aws

사용 설명서

AWS 인증서 관리자

버전 1.0

AWS 인증서 관리자: 사용 설명서

Table of Contents

AWS Certificate Manager란 무엇인가요?

AWS Certificate Manager (ACM)는 AWS 웹 사이트와 애플리케이션을 보호하는 퍼블릭 및 프라이빗 SSL/TLS X.509 인증서와 키를 생성, 저장 및 갱신하는 복잡성을 처리합니다. ACM에서 직접 발급하거나 서드 파티 인증서를 ACM 관리 시스템으로 가져오는 방법으로 통합 AWS 서비스에 대한 인증서를 제공할 수 있습니다. ACM 인증서는 단일 도메인 이름, 여러 특정 도메인 이름, 와일드카드 도메인 또는 이러한 도메인의 조합을 보호할 수 있습니다. ACM 와일드카드 인증서는 원하는 만큼의 하위 도메인을 보호할 수 있습니다. 내부 PKI의 어느 곳에서나 사용할 수 AWS Private CA 있도록에서 서명한 ACM 인증서를 내보낼 수도 있습니다.

> **Note**
>
> ACM은 독립형 웹 서버에 사용하기 위한 것이 아닙니다. Amazon EC2 인스턴스에 독립 실행형 보안 서버를 설정하려는 경우 다음 튜토리얼에 지침이 있습니다. Amazon Linux 2023에서 SSL/TLS 구성.

주제

지원되는 리전:

ACM은 퍼블릭 엔드포인트에서 IPv4 및 IPv6를 지원합니다. ACM에 대한 리전별 가용성은 AWS 일반 참조의 AWS 리전 및 엔드포인트 또는 AWS 리전 표를 참조하세요.

ACM의 인증서는 리전별 리소스입니다. 둘 이상의 AWS 리전에서 동일한 FQDN(정규화된 도메인 이름) 또는 FQDNs 집합에 대해 Elastic Load Balancing과 함께 인증서를 사용하려면 각 리전에 대한 인증서를 요청하거나 가져와야 합니다. ACM에서 제공하는 인증서의 경우 각 리전에 대한 인증서에서 각 도메인 이름을 다시 검증해야 합니다. 리전 간에 인증서를 복사할 수 없습니다.

Amazon CloudFront에서 ACM 인증서를 사용하려면 미국 동부(버지니아 북부) 리전에서 인증서를 요청하거나 가져와야 합니다. CloudFront 배포와 연결되는 이 리전의 ACM 인증서는 해당 배포에 대해 구성된 모든 지리적 위치에 배포됩니다.

에 대한 요금 AWS Certificate Manager

관리하는 SSL/TLS 인증서에는 추가 요금이 부과되지 않습니다 AWS Certificate Manager. 웹 사이트 또는 애플리케이션을 실행하기 위해 생성한 AWS 리소스에 대해서만 비용을 지불합니다. 최신 ACM 요금 정보는 AWS 웹 사이트의 AWS Certificate Manager 서비스 요금 페이지를 참조하세요.

AWS Certificate Manager 개념

이 섹션에서는에서 사용하는 개념에 대한 정의를 제공합니다 AWS Certificate Manager.

주제

- ACM 인증서
- ACM 루트 CA
- Apex 도메인
- 비대칭 키 암호화 기법
- 인증 기관
- 인증서 투명성 로깅
- Domain Name System(DNS)
- 도메인 이름
- 암호화 및 암호 해독
- 정규화된 도메인 이름(FQDN)
- Hypertext Transfer Protocol(HTTP)
- 퍼블릭 키 인프라(PKI)
- 루트 인증서
- Secure Sockets Layer(SSL)
- 보안 HTTPS
- SSL 서버 인증서
- 대칭 키 암호화
- 전송 계층 보안(TLS)
- 신뢰

ACM 인증서

ACM은 X.509 버전 3 인증서를 생성합니다. 각는 198일 동안 유효하며 다음 확장을 포함합니다.

- 기본 제약 - 인증서의 주체가 인증 기관(CA)인지 여부를 지정합니다.
- 기관 키 식별자 - 인증서에 서명하는 데 사용되는 프라이빗 키에 해당하는 퍼블릭 키를 식별할 수 있습니다.
- 주체 키 식별자 - 특정 퍼블릭 키가 포함된 인증서를 식별할 수 있습니다.
- 키 사용 - 인증서에 포함된 퍼블릭 키의 용도를 정의합니다.
- 확장 키 사용 - 키 사용 확장에 지정된 용도에 추가하여 퍼블릭 키를 사용할 수 있는 하나 이상의 용도를 지정합니다.

> **Important**
>
> 2025년 6월 11일부터 웹 사이트 인증서에 대한 새 브라우저 요구 사항에 맞게 "TLS 웹 클라이언트 인증"(clientAuth) 확장 키 사용(EKU)으로 인증서를 더 이상 발급 AWS Certificate Manager 하지 않습니다.

- CRL 배포 지점 - CRL 정보를 얻을 수 있는 위치를 지정합니다.

ACM에서 발급된 인증서의 일반 텍스트는 다음 예와 유사합니다.

```
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 루트 CA

ACM에서 발급된 공인 최종 엔터티 인증서는 다음 Amazon 루트 CA에서 보안 인증을 추출합니다.

고유 이름	암호화 알고리즘
CN=Amazon Root CA 1,O=Amazon,C=US	2048비트 RSA (RSA_2048)
CN=Amazon Root CA 2,O=Amazon,C=US	4096비트 RSA (RSA_4096)
CN=Amazon Root CA 3,O=Amazon,C=US	타원 프라임 곡선 256비트 (EC_prime256v1)
CN=Amazon Root CA 4,O=Amazon,C=US	타원 프라임 곡선 384비트 (EC_secp384r1)

ACM에서 발급된 인증서에 대한 보안 인증의 기본 루트는 CN=Amazon Root CA 1,O=Amazon,C=US로 2048비트 RSA 보안을 제공합니다. 나중에 사용할 수 있도록 기타 루트가 예약되어 있습니다. 모든 루트는 Starfield Services Root Certificate Authority 인증서에서 교차 서명됩니다.

자세한 내용은 Amazon Trust Services를 참조하세요.

Apex 도메인

도메인 이름을(를) 참조하세요.

비대칭 키 암호화 기법

대칭 키 암호화과는 달리, 비대칭 암호화 기법에서는 다르지만 수학적으로 관련된 키를 사용하여 콘텐츠를 암호화 및 암호화 해제합니다. 키 중 하나는 퍼블릭 키이며 일반적으로 X.509 v3 인증서에 제공됩니다. 다른 키는 프라이빗 키이며 보안 방식으로 저장됩니다. X.509 인증서는 사용자, 컴퓨터 또는 다른 리소스(인증서 주체)의 자격 증명을 퍼블릭 키로 바인딩합니다.

ACM 인증서는 웹 사이트의 자격 증명 및 조직 세부 정보를 인증서에 포함된 퍼블릭 키와 바인딩하는 X.509 SSL/TLS 인증서입니다. ACM은 AWS KMS key 를 사용하여 프라이빗 키를 암호화합니다. 자세한 내용은 인증서 프라이빗 키 보안 단원을 참조하십시오.

인증 기관

인증 기관(CA)은 디지털 인증서를 발행하는 단체입니다. 상업적으로 가장 일반적인 유형의 디지털 인증서는 ISO X.509 표준을 기반으로 합니다. CA는 인증서 주체의 자격 증명을 확인하고 해당 자격 증명을 인증서에 포함된 퍼블릭 키와 바인딩하는 서명된 디지털 인증서를 발행합니다. CA는 일반적으로 인증서 취소도 관리합니다.

인증서 투명성 로깅

실수로 또는 훼손된 CA에서 발급되는 SSL/TLS 인증서를 방지하기 위해 일부 브라우저에서는 도메인에 대해 발급된 공인 인증서를 인증서 투명성 로그에 기록해야 합니다. 도메인 이름이 기록됩니다. 프라이빗 키는 기록되지 않습니다. 로깅되지 않는 인증서는 일반적으로 브라우저에서 오류를 생성합니다.

로그를 모니터링하여 도메인에 대해 허가한 인증서만 발급되는지 확인할 수 있습니다. 인증서 검색과 같은 서비스를 사용하여 로그를 점검할 수 있습니다.

Amazon CA는 도메인에 대해 공개적으로 신뢰할 수 있는 SSL/TLS 인증서를 발급하기 전에 최소 세 개 이상의 인증서 투명성 로그 서버에 인증서를 제출합니다. 이러한 서버는 인증서를 퍼블릭 데이터베이스에 추가하고 서명된 인증서 타임스탬프(SCT)를 Amazon CA에 반환합니다. 그러면 CA는 SCT를 인증서에 포함시키고 인증서에 서명한 다음 사용자에게 인증서를 발급합니다. 타임스탬프는 기타 X.509 확장과 함께 포함됩니다.

```
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
```

옵트아웃을 선택하지 않는 한 인증서를 요청하거나 갱신할 때 인증서 투명성 로깅이 자동으로 수행됩니다. 옵트아웃에 대한 자세한 내용은 인증서 투명성 로깅 옵트아웃 단원을 참조하세요.

Domain Name System(DNS)

Domain Name System(DNS)은 인터넷이나 프라이빗 네트워크에 연결된 컴퓨터 및 기타 리소스에 대한 계층적 분산 명명 시스템입니다. DNS는 `aws.amazon.com`과 같은 텍스트 도메인 이름을 `111.122.133.144` 형태의 숫자 IP(인터넷 프로토콜) 주소로 변환하는 데 주로 사용됩니다. 하지만 귀하 도메인의 DNS 데이터베이스는 다른 목적에 이용될 수 있는 많은 기록을 보유하고 있습니다. 예를 들어 사용자는 CNAME 기록을 사용하면 ACM에 인증서를 요청할 때 도메인 이름에 대한 소유권이나 제어 권한을 입증할 수 있습니다. 자세한 내용은 AWS Certificate Manager DNS 검증 단원을 참조하십시오.

도메인 이름

도메인 이름은 Domain Name System(DNS)에 의해 IP 주소로 변환될 수 있는 `www.example.com` 같은 텍스트 문자열입니다. 인터넷을 포함하는 컴퓨터 네트워크에는 텍스트 이름보다는 IP 주소가 사용됩니다. 도메인 이름은 마침표로 구분되는 개별 레이블로 구성됩니다.

TLD

오른쪽 끝에 있는 레이블을 TLD(최상위 도메인)라고 합니다. 일반적인 예로는 `.com`, `.net` 및 `.edu`가 있습니다. 또한 일부 국가에 등록된 TLD는 국가 이름의 약어로 국가 코드라고도 합니다. 예를 들면 영국의 경우 `.uk`, 러시아의 경우 `.ru`, 그리고 프랑스의 경우에는 `.fr`입니다. 국가 코드를 사용하는 경우 흔히 등록된 개체의 유형을 식별하기 위한 TLD에 대한 2차 수준 계층 구조가 추가됩니다. 예를 들면 `.co.uk` TLD는 영국의 상용 기업을 나타냅니다.

Apex 도메인

apex 도메인 이름은 최상위 도메인을 포함하고 확장합니다. 국가 코드를 포함하는 도메인 이름의 경우 apex 도메인은 있는 경우, 등록된 개체의 유형을 식별하는 코드와 레이블을 포함합니다. apex 도메인은 하위 도메인을 포함하지 않습니다(다음 단락 참조). `www.example.com`에서 apex 도메인의 이름

은 example.com입니다. www.example.co.uk에서 apex 도메인의 이름은 example.co.uk입니다. apex 대신에 자주 사용되는 그 밖의 이름으로는 base, root, root apex 또는 zone apex가 있습니다.

하위 도메인

하위 도메인 이름은 apex 도메인 이름 앞에 붙으며 마침표로 서로 간에 구분됩니다. 가장 일반적으로 사용되는 하위 도메인 이름은 www지만, 아무 이름이나 사용할 수 있습니다. 하위 도메인 이름에도 여러 수준을 지정할 수 있습니다. 예를 들면, jake.dog.animals.example.com에서 하위 도메인은 jake, dog 및 animals 순서대로 지정되어 있습니다.

슈퍼 도메인

하위 도메인이 속한 도메인.

FQDN

FQDN(정규화된 도메인 이름)은 네트워크 또는 인터넷에 연결된 컴퓨터, 웹 사이트 또는 기타 리소스에 대한 완전한 DNS 이름입니다. 예를 들어 aws.amazon.com은 Amazon Web Services의 FQDN입니다. FQDN에는 최상위 도메인까지의 모든 도메인이 포함됩니다. 예를 들면 [subdomain$_1$].[subdomain$_2$]...[subdomain$_n$].[apex domain].[top-level domain]은 FQDN의 일반 형식을 나타냅니다.

PQDN

정규화되지 않은 도메인 이름은 PQDN(부분 정규화된 도메인 이름)이라고 하며 모호합니다. [subdomain$_1$.subdomain$_2$.] 같은 이름은 루트 도메인을 확인할 수 없으므로 PQDN입니다.

암호화 및 암호 해독

암호화는 데이터에 기밀성을 제공하는 프로세스입니다. 암호 해독은 프로세스를 역방향으로 수행하여 원본 데이터를 복구합니다. 암호화되지 않은 데이터는 텍스트인지 여부와 상관없이 일반적으로 일반 텍스트라고 합니다. 암호화된 데이터는 일반적으로 암호 텍스트라고 합니다. 클라이언트와 서버 간 메시지의 HTTPS 암호화는 알고리즘과 키를 사용합니다. 알고리즘은 일반 텍스트 데이터가 암호 텍스트로 변환되고(암호화) 암호 텍스트가 다시 원본 일반 텍스트로 변환되는(암호 해독) 단계별 절차를 정의합니다. 키는 암호화 또는 암호 해독 프로세스 중에 알고리즘에서 사용됩니다. 키는 프라이빗이거나 퍼블릭일 수 있습니다.

정규화된 도메인 이름(FQDN)

도메인 이름을(를) 참조하세요.

Hypertext Transfer Protocol(HTTP)

Hypertext Transfer Protocol(HTTP)은 월드 와이드 웹에서의 데이터 통신의 기반입니다. 이것은 다양한 콘텐츠 유형을 교환할 수 있게 해 주는 애플리케이션 계층 프로토콜입니다. HTTP는 클라이언트-서버 모델에서 작동하며, 웹 브라우저가 일반적으로 웹 서버에서 리소스를 요청하는 클라이언트 역할을 합니다. 상태 비저장 프로토콜인 HTTP는 각 요청을 독립적으로 처리하고 이전 요청의 정보를 유지하지 않습니다.

ACM의 맥락에서 HTTP는 SSL/TLS 인증서를 발급할 때 도메인 검증에 사용할 수 있습니다. 이 프로세스에는 ACM이 도메인 소유권을 확인하기 위해 특정 HTTP 요청을 전송하는 작업이 수반됩니다. 서버가 이러한 요청에 올바르게 응답하는 기능은 도메인에 대한 제어권이 있음을 보여줍니다.

이메일 또는 DNS 검증 인증서와 달리 ACM 고객은 ACM에서 직접 HTTP 검증 인증서를 발급할 수 없습니다. 대신 이러한 인증서는 CloudFront 프로비저닝 프로세스의 일부로 자동으로 발급 및 관리됩니다. 고객은 ACM을 사용해 이러한 인증서를 보고 모니터링하고 관리할 수 있지만 초기 발급은 ACM과 CloudFront 간의 통합에 의해 처리됩니다.

HTTP는 널리 사용되지만 일반 텍스트 형식으로 데이터를 전송한다는 점에 유의해야 합니다. 보안 통신을 위해서는 HTTPS(HTTP Secure)가 사용되며, 이는 SSL/TLS 프로토콜을 사용하여 데이터를 암호화합니다. 보안 통신에 대한 자세한 내용은 보안 HTTPS 섹션을 참조하세요.

퍼블릭 키 인프라(PKI)

퍼블릭 키 인프라(PKI)는 퍼블릭 네트워크를 통한 보안 통신을 지원하는 프로세스, 기술 및 정책으로 구성된 시스템입니다. ACM의 맥락에서 PKI는 디지털 인증서의 발급, 관리 및 검증에 있어 중요한 역할을 합니다. PKI는 자유롭게 배포되는 퍼블릭 키와 소유자가 비밀로 유지하는 프라이빗 키로 구성된 한 쌍의 암호화 키를 사용합니다. 이 시스템은 디지털 엔터티의 안전한 데이터 전송, 디지털 서명 및 인증을 허용합니다.

ACM은 PKI의 여러 주요 구성 요소를 구현합니다. 이는 디지털 인증서를 발급하는 신뢰할 수 있는 타사인 인증 기관(CA)의 역할을 하여 도메인 또는 조직과 같은 엔터티에 퍼블릭 키를 바인딩합니다. ACM은 엔터티, 퍼블릭 키 및 인증서의 유효 기간에 대한 정보가 포함된 X.509 인증서를 발급합니다. 또한 발급, 갱신 및 취소를 포함한 인증서의 전체 수명 주기를 처리합니다. 인증서 요청의 적법성을 보장하기 위해 ACM은 DNS 검증 및 HTTP 검증과 같이 도메인 소유권을 검증하는 다양한 방법을 지원합니다.

ACM은 PKI를 활용하여 AWS 리소스 및 애플리케이션에 대한 보안 HTTPS 연결, 디지털 서명 및 암호화된 통신을 지원합니다. 이 인프라는 인터넷을 통해 전송되는 데이터의 기밀성, 무결성 및 신뢰성

을 유지하는 데 필수적입니다. ACM이 PKI를 구현하는 방법에 대한 자세한 내용은 AWS Certificate Manager 인증서 시작하기 섹션을 참조하세요.

루트 인증서

CA(인증 기관)는 일반적으로 명확하게 정의된 상위-하위 관계에 있는 여러 다른 CA가 포함된 계층 구조 내에 존재합니다. 하위 또는 종속 CA는 상위 CA에서 인증되어 인증서 체인을 생성합니다. 계층 구조의 최상위에 있는 CA를 루트 CA라고 하며 해당 인증서를 루트 인증서라고 합니다. 이 인증서는 일반적으로 자체 서명됩니다.

Secure Sockets Layer(SSL)

Secure Sockets Layer(SSL)와 TLS(전송 계층 보안)는 컴퓨터 네트워크에서 통신 보안을 제공하는 암호화 프로토콜입니다. TLS는 SSL의 후속 프로토콜입니다. 두 프로토콜 모두 X.509 인증서를 사용하여 서버를 인증합니다. 이 두 가지 프로토콜은 모두 클라이언트와 서버라는 두 개체 간에 전달되는 데이터를 암호화하는 데 사용되는 대칭 키를 클라이언트와 서버 간에 협상합니다.

보안 HTTPS

HTTPS는 모든 주요 브라우저와 서버에서 지원되는 보안 형식의 HTTP인 HTTP over SSL/TLS를 나타냅니다. 모든 HTTP 요청과 응답은 네트워크를 통해 전송되기 전에 암호화됩니다. HTTPS는 HTTP 프로토콜을 대칭, 비대칭 및 X.509 인증서 기반 암호화 기술과 결합합니다. HTTPS는 개방형 시스템 간 상호 연결(OSI) 모델에서 HTTP 애플리케이션 계층 아래 및 TCP 전송 계층 위에 암호화 보안 계층을 삽입하여 작동합니다. 보안 계층은 Secure Sockets Layer(SSL) 프로토콜 또는 TLS(전송 계층 보안) 프로토콜을 사용합니다.

SSL 서버 인증서

HTTPS 트랜잭션에서는 서버를 인증하기 위해 서버 인증서가 필요합니다. 서버 인증서는 인증서의 퍼블릭 키를 인증서의 보안 주체와 바인딩하는 X.509 v3 데이터 구조입니다. SSL/TLS 인증서는 인증 기관(CA)에서 서명되며 서버 이름, 유효 기간, 퍼블릭 키, 서명 알고리즘 등을 포함합니다.

대칭 키 암호화

대칭 키 암호화는 동일한 키를 사용하여 디지털 데이터를 암호화하고 암호 해독합니다. 또한 비대칭 키 암호화 기법 섹션도 참조하세요.

전송 계층 보안(TLS)

Secure Sockets Layer(SSL)을(를) 참조하세요.

신뢰

웹 브라우저가 웹 사이트의 자격 증명을 신뢰하려면 브라우저가 웹 사이트의 인증서를 확인할 수 있어야 합니다. 하지만 브라우저는 CA 루트 인증서로 알려진 소수의 인증서만 신뢰합니다. 인증 기관(CA)으로 알려진 신뢰할 수 있는 타사는 웹 사이트의 자격 증명을 확인하고 서명된 디지털 인증서를 웹 사이트 운영자에게 발행합니다. 그러면 브라우저는 디지털 서명을 점검하여 웹 사이트의 자격 증명을 확인할 수 있습니다. 확인에 성공하면 브라우저는 주소 표시줄에 자물쇠 아이콘을 표시합니다.

내 요구 사항에 적합한 AWS 인증서 서비스는 무엇입니까?

AWS 는 관리형 X.509 인증서를 배포하는 고객에게 두 가지 옵션을 제공합니다. 필요에 가장 적합한 것을 선택하세요.

1. AWS Certificate Manager (ACM) -이 서비스는 TLS를 사용하여 안전한 웹 존재가 필요한 엔터프라이즈 고객을 위한 것입니다. ACM 인증서는 Elastic Load Balancing, Amazon CloudFront, Amazon API Gateway 및 기타 통합 AWS 서비스를 통해 배포됩니다. 이러한 종류의 가장 일반적인 애플리케이션은 중요한 트래픽 요구 사항을 가진 안전한 공개 웹사이트입니다. 또한 ACM은 만료되는 인증서의 갱신을 자동화하여 보안 관리를 단순화합니다. 이 서비스에 적합한 위치에 있습니다.
2. AWS Private CA - 이 서비스는 AWS 클라우드 내부에 PKI(퍼블릭 키 인프라)를 구축하는 기업 고객을 대상으로 하며 조직 내에서 비공개로 사용할 수 있도록 고안되었습니다. 를 사용하면 자체 인증 기관(CA) 계층 구조를 생성하고 사용자 AWS Private CA, 컴퓨터, 애플리케이션, 서비스, 서버 및 기타 디바이스를 인증하기 위한 인증서를 발급할 수 있습니다. 사설 CA에서 발급한 인증서는 인터넷에서 사용할 수 없습니다. 자세한 내용은 AWS Private CA 사용 설명서를 참조하십시오.

AWS Certificate Manager 인증서 시작하기

ACM에서 퍼블릭, 프라이빗 및 가져온 인증서를 관리합니다. 인증서는 인터넷 또는 내부 네트워크 내에서 보안 통신을 설정하는 데 사용됩니다. 공개적으로 신뢰할 수 있는 인증서('ACM 인증서')를 ACM에서 직접 요청하거나 서드 파티에서 발급한 공개적으로 신뢰할 수 있는 인증서를 가져올 수 있습니다. 자체 서명된 인증서도 지원됩니다. 조직의 내부 PKI를 프로비저닝하려는 경우, Private Certificate Authority(CA)가 서명하고 AWS Private CA이(가) 생성 및 관리하는 ACM 인증서를 발급할 수 있습니다. CA는 사용자의 계정에 상주하거나 다른 계정에 의해 공유될 수 있습니다.

> **Note**
>
> 퍼블릭 ACM 인증서는 Nitro Enclave에 연결된 Amazon EC2 인스턴스에 설치할 수 있습니다. 모든 Amazon EC2 인스턴스에서 사용할 퍼블릭 인증서를 내보낼 수도 있습니다. Nitro Enclave에 연결되지 않은 Amazon EC2 인스턴스에서 독립형 웹 서버를 설정하는 방법에 대한 자세한 내용은 자습서: Amazon Linux 2에 LAMP 웹 서버 설치 또는 자습서: Amazon Linux AMI를 사용하여 LAMP 웹 서버 설치를 참조하세요.

> **Note**
>
> 프라이빗 CA가 서명한 인증서는 기본적으로 신뢰할 수 없으므로 관리자가 클라이언트 트러스트 스토어에 인증서를 설치해야 합니다.

인증서 발급을 시작하려면 AWS Management Console에 로그인하고 https://console.aws.amazon.com/acm/home ACM 콘솔을 엽니다. 소개 페이지가 나타나면 [Get Started]를 선택합니다. 그렇지 않으면 왼쪽 탐색 창에서 Certificate Manager 또는 사설 CA를 선택합니다.

주제

- 를 사용하도록 설정 AWS Certificate Manager

를 사용하도록 설정 AWS Certificate Manager

AWS Certificate Manager (ACM)을 사용하면 AWS 기반 웹 사이트 및 애플리케이션에 대한 SSL/TLS 인증서를 프로비저닝하고 관리할 수 있습니다. ACM을 사용해서 인증서를 생성하거나 가져온 다음 관리합니다. 다른 AWS 서비스를 사용하여 인증서를 웹 사이트 또는 애플리케이션에 배포해야 합니다.

ACM에 통합된 서비스에 대한 자세한 내용은 ACM에 통합된 서비스 섹션을 참조하세요. 다음 섹션에서는 ACM을 사용하기 이전에 수행해야 하는 단계를 설명합니다.

주제

- 에 가입 AWS 계정
- 관리자 액세스 권한이 있는 사용자 생성
- ACM에 대한 도메인 이름 등록
- (선택 사항) CAA 레코드 구성

에 가입 AWS 계정

이 없는 경우 다음 단계를 AWS 계정완료하여 생성합니다.

에 가입하려면 AWS 계정

1. https://portal.aws.amazon.com/billing/signup을 엽니다.
2. 온라인 지시 사항을 따르세요.

   등록 절차 중 전화 또는 텍스트 메시지를 받고 전화 키패드로 확인 코드를 입력하는 과정이 있습니다.

   에 가입하면 AWS 계정AWS 계정 루트 사용자이 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스에 액세스할 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 루트 사용자 액세스 권한이 필요한 작업을 수행하는 것입니다.

AWS 는 가입 프로세스가 완료된 후 확인 이메일을 보냅니다. 언제든지 https://aws.amazon.com/으로 이동하고 내 계정을 선택하여 현재 계정 활동을 확인하고 계정을 관리할 수 있습니다.

관리자 액세스 권한이 있는 사용자 생성

에 가입한 후 일상적인 작업에 루트 사용자를 사용하지 않도록 관리 사용자를 AWS 계정보호 AWS IAM Identity Center, AWS 계정 루트 사용자활성화 및 생성합니다.

보안 AWS 계정 루트 사용자

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 계정 소유자AWS Management Console로에 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

루트 사용자를 사용하여 로그인하는 데 도움이 필요하면 AWS 로그인 사용 설명서의 루트 사용자로 로그인을 참조하세요.

2. 루트 사용자의 다중 인증(MFA)을 활성화합니다.

   지침은 IAM 사용 설명서의 AWS 계정 루트 사용자(콘솔)에 대한 가상 MFA 디바이스 활성화를 참조하세요.

관리자 액세스 권한이 있는 사용자 생성

1. IAM Identity Center를 활성화합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 AWS IAM Identity Center설정을 참조하세요.

2. IAM Identity Center에서 사용자에게 관리 액세스 권한을 부여합니다.

   를 자격 증명 소스 IAM Identity Center 디렉터리 로 사용하는 방법에 대한 자습서는 사용 AWS IAM Identity Center 설명서의 기본값으로 사용자 액세스 구성을 IAM Identity Center 디렉터리 참조하세요.

관리 액세스 권한이 있는 사용자로 로그인

- IAM IDentity Center 사용자로 로그인하려면 IAM Identity Center 사용자를 생성할 때 이메일 주소로 전송된 로그인 URL을 사용합니다.

  IAM Identity Center 사용자를 사용하여 로그인하는 데 도움이 필요하면 AWS 로그인 사용 설명서의 AWS 액세스 포털에 로그인을 참조하세요.

추가 사용자에게 액세스 권한 할당

1. IAM Identity Center에서 최소 권한 적용 모범 사례를 따르는 권한 세트를 생성합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 Create a permission set를 참조하세요.

2. 사용자를 그룹에 할당하고, 그룹에 Single Sign-On 액세스 권한을 할당합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 그룹 추가를 참조하세요.

ACM에 대한 도메인 이름 등록

FQDN(Fully Qualified Domain Name)은 인터넷에서 최상위 수준 도메인 확장명(예: `.com` 또는 `.org`) 앞에 오는 조직 또는 개인에 대한 고유한 이름입니다. 등록된 도메인 이름이 아직 없는 경우 Amazon Route 53 또는 여러 다른 등록 대행 기관을 통해 등록할 수 있습니다. 일반적으로 등록 대행 기관의 웹 사이트로 이동하여 도메인 이름을 요청합니다. 일반적으로 도메인 이름 등록은 설정된 기간(예: 1년 또는 2년)에 유지되며 그 이후에는 갱신해야 합니다.

Amazon Route 53을 사용하여 도메인 이름을 등록하는 자세한 방법은 Amazon Route 53 개발자 가이드의 Route 53을 사용하여 도메인 이름 등록을 참조하세요.

(선택 사항) CAA 레코드 구성

CAA 레코드는 도메인 또는 하위 도메인에 대한 인증서 발급이 허용되는 인증 기관(CA)을 지정합니다. ACM에서 사용할 CAA 레코드를 생성하면 잘못된 CA가 도메인에 대한 인증서를 발급하는 것을 방지하는 데 도움이 됩니다. CAA 레코드는 인증 기관에서 지정한 보안 요구 사항(예: 도메인의 소유자임을 확인하기 위한 요구 사항) 대신 사용할 수 없습니다.

ACM은 인증서 요청 프로세스 중에 도메인을 검증한 후 인증서를 발급할 수 있도록 CAA 레코드의 존재를 확인합니다. CAA 레코드 구성은 선택 사항입니다.

CAA 레코드를 구성할 때 다음 값을 사용합니다.

flags

tag 필드의 값이 ACM에서 지원되도록 할지 여부를 지정합니다. 이 값을 0으로 설정합니다.

태그

tag 필드에는 다음 중 한 가지 값이 올 수 있습니다. iodef 필드는 현재 무시된다는 점에 유의하세요.

issue

value 필드에 지정한 ACM CA가 도메인 또는 하위 도메인에 대한 인증서를 발급하도록 승인되었음을 나타냅니다.

issuewild

value 필드에 지정한 ACM CA가 도메인 또는 하위 도메인에 대한 와일드 카드 인증서를 발급하도록 승인되었음을 나타냅니다. 와일드카드 인증서는 도메인 또는 하위 도메인 및 모든 하위 도메인에 적용됩니다. HTTP 검증을 사용할 계획인 경우, HTTP 검증이 와일드카드 인증서를 지원하지 않으므로 이 설정은 적용되지 않습니다. 와일드카드 인증서에는 대신 DNS 또는 이메일 검증을 사용하세요.

USD 상당

이 필드의 값은 tag 필드의 값에 따라 달라집니다. 이 값은 인용 부호("")로 묶어야 합니다.

tag가 issue인 경우

value 필드에는 CA 도메인 이름이 포함됩니다. 이 필드에는 Amazon CA가 아닌 다른 CA의 이름이 포함될 수 있습니다. 그러나 다음 네 가지 Amazon CA 중 하나를 지정하는 CAA 레코드가 없으면 ACM은 해당 도메인 또는 하위 도메인에 인증서를 발급할 수 없습니다.

- amazon.com
- amazontrust.com
- awstrust.com
- amazonaws.com을 위한 CNAME 별칭으로 구성해야 합니다

또한 value 필드에 세미콜론(;)을 포함하여 어떤 CA도 해당 도메인 또는 하위 도메인에 대해 인증서를 발급하도록 허용해서는 안 됨을 나타낼 수 있습니다. 어느 시점에 더는 특정 도메인에 인증서가 발급되지 않도록 하겠다고 결정을 내리게 되면 이 필드를 사용하세요.

tag가 issuewild인 경우

value 필드는 그 값이 와일드카드 인증서에 적용되는 경우를 제외하고는 tag가 issue인 경우의 값과 동일합니다.

ACM CA 값이 포함되지 않은 issuewild CAA 레코드가 있는 경우 ACM이 와일드 카드를 발급할 수 없습니다. issuewild가 없지만 ACM에 대한 issue CAA 레코드가 있는 경우 ACM이 와일드 카드를 발급할 수 있습니다.

Example CAA 레코드 예제

다음 예제에서는 도메인 이름이 먼저 나오고 뒤이어 레코드 유형(CAA)이 나옵니다. flags 필드의 값은 항상 0입니다. tags 필드는 issue 또는 issuewild일 수 있습니다. 필드가 issue이고 value 필드에 CA 서버의 도메인 이름을 입력하는 경우 CAA 레코드는 지정된 서버가 요청된 인증서를 발급할 권한이 있음을 나타냅니다. value 필드에 세미콜론(;)을 입력하면 CAA 레코드는 어떤 CA에서도 인증서를 발급할 권한이 없음을 나타냅니다. CAA 레코드의 구성은 DNS 공급자에 따라 달라집니다.

> **Important**
>
> CloudFront에서 HTTP 검증을 사용하려는 경우, HTTP 검증이 와일드카드 인증서를 지원하지 않으므로 issuewild 레코드를 구성할 필요가 없습니다. 와일드카드 인증서의 경우 DNS 또는 이메일 검증을 대신 사용합니다.

