト運営者のなりすましを防ぎ、組織情報がTLS証明書に記載）

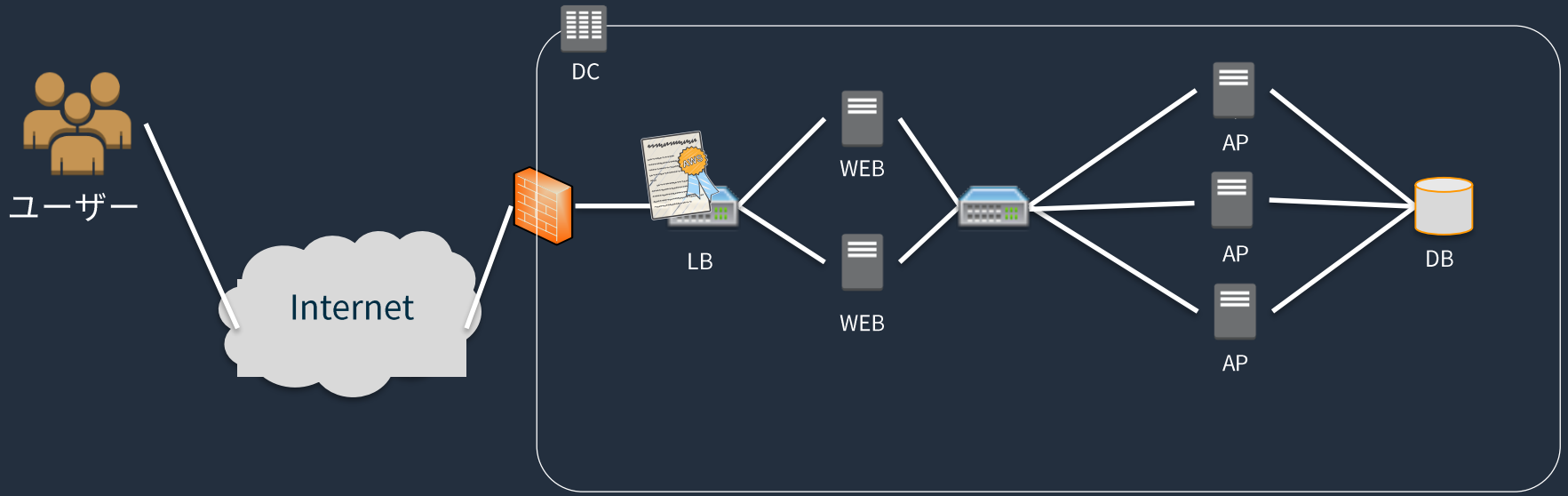
拡張認証（EV）

- OVよりも厳格な審査を経て発行

TLSサーバー証明書配置① - Web Server



TLSサーバー証明書配置② - Load Balancer



TLSサーバー証明書の運用上の課題

- 証明書にかかるコスト
  - サーバーやドメインの数だけ証明書を用意
  - クラウドライセンス（CAによってライセンスが異なる）
- 運用にかかる負荷
  - 1台ごとにCSR生成、証明書セットアップする手間
  - CAにて証明書を発行する手間
  - 更新忘れ
- SEOの判断基準にも
  - 利用しない場合に検索の順位低下も

より低コストかつ便利にTLSサーバー証明書の運用をしたい！

AWS Certificate Manager

AWS Certificate Manager （ACM）

ACMを使用すると、AWSクラウド上で
TLSサーバー証明書のプロビジョニング、管理、展開、更新が容易

ACM の機能

- 発行機能
    - パブリックDV証明書の発行（DNSとEメールによるドメイン検証）
    - プライベート 証明書の発行
- 展開更新機能
    - 発行した証明書の展開
    - インポートした証明書の展開
- 対象サービス：
    - Elastic Load Balancing （ELB）※ALB/CLB対象
    - Amazon CloudFront ディストリビューション
    - Amazon API Gateway 上のAPIのカスタムドメイン
    - EC2やオンプレサーバー等の内部リソース（プライベート証明書）
- コンソール、API / SDK、CLI

ACM のメリット

- Webサイトとアプリケーションの保護
- 迅速かつ容易に証明書を提供
- 安全な鍵管理
- AWSクラウドで証明書を集中管理
- サードパーティ証明書のインポートと展開
- 他のAWSクラウドサービスと統合
- AWSが 面倒な処理を肩代わり
    - キーペアとCSR （Certificate Signing Request）生成
    - マネージドな更新と展開
- ACMに統合されるサービスで利用の証明書は無料

ACM 証明書 ライフサイクルマネジメント

- ACMはパブリック・プライベート証明書ともに1つのインターフェースでの管理
- 「ACMに統合されるサービス（CloudFront、ELB、API Gateway）」はマネージド証明書更新と自動適用
- ベストプラクティスを用いたプライベートキーの安全な保管
- コードからプライベート証明書のエクスポートや適用をAPI呼び出し可能

ACM で発行できる証明書の種類

	ACMパブリック証明書	ACMプライベート証明書
検証方法	ドメイン所有検証が必要 （DNS検証かEメール検証）	外部検証は不要
アプリケーション, ブラウザ, OSからの信頼方法	標準で信頼される	ルートCAをトラストストアに追加する必要があり*
証明書のサブジェクト	有効なパブリックDNS名のみ	パブリックまたはプライベートのDNS名、または有効なX.509サブジェクト/ SAN
ユースケース	公開サーバー	内部リソース（サーバー、クライアント、コンテナ、IoT他）
ルートCA	パブリックルートCA	プライベートルートCA

*既存のプライベートルートCAご利用いただくか、プライベートルートCAを構築頂く必要あり

ACM デフォルトサービス上限

項目	デフォルトの制限
ACM 証明書の数	1000
1 年間の ACM 証明書の数（過去 365 日間）	アカウントの制限の 2 倍
インポートされた証明書の数	1000
1 年間にインポートされた証明書の数（過去 365 日間）	アカウントの制限の 2 倍
ACM 証明書ごとのドメイン名の数	10
プライベート CA の数	10
CA あたりのプライベート証明書の数	50,000

上限緩和についてはAWS Supportまでお問い合わせください。
https://aws.amazon.com/jp/contact-us/

ACMの認証タイプ

自己署名証明

- 自分で署名し、TLSサーバー証明書を発行

企業内証明

- 社内用にプライベートCAを作成し、TLSプライベート証明書を発行

ドメイン認証（DV）

- ドメインの所有・管理していることを確認、TLSサーバー証明書の発行
  （発行されたTLS証明書の属性には組織情報が記載されない）

組織認証（OV）

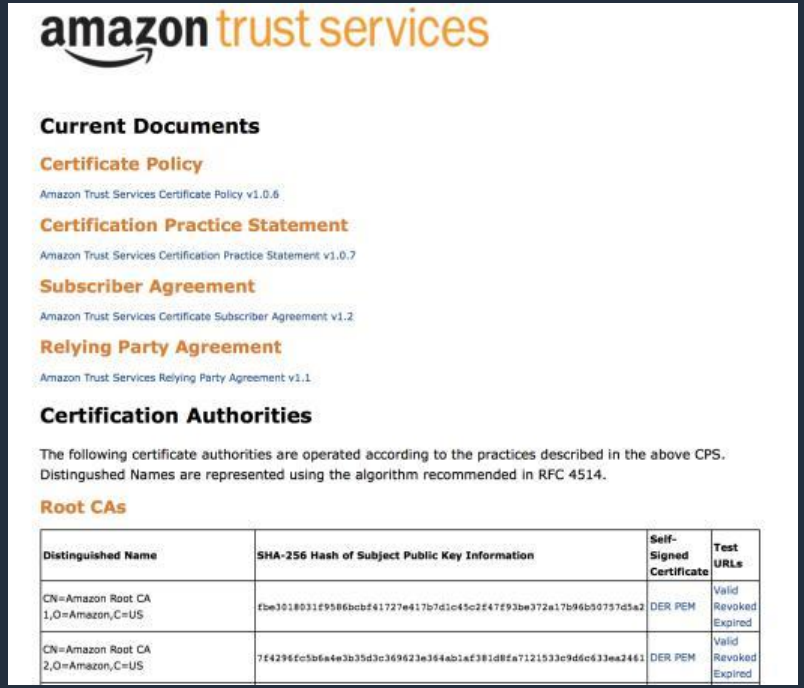
- 組織情報の審査を経てから発行
  （サイト運営者のなりすましを防ぎ、組織情報がTLS証明書に記載）