```
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 레코드를 추가 또는 수정하는 방법에 대한 자세한 정보는 DNS 공급자에게 문의하세요. Route 53은 CAA 레코드를 지원합니다. Route 53이 DNS 공급자인 경우 레코드 생성에 대한 자세한 내용은 CAA 형식을 참조하세요.

AWS Certificate Manager 퍼블릭 인증서

퍼블릭 인증서를 요청한 후에는 AWS Certificate Manager 퍼블릭 인증서에 대한 도메인 소유권 검증에서 설명한 대로 도메인 소유권을 검증해야 합니다.

퍼블릭 ACM 인증서는 X.509 표준을 따르며 다음 제한 사항이 적용됩니다.

- 이름: DNS를 준수하는 주체 이름을 사용해야 합니다. 자세한 내용은 도메인 이름 단원을 참조하십시오.
- 알고리즘: 암호화를 위해서는 인증서 프라이빗 키 알고리즘이 2048비트 RSA, 256비트 ECDSA 또는 384비트 ECDSA 중 하나에 해당해야 합니다.
- 만료: 각 인증서는 198일 동안 유효합니다.
- 갱신: ACM은 만료 45일 전에 퍼블릭 인증서를 자동으로 갱신하려고 시도합니다.

> **Note**
>
> 퍼블릭 ACM 인증서는 Nitro Enclave에 연결된 Amazon EC2 인스턴스에 설치할 수 있습니다. 모든 Amazon EC2 인스턴스에서 사용할 퍼블릭 인증서를 내보낼 수도 있습니다. Nitro Enclave에 연결되지 않은 Amazon EC2 인스턴스에서 독립형 웹 서버를 설정하는 방법에 대한 자세한 내용은 자습서: Amazon Linux 2에 LAMP 웹 서버 설치 또는 자습서: Amazon Linux AMI를 사용하여 LAMP 웹 서버 설치를 참조하세요.

관리자는 ACM 조건부 키 정책을 사용하여 최종 사용자가 새 인증서를 발급하는 방법을 제어할 수 있습니다. 이러한 조건부 키를 사용하면 도메인, 유효성 검사 방법 및 인증서 요청과 관련된 기타 속성에 제한을 둘 수 있습니다. 인증서를 요청할 때 문제가 발생하면 인증서 요청 문제 해결 단원을 참조하세요.

를 사용하여 프라이빗 PKI에 대한 인증서를 요청하려면 섹션을 AWS Private CA참조하세요에서 프라이빗 인증서 요청 AWS Certificate Manager.

주제

- AWS Certificate Manager 퍼블릭 인증서 특성 및 제한 사항
- 에서 퍼블릭 인증서 요청 AWS Certificate Manager
- AWS Certificate Manager 내보내기 가능한 퍼블릭 인증서
- AWS Certificate Manager 퍼블릭 인증서에 대한 도메인 소유권 검증

AWS Certificate Manager 퍼블릭 인증서 특성 및 제한 사항

ACM이 제공하는 퍼블릭 인증서에는 다음과 같은 특성과 제한 사항이 있습니다. 이러한 사항은 ACM에서 제공하는 인증서에만 적용되고, 가져온 인증서에는 적용되지 않습니다.

브라우저 및 애플리케이션 신뢰

Google Chrome, Microsoft Edge, Mozilla Firefox, Apple Safari 등 모든 주요 브라우저에서 ACM 인증서를 신뢰합니다. 브라우저는 TLS에 의해 ACM 인증서를 사용하여 사이트에 연결될 때 자물쇠 아이콘을 표시합니다. Java는 ACM 인증서도 신뢰합니다.

인증 기관 및 계층 구조

ACM을 통해 요청하는 공인 인증서는 아마존 관리형 퍼블릭 인증 기관(CA)인 Amazon Trust Services에서 제공합니다. Amazon Root CA 1~4는 Starfield G2 Root Certificate Authority - G2에 의해 상호 서명됩니다. Starfield 루트는 Android(최신 Gingerbread 버전) 및 iOS(버전 4.1 이상)에서 신뢰할 수 있습니다. Amazon 루트는 iOS 11 이상에서 신뢰할 수 있습니다. Amazon 또는 Starfield 루트가 포함된 모든 브라우저, 애플리케이션 또는 OS는 ACM 퍼블릭 인증서를 신뢰합니다.

ACM은 인증서 유형(RSA 또는 ECDSA)을 기반으로 무작위 할당된 중간 CA를 통해 고객에게 리프 또는 최종 엔터티 인증서를 발급합니다. 이 무작위 선택으로 인해 ACM은 중간 CA 정보를 제공하지 않습니다.

도메인 검증(DV)

ACM 인증서는 도메인 검증을 거쳐 도메인 이름만 식별합니다. ACM 인증서를 요청할 때는 지정된 모든 도메인의 소유권 또는 제어권을 증명해야 합니다. 이메일 혹은 DNS를 사용해 소유권을 검증할 수 있습니다. 자세한 내용은 AWS Certificate Manager 이메일 검증 및 AWS Certificate Manager DNS 검증 섹션을 참조하세요.

HTTP 검증

ACM은 CloudFront에서 사용할 퍼블릭 TLS 인증서를 발급할 때 도메인 소유권 확인을 위한 HTTP 검증을 지원합니다. 이 방식은 HTTP 리디렉션을 사용하여 도메인 소유권을 증명하고 DNS 검증과 유사한 자동 갱신을 제공합니다. HTTP 검증은 현재 CloudFront Distribution Tenants 기능을 통해서만 사용할 수 있습니다.

HTTP 리디렉션

HTTP 검증을 위해 ACM은 `RedirectFrom` URL 및 `RedirectTo` URL을 제공합니다. 도메인 제어권을 증명하려면 `RedirectFrom`에서 `RedirectTo`로의 리디렉션을 설정해야 합니다.

`RedirectFrom` URL에는 검증된 도메인이 포함되는 반면, `RedirectTo`는 고유한 검증 토큰이 포함된 CloudFront 인프라의 ACM 제어 위치를 가리킵니다.

에서 관리

다른 서비스에서 관리하는 ACM의 인증서는 `ManagedBy` 필드에 해당 서비스의 ID를 보여줍니다. CloudFront에서 HTTP 검증을 사용하는 인증서의 경우 이 필드에 "CLOUDFRONT"가 표시됩니다. 이러한 인증서는 CloudFront를 통해서만 사용할 수 있습니다. `ManagedBy` 필드는 DescribeCertificate 및 ListCertificates API와 ACM 콘솔의 인증서 인벤토리 및 세부 정보 페이지에 표시됩니다.

`ManagedBy` 필드는 "Can be used with" 속성과 상호 배타적입니다. CloudFront 관리형 인증서의 경우 다른 AWS 서비스를 통해 새 사용량을 추가할 수 없습니다. 이러한 인증서는 CloudFront API를 통해서만 더 많은 리소스에 사용할 수 있습니다.

중간 및 루트 CA 교체

Amazon은 복원력이 뛰어난 인증서 인프라를 유지하기 위해 통지 없이 중간 CA를 중단할 수 있습니다. 이러한 변경 사항은 고객에게 영향을 주지 않습니다. 자세한 내용은 "Amazon introduces dynamic intermediate certificate authorities(Amazon, 동적 중간 인증 기관 도입)"를 참조하세요.

Amazon이 루트 CA를 중단하는 경우, 필요에 따라 빠르게 변경이 이루어집니다. Amazon은 , Health Dashboard이메일, 기술 계정 관리자에게 연락 등 사용 가능한 모든 방법을 사용하여 AWS 고객에게 알립니다.

해지를 위한 방화벽 액세스

취소된 최종 엔터티 인증서는 OCSP 및 CRL을 사용하여 취소 정보를 확인하고 게시합니다. 이러한 메커니즘을 허용하기 위해 일부 고객 방화벽에 추가 규칙이 필요할 수 있습니다.

다음 URL 와일드카드 패턴을 사용하여 취소 트래픽을 식별합니다.

- OCSP

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

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

- CRL

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

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

별표(*) 와일드카드는 하나 이상의 영숫자 문자에 해당하고, 물음표(?)는 단일 영숫자를 나타내고, 해시 마크(#)는 숫자를 나타냅니다.

키 알고리즘

인증서는 반드시 알고리즘과 키 크기를 지정해야 합니다. ACM은 다음과 같은 RSA 및 ECDSA 퍼블릭 키 알고리즘을 지원합니다.

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

ACM에서는 별표(*)로 표시된 알고리즘을 사용하여 새 인증서를 요청할 수 있습니다. 기타 알고리즘은 가져온 인증서에만 사용됩니다.

> **Note**
>
> AWS Private CA CA에서 서명한 프라이빗 PKI 인증서의 경우 서명 알고리즘 패밀리(RSA 또는 ECDSA)가 CA의 보안 키 알고리즘 패밀리와 일치해야 합니다.

ECDSA 키는 유사한 보안 수준의 RSA 키보다 작고 컴퓨팅 효율성이 뛰어나지만 모든 네트워크 클라이언트가 ECDSA를 지원하는 것은 아닙니다. NIST에서 발췌한 이 표는 RSA 및 ECDSA 키 크기(비트)를 비교하여 동등한 보안 강도를 제공합니다.

알고리즘과 키의 보안 비교

보안 강도	RSA 키 크기	ECDSA 키 크기
128	3072	256
192	7680	384
256	15360	521

2의 거듭제곱으로 표현되는 보안 강도는 암호를 해제하는 데 필요한 추측의 수와 연관되어 있습니다. 예를 들어 3072비트 RSA 키와 256비트 ECDSA 키는 모두 2^{128}회 이하의 추측으로 검색이 가능합니다.

알고리즘 선택에 도움이 필요하면 AWS 블로그 게시물에서 ECDSA 인증서를 평가하고 사용하는 방법을 참조하세요 AWS Certificate Manager.

> **Important**
>
> 통합 서비스는 리소스에 대해 지원되는 알고리즘 및 키 크기만 허용합니다. 지원은 인증서를 IAM으로 가져오는지 ACM으로 가져오는지에 따라 달라집니다. 세부 정보는 각 서비스 문서를 참조하세요.
>
> - Elastic Load Balancing의 경우 Application Load Balancer를 위한 HTTPS 리스너를 참조하세요.
> - CloudFront의 경우 지원되는 SSL/TLS 프로토콜 및 암호를 참조하세요.

관리형 갱신 및 배포

ACM은 ACM 인증서의 갱신 및 프로비저닝을 관리합니다. 자동 갱신은 잘못된 구성, 취소 또는 만료된 인증서로 인한 가동 중지를 방지하는 데 도움이 됩니다. 자세한 내용은 에서 관리형 인증서 갱신 AWS Certificate Manager 단원을 참조하십시오.

여러 도메인 이름

각 ACM 인증서에는 하나 이상의 정규화된 도메인 이름(FQDN)이 포함되어 있어야 하며 이름을 추가할 수 있습니다. 예를 들어 `www.example.com`에 대한 인증서에는 `www.example.net`도 포함될 수 있습니다. 이는 베어 도메인(zone apex 또는 네이키드 도메인)에도 적용됩니다. www.example.com에 대한 인증서를 요청하고 example.com을 추가할 수 있습니다. 자세한 내용은 AWS Certificate Manager 퍼블릭 인증서 단원을 참조하십시오.

퓨니코드

다국어 도메인 이름에 대한 다음 Punycode 요구 사항을 충족해야 합니다.

1. '<character><character>--' 패턴으로 시작하는 도메인 이름은 'xn--'과 일치해야 합니다.
2. 'xn--'으로 시작하는 도메인 이름도 유효한 다국어 도메인 이름이어야 합니다.

Punycode 예제

도메인 이름	#1 충족	#2 충족	허용	Note
example.com	해당 사항 없음	해당 사항 없음	✓	'<character><character>--'로 시작하지 않음
a--example.com	해당 사항 없음	해당 사항 없음	✓	'<character><character>--'로 시작하지 않음
abc--example.com	해당 사항 없음	해당 사항 없음	✓	'<character><character>--'로 시작하지 않음
xn—xyz.com	예	예	✓	유효한 다국어 도메인 이름(简.com으로 확인)
xn--example.com	예	아니요	✗	유효한 다국어 도메인 이름이 아님
ab--example.com	아니요	아니요	✗	'xn--'으로 시작해야 함

유효성 기간

ACM 인증서는 198일 동안 유효합니다.

와일드카드 이름

ACM은 도메인 이름에 별표(*)를 사용하여 동일한 도메인에서 여러 사이트를 보호할 수 있는 와일드카드 인증서를 생성할 수 있게 해 줍니다. 예를 들어 `*.example.com`은 `www.example.com` 및 `images.example.com`을 보호합니다.

와일드카드 인증서에서 별표(*)는 도메인 이름의 맨 왼쪽에 와야 하며 하나의 하위 도메인 수준만 보호합니다. 예를 들어, `*.example.com`은 `login.example.com` 및 `test.example.com`을 보호하지만 `test.login.example.com`은 보호하지 않습니다. 또한 `*.example.com`은 의 하위 도메인만 보호하고 베어 또는 apex 도메인(`example.com`)은 보호하지 않습니다. `example.com` 및

*.example.com과 같은 여러 도메인 이름을 지정하여 베어 도메인과 해당 하위 도메인 모두에 대한 인증서를 요청할 수 있습니다.

> **Important**
>
> 참고로, CloudFront를 사용하는 경우 HTTP 검증은 와일드카드 인증서를 지원하지 않습니다. 와일드카드 인증서의 경우 DNS 검증 또는 이메일 검증을 사용해야 합니다. 자동 인증서 갱신을 지원하는 DNS 검증을 사용하는 것이 좋습니다.

에서 퍼블릭 인증서 요청 AWS Certificate Manager

ACM 콘솔 AWS CLI또는 API에서 AWS Certificate Manager 퍼블릭 인증서를 요청할 수 있습니다. 이러한 인증서를 통합과 함께 AWS 서비스 사용하거나 외부에서 사용할 수 있도록 내보낼 수 있습니다 AWS 클라우드.

다음 목록은 퍼블릭 인증서와 익스포터블 퍼블릭 인증서의 차이점을 설명하고 있습니다.

퍼블릭 인증서

Elastic Load Balancing, Amazon CloudFront 및 Amazon API Gateway와 AWS 서비스 같이 통합된와 함께 ACM 퍼블릭 인증서를 사용합니다. 자세한 내용은 ACM에 통합된 서비스 단원을 참조하십시오.

> **Note**
>
> 2025년 6월 17일 이전에 생성된 ACM 퍼블릭 인증서는 내보낼 수 없습니다.

익스포터블 퍼블릭 인증서

내보내기 가능한 퍼블릭 인증서는 통합에서 작동 AWS 서비스 하며 외부에서도 사용할 수 있습니다 AWS 클라우드. 자세한 내용은 AWS Certificate Manager 내보내기 가능한 퍼블릭 인증서 및 ACM에 통합된 서비스 섹션을 참조하세요. 퍼블릭 인증서를 내보낼 수 있으려면 새 ACM 퍼블릭 인증서를 생성하고 익스포터블을 활성화해야 합니다.

다음 섹션에서는 퍼블릭 ACM 인증서를 요청, 내보내기 및 취소하는 방법을 설명합니다.

주제

콘솔을 사용하여 퍼블릭 인증서 요청

ACM 퍼블릭 인증서를 요청하려면(콘솔)

1. AWS Management Console에 로그인하고 https://console.aws.amazon.com/acm/home ACM 콘솔을 엽니다.

   인증서 요청을 선택합니다.

2. 도메인 이름(Domain names) 섹션에서 도메인 이름을 입력합니다.

   www.example.com 같은 FQDN(Fully Qualified Domain Name)이나 **example.com** 같은 베어 또는 apex 도메인 이름을 사용할 수 있습니다. 맨 왼쪽에서 별표(*)를 와일드카드로 사용하여 동일한 도메인 내에서 여러 사이트 이름을 보호할 수도 있습니다. 예를 들어 ***.example.com**은 **corp.example.com** 및 **images.example.com**을 보호합니다. 와일드카드 이름은 주체(Subject) 필드와 ACM 인증서의 주체 대체 이름(Subject Alternative Name) 확장에 표시됩니다.

   와일드카드 인증서를 요청할 때 별표(*)는 도메인 이름의 맨 왼쪽에 와야 하며 하나의 하위 도메인 수준만 보호할 수 있습니다. 예를 들어 ***.example.com**은 **login.example.com** 및 **test.example.com**을 보호할 수 있지만 **test.login.example.com**은 보호할 수 없습니다. 또한 ***.example.com**은 **example.com**의 하위 도메인만 보호하고 베어 또는 apex 도메인(**example.com**)은 보호하지 못합니다. 둘 모두를 보호하려면 다음 단계를 참조하세요.

   > **Note**
   >
   > RFC 5280에 따라, 이 단계에서 입력하는 도메인 이름(일반 이름)의 길이는 마침표를 포함하여 64 옥텟(자)을 초과할 수 없습니다. 다음 단계에서 보듯이 사용자가 제공하는 각 주체 대체 이름(SAN)의 길이는 최대 253옥텟입니다.

   - 다른 이름을 추가하려면 이 인증서에 다른 이름 추가를 선택하고 텍스트 상자에 이름을 입력합니다. 이렇게 하면 베어 또는 apex 도메인(예: **example.com**)과 하위 도메인(예: ***.example.com**)을 보호하는 데 유용합니다.

3. ACM 익스포터블 퍼블릭 인증서를 생성하려면 내보내기 활성화 옵션을 선택합니다. 인증서의 프라이빗 키에 액세스하여 AWS 클라우드외부에서 사용할 수 있습니다. 자세한 내용은 AWS Certificate Manager 내보내기 가능한 퍼블릭 인증서 단원을 참조하십시오.
4. 검증 방법(Validation method) 섹션에서 필요에 따라 DNS 검증 - 권장(DNS validation - recommended) 또는 이메일 검증(Email validation)을 선택합니다.

   > **Note**
   >
   > 사용자 DNS 환경 설정을 편집할 수 있다면, 이메일 검증보다는 DNS 검증을 사용하는 것을 권장합니다. DNS 검증은 이메일 검증에 비해 다양한 이점이 있습니다. AWS Certificate Manager DNS 검증을(를) 참조하세요.