拡張認証（EV）

- OVよりも厳格な審査を経て発行

Amazon Trust Services 自社認証局

- AWS は Amazon Trust Services 認証局 があらゆるところで確実に使えるようにするために、2005年以降のほとんどのブラウザで信頼されているルート認証局である Starfield Services の認証局の一つを購入
- Amazon Trust Services で発行された証明書を使うために何もする必要が無い
- 各AWSサービスのエンドポイントで利用されるサーバー証明書がATSへ移行中



amazon trust services

Current Documents

Certificate Policy

Amazon Trust Services Certificate Policy v1.0.6

Certification Practice Statement

Amazon Trust Services Certification Practice Statement v1.0.7

Subscriber Agreement

Amazon Trust Services Certificate Subscriber Agreement v1.2

Relying Party Agreement

Amazon Trust Services Relying Party Agreement v1.1

Certification Authorities

The following certificate authorities are operated according to the practices described in the above CPS. Distinguished Names are represented using the algorithm recommended in RFC 4514.

Root CAs

Distinguished Name	SHA-256 Hash of Subject Public Key Information	Self-Signed Certificate	Test URLs
CN=Amazon Root CA 1,O=Amazon,C=US	fbe3018031f9586bcbf41727e417b7d1c45c2f47f93be372a17b96b50757d5a2	DER PEM	Valid Revoked Expired
CN=Amazon Root CA 2,O=Amazon,C=US	7f4296fc5b6a4e3b35d3c369623e364ab1af381d8fa7121533c9d6c633ea2461	DER PEM	Valid Revoked Expired

https://www.amazontrust.com/repository/

ACMに統合されるサービス

証明書配置可能

- Elastic Load Balancing（ELB）※ALB,CLB
- Amazon CloudFront
- Amazon API Gateway

連携利用可能

- AWS Elastic Beanstalk
- AWS CloudFormation

利用可能なリージョン

- 証明書のエクスポートやリージョン間でのコピー不可
- Amazon CloudFront に関連付けるには米国東部 （バージニア北部）で設定

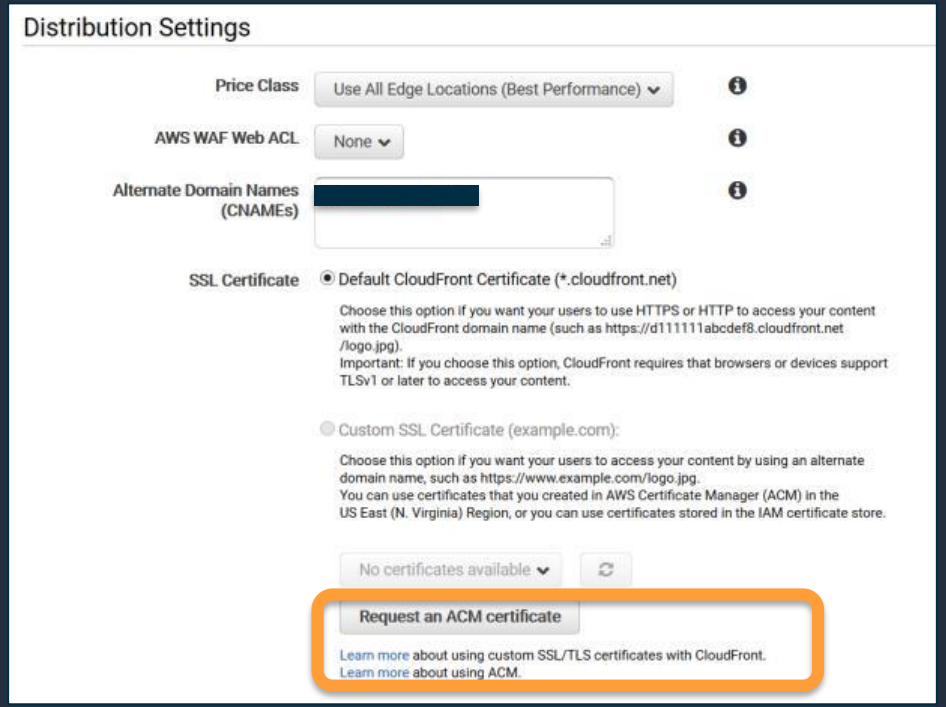
オハイオ	ソウル	アイルランド
バージニア北部	シンガポール	ロンドン
北カリフォルニア	シドニー	パリ
オレゴン	東京	サンパウロ
ムンバイ	カナダ	GovCloud(US−EAST)
大阪ローカル	フランクフルト	GovCloud(US)

(2018年12月19日現在)

CloudFrontにおける設定の注意点

- 設定リージョン
  Amazon CloudFrontを利用する場合は、米国東部（バージニア北部）リージョンにて設定を行う

- 配信
  米国東部 （バージニア北部）リージョンでプロビジョンされ、Amazon CloudFront ディストリビューションに関連付けられた証明書が、お客様のディストリビューションに設定された地理的場所に配信される



料金以外のACM利用メリット

証明書の更新とデプロイが自動化

- ACM により、安全なウェブサービスやアプリケーションに対する TLS の設定と維持が自動化されるため、エラーが発生しやすい手動プロセスと比較して、運用の信頼性が向上（証明書の期限切れによるダウンタイムがなくなる）
- TLS 証明書の購入、アップロード、および更新という時間のかかるプロセスを手動で行う必要がなくなる

ACM による証明書やキーの更新や古い証明書の差し替えは、事前の通知なしに行われる可能性あり

複数環境での利用

- 同じリージョンの複数の Elastic Load Balancing ロードバランサーや複数の CloudFront ディストリビューションで同じ一枚の証明書を使用可能

CloudTrailへの対応

- 操作の証跡を記録することが可能

マネージド型自動更新の注意点

- 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

- インポートした証明書には自動更新不可

- 自動更新の前提条件
    - ACM によって証明書の各ドメインと HTTPS 接続を確立できる
    - 各接続では、返される証明書が ACM が更新している証明書と一致
    - 証明書は 「ACMに統合されるサービス」に関連付けられている
    - ACM が証明書に記載されているドメイン名を検証できる

https://docs.aws.amazon.com/ja_jp/acm/latest/userguide/troubleshooting-renewal.html

サードパーティ証明書のインポート

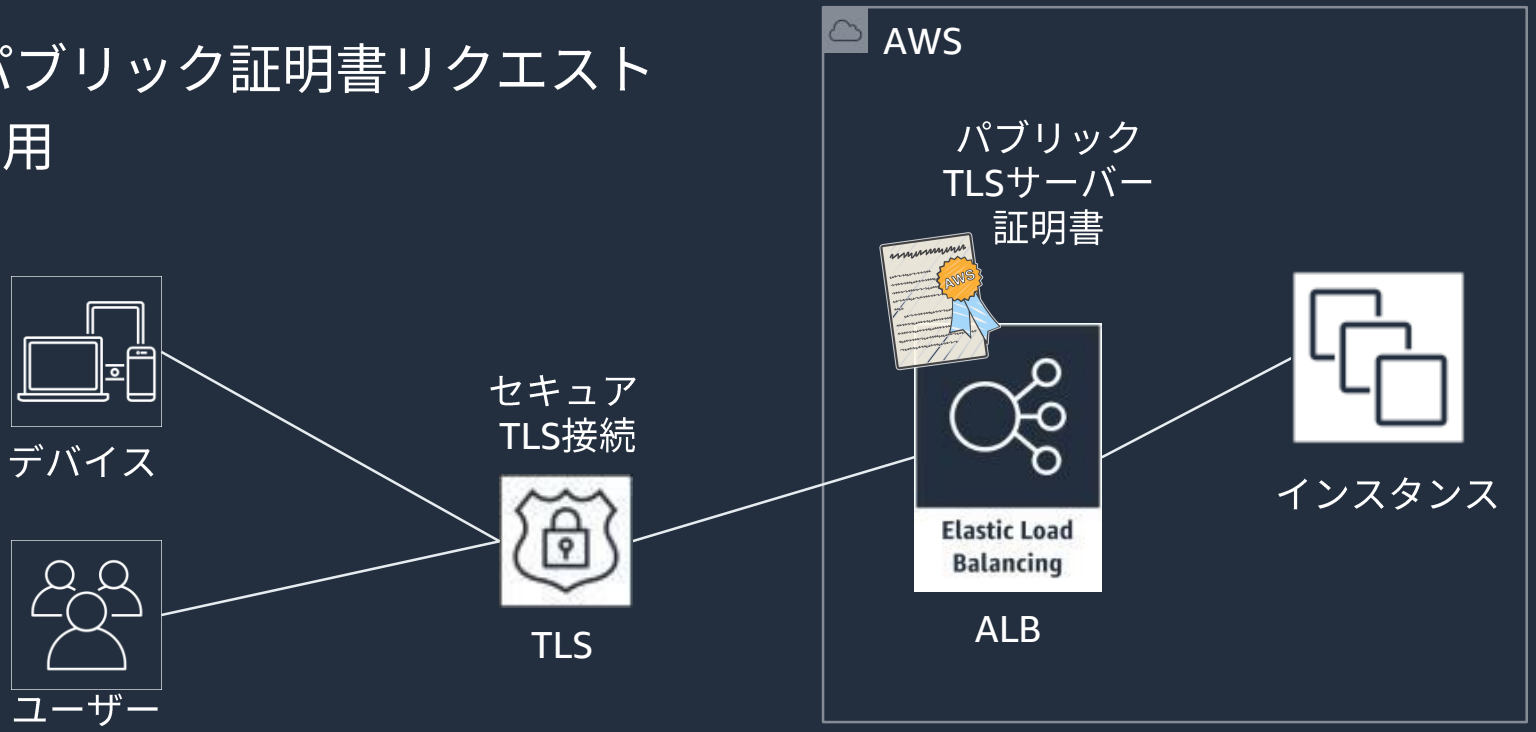
※インポートされた証明書には
マネージド型更新不可

https://docs.aws.amazon.com/ja_jp/acm/latest/userguide/import-certificate.html

AWS Certificate Manager
デモ

ACM設定デモ（ALB）

- ACMにパブリック証明書リクエスト
- ALBに適用



AWS Certificate Manager トップページ

事前準備

- 自分でドメインを取得していること

- ドメイン検証準備
    - DNS レコードを追加できること

    or

    - Eメールが受信できること

ドメイン名の追加

複数ドメイン名、ワイルドカードドメイン名対応のメリット

一枚の証明書でサブドメインなどを含めた証明書の利用が必要な場合

- 例えば、*.example.com というドメイン名を使用すると、www.example.com、images.example.com など、任意のホスト名 （一番左のサブドメイン） に .example.com が続くすべてのドメイン名を保護可能
- example.net、example.comといった範囲としたいドメインを保護可能

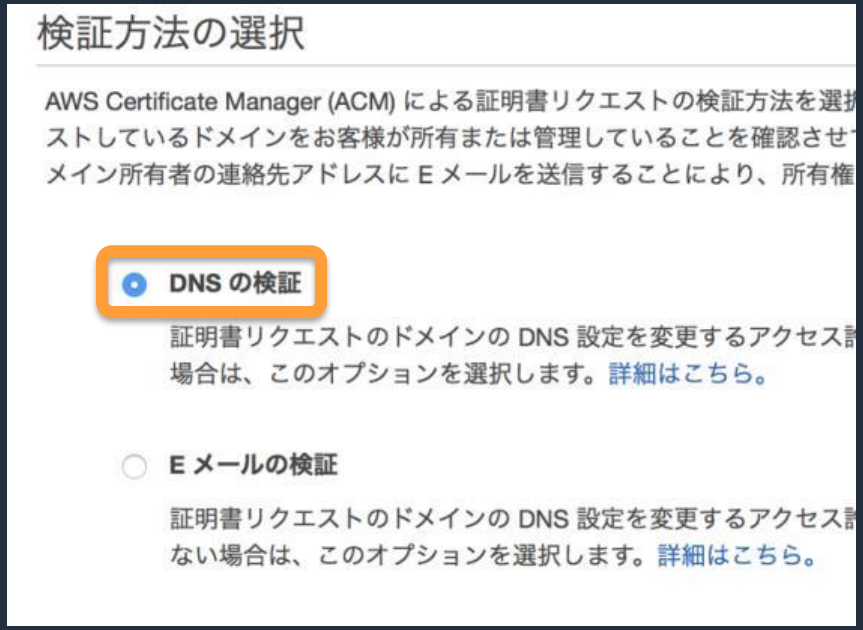
証明書のリクエスト

検証方法の選択

- DNS検証（推奨）
    - DNS にCNAMEレコードを追加し、ACMが自動的に検証
        - DNSは外部DNS、Route 53 どちらも利用可能
        - Route 53 をご利用であればワンクリック