   ACM은 인증서 요청 시 인증서를 발급하기 전에 귀사가 도메인 이름을 소유하거나 관리 권한을 보유하고 있는지 검증합니다. 이메일 검증 혹은 DNS 검증을 사용할 수 있습니다.

   a. 이메일 검증을 선택하면 ACM은 도메인 이름 필드에 지정한 도메인으로 검증 이메일을 전송합니다. 검증 도메인을 지정하면 ACM이 대신 해당 검증 도메인으로 이메일을 전송합니다. 이메일 검증에 대한 자세한 내용은 AWS Certificate Manager 이메일 검증를 참조하세요.
   b. DNS 검증을 선택하면 사용자는 사용자 DNS 환경 설정을 위해 ACM이 제공한 CNAME 기록을 추가하기만 하면 됩니다. DNS 검증에 대한 자세한 내용은 AWS Certificate Manager DNS 검증 단원을 참조하세요.
5. 키 알고리즘 섹션에서 알고리즘을 선택합니다.
6. 태그 페이지에서 선택 사항으로 인증서에 태그를 지정할 수 있습니다. 태그는 AWS 리소스를 식별하고 구성하기 위한 메타데이터 역할을 하는 키-값 페어입니다. ACM 태그 파라미터 목록과 생성 후 인증서에 태그를 추가하는 방법에 대한 지침은 AWS Certificate Manager 리소스 태그 지정 섹션을 참조하세요.

   태그 추가를 마치면 요청(Request)을 선택합니다.
7. 요청이 처리되면 콘솔에서 인증서 목록으로 돌아가고, 여기에 새 인증서의 정보가 표시됩니다.

   인증서 요청 실패 문제 해결 주제에 나와 있는 이유 중의 하나에 의해 실패하지 않는 한, 인증서는 요청을 받으면 검증 보류(Pending validation) 상태에 들어갑니다. ACM이 72시간 동안 인증서의 유효성 검증을 반복적으로 시도한 다음 시간이 초과됩니다. 인증서에 실패 또는 검증 시간 초과 상태가 표시되는 경우 요청을 삭제하고 DNS 검증 또는 이메일 검증으로 문제를 수정한 다음 다시 시도하세요. 검증에 성공한 경우에는 인증서가 발급 완료 상태에 들어갑니다.

> **Note**
>
> 목록을 정렬한 방법에 따라 찾고 있는 인증서가 즉시 표시되지 않을 수 있습니다. 오른쪽의 검은색 삼각형을 클릭하여 순서를 변경할 수 있습니다. 오른쪽 상단의 페이지 번호를 사용하여 여러 페이지의 인증서를 탐색할 수도 있습니다.

CLI를 사용하여 퍼블릭 인증서 요청

request-certificate 명령을 사용하여 명령줄에서 새 퍼블릭 ACM 인증서를 요청합니다. 검증 방법에 대해 선택 가능한 값은 DNS와 이메일입니다. 키 알고리즘에 대해 선택 가능한 값은 RSA_2048(파라미터가 명시적으로 제공되지 않은 경우 기본값), EC_prime256v1 및 EC_secp384r1입니다.

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

이 명령은 새 공용 인증서의 Amazon 리소스 이름(ARN)을 출력합니다.

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

AWS Certificate Manager 내보내기 가능한 퍼블릭 인증서

AWS Certificate Manager 내보내기 가능한 퍼블릭 인증서를 사용하면 Amazon EC2 인스턴스, 컨테이너 및 온프레미스 호스트를 포함하여 어디서나 SSL/TLS 인증서를 프로비저닝, 관리 및 배포할 수 있습니다. 이 기능은 ACM에서 발급한 퍼블릭 인증서를 통합 이상으로 확장 AWS 서비스하여 전체 인프라에서 인증서를 중앙 집중식으로 제어할 수 있습니다.

이점

다음은 ACM 익스포터블 퍼블릭 인증서의 이점을 간략하게 설명합니다.

- 간소화된 인증서 관리: ACM을 사용하여 모든 리소스의 인증서를 중앙에서 관리합니다.

- 더 빠른 인증서 발급: 더 짧은 시간 내에 인증서를 액세스하고 사용합니다.
- 자동 갱신: ACM은 인증서 갱신을 자동으로 처리하고 새 인증서를 배포할 준비가 되면 알려줍니다. 자세한 내용은 ACM에 대한 Amazon EventBridge 지원 단원을 참조하십시오.
- 비용 효율성: 생성한 익스포터블 퍼블릭 인증서에 대해서만 비용을 지불합니다.
- 유연한 배포: 표준 SSL/TLS 인증서를 지원하는 모든 서버 또는 애플리케이션에 인증서를 사용합니다.

ACM 익스포터블 퍼블릭 인증서의 작동 방식

다음은 ACM 익스포터블 퍼블릭 인증서의 작동 방식을 간략하게 설명합니다.

1. ACM을 통해 도메인에 대한 익스포터블 인증서를 요청합니다.
2. DNS 또는 이메일 검증을 사용하여 도메인 소유권을 검증합니다.
3. 인증서, 프라이빗 키 및 인증서 체인을 내보냅니다.
4. 서버 또는 애플리케이션에 인증서를 배포합니다.
5. ACM은 갱신을 관리하고 새 인증서를 사용할 수 있을 때 알림을 보냅니다.

보안 고려 사항

다음은 ACM 익스포터블 퍼블릭 인증서를 사용할 때의 보안 고려 사항입니다. 자세한 내용은 의 데이터 보호 AWS Certificate Manager 단원을 참조하십시오.

- 내보낸 프라이빗 키를 보안 스토리지 및 액세스 제어를 사용하여 보호합니다.
- 키 손상이 의심되는 경우 ACM의 취소 기능을 사용합니다.
- 갱신된 인증서를 배포할 때 적절한 키 교체 절차를 구현합니다.

제한 사항

다음은 몇 가지 ACM 인증서 제한 사항입니다.

- 인증서의 유효 기간은 198일입니다.
- ACM은 만료 날짜 45일 전에 만료되도록 설정된 인증서를 갱신합니다.
- 내보낸 인증서의 배포 프로세스를 관리해야 합니다.

가격 책정

생성한 내보내기 가능한 퍼블릭 SSL/TLS 인증서에는 추가 요금이 부과됩니다 AWS Certificate Manager. 최신 ACM 요금 정보는 AWS 웹 사이트의 AWS Certificate Manager 서비스 요금 페이지를 참조하세요.

모범 사례

다음은 ACM 인증서를 사용할 때의 몇 가지 모범 사례입니다.

- 인증서가 갱신되면 즉시 사용을 시작해야 합니다.
- 갱신된 인증서에 대한 자동 배포 프로세스를 테스트하고 구현합니다.
- Amazon EventBridge 지표 및 경보를 사용하여 인증서 배포를 모니터링합니다.

AWS Certificate Manager 퍼블릭 인증서 내보내기

다음 절차에서는 ACM 콘솔에서 ACM 퍼블릭 인증서를 내보내는 방법을 안내합니다. 또는 `export-certificate` AWS CLI ExportCertificate API 작업을 사용할 수도 있습니다.

> **Note**
>
> 2025년 6월 17일 이전에 생성된 ACM 퍼블릭 인증서는 내보낼 수 없습니다.

퍼블릭 인증서 내보내기(콘솔)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/acm/ ACM 콘솔을 엽니다.
2. 인증서 나열을 선택하고 내보낼 인증서의 확인란을 선택합니다.
   - 또는 인증서를 선택할 수 있습니다. 인증서 세부 정보 페이지에서 내보내기를 선택합니다.
3. 추가 작업을 선택한 다음 내보내기를 선택합니다.
4. 프라이빗 키의 암호를 입력하고 확인합니다.
5. 인증서 파일을 다운로드하거나 복사할 수 있습니다.

> **Note**
>
> ACM 콘솔에서 .pem 인증서 파일을 내보낼 수 있습니다. .pem 파일을 .ppk와 같은 다른 파일 형식으로 변환할 수 있습니다. 자세한 내용은 이 re:Post 문서를 참조하세요.

퍼블릭 인증서 내보내기(AWS CLI)

`export-certificate` AWS CLI 명령 또는 ExportCertificate API 작업을 사용하여 퍼블릭 인증서와 프라이빗 키를 내보냅니다. 명령을 실행할 때 암호를 할당해야 합니다. 보안을 강화하려면 파일 편집기를 사용하여 파일에 암호를 저장한 다음 파일을 제공하여 암호를 제공합니다. 이렇게 하면 암호가 명령 레코드에 저장되지 않으며 암호를 입력할 때 다른 사람이 암호를 볼 수 없습니다.

> **Note**
>
> 암호가 포함된 파일은 행 종결자로 끝나지 않아야 합니다. 다음과 같은 암호 파일을 확인할 수 있습니다.

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

다음 예제는 명령 출력을 `jq`로 파이핑하여 PEM 형식 지정을 적용합니다.

```
[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)"'
```

이것은 base64로 인코딩된, PEM 형식 인증서를 출력하며, 다음 축약된 예에서와 같이 인증서 체인과 프라이빗 키도 포함합니다.

```
-----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-----
```

모든 요소를 파일로 출력하려면 이전 예제에 > 리디렉션을 추가하여 다음 명령을 출력합니다.

```
$ 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 인증서를 사용하여 Kubernetes 워크로드 보호

Kubernetes용 AWS 컨트롤러(ACK)와 함께 AWS Certificate Manager 내보내기 가능한 퍼블릭 인증서를 사용하여 ACM에서 Kubernetes 워크로드로 퍼블릭 TLS 인증서를 발급하고 내보낼 수 있습니다. 이 통합을 통해 Amazon Elastic Kubernetes Service(Amazon EKS) 포드를 보호하고 Kubernetes 수신 시 TLS를 종료할 수 있습니다. 시작하려면 GitHub의 Kubernetes용 ACM 컨트롤러를 참조하세요.

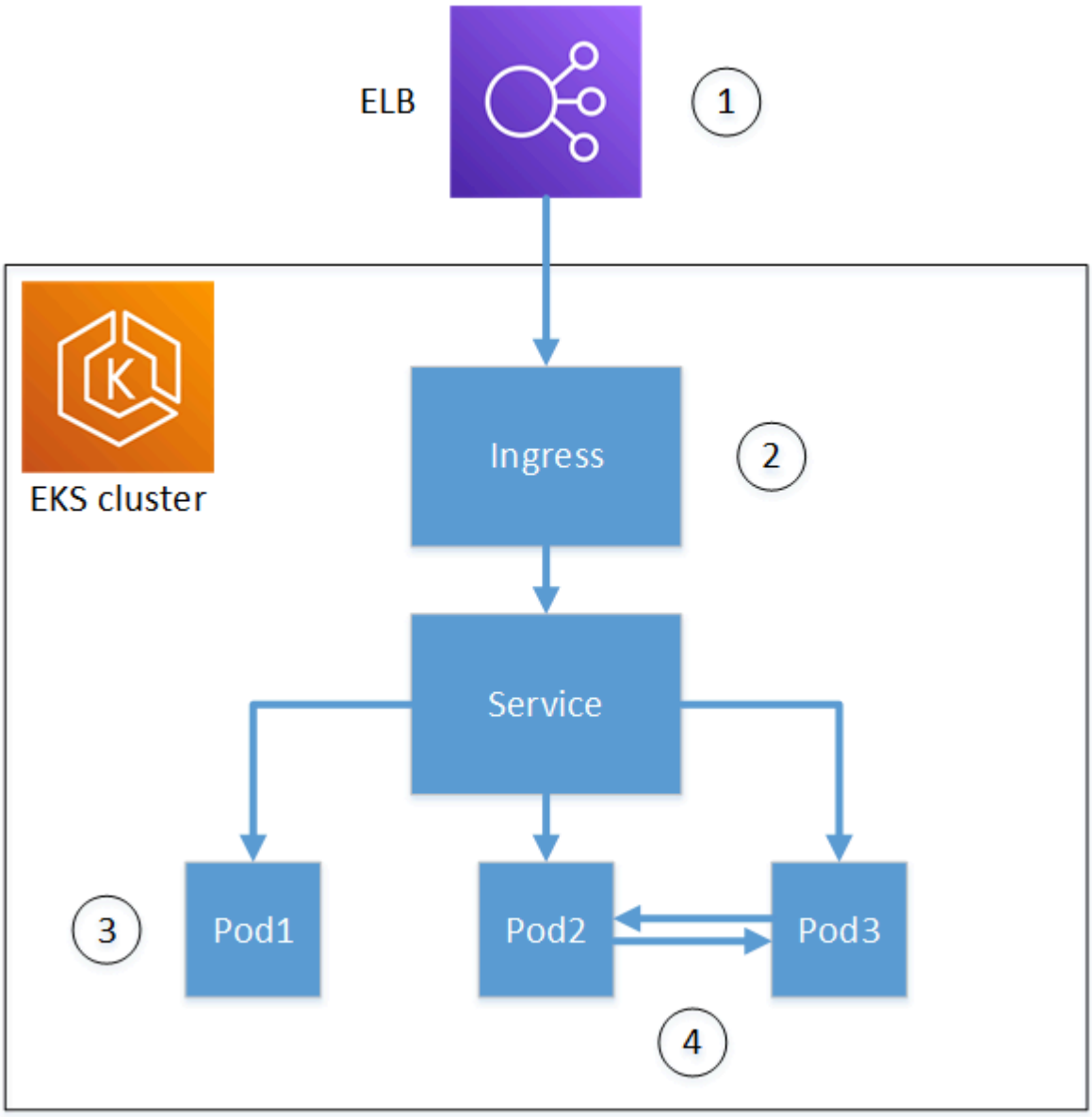
AWS Kubernetes용 컨트롤러(ACK)는 Kubernetes API를 확장하여 네이티브 Kubernetes 매니페스트를 사용하여 AWS 리소스를 관리합니다. ACM용 ACK 서비스 컨트롤러는 Kubernetes 워크플로 내에서 자동화된 인증서 수명 주기 관리를 제공합니다. Kubernetes에서 ACM 인증서 리소스를 생성하면 ACK 컨트롤러는 다음 작업을 수행합니다.

1. 인증서 서명 요청(CSR)을 생성하는 ACM에서 인증서를 요청합니다.
2. 도메인 검증이 완료되고 ACM이 인증서를 발급할 때까지 기다립니다.
3. `exportTo` 필드가 지정된 경우는 발급된 인증서와 프라이빗 키를 내보내고 지정된 Kubernetes 보안 암호에 저장합니다.
4. `exportTo` 필드를 지정하고 인증서를 갱신할 수 있는 경우는 만료 전에 Kubernetes 보안 암호를 갱신된 인증서로 업데이트합니다.

공개적으로 발급된 인증서는 ACM이 발급하기 전에 도메인 검증이 필요합니다. Amazon Route 53용 ACK 서비스 컨트롤러를 사용하여 호스팅 영역에 필요한 DNS 검증 CNAME 레코드를 자동으로 생성할 수 있습니다.

인증서 사용 옵션

다음과 같은 몇 가지 방법으로 Kubernetes에서 ACM 인증서를 사용할 수 있습니다.



1. 로드 밸런서 종료(내보내기 제외): ACK를 통해 인증서를 발급하고 이를 사용하여 AWS 로드 밸런서에서 TLS를 종료합니다. 인증서는 ACM에 남아 있으며 AWS Load Balancer 컨트롤러에서 자동으로 검색됩니다. 이 접근 방식은 인증서를 내보낼 필요가 없습니다.
2. 수신 종료(내보내기 포함): ACM에서 인증서를 내보내고 수신 수준에서 TLS 종료를 위해 Kubernetes 보안 암호에 저장합니다. 이렇게 하면 Kubernetes 워크로드 내에서 직접 인증서를 사용할 수 있습니다.

> **Note**
>
> 프라이빗 인증서가 필요한 사용 사례는 cert-manager 플러그인인 AWS Kubernetes용 프라이빗 CA 커넥터를 참조하세요.

사전 조건

ACM용 ACK 서비스 컨트롤러를 설치하기 전에 다음이 있는지 확인합니다.

- Kubernetes 클러스터.
- Helm이 설치되었습니다.
- 클러스터와 통신하도록 구성된 `kubectl`.
- `eksctl` EKS에서 포드 자격 증명 연결을 구성하기 위해 설치됩니다.

ACM용 ACK 서비스 컨트롤러 설치

Helm을 사용하여 Amazon EKS 클러스터에 ACM용 ACK 서비스 컨트롤러를 설치합니다.

1. ACK 컨트롤러의 네임스페이스를 생성합니다.

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

2. ACK 컨트롤러에 대한 포드 자격 증명 연결을 생성합니다. *CLUSTER_NAME*을 클러스터 이름으로 바꾸고 *REGION*을 AWS 리전으로 바꿉니다.

```
$ 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 퍼블릭 레지스트리에 로그인합니다.

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

4. ACM용 ACK 서비스 컨트롤러를 설치합니다. *REGION*을 해당 AWS 리전으로 바꿉니다.

```
$ 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. 컨트롤러가 실행 중인지 확인합니다.

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

포드 자격 증명 연결에 대한 자세한 내용은 Amazon EKS 사용 설명서의 EKS 포드 자격 증명을 참조하세요.

예: 수신 시 TLS 종료

다음 예제에서는 ACM 인증서를 내보내고 이를 사용하여 Kubernetes 수신 수준에서 TLS를 종료하는 방법을 보여줍니다. 이 구성은 ACM 인증서를 생성하여 Kubernetes 보안 암호로 내보내고 TLS 종료에 인증서를 사용하도록 수신 리소스를 구성합니다.

이 예시는 다음과 같이 설정되어 있습니다.

- 내보낸 인증서를 저장하기 위해 보안 암호가 생성됩니다(`exported-cert-secret`).
- ACK 인증서 리소스는 도메인에 대해 ACM에서 인증서를 요청하고 `exported-cert-secret` 보안 암호로 내보냅니다.
- 수신 리소스는 `exported-cert-secret`를 참조하여 수신 트래픽에 대한 TLS를 종료합니다.

`${HOSTNAME}`을 사용자 이름으로 바꿉니다.