- Eメール検証
    - 何かの理由でDNS検証が行えない場合に選択
    - 特定メールアカウントに対する受信するメールで承認が必要

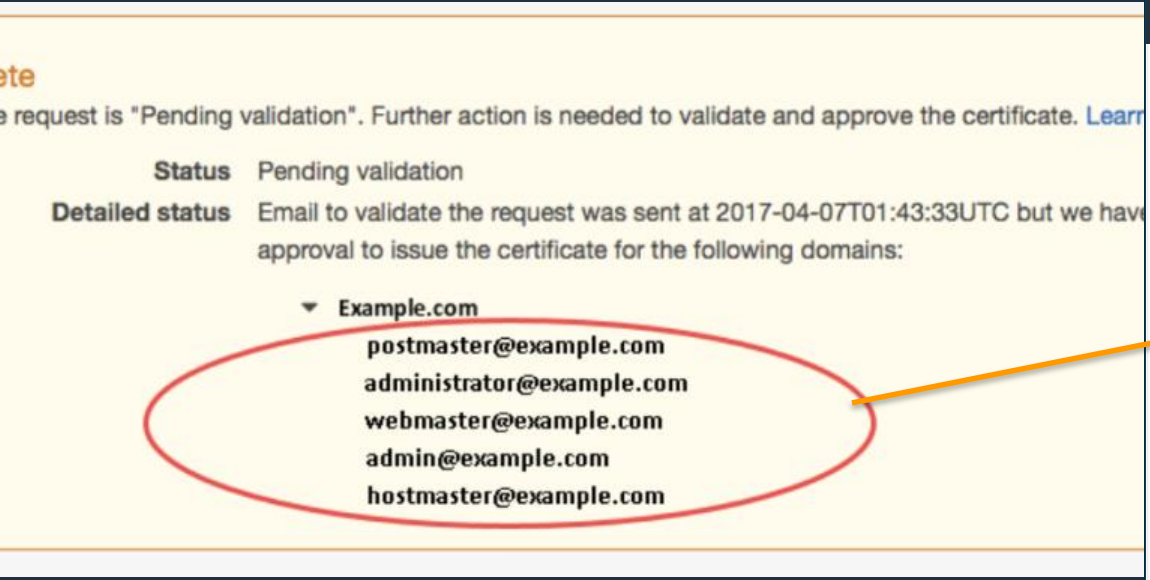
DNS検証 – DNSを使用した、証明書の簡単な検証

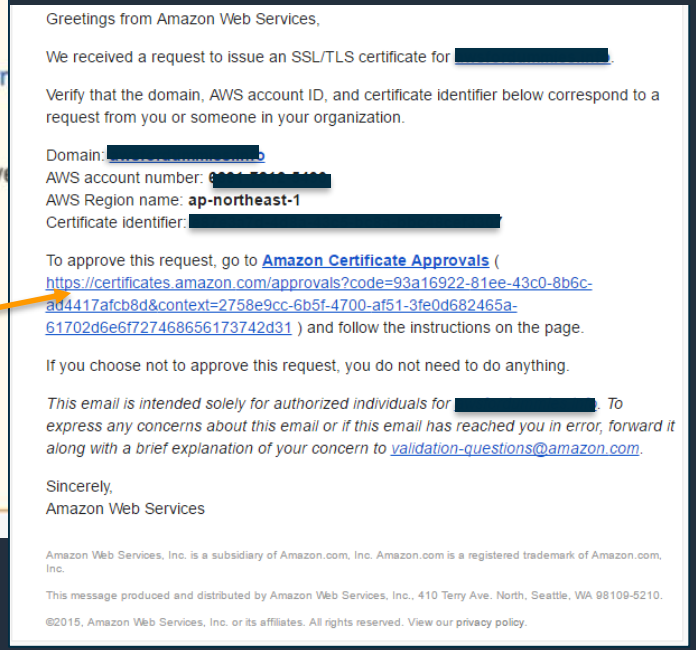


- DNS 認証を使用して ACM に認証をリクエストするときに、お客様がドメイン名の所有者または管理者であることを証明可能
- 設定方法：DNS 設定に CNAME レコードを書き込み
  （Amazon Route 53 を使用して DNS レコードを管理している場合、ワンクリックでDNS レコードを設定も可）
- DNS で検証済みの証明書を完全に自動更新可能

DNS検証 - Route 53 でのレコード作成

※Eメール検証の場合 – メール送信先




Greetings from Amazon Web Services,

We received a request to issue an SSL/TLS certificate for [redacted].

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

Domain: [redacted]
AWS account number: [redacted]
AWS Region name: **ap-northeast-1**
Certificate identifier: [redacted]

To approve this request, go to **Amazon Certificate Approvals** (https://certificates.amazon.com/approvals?code=93a16922-81ee-43c0-8b6c-ad4417afcb8d&context=2758e9cc-6b5f-4700-af51-3fe0d682465a-61702d6e6f727468656173742d31) and follow the instructions on the page.

If you choose not to approve this request, you do not need to do anything.

This email is intended solely for authorized individuals for [redacted]. 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

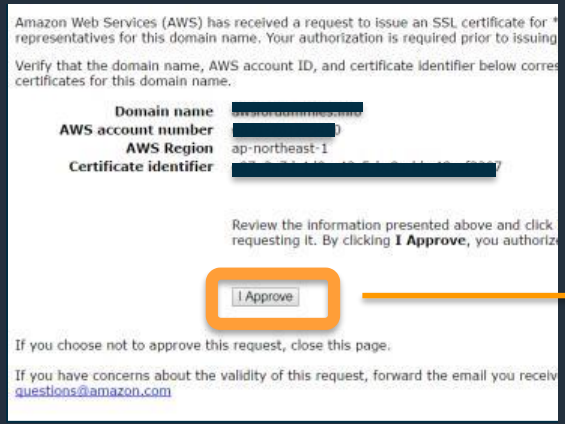
Amazon Web Services, Inc. is a subsidiary of Amazon.com, Inc. Amazon.com is a registered trademark of Amazon.com, Inc.

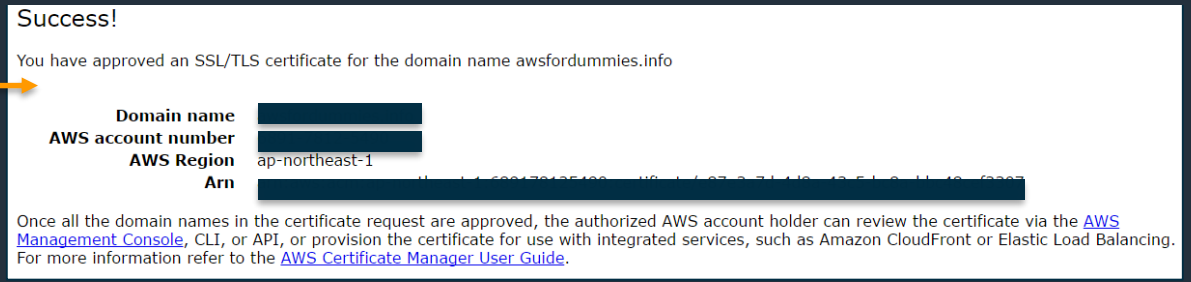
This message produced and distributed by Amazon Web Services, Inc., 410 Terry Ave. North, Seattle, WA 98109-5210.

©2015, Amazon Web Services, Inc. or its affiliates. All rights reserved. View our privacy policy.

差出人：**Amazon Certificates**
<no-reply@certificates.amazon.com>

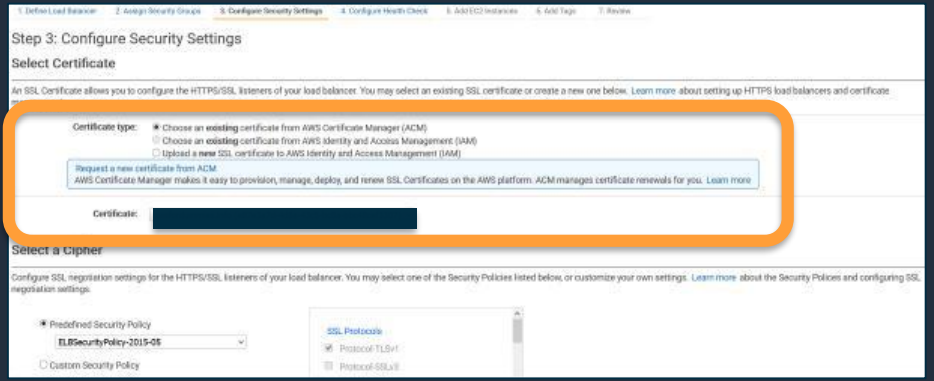
※Eメール検証の場合 – メールよる承認

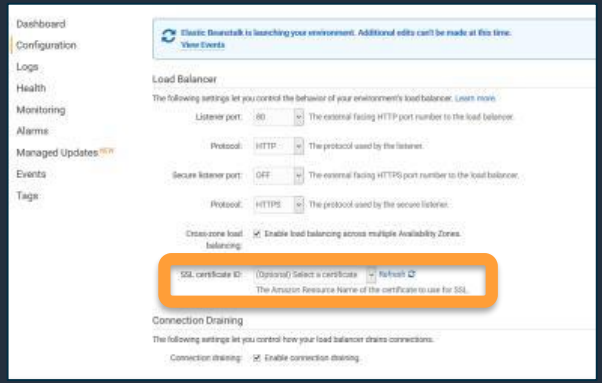




ELBのACM設定

- 設定画面において任意のACM証明書を選択
- セキュリティポリシー設定
  - 利用しないバージョン、脆弱なバージョンは許可しないことが原則





AWS Certificate Manager

AWS Certificate Manager (ACM) により、AWS プラットフォームでの SSL/TLS 証明書のプロビジョニング、管理、デプロイ、更新が容易になります。