```
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
```

배포되면 ACM용 ACK 서비스 컨트롤러는 갱신을 포함하여 인증서 수명 주기를 자동으로 관리합니다. ACM이 인증서를 갱신하면 컨트롤러는 `exported-cert-secret` 보안 암호를 새 인증서로 업데이트하여 수신이 수동 개입 없이 유효한 인증서를 계속 사용하도록 합니다.

AWS Certificate Manager 퍼블릭 인증서 취소

ACM 콘솔 AWS CLI또는 API 작업을 사용하여 AWS Certificate Manager 내보낼 수 있는 퍼블릭 인증서를 취소할 수 있습니다.

> **Warning**
> 인증서가 취소된 후에는 인증서를 재사용할 수 없습니다. 인증서 취소는 영구적입니다.

조직의 정책을 준수하거나 키 손상을 완화하기 위해 인증서를 취소해야 할 수 있습니다. 인증서를 취소하려면 사유가 필요합니다. 다음과 같은 사유를 사용할 수 있습니다.

- 지정 안 함
- 소속 변경
- 대체됨
- 작업 중단

자세한 내용은 Amazon Trust Services Certificate Subscriber Agreement 및 Amazon Trust Service를 참조하세요.

AWS 는 인증서 취소를 확인하기 위한 두 가지 서비스인 온라인 인증서 상태 프로토콜(OCSP)과 인증서 취소 목록을 제공합니다. OCSP를 사용하면 클라이언트가 상태를 실시간으로 반환하는 신뢰할 수 있는 해지 데이터베이스를 쿼리합니다. OCSP는 모두 인증서에 포함된 검증 정보를 기반으로 합니다.

고려 사항

다음은 인증서를 취소하기 전에 고려해야 할 사항입니다.

- 이전에 내보낸 인증서만 취소할 수 있습니다.
- 내보낼 수 없는 퍼블릭 인증서는 취소할 수 없습니다. 이러한 인증서가 더 이상 필요하지 않은 경우, 대신 삭제해야 합니다.
- 더 이상 인증서가 필요하지 않은 경우 인증서를 취소하는 대신 인증서를 삭제해야 합니다.
- 인증서 취소 프로세스는 전역적입니다. 취소하기로 선택한 모든 유효한 인증서는 연결된 ARN과 함께 취소됩니다.
- 인증서 취소는 영구적입니다. 취소된 인증서는 재사용할 수 없습니다.
- 인증서 취소가 적용되려면 최대 24시간이 걸릴 수 있습니다.

인증서 취소(콘솔)

다음 절차에서는 ACM 퍼블릭 또는 프라이빗 인증서를 취소하는 방법을 안내합니다.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/acm/ ACM 콘솔을 엽니다.
2. 인증서 나열을 선택하고 취소할 인증서의 확인란을 선택합니다.
   - 또는 인증서를 선택할 수 있습니다. 인증서 세부 정보 페이지에서 취소를 선택합니다.
3. 추가 작업을 선택한 다음 취소를 선택합니다.
4. 대화 상자가 나타나면 취소 사유를 제공하고 **revoke**를 입력한 다음 취소를 선택해야 합니다.

인증서 취소(AWS CLI)

`revoke-certificate` AWS CLI 명령 또는 `RevokeCertificate` API 작업을 사용하여 ACM 퍼블릭 또는 프라이빗 인증서를 취소합니다. `list-certificates` 명령을 호출하여 인증서의 ARN을 검색할 수 있습니다.

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

> **Warning**
>
> 인증서가 취소된 후에는 인증서를 재사용할 수 없습니다. 인증서 취소는 영구적입니다.

다음은 `revoke-certificate` 명령의 출력을 보여줍니다.

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

자동 갱신 이벤트 구성

AWS Certificate Manager 내보내기 가능한 퍼블릭 인증서와 Amazon EventBridge를 사용하여 자동 인증서 갱신 이벤트를 구성할 수 있습니다.

1. 인증서 갱신을 모니터링하도록 Amazon EventBridge 이벤트를 설정합니다. 자세한 내용은 ACM에 대한 Amazon EventBridge 지원을 참조하세요.
2. 갱신 시 인증서 배포를 처리하는 자동화를 생성합니다. 자세한 내용은 ACM에서 Amazon EventBridge를 사용하여 작업 시작 단원을 참조하십시오.

3. 갱신 또는 배포 실패를 알리도록 EventBridge 이벤트를 구성합니다.

인증서 강제 갱신

ACM 콘솔, 갱신 인증서 또는 RenewCertificate API 작업을 사용하여 ACM 퍼블릭 및 프라이빗 인증서를 갱신 AWS CLI할 수 있습니다. 이전에 내보낸 인증서만 갱신할 수 있습니다.

> **Important**
>
> ACM 익스포터블 퍼블릭 인증서를 갱신하면 추가 요금이 부과됩니다. 최신 ACM 요금 정보는 AWS 웹 사이트의 AWS Certificate Manager 서비스 요금 페이지를 참조하세요.

인증서 갱신(콘솔)

다음 절차에서는 ACM 퍼블릭 또는 프라이빗 인증서를 강제로 갱신하는 방법을 안내합니다.

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/acm/ ACM 콘솔을 엽니다.
2. 인증서 나열을 선택하고 갱신할 인증서의 확인란을 선택합니다.
   - 또는 인증서를 선택할 수 있습니다. 인증서 세부 정보 페이지에서 갱신을 선택합니다.
3. 추가 작업을 선택한 다음 갱신을 선택합니다.
4. 대화 상자가 나타나면 **renew**를 입력한 다음 갱신을 선택해야 합니다.

인증서를 갱신하려면(AWS CLI)

renew-certificate AWS CLI 명령 또는 RenewCertificate API 작업을 사용하여 ACM 퍼블릭 또는 프라이빗 인증서를 갱신합니다. list-certificates 명령을 호출하여 인증서의 ARN을 검색할 수 있습니다. renew-certificate 명령은 응답을 반환하지 않습니다.

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

AWS Certificate Manager 퍼블릭 인증서에 대한 도메인 소유권 검증

Amazon 인증 기관(CA)에서 사용자 사이트에 대한 인증서를 발급하려면 AWS Certificate Manager (ACM)에서 요청에 지정하는 모든 도메인 이름에 대한 소유권 또는 제어권이 사용자에게 있음을 증명해야 합니다. 인증서 요청 시 도메인 이름 시스템(DNS) 검증, 이메일 검증 또는 HTTP 검증을 통해 소유권을 증명하도록 선택할 수 있습니다.

> **Note**
>
> 검증은 ACM에서 발급한 공개적으로 신뢰할 수 있는 인증서에만 적용됩니다. ACM은 가져온 인증서 또는 프라이빗 CA에서 서명한 인증서에 대해 도메인 소유권을 검증하지 않습니다. ACM은 Amazon VPC 프라이빗 호스팅 영역 또는 기타 비공개 도메인의 리소스를 검증할 수 없습니다. 자세한 내용은 인증서 검증 문제 해결 단원을 참조하십시오.

다음과 같은 이유로 이메일 검증을 통한 DNS 검증을 사용하는 것이 좋습니다.

- Amazon Route 53 사용하여 퍼블릭 DNS 레코드를 관리하는 경우 ACM을 통해 레코드를 직접 업데이트할 수 있습니다.
- ACM은 인증서를 아직 사용 중이고 DNS 레코드가 존재하는 한 DNS에서 확인한 인증서를 자동으로 갱신합니다.
- 이메일로 검증된 인증서를 갱신하려면 도메인 소유자의 조치가 필요합니다. ACM은 만료 45일 전에 갱신 통지를 전송하기 시작합니다. 이러한 알림은 도메인의 공통 관리자 주소 5개 중 하나 이상으로 이동합니다. 이 알림에는 도메인 소유자가 클릭하여 손쉽게 갱신 작업을 수행할 수 있는 링크가 포함되어 있습니다. 나열된 모든 도메인이 검증되면 ACM은 동일한 ARN 사용하여 갱신된 인증서를 발급합니다.

도메인의 DNS 데이터베이스를 편집할 수 없는 경우 이메일 검증을 대신 사용해야 합니다.

HTTP 검증은 CloudFront에서 사용되는 인증서에 사용할 수 있습니다. 이 방식은 HTTP 리디렉션을 사용하여 도메인 소유권을 증명하고 DNS 검증과 유사한 자동 갱신을 제공합니다.

> **Note**
>
> 이메일 검증을 통해 인증서를 생성한 후에는 해당 인증서를 DNS를 통한 인증으로 전환할 수 없습니다. DNS 검증을 사용하려면 인증서를 삭제한 다음, DNS 검증을 사용하는 새 인증서를 생성합니다.

주제

AWS Certificate Manager DNS 검증

도메인 이름 시스템(DNS)은 네트워크에 연결되는 리소스를 위한 디렉터리 서비스입니다. DNS 공급자는 도메인을 정의하는 레코드가 포함된 데이터베이스를 유지 관리합니다. DNS 검증을 선택하면 ACM은 이 데이터베이스에 추가해야 하는 하나 이상의 CNAME 레코드를 제공합니다. 이 레코드에는 사용자가 도메인을 통제함을 증명하는 역할을 하는 고유한 키-값 페어가 포함되어 있습니다.

> **Note**
>
> 이메일 검증을 통해 인증서를 생성한 후에는 해당 인증서를 DNS를 통한 인증으로 전환할 수 없습니다. DNS 검증을 사용하려면 인증서를 삭제한 다음, DNS 검증을 사용하는 새 인증서를 생성합니다.

예를 들면 추가 이름이 `www.example.com`인 `example.com` 도메인에 대해 인증서를 요청할 경우 ACM은 CNAME 레코드 두 개를 자동으로 생성합니다. 사용자의 도메인 및 계정용으로 특별히 생성된 각 레코드에는 이름과 값이 포함되어 있습니다. 값은 ACM이 인증서를 자동으로 갱신하는 데 사용하는 AWS 도메인을 가리키는 별칭입니다. CNAME 레코드는 DNS 데이터베이스에 한 번만 추가해야 합니다. 인증서가 사용 중이고 CNAME 레코드가 여전히 존재하는 경우에 한해 ACM은 인증서를 자동으로 갱신합니다.

> **Important**
>
> Amazon Route 53을 사용하여 퍼블릭 DNS 레코드를 관리하지 않는 경우 DNS 공급자에게 문의하여 레코드를 추가하는 방법을 확인하세요. 도메인의 DNS 데이터베이스를 편집할 권한이 없는 경우 이메일 검증을 대신 사용해야 합니다.

검증을 반복할 필요 없이 CNAME 레코드가 유지되는 동안 정규화된 도메인 이름(FQDN)에 대한 추가 ACM 인증서를 요청할 수 있습니다. 즉, 도메인 이름이 동일한 대체 인증서 또는 다른 하위 도메인을 포함하는 인증서를 만들 수 있습니다. CNAME 검증 토큰은 모든 AWS 리전에서 작동하므로 여러 리전에서 동일한 인증서를 다시 생성할 수 있습니다. 또한 삭제된 인증서를 교체할 수도 있습니다.

연결된 AWS 서비스에서 인증서를 제거하거나 CNAME 레코드를 삭제하여 자동 갱신을 중지할 수 있습니다. Route 53이 DNS 공급자가 아닌 경우에는 공급자에게 연락하여 레코드를 삭제하는 방법을 알아보세요. Route 53이 공급자인 경우 Route 53 개발자 가이드에서 리소스 레코드 세트 삭제를 참조하세요. 관리형 인증서 갱신에 대한 자세한 내용은 에서 관리형 인증서 갱신 AWS Certificate Manager 섹션을 참조하세요.

> **Note**
>
> DNS 구성에서 5개 이상의 CNAME 하나로 연결되어 있으면 CNAME 확인이 실패합니다. 더 긴 체인이 필요한 경우 이메일 검증을 사용하는 것이 좋습니다.

ACM에 대한 CNAME 레코드의 작동 방식

> **Note**
>
> 이 섹션은 Route 53을 DNS 공급자로 사용하지 않는 고객에게 적용됩니다.

Route 53을 DNS 공급자로 사용하지 않는 경우, ACM에서 제공하는 CNAME 레코드를 일반적으로 웹 사이트를 통해 공급자의 데이터베이스에 수동으로 입력해야 합니다. CNAME 레코드는 리디렉션 메커니즘, 공급업체별 메타데이터의 컨테이너 등 다양한 용도로 사용됩니다. ACM의 경우 이러한 레코드를 통해 초기 도메인 소유권 확인 및 지속적인 자동 인증서 갱신을 수행할 수 있습니다.

다음 테이블에는 6개 도메인 이름에 대한 CNAME 레코드의 예제가 나와 있습니다. 각 레코드의 레코드 이름-레코드 값 페어는 도메인 이름 소유권을 인증하는 역할을 합니다.

이 표에서 처음 2개의 레코드 이름-레코드 값 페어는 동일합니다. 이는 `*.example.com`과 같은 와일드 카드 도메인의 경우 ACM이 생성한 문자열이 기본 도메인 `example.com`에 대해 생성된 문자열과 동일함을 보여줍니다. 그렇지 않으면 레코드 이름 및 레코드 값 페어는 각 도메인 이름별로 다릅니다.

CNAME 레코드 예

도메인 이름	레코드 이름	레코드 값	설명
*.example.com	*_x1*.example.com.	*_x2*.acm-validations.aws.	Identical
example.com	*_x1*.example.com.	*_x2*.acm-validations.aws.	

도메인 이름	레코드 이름	레코드 값	설명
www.example.com	*_x3*.www.example.com.	*_x4*.acm-validations.aws.	고유
host.example.com	*_x5*.host.example.com.	*_x6*.acm-validations.aws.	고유
subdomain.example.com	*_x7*.subdomain.example.com.	*_x8*.acm-validations.aws.	고유
host.subdomain.example.com	*_x9*.host.subdomain.example.com.	*_x10*.acm-validations.aws.	고유

밑줄(_) 다음에 오는 *xN* 값은 ACM이 생성한 긴 문자열입니다. 예를 들면 다음과 같습니다.

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

이 예는 결과로 생성된 레코드 이름을 보여줍니다. 연결된 레코드 값은 다음과 같은 값일 수 있습니다.

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

위의 예는 동일한 DNS 레코드에 대한 결과입니다.

> **Note**
>
> DNS 공급자가 앞에 밑줄이 붙은 CNAME 값을 지원하지 않을 경우 DNS 검증 문제 해결을 참조하세요.

인증서를 요청하고 DNS 검증을 지정하면, ACM이 CNAME 정보를 다음 형식으로 제공합니다.

도메인 이름	레코드 이름	레코드 형식	레코드 값
example.com	_a79865eb4cd1a6ab990a45779b4e0b96.example.com.	CNAME	_424c7224e9b0146f9a8808af955727d0.acm-validations.aws.

도메인 이름은 인증서와 연결된 FQDN입니다. 레코드 이름은 키-값 페어의 키로, 레코드를 고유하게 식별합니다. 레코드 값은 키-값 페어의 값으로 사용됩니다.

DNS 레코드를 추가하려면 이 세 가지 값(도메인 이름, 레코드 이름, 레코드 값)을 모두 DNS 공급자 웹 인터페이스의 해당 필드에 입력해야 합니다. 레코드 이름(또는 '이름') 필드를 처리하는 방식이 공급자 간에 일관되지 않습니다. 경우에 따라 위에 표시된 것처럼 전체 문자열을 제공해야 합니다. 일부 공급자의 경우에는 사용자가 입력한 문자열에 도메인 이름이 자동으로 추가됩니다. 즉, 이 예에서는 이름 필드에

```
_a79865eb4cd1a6ab990a45779b4e0b96
```

만 입력해야 합니다. 이 규칙을 잘못 알고 도메인 이름이 포함된 레코드 이름(예: *.example.com*)을 입력하면 다음과 같이 될 수 있습니다.

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

이 경우 검증이 실패합니다. 따라서 공급자에 맞는 입력 유형을 미리 확인해야 합니다.

DNS 검증 설정

이 섹션에서는 DNS 검증을 사용하도록 퍼블릭 인증서를 구성하는 방법에 대해서 설명합니다.

콘솔에서 DNS 검증을 설정하려면

> **Note**
>
> 이 절차에서는 이미 하나 이상의 인증서를 생성했으며 인증서를 생성한 AWS 리전에서 작업하고 있다고 가정합니다. 콘솔을 열려고 하는데 처음 사용 화면이 대신 표시되거나 콘솔을 성공적으로 열었지만 목록에 인증서가 표시되지 않는 경우, 올바른 리전을 지정했는지 확인합니다.

1. https://console.aws.amazon.com/acm/에서 ACM 콘솔을 엽니다.
2. 인증서 목록에서 검증 보류 중(Pending validation) 상태인 구성하려는 인증서 ID(Certificate ID)를 선택합니다. 그러면 인증서에 대한 세부 정보 페이지가 열립니다.
3. 도메인(Domains) 섹션에서 다음 두 절차 중 하나를 완료합니다.

    a. (선택 사항) Route 53로 검증합니다.

다음 조건에 해당할 경우 활성 상태의 Route 53에 레코드 생성(Create record in Route 53) 버튼이 표시됩니다.

- Route 53을 DNS 공급자로 사용합니다.
- Route 53에서 호스팅한 영역에 대한 쓰기 권한이 있습니다.
- FQDN이 아직 검증되지 않았습니다.

> **Note**
>
> Route 53을 사용하고 있지만 Route 53에 레코드 생성이 표시되지 않거나 비활성화된 경우, ACM 콘솔에서 'Route 53에 레코드 생성' 버튼이 나타나지 않음 섹션을 참조하세요.

Route 53에 레코드 생성을 선택한 다음 레코드 생성을 선택합니다. 인증서 상태(Certificate status) 페이지가 DNS 레코드가 생성됨(Successfully created DNS records)이라고 표시되는 상태 배너와 함께 열립니다.

새 인증서의 상태가 최대 30분 동안 계속 검증 보류 중(Pending validation)으로 표시될 수 있습니다.

> **Tip**
>
> ACM이 Route 53에서 레코드를 자동으로 생성하도록 프로그래밍 방식으로 요청할 수 없습니다. 그러나 Route 53 DNS 데이터베이스에 레코드를 생성하기 위해 Route 53에 AWS CLI 또는 API를 호출할 수 있습니다. Route 53 레코드 세트에 대한 자세한 내용은 리소스 레코드 세트 관련 작업을 참조하세요.

b. (선택 사항) Route 53을 DNS 공급자로 사용하지 않는 경우 CNAME 정보를 검색하여 DNS 데이터베이스에 추가해야 합니다. 새 인증서의 세부 정보 페이지에서 이 작업은 다음 두 가지 방법 중 하나로 수행할 수 있습니다.

- 도메인(Domains) 섹션에 표시된 CNAME 구성 요소를 복사합니다. 이 정보를 DNS 데이터베이스에 수동으로 추가해야 합니다.
- 또는 CSV로 내보내기(Export to CSV)를 선택합니다. 결과 파일의 정보를 DNS 데이터베이스에 수동으로 추가해야 합니다.

> **Important**
>
> 검증 문제를 방지하려면 DNS 공급자의 데이터베이스에 정보를 추가하기 전에 ACM에 대한 CNAME 레코드의 작동 방식를 검토하세요. 문제가 발생하면 DNS 검증 문제 해결 섹션을 참조하세요.

ACM이 사용자를 위해 CNAME 값을 생성한 시각으로부터 72시간 내에 도메인 이름을 검증할 수 없는 경우, ACM은 인증서 상태를 [검증 시간 초과(Validation timed out)]로 변경합니다. 이러한 결과가 발생하는 가장 큰 이유는 ACM이 생성한 값으로 DNS 구성을 성공적으로 업데이트하지 않았기 때문입니다. 이 문제를 해결하려면 CNAME 지침을 검토한 후 새 인증서를 요청해야 합니다.

AWS Certificate Manager 이메일 검증

Amazon 인증 기관(CA)에서 사용자 사이트에 대한 인증서를 발급하려면 AWS Certificate Manager (ACM)는 요청에 지정한 모든 도메인에 대한 소유권 또는 제어권이 사용자에게 있는지 확인해야 합니다. 이메일 혹은 DNS를 사용하여 확인할 수 있습니다. 이 주제에서는 이메일 검증에 대해 설명합니다.

이메일 검증 사용 시 문제가 발생하면 이메일 검증 문제 해결 단원을 참조하세요.

이메일 검증 작동 방식

ACM은 각 도메인에 대해 다음 5개의 공통 시스템 이메일로 검증 이메일 메시지를 전송합니다. 또는 대신 해당 도메인에서 이러한 이메일을 수신하려는 경우 슈퍼 도메인을 검증 도메인으로 지정할 수 있습니다. 최소 웹 사이트 주소까지의 모든 하위 도메인이 유효하며, 이메일 주소의 도메인으로 @ 뒤에 접미사로 사용됩니다. 예를 들어, example.com을 subdomain.example.com의 검증 도메인으로 지정하는 경우 admin@example.com으로 이메일을 받을 수 있습니다.

- administrator@해당_도메인_이름
- hostmaster@해당_도메인_이름.
- postmaster@해당_도메인_이름
- webmaster@해당_도메인_이름
- admin@해당_도메인_이름

도메인을 소유하고 있음을 증명하려면 이러한 이메일에 포함된 검증 링크를 선택해야 합니다. 또한 ACM은 인증서가 만료된 지 45일이 되면 인증서를 갱신하기 위해 동일한 주소로 검증 이메일을 전송합니다.

ACM API 또는 CLI를 사용하여 다중 도메인 인증서 요청에 대한 이메일 검증을 수행하면 요청이 요청에 있는 다른 도메인의 하위 도메인을 포함해도 요청된 각 도메인에서 이메일 메시지를 전송합니다. ACM이 인증서를 발급하려면 도메인 소유자가 이러한 각 도메인의 이메일 메시지를 확인해야 합니다.

이 프로세스의 예외

www 또는 와일드카드 별표(*)로 시작하는 도메인 이름에 대한 ACM 인증서를 요청하는 경우, ACM은 맨 앞의 **www** 또는 별표를 제외하고 관리 주소로 이메일을 전송합니다. 이 주소는 도메인 이름의 나머지 부분에 admin@, administrator@, hostmaster@, postmaster@ 및 webmaster@를 추가한 양식입니다. 예를 들어 www.example.com에 대한 ACM 인증서를 요청하는 경우, admin@www.example.com이 아닌 admin@example.com으로 이메일을 보냅니다. 마찬가지로 *.test.example.com에 대한 ACM 인증서를 요청하는 경우 admin@test.example.com으로 이메일을 보냅니다. 나머지 공통 관리 주소도 비슷하게 구성됩니다.

> **Important**
>
> ACM은 더 이상 새 인증서 또는 갱신에 대한 WHOIS 이메일 검증을 지원하지 않습니다. 공통 시스템 주소는 계속 지원됩니다. 자세한 내용은 블로그 게시물을 참조하세요.

고려 사항

이메일 검증에 대한 다음과 같은 고려 사항을 따르세요.

- 이메일 검증을 사용하려면 도메인에 등록된 유효한 이메일 주소가 필요합니다. 이메일 주소 설정 절차는 이 설명서에서는 다루지 않습니다.
- 검증은 ACM에서 발급한 공개적으로 신뢰할 수 있는 인증서에만 적용됩니다. ACM은 가져온 인증서 또는 프라이빗 CA에서 서명한 인증서에 대해 도메인 소유권을 검증하지 않습니다. ACM은 Amazon VPC 프라이빗 호스팅 영역 또는 기타 비공개 도메인의 리소스를 검증할 수 없습니다. 자세한 내용은 인증서 검증 문제 해결 단원을 참조하십시오.
- 이메일 검증을 통해 인증서를 생성한 후에는 해당 인증서를 DNS를 통한 인증으로 전환할 수 없습니다. DNS 검증을 사용하려면 인증서를 삭제한 다음, DNS 검증을 사용하는 새 인증서를 생성합니다.

인증서 만료 및 갱신

ACM 인증서는 198일 동안 유효합니다. 인증서를 갱신하려면 도메인 소유자의 작업이 필요합니다. ACM은 만료 45일 전에 도메인과 연결된 이메일 주소로 갱신 알림을 전송하기 시작합니다. 이 알림에는 도메인 소유자가 갱신을 위해 클릭할 수 있는 링크가 포함되어 있습니다. 나열된 모든 도메인이 검증되면 ACM은 동일한 ARN 사용하여 갱신된 인증서를 발급합니다.

(선택 사항) 검증 이메일 재전송

각 검증 이메일에는 인증서 요청을 승인하는 데 사용할 수 있는 토큰이 포함되어 있습니다. 하지만 승인 프로세스에 필요한 검증 이메일이 스팸 필터에 의해 차단되거나 전송 중에 손실될 수 있으므로 토큰은 72시간이 경과하면 자동으로 만료됩니다. 원래 이메일을 수신하지 못하거나 토큰이 만료된 경우 이메일을 재전송하도록 요청할 수 있습니다. 검증 이메일을 재전송하는 방법에 대한 자세한 내용은 검증 이메일 재전송 섹션을 참조하세요.

이메일 검증과 관련된 지속적인 문제는 의 문제 해결 AWS Certificate Manager의 이메일 검증 문제 해결 단원을 참조하세요.

AWS Certificate Manager 이메일 검증 자동화

이메일 검증 ACM 인증서의 경우 일반적으로 도메인 소유자의 수작업이 필요합니다. 많은 수의 이메일 검증 인증서를 사용하는 조직의 경우 필요한 응답을 자동화하는 파서를 만드는 것이 좋을 수 있습니다. 이메일 검증을 사용할 수 있도록, 이 섹션의 정보에서는 도메인 검증 이메일 메시지에 사용되는 템플릿, 그리고 검증 프로세스 수행과 관련한 워크플로에 대해 설명합니다.

검증 이메일 템플릿

검증 이메일 메시지의 형식은 새 인증서를 요청하는지 또는 기존 인증서를 갱신하는지에 따라 다음 두 가지 중 하나로 정해집니다. 강조 표시된 문자열의 내용은 검증되는 도메인에 특정 값으로 바꾸어야 합니다.

새 인증서 생성

이메일 템플릿 텍스트:

```
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
```

갱신을 위한 인증서 검증

이메일 템플릿 텍스트:

```
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
```

에서 새 검증 메시지를 받으면 구문 분석기에 대한 가장 up-to-date의 신뢰할 수 있는 템플릿으로 사용하는 것이 AWS좋습니다. 2020년 11월 이전에 설계된 메시지 파서를 사용하는 고객은 템플릿에 대해 다음과 같은 변경 사항을 확인해야 합니다.

- 이제 이메일 제목 줄에 '"Certificate approval for *domain name*' 대신 'Certificate request for *domain name*'이 표시됩니다.
- AWS account ID가 이제 대시 또는 하이픈 없이 표시됩니다.
- 이제 Certificate Identifier가 단축된 양식 대신 전체 인증서 ARN을 표시합니다(예: *3b4d78e1-0882-4f51-954a-298ee44ff369* 대신 *arn:aws:acm:us-east-1:000000000000:certificate/3b4d78e1-0882-4f51-954a-298ee44ff369* 표시).
- 이제 인증서 승인 URL에 certificates.amazon.com 대신 acm-certificates.amazon.com이 포함됩니다.
- 인증서 승인 URL을 클릭하여 연 승인 양식에 이제 승인 버튼이 포함됩니다. 이제 승인 버튼 div의 이름이 approval_button 대신 approve-button이 됩니다.
- 새로 요청된 인증서와 갱신 인증서 모두의 검증 메시지 이메일 형식이 동일합니다.

검증 워크플로

이 섹션에서는 이메일 검증 인증서의 갱신 워크플로에 대한 정보를 제공합니다.

- ACM 콘솔이 다중 도메인 인증서 요청을 처리하면 퍼블릭 인증서를 요청할 때 지정한 도메인 이름 또는 검증 도메인으로 검증 이메일 메시지를 전송합니다. ACM이 인증서를 발급하려면 도메인 소유자가 각 도메인의 이메일 메시지를 확인해야 합니다. 자세한 내용은 이메일을 사용하여 도메인 소유권 확인을 참조하세요.
- ACM API 또는 CLI를 사용하여 다중 도메인 인증서 요청에 대한 이메일 검증을 수행하면 요청이 요청에 있는 다른 도메인의 하위 도메인을 포함해도 요청된 각 도메인에서 이메일 메시지를 전송합니다. ACM이 인증서를 발급하려면 도메인 소유자가 이러한 각 도메인의 이메일 메시지를 확인해야 합니다.

  ACM 콘솔을 통해 기존 인증서에 대한 이메일을 재전송하면 원본 인증서 요청에 지정된 검증 도메인 또는 검증 도메인이 지정되지 않은 경우 정확한 도메인으로 이메일이 전송됩니다. 다른 도메인에서 검증 이메일을 수신하려면 검증에 사용할 검증 도메인을 지정하여 새 인증서를 요청할 수 있습니다. 또는 API, SDK 또는 CLI를 사용하여 `ValidationDomain` 파라미터를 통해 ResendValidationEmail을 직접 호출할 수 있습니다. 그러나 `ResendValidationEmail` 요청에 지정된 검증 도메인은 해당 직접 호출에만 사용되며 향후 검증 이메일을 위해 인증서 Amazon 리소스 이름(ARN)에 저장되지 않습니다. 원본 인증서 요청에 지정되지 않은 도메인 이름으로 검증 이메일을 수신하려면 매번 `ResendValidationEmail`을 직접 호출해야 합니다.

  > **Note**
  >
  > 2020년 11월 이전에는 고객이 apex 도메인만 검증해야 했으며 ACM은 모든 하위 도메인에도 적용되는 인증서를 발급했습니다. 이 시점 이전에 설계된 메시지 파서를 사용하는 고객은 이메일 검증 워크플로의 변경 사항을 확인해야 합니다.

- ACM API 또는 CLI를 사용하면 다중 도메인 인증서 요청에 대한 모든 검증 이메일 메시지를 apex 도메인으로 보낼 수 있습니다. API에서 RequestCertificate 작업의 `DomainValidationOptions` 파라미터를 사용하여 DomainValidationOption 유형의 멤버인 `ValidationDomain`의 값을 지정합니다. CLI에서 request-certificate 명령의 --domain-validation-options 파라미터를 사용하여 `ValidationDomain`의 값을 지정합니다.

AWS Certificate Manager HTTP 검증

Hypertext Transfer Protocol(HTTP)은 월드 와이드 웹에서의 데이터 통신을 위한 기본 프로토콜입니다. CloudFront에 사용되는 인증서에 대해 HTTP 검증을 선택하면 ACM은 이 프로토콜을 활용하여 도메인

소유권을 검증합니다. ACM은 CloudFront와 함께 작동하여 특정 URL을 제공하고 도메인의 해당 URL에서 액세스할 수 있어야 하는 고유한 토큰을 제공합니다. 이 토큰은 도메인에 대한 통제권이 있다는 증거 역할을 합니다. 도메인에서 CloudFront 인프라 내의 ACM 제어 위치로 리디렉션을 설정하는 것은 도메인의 콘텐츠를 수정할 수 있는 기능을 보여주는 것이며 결과적으로 소유권을 검증해 줍니다. ACM과 CloudFront 간의 원활한 통합은 특히 CloudFront 배포의 경우 인증서 발급 프로세스를 간소화합니다.

> **Important**
>
> HTTP 검증은 와일드카드 도메인 인증서(예: *.example.com)를 지원하지 않습니다. 와일드카드 인증서의 경우 DNS 검증 또는 이메일 검증을 대신 사용해야 합니다.

예를 들어 CloudFront로 `www.example.com`을 추가 이름으로 사용하는 `example.com` 도메인에 대한 인증서를 요청하는 경우 ACM은 HTTP 검증을 위한 두 세트의 URL을 제공합니다. 각 세트에는 도메인 및 AWS 계정을 위해 특별히 생성된 `redirectFrom` URL과 `redirectTo` URL이 포함됩니다. `redirectFrom` URL은 구성해야 하는 도메인의 경로(예: `http://example.com/.well-known/pki-validation/example.txt`)입니다. `redirectTo` URL은 고유한 검증 토큰이 저장된 CloudFront 인프라 내의 ACM 제어 위치를 가리킵니다. 이러한 리디렉션은 한 번만 설정하면 됩니다. 인증 기관은 도메인 소유권의 검증을 시도할 때 `redirectFrom` URL에서 파일을 요청하고, 이 URL은 CloudFront에서 `redirectTo` URL로 리디렉션하여 검증 토큰에 대한 액세스를 허용합니다. 인증서가 CloudFront에서 사용 중이고 리디렉션이 여전히 존재하는 한 ACM은 인증서를 자동으로 갱신합니다.

CloudFront를 사용하여 정규화된 도메인 이름(FQDN)에 대한 HTTP 검증을 설정했으면 HTTP 리디렉션이 그대로 유지되는 한 검증 프로세스를 반복하지 않고 해당 FQDN에 대한 추가 ACM 인증서를 요청할 수 있습니다. 즉, 동일한 도메인 이름으로 대체 인증서를 생성할 수 있습니다. 리디렉션이 여전히 활성 상태인 한 검증 프로세스를 다시 거치지 않고 삭제된 인증서를 교체할 수도 있습니다.

HTTP 검증 인증서의 자동 갱신을 중지하려면 두 가지 옵션이 있습니다. 연결된 CloudFront 배포에서 인증서를 제거하거나 검증을 위해 설정한 HTTP 리디렉션을 삭제할 수 있습니다. CloudFront 이외의 콘텐츠 전송 네트워크(CDN) 또는 웹 서버를 사용하여 리디렉션을 관리하는 경우 해당 설명서를 참조하여 리디렉션을 제거하는 방법을 알아봅니다. CloudFront를 사용하여 리디렉션을 관리하는 경우 배포의 구성을 업데이트하여 리디렉션을 제거할 수 있습니다. 관리형 인증서 갱신에 대한 자세한 내용은 에서 관리형 인증서 갱신 AWS Certificate Manager 섹션을 참조하세요. 자동 갱신을 중지하면 인증서 만료로 이어져 HTTPS 트래픽이 중단될 수 있다는 점을 기억하세요.

ACM에 대한 HTTP 리디렉션 작동 방식

> **Note**
>
> 이 섹션은 콘텐츠 전송을 위해 CloudFront를 사용하고 SSL/TLS 인증서 관리를 위해 ACM을 사용하는 고객을 위한 것입니다.

ACM 및 CloudFront에서 HTTP 검증을 사용하는 경우 HTTP 리디렉션을 설정해야 합니다. 이러한 리디렉션을 통해 ACM은 초기 인증서 발급 및 지속적인 자동 갱신에 대한 도메인 소유권을 검증할 수 있습니다. 이 리디렉션 메커니즘은 도메인의 특정 URL이 고유한 검증 토큰이 저장된 CloudFront 인프라 내의 ACM 제어 위치를 가리키게 하는 방식으로 작동합니다.

다음 표는 도메인 이름에 대한 리디렉션 구성의 예를 보여줍니다. 참고로 HTTP 검증은 와일드카드 도메인(예: *.example.com)을 지원하지 않습니다. 각 구성의 리디렉션 소스-리디렉션 대상 페어는 도메인 이름 소유권을 인증하는 역할을 합니다.

HTTP 리디렉션 구성의 예

도메인 이름	리디렉션 소스	리디렉션 대상	설명
example.com	`http://example.com/.well-known/pki-validation/ x2.txt`	`https://validation. region.acm-validations.aws/ y2/.well-known/pki-validation/ x2.txt`	고유
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`	고유
host.example.com	`http://host.example.com/.well-known/pki-validation/ x4.txt`	`https://validation. region.acm-validations.aws/ y4/.well-kn`	고유

도메인 이름	리디렉션 소스	리디렉션 대상	설명
		own/pki-validation / *x4*.txt	
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	고유
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	고유

파일 이름의 *xN* 값과 ACM 제어 도메인의 *yN* 값은 ACM에서 생성하는 고유 식별자입니다. 예:

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

은(는) 결과로 생성된 리디렉션 소스 URL를 나타냅니다. 연결된 리디렉션 대상 URL은 동일한 검증 레코드에 대해

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

일 수 있습니다.

> **Note**
>
> 웹 서버 또는 콘텐츠 전송 네트워크가 지정된 경로에서 리디렉션 설정을 지원하지 않는 경우 HTTP 검증 문제 해결을 참조하세요.

인증서를 요청하고 HTTP 검증을 지정하면, ACM이 리디렉션 정보를 다음 형식으로 제공합니다.

도메인 이름	리디렉션 대상
example.com	https://validation.*region*.acm-validations.aws/*a424c7224e9b* /.well-known/pki-validation/*a79865eb4cd1a6ab990a45779b4e0b96* .txt

도메인 이름은 인증서와 연결된 FQDN입니다. 리디렉션 소스는 ACM이 검증 파일을 찾을 도메인의 URL입니다. 리디렉션 대상은 실제 검증 파일이 호스팅되는 ACM 제어 URL입니다.

리디렉션 소스 URL에서 리디렉션 대상 URL로 요청을 리디렉션하도록 웹 서버 또는 CloudFront 배포를 구성해야 합니다. 이 리디렉션을 설정하는 정확한 방법은 웹 서버 소프트웨어 또는 CloudFront 구성에 따라 다릅니다. ACM이 도메인 소유권을 검증하고 인증서를 발급하거나 갱신할 수 있도록 리디렉션이 올바르게 설정되어 있는지 확인합니다.

HTTP 검증 설정

ACM은 CloudFront에서 사용할 퍼블릭 SSL/TLS 인증서를 발급할 때 HTTP 검증을 사용하여 도메인 소유권을 확인합니다. 이 섹션에서는 HTTP 검증을 사용하도록 퍼블릭 인증서를 구성하는 방법에 대해서 설명합니다.

콘솔에서 HTTP 검증을 설정하려면

> **Note**
>
> 이 절차에서는 이미 CloudFront를 통해 인증서를 요청했으며 인증서를 생성한 AWS 리전에서 작업하고 있다고 가정합니다. HTTP 검증은 CloudFront Distribution Tenants 기능을 통해서만 사용할 수 있습니다.

1. https://console.aws.amazon.com/acm/에서 ACM 콘솔을 엽니다.
2. 인증서 목록에서 검증 보류 중(Pending validation) 상태인 구성하려는 인증서 ID(Certificate ID)를 선택합니다. 그러면 인증서에 대한 세부 정보 페이지가 열립니다.
3. 도메인 섹션에서 인증서 요청의 각 도메인에 대한 리디렉션 소스 및 리디렉션 대상 값을 볼 수 있습니다.
4. 각 도메인에 대해 리디렉션 소스 URL에서 리디렉션 대상 URL로의 HTTP 리디렉션을 설정합니다. CloudFront 배포 구성을 통해이 작업을 수행할 수 있습니다.
5. 리디렉션 소스 URL에서 리디렉션 대상 URL로 요청을 리디렉션하도록 CloudFront 배포를 구성합니다. 이 리디렉션을 설정하는 방법은 CloudFront 구성에 따라 다릅니다.
6. 리디렉션을 설정한 후에는 ACM이 도메인 소유권 검증을 자동으로 시도합니다. 이 프로세스는 최대 30분이 걸릴 수 있습니다.

ACM이 리디렉션 값을 생성한 시각으로부터 72시간 내에 도메인 이름을 검증할 수 없는 경우, ACM은 인증서 상태를 검증 시간 초과로 변경합니다. 이 결과의 가장 가능성이 높은 이유는 HTTP 리디렉션을 성공적으로 설정하지 않았기 때문입니다. 이 문제를 해결하려면 리디렉션 지침을 검토한 후 새 인증서를 요청해야 합니다.

> **Important**
>
> 검증 문제를 방지하려면 리디렉션 소스 위치의 콘텐츠가 리디렉션 대상 위치의 콘텐츠와 일치해야 합니다. 문제가 발생하면 HTTP 검증 문제 해결 단원을 참조하세요.

> **Note**
>
> DNS 검증과 달리 ACM이 HTTP 리디렉션의 자동 생성을 프로그래밍 방식으로 요청할 수 없습니다. CloudFront 배포 설정을 통해 이러한 리디렉션을 구성해야 합니다.

HTTP 검증 작업 방법에 대한 자세한 내용은 ACM에 대한 HTTP 리디렉션 작동 방식 섹션을 참조하세요.

의 프라이빗 인증서 AWS Certificate Manager

에서 생성한 기존 프라이빗 CA에 액세스할 수 있는 경우 AWS Private CA, AWS Certificate Manager (ACM)은 프라이빗 키 인프라(PKI)에 사용하기에 적합한 인증서를 요청할 수 있습니다. CA는 사용자의 계정에 상주하거나 다른 계정에 의해 공유될 수 있습니다. 사설 CA 생성에 대한 자세한 내용은 사설 인증 기관 생성을 참조하세요.

프라이빗 CA가 서명한 인증서는 기본적으로 신뢰할 수 없으며 ACM에서는 인증서에 대한 어떤 형태의 검증도 지원하지 않습니다. 따라서 관리자는 조직의 클라이언트 트러스트 스토어에 인증서를 설치하기 위한 조치를 취해야 합니다.

프라이빗 ACM 인증서는 X.509 표준을 따르며 다음 제한 사항이 적용됩니다.

- 이름: DNS를 준수하는 주체 이름을 사용해야 합니다. 자세한 내용은 도메인 이름 단원을 참조하십시오.
- 알고리즘: 암호화를 위해서는 인증서 프라이빗 키 알고리즘이 2048비트 RSA, 256비트 ECDSA 또는 384비트 ECDSA 중 하나에 해당해야 합니다.

  > **Note**
  >
  > 지정된 서명 알고리즘 패밀리(RSA 또는 ECDSA)는 CA의 보안 암호 키의 알고리즘 패밀리와 일치해야 합니다.

- 만료: 각 프라이빗 인증서는 13개월(395일) 동안 유효합니다. 서명한 CA 인증서의 만료 날짜가 요청된 인증서의 만료 날짜를 초과해서는 안 되며, 초과할 경우 인증서 요청이 실패하게 됩니다.

  > **Note**
  >
  > 프라이빗 인증서의 유효 기간은 퍼블릭 인증서보다 깁니다. 퍼블릭 ACM 인증서는 198일 동안 유효합니다. 퍼블릭 인증서에 대한 자세한 내용은 섹션을 참조하세요에서 퍼블릭 인증서 요청 AWS Certificate Manager.

- 갱신: ACM은 11개월 후에 프라이빗 인증서를 자동으로 갱신하려고 시도합니다.

최종 엔티티 인증서에 서명하는 데 사용되는 사설 CA에는 다음과 같은 자체 제한이 적용됩니다.

- CA는 활성 상태여야 합니다.

> **Note**
>
> 공개적으로 신뢰할 수 있는 인증서와 달리, 프라이빗 CA에서 서명한 인증서는 검증이 필요하지 않습니다.

주제

를 사용하여 ACM 프라이빗 인증서에 서명 AWS Private CA 하는 조건

다음 두 경우 중 하나에서 AWS Private CA 를 사용하여 ACM 인증서에 서명할 수 있습니다.

- 단일 계정: 서명하는 CA와 발급된 AWS Certificate Manager (ACM) 인증서가 동일한 AWS 계정에 상주합니다.

  단일 계정 발급 및 갱신을 사용하려면, AWS Private CA 관리자가 ACM 서비스 보안 주체에 인증서를 생성, 검색 및 나열할 수 있는 권한을 부여해야 합니다. 이는 API 작업 CreatePermission 또는 AWS CLI 명령 create-permission을 사용하여 AWS Private CA 수행됩니다. 계정 소유자는 인증서 발급을 담당하는 IAM 사용자, 그룹 또는 역할에 이러한 권한을 할당합니다.
- 교차 계정: 서명 CA와 발급된 ACM 인증서는 서로 다른 AWS 계정에 있으며 인증서가 있는 계정에 CA에 대한 액세스 권한이 부여되었습니다.

  교차 계정 발급 및 갱신을 AWS Private CA 활성화하려면 관리자가 AWS Private CA API 작업 PutPolicy 또는 AWS CLI 명령 put-policy를 사용하여 CA에 리소스 기반 정책을 연결해야 합니다. 이 정책은 CA에 대해 제한된 액세스가 허용되는 다른 계정의 보안 주체를 지정합니다. 자세한 내용은 ACM Private CA에서 리소스 기반 정책 사용을 참조하세요.

  계정 간 시나리오에서는 ACM이 PCA 정책과 보안 주체로서 상호 작용하는 서비스 연결 역할(SLR)을 설정해야 합니다. ACM은 첫 번째 인증서를 발급하는 동안 SLR을 자동으로 생성합니다.

  이 경우 ACM은 계정에 SLR이 있는지 여부를 확인할 수 없다는 알림을 표시할 수 있습니다. 필요한 `iam:GetRole` 권한이 이미 계정의 ACM SLR에 부여된 경우, SLR이 생성된 후 알림이 다시 표시되지 않습니다. 알림이 다시 표시될 경우 사용자 또는 계정 관리자가 ACM에 `iam:GetRole` 권한을 부

여하거나 계정을 ACM 관리형 정책 `AWSCertificateManagerFullAccess`에 연결해야 할 수 있습니다.

자세한 내용은 ACM에서 서비스 연결 역할 사용을 참조하세요.

> **Important**
>
> ACM 인증서를 자동으로 갱신하려면 지원되는 AWS 서비스와 적극적으로 연결해야 합니다. ACM이 지원하는 리소스에 대한 자세한 내용은 ACM에 통합된 서비스 섹션을 참조하세요.

에서 프라이빗 인증서 요청 AWS Certificate Manager

프라이빗 인증서 요청(콘솔)

1. AWS Management Console에 로그인하고 https://console.aws.amazon.com/acm/home ACM 콘솔을 엽니다.

   인증서 요청을 선택합니다.
2. 인증서 요청 페이지에서 프라이빗 인증서 요청(Request a private certificate)을 선택하고 다음(Next)을 선택하여 계속합니다.
3. 인증 기관 세부 정보(Certificate authority details) 섹션에서 인증 기관(Certificate authority) 메뉴를 선택하고 사용할 수 있는 프라이빗 CA 중 하나를 선택합니다. 다른 계정에서 CA를 공유한 경우 ARN 앞에 소유권 정보가 표시됩니다.

   CA에 대한 세부 정보가 표시되므로 올바른 CA를 선택했는지 확인할 수 있습니다.

   - 소유자
   - 유형
   - 일반 이름(CN)
   - 조직(O)
   - 조직 단위(OU)
   - 국가 이름(C)
   - 주 또는 지방
   - 시 이름

4. 도메인 이름(Domain names) 섹션에서 도메인 이름을 입력합니다. **www.example.com** 같은 FQDN(Fully Qualified Domain Name)이나 **example.com** 같은 베어 또는 apex 도메인 이름을 사용할 수 있습니다. 맨 왼쪽에서 별표(*)를 와일드카드로 사용하여 동일한 도메인 내에서 여러 사이트 이름을 보호할 수도 있습니다. 예를 들어 ***.example.com**은 **corp.example.com** 및 **images.example.com**을 보호합니다. 와일드카드 이름은 주체(Subject) 필드와 ACM 인증서의 주체 대체 이름(Subject Alternative Name) 확장에 표시됩니다.

   > **Note**
   >
   > 와일드카드 인증서를 요청할 때 별표(*)는 도메인 이름의 맨 왼쪽에 와야 하며 하나의 하위 도메인 수준만 보호할 수 있습니다. 예를 들어 ***.example.com**은 **login.example.com** 및 **test.example.com**을 보호할 수 있지만 **test.login.example.com**은 보호할 수 없습니다. 또한 ***.example.com**은 **example.com**의 하위 도메인만 보호하고 베어 또는 apex 도메인(**example.com**)은 보호하지 못합니다. 둘 모두를 보호하려면 다음 단계를 참조하세요.

   또는 이 인증서에 다른 이름 추가(Add another name to this certificate)를 선택하고 텍스트 상자에 이름을 입력합니다. 이렇게 하면 베어 또는 apex 도메인(예: **example.com**)과 하위 도메인(예: ***.example.com**)을 인증하는 데 유용합니다.

5. 키 알고리즘 섹션에서 알고리즘을 선택합니다.

   알고리즘 선택에 도움이 되는 자세한 내용은 AWS 블로그 게시물 에서 ECDSA 인증서를 평가하고 사용하는 방법을 AWS Certificate Manager참조하세요.

6. 태그 페이지에서 선택 사항으로 인증서에 태그를 지정할 수 있습니다. 태그는 AWS 리소스를 식별하고 구성하기 위한 메타데이터 역할을 하는 키-값 페어입니다. ACM 태그 파라미터 목록과 생성 후 인증서에 태그를 추가하는 방법에 대한 지침은 AWS Certificate Manager 리소스 태그 지정 섹션을 참조하세요.

7. 인증서 갱신 권한(Certificate renewal permissions) 섹션에서 인증서 갱신 권한에 대한 통지를 확인합니다. 이러한 권한을 사용하면 선택한 CA로 서명된 프라이빗 PKI 인증서를 자동으로 갱신할 수 있습니다. 자세한 내용은 ACM에서 서비스 연결 역할 사용을 참조하세요.

8. 필수 정보를 모두 제공한 후 요청(Request)을 선택합니다. 콘솔에서 새 인증서를 볼 수 있는 인증서 목록으로 돌아갑니다.

> **Note**
>
> 목록을 정렬한 방법에 따라 찾고 있는 인증서가 즉시 표시되지 않을 수 있습니다. 오른쪽의 검은색 삼각형을 클릭하여 순서를 변경할 수 있습니다. 오른쪽 상단의 페이지 번호를 사용하여 여러 페이지의 인증서를 탐색할 수도 있습니다.

프라이빗 인증서 요청(CLI)

request-certificate 명령을 사용하여 ACM에서 프라이빗 인증서를 요청합니다.

> **Note**
>
> CA가 서명한 프라이빗 PKI 인증서를 요청할 때 AWS Private CA지정된 서명 알고리즘 패밀리(RSA 또는 ECDSA)는 CA 보안 키의 알고리즘 패밀리와 일치해야 합니다.

```
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
```

이 명령은 새 사설 인증서의 Amazon 리소스 이름(ARN)을 출력합니다.

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

대부분의 경우 ACM은 공유 CA를 처음 사용할 때 서비스 연결 역할(SLR)을 계정에 자동으로 연결합니다. SLR은 발급한 최종 엔터티 인증서를 자동으로 갱신할 수 있도록 합니다. SLR이 있는지 확인하려면 다음 명령을 사용하여 IAM을 쿼리하면 됩니다.

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

SLR이 있으면 명령 출력이 다음과 같이 나타납니다.

```
{
```

```
   "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"
      }
   }
}
```

SLR이 없으면 ACM에서 서비스 연결 역할 사용을 참조하세요.

AWS Certificate Manager 프라이빗 인증서 내보내기

프라이빗 PKI 환경의 어디에서나 사용할 수 AWS Private CA 있도록에서 발급한 인증서를 내보낼 수 있습니다. 내보낸 파일에는 인증서, 인증서 체인 및 암호화된 프라이빗 키가 포함됩니다. 이 파일을 안전하게 저장해야 합니다. 에 대한 자세한 내용은 AWS Private Certificate Authority 사용 설명서를 AWS Private CA참조하세요.

> **Note**
>
> ACM을 통해 발급된 퍼블릭 인증서를 내보내려면 ACM 내보내기 가능 퍼블릭 인증서를 참조하세요.

주제

프라이빗 인증서 내보내기(콘솔)

1. AWS Management Console에 로그인하고 https://console.aws.amazon.com/acm/home ACM 콘솔을 엽니다.
2. Certificate Manager를 선택합니다.
3. 내보내려는 인증서의 링크를 선택하세요.
4. 내보내기를 선택합니다.
5. 프라이빗 키의 암호를 입력하고 확인합니다.

> **Note**
>
> 암호를 만들 때 #, $ 또는%를 제외한 모든 ASCII 문자를 사용할 수 있습니다.

6. PEM 인코딩 생성을 선택합니다.
7. 인증서, 인증서 체인 및 암호화된 키를 메모리에 복사하거나 각각에 대해 파일로 내보내기를 선택할 수 있습니다.
8. 완료를 선택합니다.

프라이빗 인증서 내보내기(CLI)

export-certificate 명령을 사용하여 사설 인증서와 프라이빗 키를 내보냅니다. 명령을 실행할 때 암호를 할당해야 합니다. 보안을 강화하려면 파일 편집기를 사용하여 파일에 암호를 저장한 다음 파일을 제공하여 암호를 제공합니다. 이렇게 하면 암호가 명령 레코드에 저장되지 않으며 암호를 입력할 때 다른 사람이 암호를 볼 수 없습니다.

> **Note**
>
> 암호가 포함된 파일은 행 종결자로 끝나지 않아야 합니다. 다음과 같은 암호 파일을 확인할 수 있습니다.
>
> ```
> $ file -k passphrase.txt
> ```

```
passphrase.txt: ASCII text, with no line terminators
```

다음 예제는 명령 출력을 jq로 파이핑하여 PEM 형식 지정을 적용합니다.

```
[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)"'
```

이것은 base64로 인코딩된, PEM 형식 인증서를 출력하며, 다음 축약된 예에서와 같이 인증서 체인과 프라이빗 키도 포함합니다.

```
-----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-----
```

모든 요소를 파일로 출력하려면 이전 예제에 > 리디렉션을 추가하여 다음을 출력합니다.

```
$ 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
```

로 인증서 가져오기 AWS Certificate Manager

AWS Certificate Manager (ACM)에서 제공하는 SSL/TLS 인증서를 요청하는 것 외에도 외부에서 얻은 인증서를 가져올 수 있습니다 AWS. 서드 파티 인증 기관(CA)의 인증서가 이미 있거나 ACM에서 발급한 인증서로 충족되지 않는 애플리케이션별 요구 사항이 있는 경우에 이렇게 할 수 있습니다.

가져온 인증서를 ACM과 통합된 모든AWS 서비스와 함께 사용할 수 있습니다. 가져온 인증서는 ACM에서 제공한 인증서와 동일하게 작동하지만, ACM에서는 가져온 인증서에 대한 관리형 갱신을 제공하지 않습니다.

가져온 인증서를 갱신하려면 인증서 발급자로부터 새 인증서를 취득한 다음 수동으로 ACM에 다시 가져오면 됩니다. 이 작업은 인증서의 연결 및 Amazon 리소스 이름(ARN)을 보존합니다. 또는 완전히 새로운 인증서를 가져올 수 있습니다. 도메인 이름이 같은 여러 개의 인증서를 가져올 수 있지만 이 경우 한 번에 가져와야 합니다.

> **Important**
>
> 따라서 가져온 인증서의 만료 날짜를 모니터링하여 만료되기 이전에 갱신하는 것은 고객의 책임입니다. 가져온 인증서가 만료될 때 Amazon CloudWatch Events를 사용하여 알림을 보냄으로써 이 태스크를 간소화할 수 있습니다. 자세한 내용은 Amazon EventBridge 사용 단원을 참조하십시오.

가져온 인증서를 포함하여 ACM의 모든 인증서는 리전별 리소스입니다. 여러 AWS 리전에서 Elastic Load Balancing 로드 밸런서를 통해 동일한 인증서를 사용하려면 인증서를 사용하려는 각 리전으로 인증서를 가져와야 합니다. Amazon CloudFront에서 인증서를 사용하려면 미국 동부(버지니아 북부) 리전으로 인증서를 가져와야 합니다. 자세한 내용은 지원되는 리전: 단원을 참조하십시오.

ACM으로 인증서를 가져오는 방법에 대한 자세한 내용은 다음 주제를 참조하세요. 인증서를 가져올 때 문제가 발생하면 인증서 가져오기 문제를 참조하세요.

주제

- ACM 인증서를 가져오기 위한 사전 조건
- 가져올 인증서 및 키 형식
- 인증서 가져오기
- 인증서 다시 가져오기

ACM 인증서를 가져오기 위한 사전 조건

자체 서명된 SSL/TLS 인증서를 ACM으로 가져오려면 해당 인증서와 프라이빗 키를 모두 제공해야 합니다. 비 AWS 인증 기관(CA)에서 서명한 인증서를 가져오려면 인증서의 프라이빗 키와 퍼블릭 키도 포함해야 합니다. 인증서는 이 주제에 설명된 모든 기준을 충족해야 합니다.

모든 가져온 인증서에 대해 암호화 알고리즘과 키 크기를 지정해야 합니다. ACM은 다음 알고리즘을 지원합니다(괄호 안은 API 이름).

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

또한 다음 추가 요구 사항에 유의하세요.

- ACM 통합 서비스에서는 리소스와의 연결이 지원되는 알고리즘과 키 크기만 허용합니다. 예를 들어 CloudFront는 1,024비트 RSA, 2,048비트 RSA, 3,072비트 RSA, 4096비트 RSA 및 타원 프라임 곡선 256비트 키만 지원하는 반면, Application Load Balancer는 ACM에서 사용할 수 있는 모든 알고리즘을 지원합니다. 자세한 내용은 사용 중인 서비스의 설명서를 참조하세요.
- 인증서는 SSL/TLS X.509 버전 3 인증서여야 합니다. 인증서에는 퍼블릭 키, 웹 사이트의 FQDN(정규화된 도메인 이름) 또는 IP 주소 및 발급 기관에 대한 정보가 포함되어야 합니다.
- 인증서는 사용자가 소유한 프라이빗 키로 자체 서명되거나 발급하는 CA의 프라이빗 키로 서명될 수 있습니다. 5KB(5,120바이트)를 초과하지 않고 암호화되지 않은 프라이빗 키를 제공해야 합니다.
- 인증서가 CA에서 서명되고 인증서 체인을 제공하기로 선택한 경우, 체인은 PEM으로 인코딩되어야 합니다.
- 인증서는 가져오는 시점에 유효해야 합니다. 유효 기간이 시작되기 전 또는 만료된 후에는 인증서를 가져올 수 없습니다. `NotBefore` 인증서 필드에는 유효 기간 시작 날짜가 포함되어 있으며 `NotAfter` 필드에는 종료 날짜가 포함되어 있습니다.
- 모든 필수 인증서 자료(인증서, 프라이빗 키 및 인증서 체인)는 PEM 인코딩되어야 합니다. DER 인코딩된 자료를 업로드하면 오류가 발생합니다. 자세한 내용과 예제는 가져올 인증서 및 키 형식 섹션을 참조하세요.

- 인증서를 갱신(다시 가져오기)할 때 이전에 가져온 인증서에 KeyUsage 또는 ExtendedKeyUsage 확장이 없는 경우 이러한 확장을 추가할 수 없습니다.

  예외: 이전 인증서와 비교할 때 클라이언트 인증 ExtendedKeyUsage가 누락된 인증서를 다시 가져올 수 있습니다. 이는 인증 기관이 더 이상 Chrome의 루트 프로그램 요구 사항을 준수하기 위해 ClientAuth EKU로 인증서를 발급하지 않는 업계 변화를 수용합니다.

  > **Important**
  >
  > 클라이언트 인증 기능이 필요한 경우, ACM은 이전에 가져온 인증서로의 롤백을 지원하지 않으므로 사용자 측에서 추가 검증을 구현해야 합니다.

- AWS CloudFormation 는 ACM으로 인증서 가져오기를 지원하지 않습니다.

가져올 인증서 및 키 형식

ACM에서는 인증서, 인증서 체인 및 프라이빗 키(있는 경우)를 별도로 가져와야 하고, 각 구성 요소를 PEM 형식으로 인코딩해야 합니다. PEM은 Privacy Enhanced Mail의 약자입니다. PEM 형식은 흔히 인증서, 인증서 요청, 인증서 체인 및 키를 표시하는 데 사용됩니다. PEM 형식 파일의 일반적인 확장자는 `.pem`이지만, 필수는 아닙니다.

> **Note**
>
> AWS 는 PEM 파일 또는 기타 인증서 형식을 조작하기 위한 유틸리티를 제공하지 않습니다. 다음 예제에서는 간단한 작업에 일반 텍스트 편집기를 사용합니다. 보다 복잡한 작업(예: 파일 형식 변환 또는 키 추출)을 수행해야 하는 경우 OpenSSL과 같은 무료 오픈 소스 도구를 손쉽게 사용할 수 있습니다.

다음 예제에서는 가져올 파일의 형식을 보여줍니다. 구성 요소가 단일 파일로 제공되는 경우, 텍스트 편집기를 사용하여 3개의 파일로 구분합니다. PEM 파일의 문자를 잘못 편집하거나 두 개 이상의 공백을 줄 끝에 추가할 경우 인증서, 인증서 체인 또는 프라이빗 키가 유효하지 않습니다.

Example 1. PEM으로 인코딩된 인증서

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

Example 2. PEM으로 인코딩된 인증서 체인

인증서 체인에는 한 개 이상의 인증서가 포함되어 있습니다. 텍스트 편집기, Windows의 copy 명령 또는 Linux cat 명령을 사용하여 여러 인증서 파일을 하나의 체인으로 연결할 수 있습니다. 인증서가 각각의 이전 인증서를 인증하도록 인증서를 순서대로 연결해야 합니다. 사설 인증서를 가져오는 경우 루트 인증서를 마지막에 복사합니다. 다음 예시의 경우에는 세 개의 인증서가 포함되어 있지만, 사용자에 따라 인증서 체인에 포함된 인증서가 그보다 많거나 적을 수 있습니다.

> **Important**
>
> 인증서를 인증서 체인으로 복사하지 마세요.

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

Example 3. PEM 인코딩 프라이빗 키

X.509 버전 3 인증서에는 퍼블릭 키 알고리즘이 사용됩니다. X.509 인증서 또는 인증서 요청을 생성할 때는 프라이빗-퍼블릭 키 페어를 생성하는 데 사용할 알고리즘 및 키 비트 크기를 지정합니다. 퍼블릭 키는 인증서 또는 요청에 있습니다. 연결된 프라이빗 키를 비밀로 유지해야 합니다. 인증서를 가져올 때 프라이빗 키를 지정합니다. 이 키는 암호화되지 않은 것이어야 합니다. 다음 예는 RSA 프라이빗 키를 보여 줍니다.

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

다음 예는 PEM으로 인코딩된 타원 곡선 프라이빗 키를 보여 줍니다. 이 키의 생성 방식에 따라 파라미터 블록이 포함되지 않을 수도 있습니다. 파라미터 블록이 포함된 경우 ACM은 이 파라미터 블록을 가져오기 프로세스 중에 해당 키를 사용하기 전에 제거합니다.