今すぐ始める

ユーザーガイド

証明書のプロビジョニング

お客様のサイトの名前を指定し、ID を設定してください。残りの手順は ACM が行います。ACM は Amazon が発行す

SSL/TLS ベースのサイトとアプリケーションのデプロイ

Elastic Load Balancer または Amazon CloudFront ディストリビューションを作成し、ACM が提供またはインポートした

証明書の管理

ACM が提供またはインポートした証明書は、すべて AWS マネジメントコンソールでご覧いただけます。ACM API、

AWS Certificate Manager まとめ

- ✓ パブリック証明書とプライベート証明書のライフサイクルを集中管理可能
- ✓ AWS の各種サービスで使用するTLSサーバー証明書のプロビジョニング、管理、およびデプロイを簡単に
- ✓ AWS Certificate Manager でプロビジョニングされた TLSサーバー証明書は無料（お支払いいただくのは、アプリケーションを実行するために作成した AWS リソースの料金のみ）

AWS Certificate Manager Private CA

プライベート CA 運用における課題

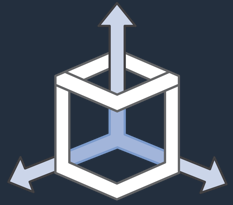
- プライベートCAの維持は複雑で高価
- プライベートCAにはインフラストラクチャだけでなく、セキュリティの専門知識も必要
- 複数のCAを運用すると複雑さがさらに増大
- 組織は、セキュリティ、説明責任およびプライベートCAの可用性を考慮必要
- 動的リソースで利用したい場合スケーラビリティ考慮必要

動的リソースに証明書でできること

- 課題
  - 動的クラウドリソースを認証する必要あり
    - Autoscaling, CloudFormation, その他自動化
  - エンタープライズ プライベートCAでの動的リソース利用
    - 遅い、柔軟性が低い、運用作業の煩雑さ
- 解決策
  - 証明書を発行し、失効情報（CRL）を自動発行するための、APIに基づくセキュアなプライベートCAサービス