```
-----BEGIN EC PARAMETERS-----
```

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

인증서 가져오기

, 또는 ACM API를 사용하여 외부에서 획득한 인증서(즉 AWS CLI, 타사 신뢰 서비스 공급자가 제공하는 인증서) AWS Management Console를 ACM으로 가져올 수 있습니다. 다음 주제에서는 AWS Management Console 및를 사용하는 방법을 보여줍니다 AWS CLI. 비AWS 발급자로부터 인증서를 받는 절차는이 가이드의 범위를 벗어납니다.

> **Important**
>
> 선택한 서명 알고리즘은 ACM 인증서를 가져오기 위한 사전 조건을 충족해야 합니다.

주제

- 가져오기(콘솔)
- 가져오기(AWS CLI)

가져오기(콘솔)

다음 예제에서는 AWS Management Console을 사용하여 인증서를 가져오는 방법을 보여 줍니다.

1. https://console.aws.amazon.com/acm/home에서 ACM 콘솔을 엽니다. ACM을 처음 사용하는 경우 AWS Certificate Manager 머리글을 찾아 그 아래에 있는 시작하기 버튼을 선택하세요.
2. [Import a certificate]을 선택합니다.
3. 해결 방법:

   a. [Certificate body]에서 PEM 인코딩된 인증서를 붙여넣어 가져옵니다. `-----BEGIN CERTIFICATE-----`으로 시작하고 `-----END CERTIFICATE-----`으로 끝나야 합니다.

   b. [인증서 프라이빗 키(Certificate private key)]에 인증서의 암호화되지 않은 PEM 인코딩 형식 프라이빗 키를 붙여 넣습니다. `-----BEGIN PRIVATE KEY-----`으로 시작하고 `-----END PRIVATE KEY-----`으로 끝나야 합니다.

   c. (선택 사항) [Certificate chain]에 PEM 인코딩된 인증서 체인을 붙여넣습니다.

4. (선택 사항) 가져온 인증서에 태그를 추가하려면 태그를 선택합니다. 태그는 AWS 리소스에 할당하는 레이블입니다. 각 태그는 사용자가 정의하는 키와 선택적 값으로 구성됩니다. 태그를 사용하여 리소스를 정리하거나 AWS 비용을 추적할 수 있습니다.
5. 가져오기를 선택합니다.

가져오기(AWS CLI)

다음 예제에서는 AWS Command Line Interface (AWS CLI)을 사용하여 인증서를 가져오는 방법을 보여 줍니다. 이 예시에서는 다음과 같이 가정합니다.

- PEM 인코딩된 인증서가 `Certificate.pem`이라는 파일에 저장되어 있다.
- PEM 인코딩된 인증서 체인이 `CertificateChain.pem`이라는 파일에 저장되어 있다.
- PEM 인코딩된 비암호화 프라이빗 키가 `PrivateKey.pem`이라는 파일에 저장되어 있다.

다음 예시를 사용하려면 파일의 이름을 바꾸고 명령을 한 줄로 이어서 입력합니다. 다음 예시에는 가독성을 높여주는 줄바꿈과 추가 공백이 포함되어 있습니다.

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

`import-certificate` 명령이 성공적으로 실행되는 경우 가져온 인증서의 Amazon 리소스 이름(ARN)이 반환됩니다.

인증서 다시 가져오기

인증서를 가져와 다른 AWS 서비스와 연결한 경우 원래 인증서의 AWS 서비스 연결을 유지하면서 만료되기 전에 해당 인증서를 다시 가져올 수 있습니다. ACM과 통합된 AWS 서비스에 대한 자세한 내용은 섹션을 참조하세요ACM에 통합된 서비스.

인증서를 다시 가져오려면 조건은 다음과 같습니다.

- 도메인 이름을 추가 및 제거할 수 있습니다.
- 인증서에서 모든 도메인 이름을 제거할 수 없습니다.
- 키 사용 확장이 원래 가져온 인증서에 있는 경우, 새 확장 값을 추가할 수 있지만 기존 값은 제거할 수 없습니다.

- 확장 키 사용 확장이 원래 가져온 인증서에 있는 경우, 새 확장 값을 추가할 수 있지만 기존 값은 제거할 수 없습니다.

  예외: 클라이언트 인증 확장 키 사용을 제거할 수 있습니다. 이는 인증 기관이 더 이상 Chrome의 루트 프로그램 요구 사항을 준수하기 위해 ClientAuth EKU로 인증서를 발급하지 않는 업계 변화를 수용합니다.

  > **Important**
  >
  > 클라이언트 인증 기능이 필요한 경우, ACM은 이전에 가져온 인증서로의 롤백을 지원하지 않으므로 사용자 측에서 추가 검증을 구현해야 합니다.

- 키 종류 및 크기는 변경할 수 없습니다.
- 인증서를 다시 가져올 때는 리소스 태그를 적용할 수 없습니다.

주제

다시 가져오기(콘솔)

다음 예제에서는 AWS Management Console을 사용하여 인증서를 다시 가져오는 방법을 보여 줍니다.

1. https://console.aws.amazon.com/acm/home에서 ACM 콘솔을 엽니다.
2. 다시 가져올 인증서를 선택하거나 확장합니다.
3. 인증서의 세부 정보 창을 열고 [Reimport certificate] 버튼을 선택합니다. 이름 옆의 확인란을 선택하여 인증서를 선택한 경우, [작업(Actions)] 메뉴에서 [인증서 다시 가져오기(Reimport certificate)]를 선택합니다.
4. [인증서 본문(Certificate body)]에 PEM으로 인코딩된 최종 엔터티 인증서를 붙여넣습니다.
5. [Certificate private key]에 인증서의 퍼블릭 키와 연결된 암호화되지 않은 PEM 인코딩 프라이빗 키를 붙여넣습니다.
6. (선택 사항) [Certificate chain]에 PEM 인코딩된 인증서 체인을 붙여넣습니다. 인증서 체인에는 모든 중간 발급 인증 기관에 대한 하나 이상의 인증서 및 루트 인증서가 포함됩니다. 가져올 인증서에 자체 서명이 되어 있는 경우, 인증서 체인을 제공하지 않아도 됩니다.

7. 인증서에 대한 정보를 검토합니다. 오류가 없으면 [Reimport]를 클릭합니다.

다시 가져오기(AWS CLI)

다음 예제에서는 AWS Command Line Interface (AWS CLI)을 사용하여 인증서를 다시 가져오는 방법을 보여 줍니다. 이 예시에서는 다음과 같이 가정합니다.

- PEM 인코딩된 인증서가 `Certificate.pem`이라는 파일에 저장되어 있다.
- PEM 인코딩된 인증서 체인이 `CertificateChain.pem`이라는 파일에 저장되어 있다.
- PEM 인코딩된 비암호화 프라이빗 키가 `PrivateKey.pem`이라는 파일에 저장되어 있다.
- 다시 가져오려는 인증서의 ARN이 있다.

다음 예시를 사용하려면 파일의 이름과 ARN을 바꾸고 명령을 한 줄로 이어서 입력합니다. 다음 예시에는 가독성을 높여주는 줄바꿈과 추가 공백이 포함되어 있습니다.

> **Note**
>
> 인증서를 다시 가져오려면 인증서 ARN을 지정해야 합니다.

```
$ 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` 명령이 성공적으로 실행되는 경우 인증서의 Amazon 리소스 이름(ARN)이 반환됩니다.

인증서 관리

ACM 콘솔 또는를 사용하여 계정의 인증서를 AWS CLI 관리할 수 있습니다.

- 인증서 나열 - ACM에서 관리하는 인증서를 확인합니다. 목록에는 각 인증서에 대한 요약 정보가 포함되어 있습니다.
- 인증서 세부 정보 보기 - 개별 인증서의 세부 정보를 확인합니다.
- 인증서 삭제 - 계정에서 인증서를 제거합니다. 삭제된 인증서는 삭제된 후 잠시 동안 목록에 표시될 수 있습니다.

에서 관리하는 인증서 나열 AWS Certificate Manager

ACM 콘솔 또는를 사용하여 ACM에서 관리하는 인증서를 나열 AWS CLI 할 수 있습니다. 콘솔에서는 한 페이지에 최대 500개의 인증서를 나열하고 CLI에서는 최대 1,000개의 인증서를 나열할 수 있습니다.

콘솔을 사용하여 인증서를 나열하려면

1. https://console.aws.amazon.com/acm/에서 ACM 콘솔을 엽니다.
2. 인증서 목록의 정보를 검토합니다. 오른쪽 상단의 페이지 번호를 사용하여 여러 페이지의 인증서를 탐색할 수 있습니다. 각 인증서는 인증서별로 기본적으로 표시되는 다음 열이 있는 행을 하나씩 차지합니다.

- 도메인 이름 - 인증서의 정규화된 도메인 이름(FQDN)입니다.
- 유형 - 인증서의 유형입니다. 가능한 값: Amazon 발급(Amazon Issued) | 프라이빗(Private) | 가져옴(Imported)
- 상태 - 인증서 상태입니다. 가능한 값: 검증 보류 중(Pending validation) | 발급됨(Issued) | 비활성(Inactive) | 만료됨(Expired) | 취소됨(Revoked) | 실패(Failed) | 검증 시간 초과(Validation timed out)
- 사용 중입니까? - ACM 인증서가 Elastic Load Balancing 또는 CloudFront와 같은 AWS 서비스와 적극적으로 연결되어 있는지 여부입니다. 값은 아니요(No) 또는 예(Yes)일 수 있습니다.
- 갱신 자격 – 인증서 만료가 가까워지면 ACM에서 인증서를 자동으로 갱신할 수 있는지 여부. 가능한 값: 적격(Eligible) | 부적격(Ineligible) 자격 규칙은 에서 관리형 인증서 갱신 AWS Certificate Manager 섹션을 참조하세요.

콘솔 상단 오른쪽의 설정 아이콘을 선택하면 페이지에 표시되는 인증서 수를 사용자 지정하고 셀 콘텐츠의 줄 바꿈 동작을 지정하며 추가 정보 필드를 표시할 수 있습니다. 다음과 같은 선택적 필드를 사용할 수 있습니다.

- 추가 도메인 이름 - 인증서에 포함된 하나 이상의 도메인 이름(주체 대체 이름)입니다.
- 요청 시간 - ACM이 인증서를 요청한 시간입니다.
- 발급 시간 - 인증서가 발급된 시간입니다. 이 정보는 Amazon에서 발급한 인증서에만 사용할 수 있으며 가져온 인증서에는 사용할 수 없습니다.
- 다음 이전에 적용되지 않음 — 이 시간 이전에는 인증서가 유효하지 않습니다.
- 다음 이후에 적용되지 않음 — 이 시간 이후에는 인증서가 유효하지 않습니다.
- 해지 시간 - 해지된 인증서의 경우 해지 시간입니다.
- Name 태그 — 해당 태그가 있는 경우 이 인증서에 대한 태그의 값을 Name이라고 합니다
- Renewal status(갱신 상태) - 요청된 인증서 갱신 상태입니다. 이 필드는 갱신을 요청한 경우에만 표시되며 값이 있습니다. 가능한 값: Pending automatic renewal(자동 갱신 보류) | Pending validation(검증 보류) | Success(성공) | Failure(실패)

> **Note**
>
> 인증서 상태 변경이 사용 가능한 상태가 되는 데 최대 몇 시간이 소요될 수 있습니다. 문제가 발생한 경우, 72시간 후에 인증서 요청 시간이 경과되므로 처음부터 발급 또는 갱신 프로세스를 다시 반복해야 합니다.

페이지 크기(Page size) 기본 설정은 각 콘솔 페이지에서 반환되는 인증서 수를 지정합니다.

사용 가능한 인증서 세부 정보에 대한 자세한 내용은 AWS Certificate Manager 인증서 세부 정보 보기 섹션을 참조하세요.

를 사용하여 인증서를 나열하려면 AWS CLI

list-certificates 명령을 사용하여 다음 예와 같이 ACM 관리형 인증서를 나열합니다.

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

이 명령은 다음과 유사한 정보를 반환합니다.

```
{
```

```
    "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"
        },...
    ]
}
```

기본적으로 keyTypes가 RSA_1024 또는 RSA_2048이고 하나 이상의 지정된 도메인이 있는 인증서만 반환됩니다. 도메인이 없는 인증서나 다른 알고리즘 또는 비트 크기를 사용하는 인증서와 같이 제어하는 다른 인증서를 보려면 다음 예와 같이 `--includes` 매개변수를 제공합니다. 파라미터를 사용하여 필터 구조의 멤버를 지정할 수 있습니다.

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

AWS Certificate Manager 인증서 세부 정보 보기

ACM 콘솔 또는를 사용하여 인증서에 대한 자세한 메타데이터를 나열 AWS CLI 할 수 있습니다.

콘솔에서 인증서 세부 정보를 보려면

1. https://console.aws.amazon.com/acm/에서 ACM 콘솔을 열고 인증서를 표시합니다. 오른쪽 상단의 페이지 번호를 사용하여 여러 페이지의 인증서를 탐색할 수 있습니다.
2. 나열된 인증서에 대한 자세한 메타데이터를 표시하려면 인증서 ID를 선택합니다. 다음 정보를 표시하는 페이지가 열립니다.

   - 인증서 상태
     - 식별자 - 인증서의 16진수로 된 32바이트 고유 식별자입니다.
     - ARN - `arn:aws:acm:`*`Region`*`:`*`444455556666`*`:certificate/`*`certificate_ID`* 형식의 Amazon 리소스 이름(ARN)입니다.
     - 유형 - ACM 인증서의 관리 범주를 식별합니다. 가능한 값은 [Amazon 발급(Amazon Issued)] | [프라이빗(Private)] | [가져옴(Imported)]입니다. 자세한 내용은 AWS Certificate Manager 퍼블릭 인증서, 에서 프라이빗 인증서 요청 AWS Certificate Manager 또는 로 인증서 가져오기 AWS Certificate Manager 섹션을 참조하세요.
     - 상태 - 인증서의 상태입니다. 가능한 값: 검증 보류 중(Pending validation) | 발급됨(Issued) | 비활성(Inactive) | 만료됨(Expired) | 취소됨(Revoked) | 실패(Failed) | 검증 시간 초과(Validation timed out)
     - 세부 상태 - 인증서를 요청하거나 가져온 날짜 및 시간입니다.
   - 도메인
     - 도메인 - 인증서의 정규화된 도메인 이름(FQDN)입니다.
     - 상태 - 도메인 검증 상태입니다. 가능한 값: 검증 보류 중(Pending validation) | 취소됨(Revoked) | 실패(Failed) | 검증 시간 초과(Validation timed out) | 성공(Success)
   - 세부 정보
     - 사용 중입니까? - 인증서가AWS 통합 서비스에 연결되어 있는지 여부를 나타냅니다. 가능한 값은 [예(Yes) | [아니요(No)]입니다.
     - 도메인 이름 - 인증서의 첫 번째 FQDN(Fully Qualified Domain Name)입니다.
     - 관리형 - ACM으로 인증서를 관리하는 AWS 서비스를 식별합니다.
     - 추가 이름 수 - 인증서가 유효한 도메인 이름의 수입니다.
     - 일련번호 - 16바이트 16진수로 된 인증서의 일련번호입니다.
     - 퍼블릭 키 정보 - 키 페어를 생성한 암호화 알고리즘입니다.
     - 서명 알고리즘 - 인증서에 서명하는 데 사용된 암호화 알고리즘입니다.
     - Can be used with(다음에 사용 가능) - ACM 통합 서비스의 목록으로, 다음 파라미터가 있는 인증서를 지원합니다.

- 요청 시기 - 발급 요청 날짜 및 시간입니다.
- 발급 시기 - 해당되는 경우 발급 날짜 및 시간입니다.
- 가져온 시기 - 해당되는 경우 가져온 날짜 및 시간입니다.
- 다음 이전에 적용되지 않음 - 인증서 유효 기간의 시작일입니다.
- 다음 이후에 적용되지 않음 - 인증서의 만료 날짜 및 시간입니다.
- Renewal eligibility(갱신 자격) - 가능한 값: Eligible(적격) | Ineligible(부적격) 자격 규칙은 에서 관리형 인증서 갱신 AWS Certificate Manager 섹션을 참조하세요.
- Renewal status(갱신 상태) - 요청된 인증서 갱신 상태입니다. 이 필드는 갱신을 요청한 경우에만 표시되며 값이 있습니다. 가능한 값: Pending automatic renewal(자동 갱신 보류) | Pending validation(검증 보류) | Success(성공) | Failure(실패)

> **Note**
>
> 인증서 상태 변경이 사용 가능한 상태가 되는 데 최대 몇 시간이 소요될 수 있습니다. 문제가 발생한 경우, 72시간 후에 인증서 요청 시간이 경과되므로 처음부터 발급 또는 갱신 프로세스를 다시 반복해야 합니다.

- CA - 서명 CA의 ARN

- Tags
  - Key(키)
  - 값
- 검증 상태 - 해당하는 경우 가능한 값은 다음과 같습니다.
  - 검증 보류 중 - 검증이 요청되었으며 완료되지 않았습니다.
  - 검증 시간 초과 - 요청된 검증이 시간 초과되었지만 요청을 반복할 수 있습니다.
  - 없음 - 인증서가 프라이빗 PKI용이거나 자체 서명되어 있어 검증이 필요하지 않습니다.

를 사용하여 인증서 세부 정보를 보려면 AWS CLI

다음 명령과 같이의 describe-certificate AWS CLI 를 사용하여 인증서 세부 정보를 표시합니다.

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

이 명령은 다음과 유사한 정보를 반환합니다.

```
{
    "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"
    }
}
```

에서 관리하는 인증서 삭제 AWS Certificate Manager

ACM 콘솔 또는 AWS CLI 를 사용하여 인증서를 삭제할 수 있습니다. 티켓 삭제는 최종적으로 일관됩니다. 인증서는 삭제된 후 잠시 동안 목록에 표시될 수 있습니다.

⚠ Important

- 다른 AWS 서비스가 사용 중인 ACM 인증서는 삭제할 수 없습니다. 사용 중인 인증서를 삭제하려면 먼저 인증서 연결을 제거해야 합니다. 이 작업은 연결된 서비스에 대한 콘솔 또는 CLI 를 사용하여 수행합니다.

- Private Certificate Authority(CA)에서 발급한 인증서를 삭제해도 CA에 영향을 주지 않습니다. CA가 삭제될 때까지 CA에 대한 요금이 계속 청구됩니다. 자세한 정보는 AWS Private Certificate Authority 사용 설명서의 Private CA 삭제를 참조하세요.

콘솔을 사용하여 인증서를 삭제하려면

1. https://console.aws.amazon.com/acm/에서 ACM 콘솔을 엽니다.
2. 인증서 목록에서 ACM 인증서의 확인란을 선택하고 삭제(Delete)를 선택합니다.

> **Note**
>
> 목록을 정렬한 방법에 따라 찾고 있는 인증서가 즉시 표시되지 않을 수 있습니다. 오른쪽의 검은색 삼각형을 클릭하여 순서를 변경할 수 있습니다. 오른쪽 상단의 페이지 번호를 사용하여 여러 페이지의 인증서를 탐색할 수도 있습니다.

를 사용하여 인증서를 삭제하려면 AWS CLI

다음 명령과 같이, delete-certificate를 사용하여 인증서를 삭제합니다.

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

에서 관리형 인증서 갱신 AWS Certificate Manager

ACM은 Amazon 발급 SSL/TLS 인증서에 대한 관리형 갱신을 제공합니다. 즉, ACM이 인증서를 자동으로 갱신(DNS 검증을 사용하는 경우)하거나 만료 시점이 다가오면 이메일 알림을 보냅니다. 이러한 서비스는 퍼블릭 및 프라이빗 ACM 인증서 모두에 대해 제공됩니다.

인증서는 다음 고려 사항에 따라 자동 갱신이 가능합니다.

- Elastic Load Balancing 또는 CloudFront와 같은 다른 AWS 서비스와 연결된 경우 자격이 있습니다.
- 발급되거나 마지막으로 갱신된 이후 내보내진 경우 자격이 있습니다.
- ACM RequestCertificate API를 직접 호출하여 프라이빗 인증서를 발급한 다음, 다른 AWS 서비스와 연결하거나 내보낸 경우 자격이 있습니다.
- 관리 콘솔을 통해 발급한 다음 다른 AWS 서비스로 내보내거나 다른 서비스와 연결한 프라이빗 인증서인 경우 자격이 있습니다.
- AWS Private CA IssueCertificate API를 직접 호출하여 프라이빗 인증서를 발급한 경우 자격이 없습니다.
- 가져온 경우 자격이 없습니다.
- 이미 만료된 경우 자격이 없습니다.

또한, 다국어 도메인 이름과 관련된 다음 Punycode 요구 사항을 충족해야 합니다.

1. ‘<character><character>--’ 패턴으로 시작하는 도메인 이름은 ‘xn--’과 일치해야 합니다.
2. ‘xn--’으로 시작하는 도메인 이름도 유효한 다국어 도메인 이름이어야 합니다.

Punycode 예제

도메인 이름	#1 충족	#2 충족	허용	Note
example.com	해당 사항 없음	해당 사항 없음	✓	‘<character><character>--’로 시작하지 않음
a--example.com	해당 사항 없음	해당 사항 없음	✓	‘<character><character>--’로 시작하지 않음

도메인 이름	#1 충족	#2 충족	허용	Note
abc--example.com	해당 사항 없음	해당 사항 없음	✓	'<character><character>--'로 시작하지 않음
xn—xyz.com	예	예	✓	유효한 다국어 도메인 이름(简.com으로 확인)
xn--example.com	예	아니요	✗	유효한 다국어 도메인 이름이 아님
ab--example.com	아니요	아니요	✗	'xn--'으로 시작해야 함

ACM에서 인증서를 갱신해도 인증서의 Amazon 리소스 이름(ARN)은 동일하게 유지됩니다. 또한 ACM 인증서는 리전별 리소스입니다. 여러 AWS 리전에 동일한 도메인 이름의 인증서가 있는 경우 각 인증서를 독립적으로 갱신해야 합니다.

주제

- ACM 퍼블릭 인증서 갱신
- 에서 프라이빗 인증서 갱신 AWS Certificate Manager
- 인증서의 갱신 상태 확인

ACM 퍼블릭 인증서 갱신

공개적으로 신뢰할 수 있는 관리형 인증서를 발급할 때는 도메인 소유자임을 증명해야 AWS Certificate Manager 합니다. 이 과정은 DNS 검증 또는 이메일 검증을 통해 진행됩니다. 인증서 갱신이 시작되면 ACM은 이전에 사용자가 선택한 것과 동일한 방법을 사용하여 소유권을 다시 검증합니다. 다음 주제에서는 각 사례에서 갱신 프로세스가 작동하는 방식에 대해 설명합니다.

주제

- DNS로 검증된 도메인의 갱신
- 이메일로 검증된 도메인에 대한 갱신
- HTTP로 검증된 도메인의 갱신

DNS로 검증된 도메인의 갱신

관리형 갱신은 원래 DNS 검증을 사용하여 발급된 ACM 인증서에 대해 완전 자동화됩니다.

만료 45일 전에 ACM은 다음 갱신 기준을 확인합니다.

> **Note**
>
> 395일의 유효 기간이 있는 이전에 발급된 인증서는 만료 60일 전에 갱신되며 198일의 갱신된 유효 기간을 받습니다. 유효 기간이 198일인 인증서는 만료 45일 전에 갱신됩니다.

- 인증서는 현재 AWS 서비스에서 사용 중입니다.
- ACM에서 제공하는 모든 필수 DNS CNAME 레코드(각 고유 주체 대체 이름에 대해 하나씩)가 존재하며 공용 DNS를 통해 액세스할 수 있습니다.

이러한 조건이 충족되면 ACM은 도메인 이름을 검증한 후 인증서를 갱신합니다.

ACM은 갱신 중에 도메인을 자동으로 검증할 수 없는 경우 AWS Health 이벤트와 Amazon EventBridge 이벤트를 전송합니다. 이러한 이벤트는 만료 30일, 15일, 7일, 3일 및 1일 전에 전송됩니다. 자세한 내용은 ACM에 대한 Amazon EventBridge 지원 단원을 참조하십시오.

이메일로 검증된 도메인에 대한 갱신

ACM 인증서는 198일 동안 유효합니다. 인증서를 갱신하려면 도메인 소유자의 작업이 필요합니다. ACM은 만료 45일 전에 도메인과 연결된 이메일 주소로 갱신 알림을 전송하기 시작합니다. 이 알림에는 도메인 소유자가 갱신을 위해 클릭할 수 있는 링크가 포함되어 있습니다. 나열된 모든 도메인이 검증되면 ACM은 동일한 ARN 사용하여 갱신된 인증서를 발급합니다.

ACM은 갱신 중에 도메인을 자동으로 검증할 수 없는 경우 AWS Health 이벤트와 Amazon EventBridge 이벤트를 전송합니다. 이러한 이벤트는 만료 30일, 15일, 7일, 3일 및 1일 전에 전송됩니다. 자세한 내용은 ACM에 대한 Amazon EventBridge 지원 단원을 참조하십시오.

검증 이메일 메시지에 대한 자세한 내용은 AWS Certificate Manager 이메일 검증 섹션을 참조하세요.

검증 이메일에 프로그래밍 방식으로 응답하는 방법을 알아보려면 AWS Certificate Manager 이메일 검증 자동화 섹션을 참조하세요.

검증 이메일 재전송

인증서를 요청할 때 도메인에 대한 이메일 검증을 구성한 후(참조AWS Certificate Manager 이메일 검증) AWS Certificate Manager API를 사용하여 ACM이 인증서 갱신을 위한 도메인 검증 이메일을 보내도록 요청할 수 있습니다. 이 작업은 다음과 같은 경우에 수행해야 합니다.

- ACM 인증서를 처음 요청할 때 이메일 검증을 사용했습니다.
- 인증서의 갱신 상태가 검증 보류 중인 경우. 인증서의 갱신 상태를 확인하는 방법에 대한 자세한 내용은 인증서의 갱신 상태 확인 단원을 참조하세요.
- ACM이 인증서 갱신에 대해 전송한 원본 도메인 검증 이메일 메시지를 받지 못했거나 찾을 수 없는 경우

인증서 요청에서 원래 구성한 것과 다른 도메인으로 검증 이메일을 보내려면 ACM API AWS CLI또는 AWS SDK에서 ResendValidationEmail 작업을 사용할 수 있습니다. SDKs ACM은 지정된 검증 도메인으로 이메일을 전송합니다. 지원되는 리전 AWS CLI 에서를 사용하여 브라우저 AWS CloudShell 에서에 액세스할 수 있습니다.

ACM이 도메인 검증 이메일 메시지을 재전송하도록 요청하려면(콘솔)

1. https://console.aws.amazon.com/acm/home AWS Certificate Manager 콘솔을 엽니다.
2. 검증이 필요한 인증서의 인증서 ID를 선택합니다.
3. 검증 이메일 재전송(Resend validation email)을 선택합니다.

ACM이 도메인 검증 이메일(ACM API)을 재전송하도록 요청하려면

ACM API에서 ResendValidationEmail 작업을 사용합니다. 이때 인증서의 ARN, 수동 검증이 필요한 도메인, 도메인 검증 이메일을 수신하려는 도메인을 전달합니다. 다음 예에서는 AWS CLI에서 이 작업을 수행하는 방법을 보여줍니다. 이 예에는 이해하기 쉽도록 줄 바꿈이 포함되어 있습니다.

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

HTTP로 검증된 도메인의 갱신

ACM은 원래 CloudFront를 통한 HTTP 검증을 사용하여 발급된 인증서에 대해 자동화된 관리형 갱신을 제공합니다.

만료 45일 전에 ACM은 다음 갱신 기준을 확인합니다.

> **Note**
>
> 395일의 유효 기간이 있는 이전에 발급된 인증서는 만료 60일 전에 갱신되며 198일의 갱신된 유효 기간을 받습니다. 유효 기간이 198일인 인증서는 만료 45일 전에 갱신됩니다.

- 인증서가 현재 CloudFront에서 사용되고 있습니다.
- 필요한 모든 HTTP 검증 레코드에 액세스할 수 있으며 예상된 콘텐츠를 포함하고 있습니다.

이러한 조건이 충족되면 ACM은 도메인 이름을 검증한 후 인증서를 갱신합니다.

ACM은 갱신 중에 도메인을 자동으로 검증할 수 없는 경우 AWS Health 이벤트와 Amazon EventBridge 이벤트를 전송합니다. 이러한 이벤트는 만료 30일, 15일, 7일, 3일 및 1일 전에 전송됩니다. 자세한 내용은 ACM에 대한 Amazon EventBridge 지원 단원을 참조하십시오.

성공적인 갱신을 위해 `RedirectFrom` 위치의 콘텐츠가 인증서의 각 도메인에 대한 `RedirectTo` 위치의 콘텐츠와 일치하는지 확인해야 합니다.

에서 프라이빗 인증서 갱신 AWS Certificate Manager

에서 프라이빗 CA가 서명한 ACM 인증서 AWS Private CA 는 관리형 갱신이 가능합니다. 공개적으로 신뢰할 수 있는 ACM 인증서와 달리 프라이빗 PKI에 대한 인증서에는 검증이 필요하지 않습니다. 트러스트는 관리자가 클라이언트 트러스트 스토어에 적절한 루트 CA 인증서를 설치할 때 설정됩니다.

> **Note**
>
> ACM 콘솔 또는 ACM API의 RequestCertificate 작업을 사용하여 획득한 인증서만 관리형 갱신 자격이 있습니다. AWS Private CA API의 IssueCertificate 작업을 AWS Private CA 사용하여에서 직접 발급된 인증서는 ACM에서 관리하지 않습니다.

관리형 인증서의 만료일이 60일 앞으로 다가오면 ACM은 인증서 자동 갱신을 시도합니다. 여기에는 수동으로 내보내고 설치한 인증서(예: 온프레미스 데이터 센터의 경우)가 포함됩니다. 또한 고객은 언제든지 ACM API의 RenewCertificate 작업을 사용하여 강제로 갱신할 수 있습니다. 강제 갱신의 Java 구현 샘플은 인증서 갱신 섹션을 참조하세요.

갱신 후 다음 중 한 가지 방법으로 인증서의 서비스가 배포됩니다.

- 인증서가 ACM 통합 서비스와 연결된 경우 새 인증서가 추가 고객 작업 없이 이전 인증서를 대체합니다.
- 인증서가 ACM 통합 서비스와 연결되지 않은 경우 갱신된 인증서를 내보내고 설치하는 고객 작업이 필요합니다. 이러한 작업은 수동으로 수행하거나 다음과 같이 AWS Health, Amazon EventBridge 및 AWS Lambda의 도움을 받아 수행할 수 있습니다. 자세한 내용은 갱신된 인증서 내보내기 자동화 섹션을 참조하세요.

갱신된 인증서 내보내기 자동화

다음 절차에서는 ACM이 인증서를 갱신할 때 프라이빗 PKI 인증서의 내보내기를 자동화하는 예제 솔루션을 제공합니다. 이 예제에서는 인증서와 해당 프라이빗 키만 ACM에서 내보냅니다. 내보낸 후에는 인증서를 대상 디바이스에 설치해야 합니다.

콘솔을 사용하여 인증서 내보내기 자동화

1. AWS Lambda 개발자 안내서의 절차에 따라 ACM 내보내기 API를 호출하는 Lambda 함수를 생성하고 구성합니다.

   a. Lambda 함수를 생성합니다.

   b. 함수에 대해 Lambda 실행 역할을 생성하고 다음 신뢰 정책을 추가합니다. 정책은 ACM API의 ExportCertificate 작업을 호출하여 함수의 코드에 갱신된 인증서 및 프라이빗 키를 검색할 수 있는 권한을 부여합니다.

      JSON

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

2.

Amazon EventBridge에서 규칙을 생성하여 ACM 상태 이벤트를 수신하고 Lambda 함수가 감지되면 Lambda 함수를 호출합니다. ACM은 인증서 갱신을 시도할 때마다 AWS Health 이벤트에 씁니다. 이러한 알림에 대한 자세한 내용은 Personal Health Dashboard(PHD)를 사용하여 상태 확인 섹션을 참조하세요.