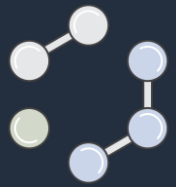
AWS Certificate Manager（ACM）Private CA

セキュアかつマネージド
なプライベートCA



証明書の集中管理

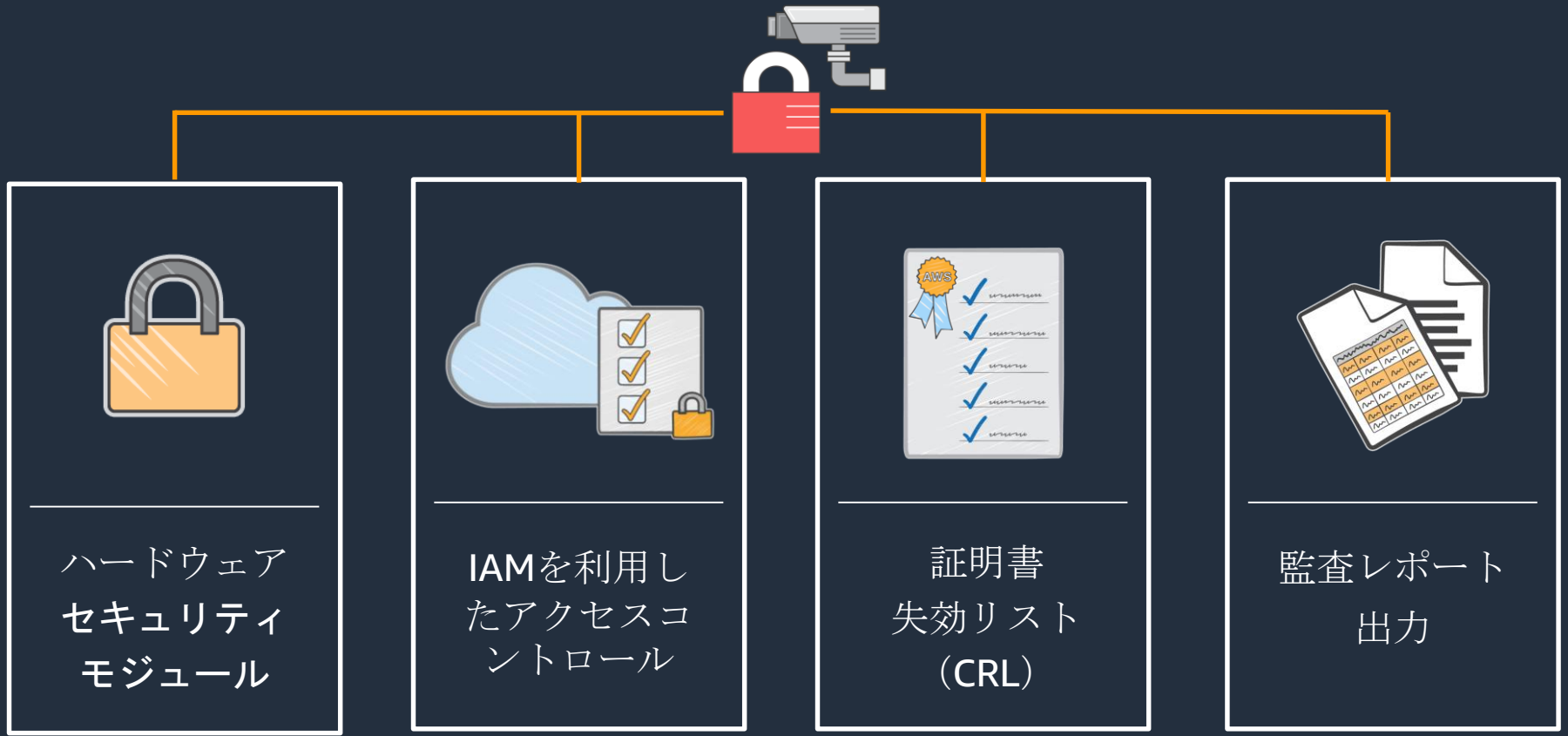
API



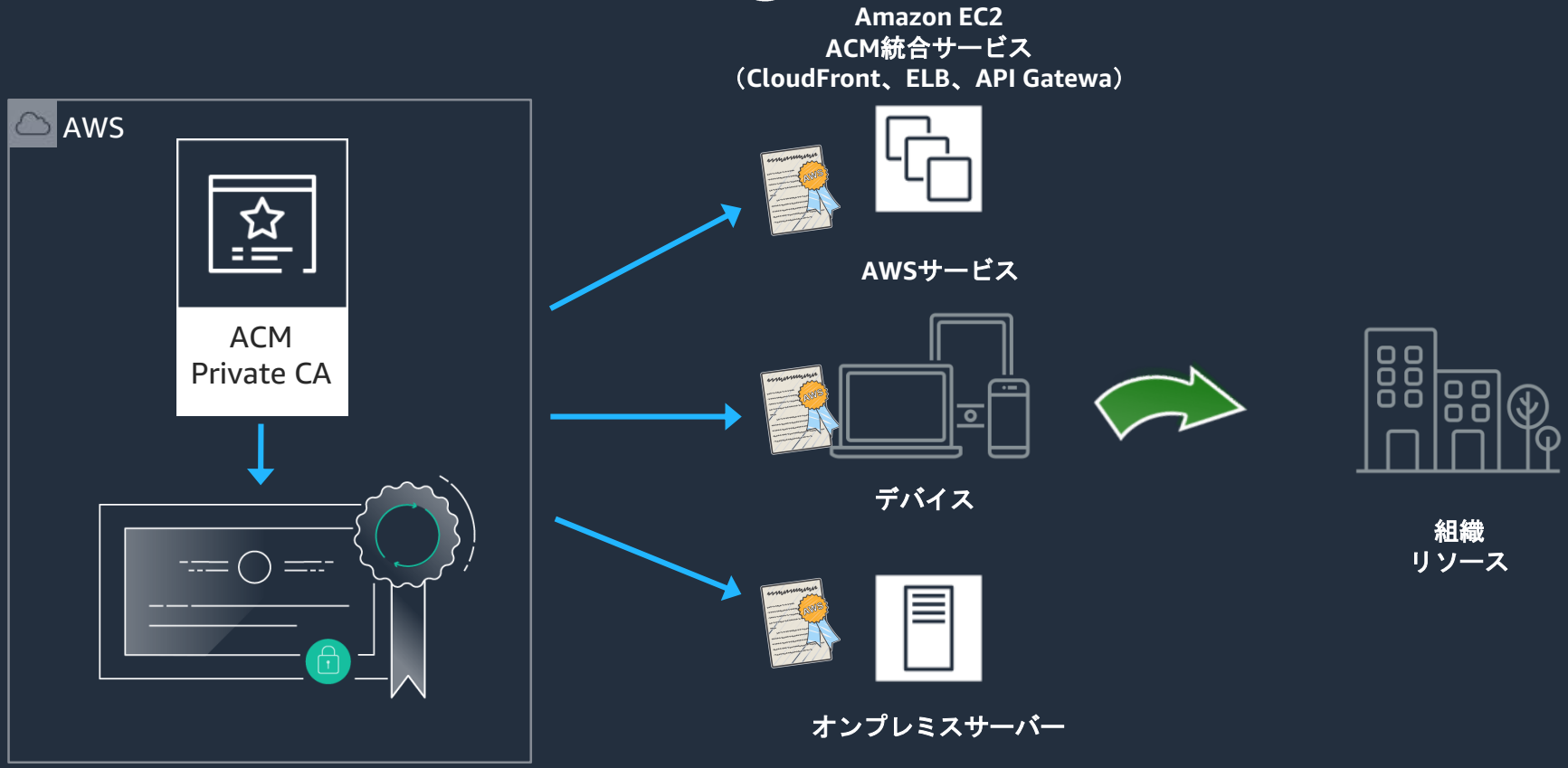
プライベート証明書の
カスタマイズ

従量課金

セキュアなマネージドプライベート CA



AWS Certificate Manager（ACM） Private CA



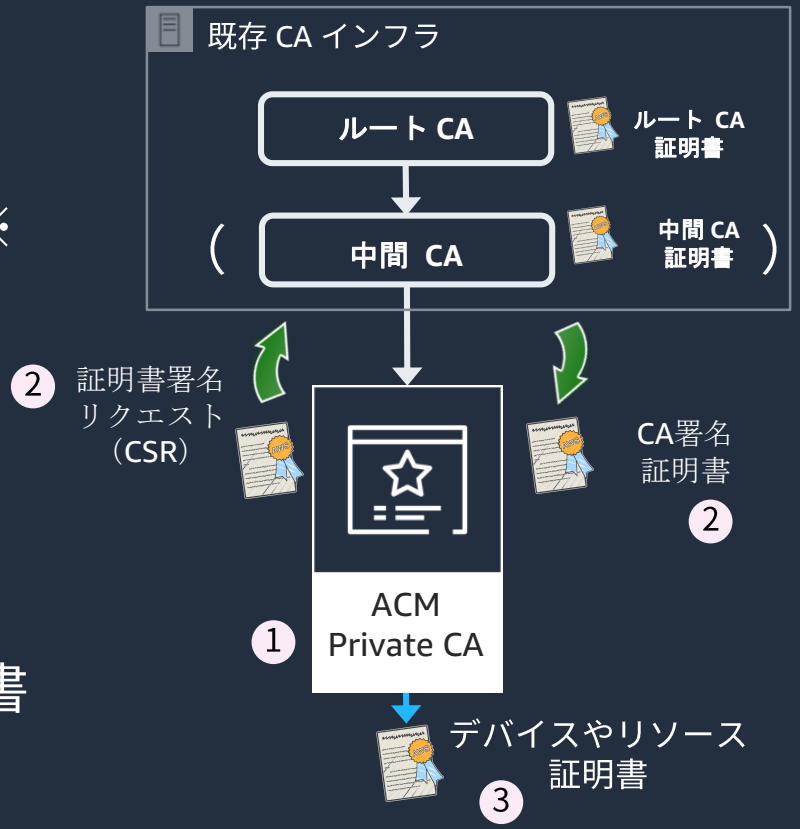
ACM Private CAの仕組み

1. ACM Private CAの作成

2. ルートCAに信頼するチェーンの構築※
   1. CSR出力し、親CAが署名
   2. 署名済CA証明書のインポート

3. デバイス、リソースの証明書発行
   1. ACMから
   2. API、CLI利用からカスタマイズ証明書

※親Privaete CAの署名証明書が必要

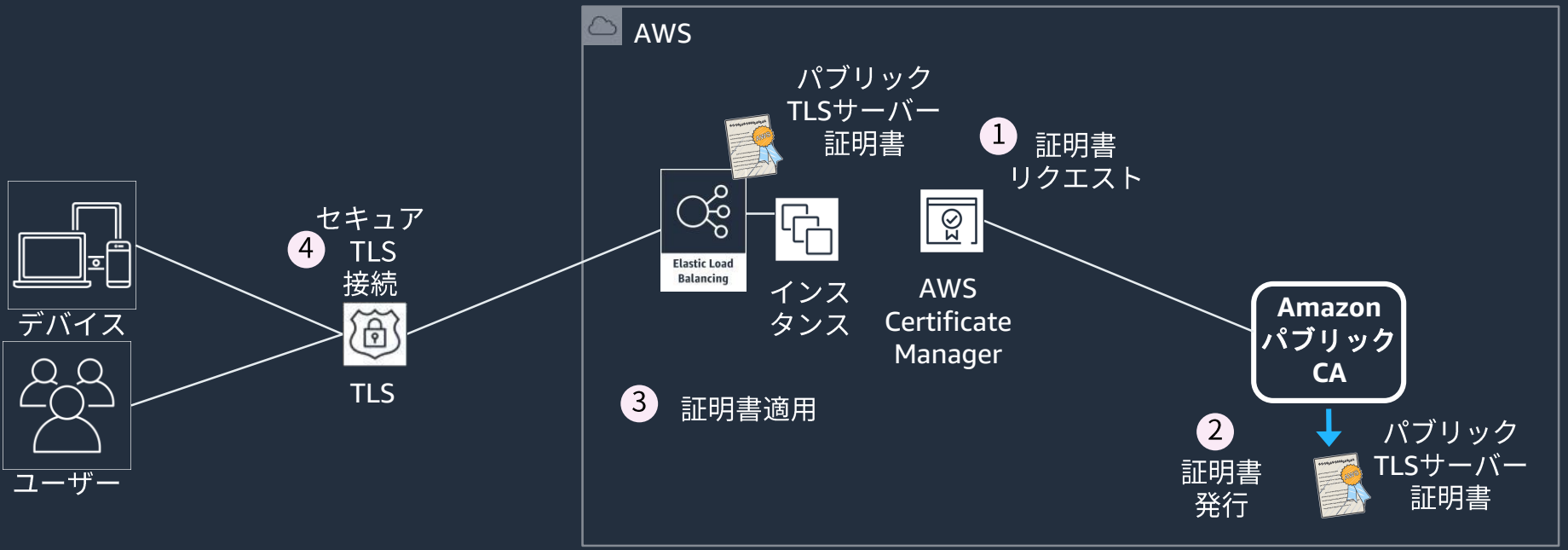


ユースケース

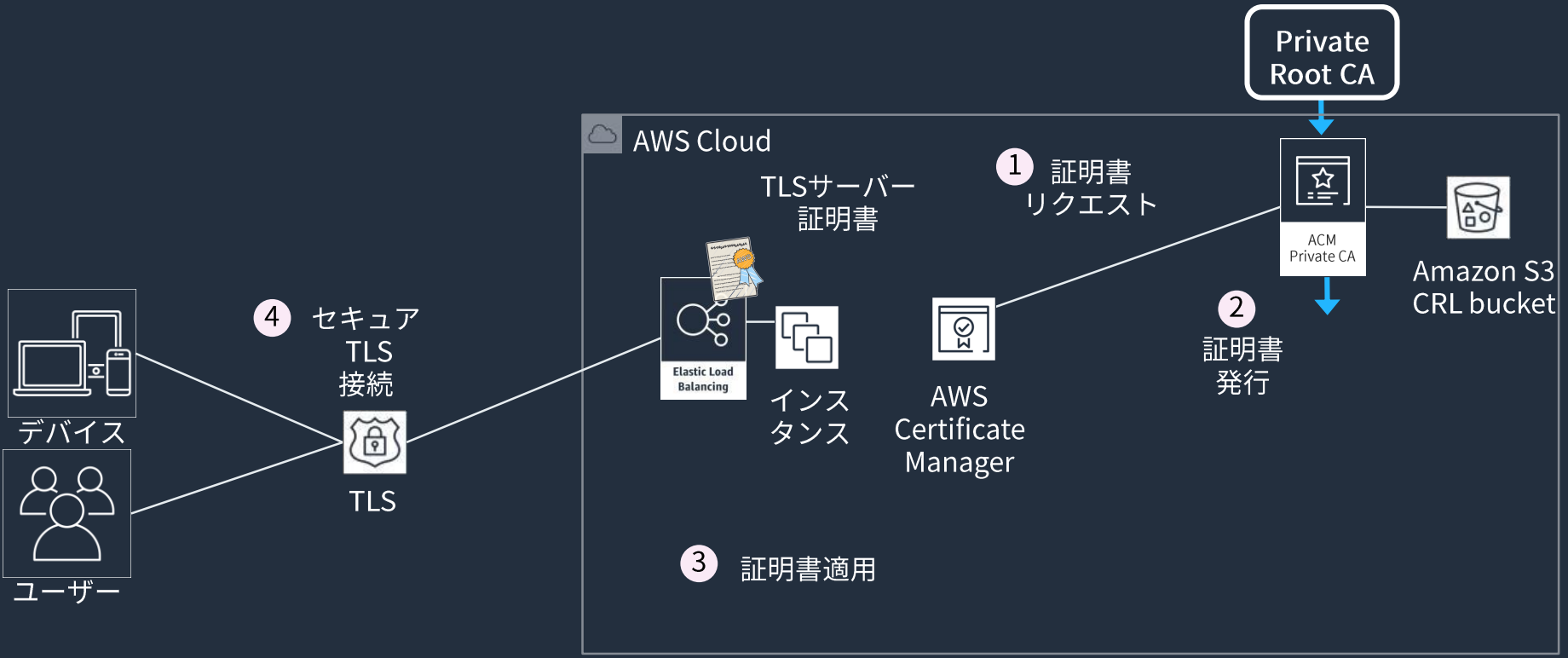
ACM Private CAはいつ使うのか？

- サーバー証明書
  - 内部サーバーの識別にプライベート証明書を利用
  - EC2, ECS, もしくはオンプレミスサーバー（例Apache, Tomcat, NGINX, IIS, WebLogic, WebSphere..）
  - AWS ELB, CloudFront, API Gatewayにて利用
- クライアント証明書
  - APIアクセスの２要素認証
  - サーバー間通信のためのTLS相互認証
- 自己署名証明書の代替
- IoT デバイス証明書

ELBでのパブリック証明書利用



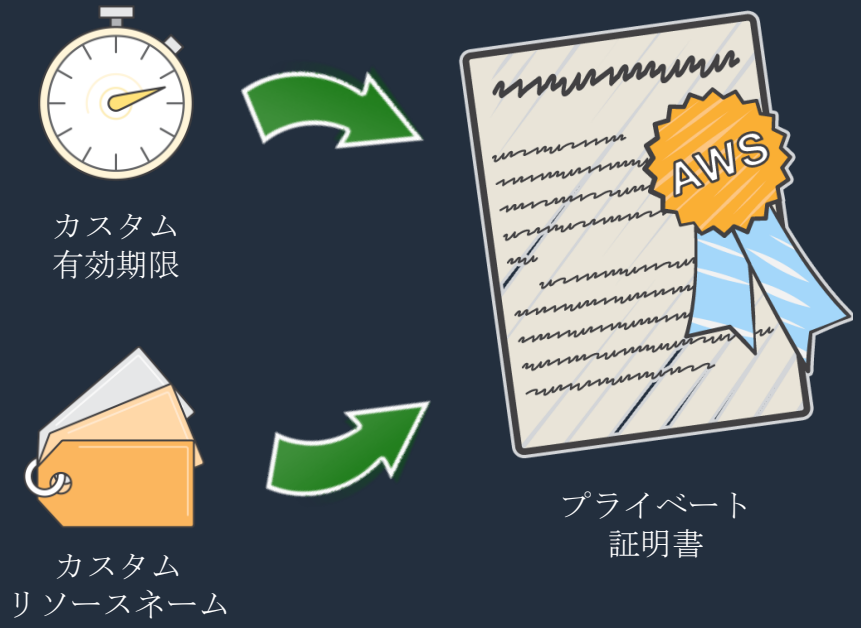
ELBでのプライベート証明書利用



ACM Private CA 証明書発行管理方法

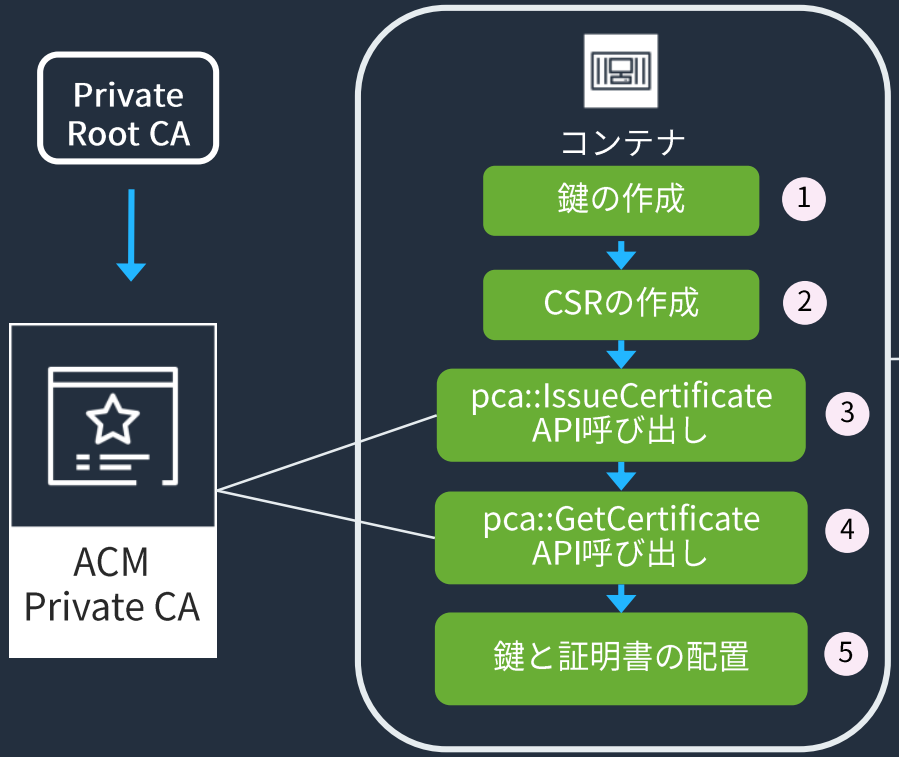
	ACM管理証明書	カスタマイズ証明書
証明書秘密鍵	ACMが秘密鍵の生成と管理	利用者が秘密鍵の生成と管理
証明サブジェクト/SAN	有効なDNS名のみ	柔軟なX.509 サブジェクト/SAN
有効期限	13 ヶ月	自由な期間
鍵と署名アルゴリズム	RSA 2048 with SHA-256 hashing	ECDSA or RSA keys SHA-256, SHA-384, SHA-512 hashing
エクスポート	プライベート証明書のみ可能 （証明書チェーン、秘密鍵含む）	n/a （利用者が プライベートキーの生成と管理）
更新	関連付けされている間、自動更新	利用者管理
適用	ACM統合サービスはACMが管理 オンプレミス、EC2、Iotは利用者管理	利用者管理
利点	集中管理	柔軟性

プライベート証明書のカスタマイズ



鍵アルゴリズム	署名アルゴリズム
RSA 2048	SHA256 with RSA
RSA 4096	SHA384 with RSA
	SHA512 with RSA
ECDSA P256	SHA256 with ECDSA
ECDSA P384	SHA384 with ECDSA
	SHA512 with ECDSA

コンテナでのTLS利用

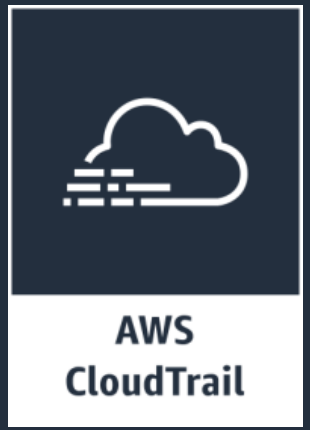


IAM Policy to allow container access to PCA

```
{
“Version”: “2012-10-17”,
“Statement”: [
{
“Effect”: “Allow”,
“Action”: [
“acm-pca:IssueCertificate”,
“acm-pca:GetCertificate”
],
“Resource”: “*”
}
]
}
```

https://aws.amazon.com/jp/blogs/compute/maintaining-transport-layer-security-all-the-way-to-your-container-part-2-using-aws-certificate-manager-private-certificate-authority/

AWS CloudTrailを利用したロギング



- ACM Private CAとの間で行われるAPI呼び出しを記録
- Amazon S3バケットに書き込む権限要設定
- ACM Private CAコンソール、AWS CLI、またはAWS SDKからのAPI呼び出しが収集

自動化ツール


Android


iOS


Java


JavaScript


.NET


Python (boto)


Ruby


Xamarin


Node.js


PHP

ACM Private CA API

CreateCertificateAuthority
IssueCertificate
GetCertificate
RevokeCertificate

```
UpdateCertificateAuthority
DeleteCertificateAuthority
ListCertificateAuthorities
DescribeCertificateAuthority
GetCertificateAuthorityCsr
CreateCertificateAuthorityAuditReport
DescribeCertificateAuthorityAuditReport
ImportCertificateAuthorityCertificate
GetCertificateAuthorityCertificate
TagCertificateAuthority
UntagCertificateAuthority
ListTags
```

ACM Private CA CLI

issue-certificate --certificate-authority-arn <value>
--csr <value>
--signing-algorithm <value>

OUTPUT: CertificateArn -> (string)

get-certificate --certificate-authority-arn <value>
--certificate-arn <value>

OUTPUT: Certificate -> (string)
CertificateChain -> (string)

ACM 汎用API

```
request-certificate --domain-name <value>
                    [--subject-alternative-names <value>]
                    [--certificate-authority-arn <value>]
  OUTPUT: CertificateArn -> (string)


export-certificate --certificate-arn <value
          --passphrase <value>
  OUTPUT: Certificate -> (string)
      CertificateChain -> (string)
      PrivateKey -> (string)
```

Private CA デフォルトサービス上限

Private Certificate Authorities （CA）の数	10
Private CA毎のプライベート証明書の数	50,000

上限緩和についてはAWS Supportまでお問い合わせください。
https://aws.amazon.com/jp/contact-us/

ACM Private CA価格

CA 運用

- $400 （1ヶ月, 1CA）
- Private CAを削除するまで月額課金

プライベート証明書発行枚数（1ヶ月, 1リージョン）	証明書毎の価格
0–1,000	$0.75
1,000–10,000	$0.35
10,000+	$0.001

証明書発行

- 証明書発行に課金
- ただし、プライベート証明書をACM統合サービス（CloudFront, ELB, API Gateway）でのご利用は無料

- Free trial –はじめてご利用で最初のCAの作成時は、最初の30日間のCA操作は無料。発行済証明書に対してのみ支払い

AWS Private CA 対応リージョン

- オハイオ
- バージニア北部
- オレゴン
- シンガポール
- シドニー
- 東京
- カナダ
- フランクフルト
- アイルランド
- ロンドン

データプライバシープロテクション関連のコンプライアンスに準拠

- PCI DSS - カード会員データを保護するための技術的および運用上の要件
- ISO 9001,27001,27017、および27018 - 最も認知されているグローバルセキュリティ基準
- HIPAA対象 – 米国における医療保険の相互運用性と説明責任に関する法令
- AICPA SOC 1,2および3 - 重要なコンプライアンス管理および目標を AWS がどのように達成したかを実証

ACM Private CA まとめ

- ✓ ACM Private CAは、完全に管理されたプライベートCAであり、複雑さと管理オーバーヘッドゼロ
- ✓ ACM Private CAは、機敏性とカスタマイズ性を提供
- ✓ 動的リソースに対して証明書発行の自動化可能
- ✓ ACM Private CAは、AWS IAM、AWS CloudTrail、タグ付けなどの他の主要なAWSサービスと統合され、より多くの機能と価値を提供

本セッションのまとめ

- ✓ ACM
  - ✓ パブリック証明書とプライベート証明書のライフサイクルを集中管理可能
  - ✓ AWS の各種サービスで使用するTLSサーバー証明書のプロビジョニング、管理、およびデプロイを簡単に
  - ✓ AWS Certificate Manager でプロビジョニングされた TLS サーバー証明書は無料
- ✓ ACM Private CA
  - ✓ 完全に管理されたプライベートCAであり、複雑さと管理オーバーヘッドゼロ
  - ✓ 機敏性とカスタマイズ性を提供

参考情報

AWS Certificate Manager メインページ:

- https://aws.amazon.com/jp/certificate-manager/

AWS Certificate Manager プライベートCA メインページ：

- https://aws.amazon.com/jp/certificate-manager/private-certificate-authority/

AWS Certificate Manager ドキュメント

- https://docs.aws.amazon.com/ja_jp/acm/

AWS Certificate Manager の料金

- https://aws.amazon.com/jp/certificate-manager/pricing/

AWS Certificate Manager のよくある質問

- https://aws.amazon.com/jp/certificate-manager/faqs/

Q&A

お答えできなかったご質問については
AWS Japan Blog 「https://aws.amazon.com/jp/blogs/news/」にて
後日掲載します。

AWS Well-Architected 個別技術相談会

毎週”W-A個別技術相談会”を実施中

- AWSのソリューションアーキテクト(SA)に対策などを相談することも可能
- 申込みはイベント告知サイトから

(https://aws.amazon.com/jp/about-aws/events/)

AWS イベント で[検索]

ご視聴ありがとうございました

AWS 公式 Webinar
https://amzn.to/JPWebinar

過去資料
https://amzn.to/JPArchive