다음 이벤트 패턴을 추가하여 규칙을 구성합니다.

```
{
   "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. 대상 시스템에 인증서를 수동으로 설치하여 갱신 프로세스를 완료합니다.

프라이빗 PKI 인증서의 관리형 갱신 테스트

ACM API 또는를 사용하여 ACM 관리형 갱신 워크플로의 구성을 수동으로 테스트 AWS CLI 할 수 있습니다. 이렇게 하면 인증서가 만료 전 ACM에 의해 자동으로 갱신되는지 확인할 수 있습니다.

> **Note**
>
> AWS Private CA에서 발급하고 내보낸 인증서의 갱신만 테스트할 수 있습니다.

아래에 설명된 API 작업 또는 CLI 명령을 사용하면 ACM이 인증서 갱신을 시도합니다. 갱신이 성공하면 ACM은 관리 콘솔 또는 API 출력에 표시된 인증서 메타데이터를 업데이트합니다. 인증서가 ACM 통합 서비스와 연결된 경우 새 인증서가 배포되고 Amazon CloudWatch Events에 갱신 이벤트가 생성됩니다. 갱신이 실패하면 ACM이 오류를 반환하고 해결 조치를 제안합니다. (이 정보는 describe-certificate 명령을 사용하여 볼 수 있음) 통합 서비스를 통해 인증서가 배포되지 않은 경우에도 인증서를 내보내고 리소스에 수동으로 설치해야 합니다.

> **Important**
>
> ACM으로 AWS Private CA 인증서를 갱신하려면 먼저 ACM 서비스 보안 주체에게 권한을 부여해야 합니다. 자세한 내용은 ACM에 인증서 갱신 권한 할당을 참조하세요.

인증서 갱신(AWS CLI)을 수동으로 테스트하려면

1. renew-certificate 명령을 사용하여 내보낸 사설 인증서를 갱신합니다.

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

2. 그런 다음 describe-certificate 명령을 사용하여 인증서의 갱신 세부 정보가 업데이트되었는지 확인합니다.

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

인증서 갱신을 수동으로 테스트하려면(ACM API)

- RenewCertificate 요청을 보내어 갱신할 사설 인증서의 ARN을 지정합니다. 그런 다음 DescribeCertificate 작업을 사용하여 인증서의 갱신 세부 정보가 업데이트되었는지 확인합니다.

인증서의 갱신 상태 확인

인증서를 갱신하려고 하면 ACM은 인증서 세부 정보에 갱신 상태 정보 필드를 제공합니다. AWS Certificate Manager 콘솔, ACM API AWS CLI, 또는를 사용하여 ACM 인증서의 갱신 상태를 Health Dashboard 확인할 수 있습니다. 콘솔 AWS CLI또는 ACM API를 사용하는 경우 갱신 상태에는 아래 나열된 네 가지 가능한 상태 값 중 하나가 있을 수 있습니다. Health Dashboard를 사용하는 경우에도 비슷한 값이 표시됩니다.

자동 갱신 보류

ACM이 인증서에서 도메인 이름을 자동으로 검증하려고 시도 중입니다. 자세한 내용은 DNS로 검증된 도메인의 갱신 단원을 참조하십시오. 추가 조치가 필요하지 않습니다.

검증 보류

ACM이 인증서에서 하나 이상의 도메인 이름을 자동으로 검증할 수 없습니다. 사용자가 이러한 도메인 이름을 검증해야 합니다. 그렇지 않으면 인증서가 갱신되지 않습니다. 해당 인증서에서 원래부터 이메일 검증을 사용한 경우에는 ACM이 이메일을 찾아서 해당 이메일의 링크를 따라 검증을 수행합니다. DNS 검증을 사용한 경우에는 DNS 레코드가 존재하고 인증서가 사용 상태로 유지되는지 확인합니다.

Success

인증서의 모든 도메인 이름이 검증되어 ACM이 인증서를 갱신했습니다. 추가 조치가 필요하지 않습니다.

실패

인증서가 만료되기 전에 하나 이상의 도메인 이름이 검증되지 않아 ACM이 인증서를 갱신하지 못했습니다. 새 인증서를 요청할 수 있습니다.

인증서가 Elastic Load Balancing 또는 CloudFront와 같은 다른 AWS 서비스와 연결되어 있거나 발급되거나 마지막으로 갱신된 이후 내보낸 경우 인증서를 갱신할 수 있습니다.

> **Note**
>
> 갱신 상태 변경이 사용 가능한 상태가 되는 데 최대 몇 시간이 소요될 수 있습니다. 문제가 발생한 경우, 72시간 후에 갱신 요청 시간이 경과되므로 처음부터 갱신 프로세스를 다시 반복해야 합니다. 문제 해결에 대한 도움말은 인증서 요청 문제 해결 섹션을 참조하세요.

주제

- 상태 확인 (콘솔)
- 상태 확인 (API)
- 상태 확인 (CLI)
- Personal Health Dashboard(PHD)를 사용하여 상태 확인

상태 확인 (콘솔)

다음 절차에서는 ACM 콘솔을 사용하여 ACM 인증서의 갱신 상태를 확인하는 방법을 설명합니다.

1. https://console.aws.amazon.com/acm/home AWS Certificate Manager 콘솔을 엽니다.
2. 인증서를 펼쳐서 세부 정보를 확인합니다.
3. Details(세부 정보) 섹션에서 Renewal Status(갱신 상태)를 찾습니다. 상태가 표시되지 않는 경우 ACM이 이 인증서에 대한 관리형 갱신 프로세스를 시작하지 않은 것입니다.

상태 확인 (API)

DescribeCertificate 작업을 사용하여 상태를 확인하는 방법을 보여주는 Java 예제는 인증서 설명 단원을 참조하세요.

상태 확인 (CLI)

다음 예에서는 AWS Command Line Interface (AWS CLI)를 사용하여 ACM 인증서 갱신 상태를 확인하는 방법을 보여 줍니다.

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

응답에서 `RenewalStatus` 필드의 값을 적어 둡니다. `RenewalStatus` 필드가 표시되지 않는 경우 ACM이 이 인증서에 대한 관리형 갱신 프로세스를 시작하지 않은 것입니다.

Personal Health Dashboard(PHD)를 사용하여 상태 확인

ACM은 퍼블릭 인증서의 경우 만료 45일 전, 프라이빗 인증서의 경우 만료 60일 전에 ACM 인증서를 자동으로 갱신하려고 시도합니다. ACM이 인증서를 자동으로 갱신할 수 없는 경우 만료 Health Dashboard 후 45일(프라이빗만 해당), 30일, 15일, 7일, 3일 및 1일 간격으로에 인증서 갱신 이벤트 알림을 보내 조치를 취해야 함을 알립니다. Health Dashboard 는 AWS Health 서비스의 일부입니다. 설정이 필요하지 않으며, 계정에서 인증된 사용자면 누구나 볼 수 있습니다. 자세한 내용은 AWS Health 사용 설명서를 참조하세요.

> **Note**
>
> ACM은 PHD 타임라인의 단일 이벤트에 연속 갱신 이벤트 알림을 기록합니다. 각 알림은 갱신이 성공할 때까지 이전 알림을 덮어씁니다.

Health Dashboard사용 방법:

1. Health Dashboard 에 로그인합니다. https://phd.aws.amazon.com/phd/home
2. [Event log]를 선택합니다.
3. [Filter by tags or attributes]에서 [Service]를 선택합니다.
4. [Certificate Manager]를 선택합니다.
5. 적용을 선택합니다.
6. [Event category]에서 [Scheduled Change]를 선택합니다.
7. 적용을 선택합니다.

AWS Certificate Manager 리소스 태그 지정

태그는 ACM 인증서에 할당할 수 있는 레이블입니다. 각 태그는 키와 값으로 구성됩니다. AWS Certificate Manager 콘솔, AWS Command Line Interface (AWS CLI) 또는 ACM API를 사용하여 ACM 인증서에 대한 태그를 추가, 확인 또는 제거할 수 있습니다. ACM 콘솔에 표시할 태그를 선택할 수 있습니다.

요구 사항을 충족하는 사용자 지정 태그를 생성할 수 있습니다. 예를 들어 여러 ACM 인증서에 `Environment = Prod` 또는 `Environment = Beta` 태그를 지정하여 각 ACM 인증서가 사용되는 환경을 식별할 수 있습니다. 다음 목록에는 다른 사용자 지정 태그에 대한 몇 가지 예시가 나와 있습니다.

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

다른 AWS 리소스도 태그 지정을 지원합니다. 따라서 동일한 태그를 다른 리소스에 할당하여 해당 리소스의 관련 여부를 나타낼 수 있습니다. 예를 들어, `Website = example.com`과 같은 태그를 ACM 인증서, 로드 밸런서 및 example.com 웹 사이트에 사용되는 기타 리소스에 할당할 수 있습니다.

주제

태그 제한 사항

ACM 인증서 태그에 적용되는 기본 제한은 다음과 같습니다.

- ACM 인증서당 최대 태그 수는 50개입니다.
- 태그 키의 최대 길이는 127자입니다.
- 태그 값의 최대 길이는 255자입니다.
- 태그 키와 값은 대소문자를 구분합니다.
- `aws:` 접두사는 AWS 사용을 위해 예약되어 있습니다. 키가 로 시작하는 태그를 추가, 편집 또는 삭제할 수 없습니다`aws:`. `aws:`로 시작하는 태그는 tags-per-resource 할당량에 포함되지 않습니다.

- 태그 지정 스키마를 여러 서비스와 리소스에서 사용하려는 경우, 서비스마다 허용되는 문자에 대한 제한이 다를 수 있음에 유의하세요. 해당 서비스에 대한 문서를 참조하세요.
- ACM 인증서 태그는 AWS Management Console의 리소스 그룹 및 태그 편집기에서 사용할 수 없습니다.

AWS 태그 지정 규칙에 대한 일반적인 내용은 AWS 리소스 태그 지정을 참조하세요.

태그 관리

AWS 관리 콘솔, 또는 AWS Certificate Manager API를 사용하여 태그를 추가 AWS Command Line Interface, 편집 및 삭제할 수 있습니다.

태그 관리(콘솔)

AWS Management Console 를 사용하여 태그를 추가, 삭제 또는 편집할 수 있습니다. 열에 태그를 표시할 수도 있습니다.

태그 추가

다음 절차에 따라 ACM 콘솔을 사용하여 태그를 추가합니다.

태그를 인증서에 추가하려면(콘솔)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/acm/home AWS Certificate Manager 콘솔을 엽니다.
2. 태그를 지정할 인증서 옆에 있는 화살표를 선택합니다.
3. 세부 정보 창에서 [Tags]가 나타날 때까지 아래로 스크롤합니다.
4. [Edit] 및 [Add Tag]를 선택합니다.
5. 태그의 키와 값을 입력합니다.
6. 저장을 선택합니다.

태그 삭제

다음 절차에 따라 ACM 콘솔을 사용하여 태그를 삭제합니다.

태그를 삭제하려면(콘솔)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/acm/home AWS Certificate Manager 콘솔을 엽니다.
2. 삭제하려는 태그가 포함된 인증서 옆의 화살표를 선택합니다.
3. 세부 정보 창에서 [Tags]가 나타날 때까지 아래로 스크롤합니다.
4. 편집을 선택합니다.
5. 삭제하려는 태그 옆의 [X]를 선택합니다.
6. 저장을 선택합니다.

태그 편집

다음 절차에 따라 ACM 콘솔을 사용하여 태그를 편집합니다.

태그를 편집하려면(콘솔)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/acm/home AWS Certificate Manager 콘솔을 엽니다.
2. 편집하려는 인증서 옆의 화살표를 선택합니다.
3. 세부 정보 창에서 [Tags]가 나타날 때까지 아래로 스크롤합니다.
4. 편집을 선택합니다.
5. 변경하려는 태그의 키 또는 값을 수정합니다.
6. 저장을 선택합니다.

열에 태그 표시

다음 절차에 따라 ACM 콘솔의 열에 태그를 표시합니다.

열에 태그를 표시하려면(콘솔)

1. 에 로그인 AWS Management Console 하고 https://console.aws.amazon.com/acm/home AWS Certificate Manager 콘솔을 엽니다.
2. 콘솔 상단 오른쪽의 기어 모양 아이콘

   을 선택하여 열로 표시하려는 태그를 선택합니다.

3. 열에 표시하려는 태그 옆의 확인란을 선택합니다.

태그 관리(API)

AWS CLI를 사용하여 태그를 추가, 나열 및 삭제하는 방법을 알아보려면 다음 주제를 참조하세요.

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

태그 관리(ACM API)

API를 사용하여 태그를 추가, 나열 및 삭제하는 방법을 알아보려면 다음 주제를 참조하세요.

- AddTagsToCertificate
- ListTagsForCertificate
- RemoveTagsFromCertificate

ACM에 통합된 서비스

AWS Certificate Manager 는 점점 더 많은 AWS 서비스를 지원합니다. AWS 기반 웹 사이트 또는 애플리케이션에 ACM 인증서 또는 프라이빗 AWS Private CA 인증서를 직접 설치할 수 없습니다.

> **Note**
>
> 퍼블릭 ACM 인증서는 Nitro Enclave에 연결된 Amazon EC2 인스턴스에 설치할 수 있습니다. 모든 Amazon EC2 인스턴스에서 사용할 퍼블릭 인증서를 내보낼 수도 있습니다. Nitro Enclave에 연결되지 않은 Amazon EC2 인스턴스에서 독립형 웹 서버를 설정하는 방법에 대한 자세한 내용은 자습서: Amazon Linux 2에 LAMP 웹 서버 설치 또는 자습서: Amazon Linux AMI를 사용하여 LAMP 웹 서버 설치를 참조하세요.

ACM 인증서는 다음 서비스에서 지원됩니다.

Elastic Load Balancing

Elastic Load Balancing은 수신 애플리케이션 트래픽을 여러 Amazon EC2 인스턴스에 자동으로 분산합니다. 또한 비정상 인스턴스를 검색하고 비정상 인스턴스가 복원될 때까지 트래픽을 정상 인스턴스로 다시 라우팅합니다. Elastic Load Balancing은 수신되는 트래픽에 맞춰 요청 처리 용량을 자동으로 조정합니다. 로드 밸런싱에 대한 자세한 내용은 Elastic Load Balancing 사용 설명서를 참조하세요.

일반적으로 SSL/TLS를 통해 보안 콘텐츠를 제공하려면 SSL/TLS 인증서를 로드 밸런서 또는 백엔드 Amazon EC2 인스턴스에 설치해야 합니다. ACM은 Elastic Load Balancing과 통합되어 로드 밸런서에 인증서를 배포합니다. 자세한 내용은 Application Load Balancer 생성을 참조하세요.

Amazon CloudFront

Amazon CloudFront는 엣지 로케이션의 전 세계 네트워크를 통해 콘텐츠를 제공하여 동적 및 정적 웹 콘텐츠를 최종 사용자에게 더 빨리 배포하도록 지원하는 웹 서비스입니다. CloudFront를 통해 서비스하는 콘텐츠를 최종 사용자가 요청하면 지연 시간이 가장 낮은 엣지 로케이션으로 라우팅됩니다. 이를 통해 최고의 성능으로 콘텐츠가 제공됩니다. 콘텐츠가 이미 엣지 로케이션에 있는 경우에는 CloudFront가 바로 전달합니다. 콘텐츠가 현재 엣지 로케이션에 없는 경우 CloudFront는 최종 콘텐츠 원본으로 식별한 Amazon S3 버킷 또는 웹 서버에서 콘텐츠를 검색합니다. CloudFront에 대한 자세한 내용은 Amazon CloudFront 개발자 가이드를 참조하세요.

SSL/TLS를 통해 보안 콘텐츠를 제공하려면 SSL/TLS 인증서를 CloudFront 배포 또는 백엔드 콘텐츠 소스에 설치해야 합니다. ACM은 CloudFront와 통합되어 CloudFront 배포에 ACM 인증서를 배포합니다. 자세한 내용은 SSL/TLS 인증서 받기 단원을 참조하세요.

> **Note**
>
> CloudFront에서 ACM 인증서를 사용하려면 미국 동부(버지니아 북부) 리전에서 인증서를 요청하거나 가져와야 합니다.

Amazon Elastic Kubernetes Service:

Amazon Elastic Kubernetes Service는 자체 Kubernetes 컨트롤 플레인을 설치, 운영 및 유지 관리할 필요 AWS 없이에서 Kubernetes를 쉽게 실행할 수 있는 관리형 Kubernetes 서비스입니다. Amazon EKS에 대한 자세한 내용은 Amazon Elastic Kubernetes Service 사용 설명서를 참조하세요.

Kubernetes용 AWS 컨트롤러(ACK)와 함께 ACM을 사용하여 Kubernetes 워크로드에 TLS 인증서를 발급하고 내보낼 수 있습니다. 이 통합을 통해 Amazon EKS 포드를 보호하고 Kubernetes Ingress 또는 AWS 로드 밸런서에서 TLS를 종료할 수 있습니다. ACM은 인증서를 자동으로 갱신하고 ACK 컨트롤러는 Kubernetes 보안 암호를 갱신된 인증서로 업데이트합니다. 자세한 내용은 ACM 인증서를 사용하여 Kubernetes 워크로드 보호 단원을 참조하십시오.

Amazon Cognito

Amazon Cognito는 웹 및 모바일 애플리케이션에 대한 인증, 권한 부여 및 사용자 관리를 제공합니다. 사용자는 자격 AWS 계정 증명으로 직접 로그인하거나 Facebook, Amazon, Google 또는 Apple과 같은 타사를 통해 로그인할 수 있습니다. Amazon Cognito에 대한 자세한 내용은 Amazon Cognito 개발자 안내서를 참조하세요.

Amazon CloudFront 프록시를 사용하도록 Cognito 사용자 풀을 구성하는 경우 CloudFront는 ACM 인증서를 배치하여 사용자 지정 도메인을 보호할 수 있습니다. 이 경우 인증서를 삭제하려면 먼저 CloudFront와의 인증서 연결을 제거해야 합니다.

AWS Elastic Beanstalk

Elastic Beanstalk를 사용하면 애플리케이션을 실행하는 인프라에 대한 걱정 없이 AWS 클라우드에서 애플리케이션을 배포하고 관리할 수 있습니다.는 관리 복잡성을 AWS Elastic Beanstalk 줄입니다. 애플리케이션을 업로드하기만 하면 Elastic Beanstalk에서 용량 프로비저닝, 로드 밸런싱, 조정, 상태 모니터링에 대한 세부 정보를 자동으로 처리합니다. Elastic Beanstalk는 Elastic Load

Balancing 서비스를 사용하여 로드 밸런서를 생성합니다. Elastic Beanstalk에 대한 자세한 내용은 AWS Elastic Beanstalk 개발자 가이드를 참조하세요.

인증서를 선택하려면 Elastic Beanstalk 콘솔에서 애플리케이션에 대한 로드 밸런서를 구성해야 합니다. 자세한 내용은 HTTPS를 종료하도록 Elastic Beanstalk 환경의 로드 밸런서 구성하기를 참조하세요.

AWS App Runner

App Runner는 소스 코드 또는 컨테이너 이미지에서 AWS 클라우드의 확장 가능하고 안전한 웹 애플리케이션으로 직접 배포할 수 있는 빠르고 간단하며 비용 효율적인 방법을 제공하는 AWS 서비스입니다. 새로운 기술을 배우거나, 사용할 컴퓨팅 서비스를 결정하거나, AWS 리소스를 프로비저닝하고 구성하는 방법을 알 필요가 없습니다. App Runner에 대한 자세한 내용은 AWS App Runner 개발자 가이드를 참조하세요.

사용자 지정 도메인 이름을 App Runner 서비스와 연결하면 App Runner는 내부적으로 도메인 유효성을 추적하는 인증서를 생성합니다. 이 인증서는 ACM에 저장됩니다. App Runner는 도메인이 서비스에서 연결 해제된 후 또는 서비스가 삭제된 후 7일 동안 이러한 인증서를 삭제하지 않습니다. 이 전체 프로세스는 자동화되어 있으므로 인증서를 직접 추가하거나 관리할 필요가 없습니다. 자세한 내용은 AWS App Runner 개발자 가이드에서 App Runner 서비스의 사용자 지정 도메인 이름 관리를 참조하세요.

Amazon API Gateway

모바일 디바이스의 급증과 사물 인터넷(IoT)의 발전으로 API를 만들어 액세스 데이터로 이용하거나 AWS에서 백엔드 시스템과 상호 작용할 수 있도록 하는 것이 점점 일반화되어가고 있습니다. API Gateway를 사용하여 API를 게시, 유지 관리, 모니터링 및 보호할 수 있습니다. API를 API Gateway에 배포한 후, 사용자 지정 도메인 이름을 설정하여 쉽게 API에 액세스할 수 있습니다. 또한 사용자 지정 도메인 이름을 설정하기 위해서 SSL/TLS 인증서를 제출해야 합니다. ACM을 사용하여 인증서를 생성하거나 가져올 수 있습니다. Amazon API Gateway에 대한 자세한 내용은 Amazon API Gateway 개발자 가이드를 참조하세요.

AWS Nitro Enclaves

AWS Nitro Enclaves는 Amazon EC2 인스턴스에서 엔클레이브라는 격리된 실행 환경을 생성할 수 있는 Amazon EC2 기능입니다. Enclaves는 별도의 강화되고 매우 제한적인 가상 시스템입니다. 즉, 상위 인스턴스와의 보안 로컬 소켓 연결만 제공합니다. 영구 스토리지, 대화형 액세스 또는 외부 네트워킹이 없습니다. 사용자는 Enclave로 SSH를 통해 연결할 수 없으며, 상위 인스턴스의 프로세스, 애플리케이션 또는 사용자(루트 또는 관리자 포함)가 Enclave 내의 데이터 및 애플리케이션에 액세스할 수 없습니다.

Nitro Enclaves에 연결된 EC2 인스턴스는 ACM 인증서를 지원합니다. 자세한 내용은 Nitro Enclaves용AWS Certificate Manager을(를) 참조하세요.

> **Note**
>
> ACM 인증서는 Nitro Enclave에 연결되지 않은 EC2 인스턴스와 연결할 수 없습니다.

AWS CloudFormation

CloudFormation 는 Amazon Web Services 리소스를 모델링하고 설정하는 데 도움이 됩니다. Elastic Load Balancing 또는 API Gateway와 같이 사용하려는 AWS 리소스를 설명하는 템플릿을 생성합니다. 그런 다음 CloudFormation 은 해당 리소스를 프로비저닝하고 구성합니다. AWS 리소스를 개별적으로 생성 및 구성하고이 모든 것을 CloudFormation 처리하는 것이 무엇인지 파악할 필요가 없습니다. ACM 인증서는 템플릿 리소스로 포함되어 있습니다. 즉, CloudFormation 는 서비스와 함께 AWS 사용하여 보안 연결을 활성화할 수 있는 ACM 인증서를 요청할 수 있습니다. 또한 ACM 인증서는 설정할 수 있는 많은 AWS 리소스에 포함됩니다 CloudFormation.

CloudFormation에 대한 일반적인 내용은 CloudFormation 사용 설명서를 참조하세요. CloudFormation에서 지원하는 ACM 리소스에 대한 내용은 AWS::CertificateManager::인증서를 참조하세요.

에서 제공하는 강력한 자동화를 사용하면 특히 새 AWS 계정에서 인증서 할당량을 CloudFormation 쉽게 초과할 수 있습니다. ACM 모범 사례를 따르는 것이 좋습니다 CloudFormation.

> **Note**
>
> 를 사용하여 ACM 인증서를 생성하는 경우 CloudFormation 스택 CloudFormation은 CREATE_IN_PROGRESS 상태로 유지됩니다. 이후 스택 작업은 사용자가 인증서 확인 이메일의 지침을 따를 때까지 지연됩니다. 자세한 내용은 Resource Failed to Stabilize During a Create, Update, or Delete Stack Operation 단원을 참조하세요.

AWS Amplify

Amplify는 프런트 엔드 웹 및 모바일 개발자가 풀 스택 애플리케이션을 빠르고 쉽게 구축할 수 있도록 특별히 제작된 도구 및 기능 세트입니다 AWS. Amplify는 Amplify Hosting 및 Amplify Studio라는 두 가지 서비스를 제공합니다. Amplify Hosting은 지속적인 배포로 풀스택 서버리스 웹 앱을 호스팅하기 위한 Git 기반 워크플로를 제공합니다. Amplify Studio는 확장 가능한 풀스택 웹 및 모바

일 앱의 생성을 간소화하는 시각적 개발 환경입니다. Studio를 사용하여 바로 사용할 수 있는 UI 구성 요소 세트로 프런트엔드 UI를 구축하고 앱 백엔드를 생성한 다음, 이 둘을 연결할 수 있습니다. Amplify에 대한 자세한 내용은 AWS Amplify 사용 설명서를 참조하세요.

사용자 지정 도메인을 애플리케이션에 연결하면 Amplify 콘솔에서 ACM 인증서를 발급하여 보안을 유지합니다.

Amazon OpenSearch Service

Amazon OpenSearch Service는 로그 분석, 실시간 애플리케이션 모니터링, 클릭스트림 분석 등의 사용 사례를 위한 검색 및 분석 엔진입니다. 자세한 내용은 Amazon OpenSearch Service 개발자 안내서를 참조하세요.

사용자 지정 도메인 및 엔드포인트를 포함하는 OpenSearch Service 클러스터를 생성할 때 ACM을 사용하여 연결된 Application Load Balancer를 인증서와 함께 프로비저닝할 수 있습니다.

AWS Network Firewall

AWS Network Firewall 는 모든 Amazon Virtual Private Cloud(VPCs. Network Firewall에 관한 자세한 내용은 AWS Network Firewall 개발자 안내서를 참조하세요.

Network Firewall 방화벽은 TLS 검사를 위해 ACM과 통합됩니다. Network Firewall 방화벽에서 TLS 검사를 사용하는 경우 방화벽을 통과하는 SSL/TLS 트래픽의 암호 해독 및 재암호화를 위한 ACM 인증서를 구성해야 합니다. Network Firewall TLS 검사를 위해 ACM과 작동하는 방식에 대한 자세한 내용은 AWS Network Firewall 개발자 안내서의 TLS 검사 구성과 함께 SSL/TLS 인증서를 사용하기 위한 요구 사항을 참조하세요.

의 보안 AWS Certificate Manager

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 는 클라우드에서 AWS AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램 일환으로 보안의 효과를 정기적으로 테스트하고 확인합니다. 에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램 AWS 제공 범위 내 서비스규정 준수 프로그램 제공 범위 내 서비스를 AWS Certificate Manager참조하세요.
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 귀사의 데이터 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 AWS Certificate Manager (ACM)를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 항목에서는 보안 및 규정 준수 목표를 충족하도록 ACM을 구성하는 방법을 보여줍니다. 또한 ACM 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

주제

의 데이터 보호 AWS Certificate Manager

AWS 공동 책임 모델의 데이터 보호에 적용됩니다 AWS Certificate Manager. 이 모델에 설명된 대로 AWS 는 모든를 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 클라우드. 사용자는 이 인프라에 호스팅되는 콘텐츠에 대한 통제 권한을 유지할 책임이 있습니다. 사용하는 AWS 서비스 의 보안 구성과 관리 태스크에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 관한 자세한 내용은 데

이터 프라이버시 FAQ를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하세요.

데이터 보호를 위해 자격 증명을 보호하고 AWS 계정 AWS IAM Identity Center 또는 AWS Identity and Access Management (IAM)를 사용하여 개별 사용자를 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다 AWS CloudTrail. CloudTrail 추적을 사용하여 AWS 활동을 캡처하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 추적 작업을 참조하세요.
- 내부의 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용합니다 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.
- 명령줄 인터페이스 또는 API를 AWS 통해에 액세스할 때 FIPS 140-3 검증 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 연방 정보 처리 표준(FIPS) 140-3을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 형식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 ACM 또는 기타 AWS 서비스 에서 콘솔 AWS CLI, API 또는 AWS SDKs를 사용하여 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명을 URL에 포함해서는 안 됩니다.

인증서 프라이빗 키 보안

퍼블릭 인증서를 요청하면 AWS Certificate Manager (ACM)가 퍼블릭/프라이빗 키 페어를 생성합니다. 가져온 인증서에 대해 키 페어를 생성합니다. 퍼블릭 키는 인증서의 일부가 됩니다. ACM은 인증서와 해당 프라이빗 키를 저장하고 AWS Key Management Service (AWS KMS)를 사용하여 프라이빗 키를 보호합니다. 프로세스는 다음과 같습니다.

1. AWS 리전에서 인증서를 처음 요청하거나 가져올 때 ACM은 별칭 aws/acm을 AWS KMS key 사용하여 관리형를 생성합니다. 이 KMS 키는 각 AWS 계정과 각 AWS 리전에서 고유합니다.
2. ACM은 이 KMS 키를 사용하여 인증서의 프라이빗 키를 암호화합니다. ACM은 암호화된 버전의 프라이빗 키만 저장합니다. ACM은 일반 텍스트 형식으로 프라이빗 키를 저장하지 않습니다. ACM은

동일한 KMS 키를 사용하여 특정 AWS 계정 및 특정 AWS 리전의 모든 인증서에 대한 프라이빗 키를 암호화합니다.

3. 인증서를 AWS Certificate Manager와(과) 통합된 서비스와 연결하면 ACM은 인증서 및 암호화된 프라이빗 키를 해당 서비스로 전송합니다. 또한 서비스가 KMS 키를 사용하여 인증서의 프라이빗 키를 복호화할 수 AWS KMS 있도록 허용하는 권한 부여가에서 생성됩니다. 권한 부여에 대한 자세한 내용은 AWS Key Management Service 개발자 가이드에서 권한 부여 사용을 참조하세요. ACM이 지원하는 서비스에 대한 자세한 내용은 ACM에 통합된 서비스 섹션을 참조하세요.

   > **Note**
   >
   > 자동으로 생성된 AWS KMS 권한 부여를 제어할 수 있습니다. 어떤 이유로든 이 권한 부여를 삭제하면 통합 서비스에 대한 ACM 기능이 손실됩니다.

4. 통합 서비스는 KMS 키를 사용하여 프라이빗 키를 복호화합니다. 그런 다음 서비스는 인증서와 암호 해독된(일반 텍스트) 프라이빗 키를 사용하여 클라이언트에 대한 보안 통신 채널(SSL/TLS 세션)을 설정합니다.
5. 인증서가 통합 서비스에서 연결 해제되면 3단계에서 생성된 권한 부여가 사용 중지됩니다. 다시 말해서 서비스가 더 이상 KMS 키를 사용하여 인증서의 프라이빗 키를 암호 복호화할 수 없습니다.

에 대한 자격 증명 및 액세스 관리 AWS Certificate Manager

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 AWS 서비스 되는 입니다. IAM 관리자는 누가 ACM 리소스를 사용하도록 인증되고(로그인됨) 권한이 부여되는지(권한 있음)를 제어합니다. IAM은 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

주제

- 대상
- ID를 통한 인증
- 정책을 사용하여 액세스 관리
- AWS Certificate Manager 에서 IAM을 사용하는 방법
- 에 대한 자격 증명 기반 정책 예제 AWS Certificate Manager
- ACM API 권한: 작업 및 리소스 참조
- AWS 에 대한 관리형 정책 AWS Certificate Manager
- ACM에서 조건 키 사용

- ACM에서 서비스 연결 역할(SLR) 사용
- AWS Certificate Manager 자격 증명 및 액세스 문제 해결

대상

AWS Identity and Access Management (IAM)를 사용하는 방법은 역할에 따라 다릅니다.

- 서비스 사용자 - 기능에 액세스할 수 없는 경우 관리자에게 권한 요청(참조AWS Certificate Manager 자격 증명 및 액세스 문제 해결)
- 서비스 관리자 - 사용자 액세스 결정 및 권한 요청 제출(AWS Certificate Manager 에서 IAM을 사용하는 방법 참조)
- IAM 관리자 - 액세스를 관리하기 위한 정책 작성(에 대한 자격 증명 기반 정책 예제 AWS Certificate Manager 참조)

ID를 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여에 로그인하는 방법입니다. AWS 계정 루트 사용자, IAM 사용자 또는 IAM 역할을 수임하여 인증되어야 합니다.

AWS IAM Identity Center (IAM Identity Center), Single Sign-On 인증 또는 Google/Facebook 자격 증명과 같은 자격 증명 소스의 자격 증명을 사용하여 페더레이션 자격 증명으로 로그인할 수 있습니다. 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의 AWS 계정에 로그인하는 방법 섹션을 참조하세요.

프로그래밍 방식 액세스를 위해는 요청에 암호화 방식으로 서명할 수 있는 SDK 및 CLI를 AWS 제공합니다. 자세한 내용은 IAM 사용 설명서의 API 요청용AWS Signature Version 4 섹션을 참조하세요.

AWS 계정 루트 사용자

를 생성할 때 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 AWS 계정 theroot 사용자라는 하나의 로그인 자격 증명으로 AWS 계정시작합니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 자격 증명이 필요한 작업은 IAM 사용 설명서의 루트 사용자 자격 증명이 필요한 작업 섹션을 참조하세요.

페더레이션 ID

가장 좋은 방법은 인간 사용자에게 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 서비스 사용하여에 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명은 엔터프라이즈 디렉터리, 웹 자격 증명 공급자 또는 자격 증명 소스의 자격 증명을 AWS 서비스 사용하여 Directory Service 에 액세스하는 사용자입니다. 페더레이션 ID는 임시 자격 증명을 제공하는 역할을 수임합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center를 추천합니다. 자세한 정보는 AWS IAM Identity Center 사용 설명서의 What is IAM Identity Center?를 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가진 ID입니다. 장기 자격 증명이 있는 IAM 사용자 대신 임시 자격 증명을 사용하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 AWS 사용하여에 액세스하도록 인간 사용자에게 요구하기를 참조하세요.

IAM 그룹은 IAM 사용자 모음을 지정하고 대규모 사용자 집합에 대한 관리 권한을 더 쉽게 만듭니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자 사용 사례 섹션을 참조하세요.

IAM 역할

IAM 역할은 임시 자격 증명을 제공하는 특정 권한이 있는 자격 증명입니다. 사용자에서 IAM 역할(콘솔)로 전환하거나 또는 API 작업을 호출하여 역할을 수임할 수 있습니다. AWS CLI AWS 자세한 내용은 IAM 사용 설명서의 역할 수임 방법을 참조하세요.

IAM 역할은 페더레이션 사용자 액세스, 임시 IAM 사용자 권한, 교차 계정 액세스, 교차 서비스 액세스 및 Amazon EC2에서 실행되는 애플리케이션에 유용합니다. 자세한 내용은 IAM 사용 설명서의 교차 계정 리소스 액세스를 참조하세요.

정책을 사용하여 액세스 관리

정책을 AWS 생성하고 자격 증명 또는 리소스에 연결하여 AWS 에서 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결될 때 권한을 정의합니다.는 보안 주체가 요청할 때 이러한 정책을 AWS 평가합니다. 대부분의 정책은에 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서에 대한 자세한 내용은 IAM 사용 설명서의 JSON 정책 개요 섹션을 참조하세요.

정책을 사용하여 관리자는 어떤 보안 주체가 어떤 리소스에 대해 어떤 조건에서 작업을 수행할 수 있는지 정의하여 누가 무엇을 액세스할 수 있는지 지정합니다.

기본적으로 사용자 및 역할에는 어떠한 권한도 없습니다. IAM 관리자는 IAM 정책을 생성하고 사용자가 수임할 수 있는 역할에 추가합니다. IAM 정책은 작업을 수행하기 위해 사용하는 방법과 관계없이 작업에 대한 권한을 정의합니다.

ID 기반 정책

ID 기반 정책은 ID(사용자, 사용자 그룹 또는 역할)에 연결하는 JSON 권한 정책 문서입니다. 이러한 정책은 자격 증명이 수행할 수 있는 작업, 대상 리소스 및 이에 관한 조건을 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

ID 기반 정책은 인라인 정책(단일 ID에 직접 포함) 또는 관리형 정책(여러 ID에 연결된 독립 실행형 정책)일 수 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책 및 인라인 정책 중에서 선택 섹션을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 예를 들어 IAM 역할 신뢰 정책 및 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

기타 정책 유형

AWS 는 보다 일반적인 정책 유형에서 부여한 최대 권한을 설정할 수 있는 추가 정책 유형을 지원합니다.

- 권한 경계 - ID 기반 정책에서 IAM 엔터티에 부여할 수 있는 최대 권한을 설정합니다. 자세한 정보는 IAM 사용 설명서의 IAM 엔터티의 권한 범위를 참조하세요.
- 서비스 제어 정책(SCP) - AWS Organizations내 조직 또는 조직 단위에 대한 최대 권한을 지정합니다. 자세한 내용은AWS Organizations 사용 설명서의 서비스 제어 정책을 참조하세요.
- 리소스 제어 정책(RCP) – 계정의 리소스에 사용할 수 있는 최대 권한을 설정합니다. 자세한 내용은 AWS Organizations 사용 설명서의 리소스 제어 정책(RCP)을 참조하세요.
- 세션 정책 – 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 자세한 내용은 IAM 사용 설명서의 세션 정책을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 에서 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

AWS Certificate Manager 에서 IAM을 사용하는 방법

IAM을 사용하여 ACM에 대한 액세스를 관리하기 전에 ACM과 함께 사용할 수 있는 IAM 기능을 알아보세요.

에서 사용할 수 있는 IAM 기능 AWS Certificate Manager

IAM 특성	ACM 지원
자격 증명 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예
정책 조건 키(서비스별)	예
ACL	아니요
ABAC(정책의 태그)	예
임시 보안 인증	예
엔터티 권한	예
서비스 역할	아니요
서비스 연결 역할	예

ACM 및 기타 AWS 서비스가 대부분의 IAM 기능과 작동하는 방식을 전체적으로 알아보려면 IAM 사용 설명서의 AWS IAM으로 작업하는 서비스를 참조하세요.

ACM에 대한 자격 증명 기반 정책

ID 기반 정책 지원: 예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서에서 고객 관리형 정책으로 사용자 지정 IAM 권한 정의를 참조하세요.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. JSON 정책에서 사용할 수 있는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

ACM에 대한 자격 증명 기반 정책 예제

ACM 자격 증명 기반 정책의 예제를 보려면 에 대한 자격 증명 기반 정책 예제 AWS Certificate Manager을(를) 참조하세요.

ACM 내 리소스 기반 정책

리소스 기반 정책 지원: 아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예제는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는이 포함될 수 있습니다 AWS 서비스.

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM에서 교차 계정 리소스 액세스를 참조하세요.

ACM 정책 작업

정책 작업 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 `Action`요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하세요.

ACM 작업 목록을 보려면 서비스 인증 참조의 AWS Certificate Manager에서 정의한 작업을 참조하세요.

ACM의 정책 작업은 작업 앞에 다음 접두사를 사용합니다.

```
acm
```

단일 문에서 여러 작업을 지정하려면 쉼표로 구분합니다.

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

ACM 자격 증명 기반 정책의 예제를 보려면 에 대한 자격 증명 기반 정책 예제 AWS Certificate Manager을(를) 참조하세요.

ACM 정책 리소스

정책 리소스 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Resource` JSON 정책 요소는 작업이 적용되는 하나 이상의 객체를 지정합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

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

ACM 리소스 유형 및 해당 ARN의 목록을 보려면 서비스 권한 부여 참조의 AWS Certificate Manager에서 정의한 리소스를 참조하세요. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 AWS Certificate Manager가 정의한 작업을 참조하세요.

ACM 자격 증명 기반 정책의 예제를 보려면 에 대한 자격 증명 기반 정책 예제 AWS Certificate Manager을(를) 참조하세요.

ACM 정책 조건 키

서비스별 정책 조건 키 지원: 예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소는 정의된 기준에 따라 문이 실행되는 시기를 지정합니다. 같음(equals) 또는 미만(less than)과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

ACM 조건 키 목록을 보려면 서비스 인증 참조의 AWS Certificate Manager을(를) 위한 조건 키를 참조하세요. 조건 키를 사용할 수 있는 작업과 리소스를 알아보려면 에서 정의한 작업을 AWS Certificate Manager 참조하세요.

ACM 자격 증명 기반 정책의 예제를 보려면 에 대한 자격 증명 기반 정책 예제 AWS Certificate Manager을(를) 참조하세요.

ACM의 ACL

ACL 지원: 아니요

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

ABAC을 통한 ACM

ABAC 지원(정책의 태그): 예

속성 기반 액세스 제어(ABAC)는 태그라고 불리는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. IAM 엔터티 및 AWS 리소스에 태그를 연결한 다음 보안 주체의 태그가 리소스의 태그와 일치할 때 작업을 허용하는 ABAC 정책을 설계할 수 있습니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*또는 `aws:TagKeys`조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 내용은 IAM 사용 설명서의 ABAC 권한 부여를 통한 권한 정의를 참조하세요. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어(ABAC) 사용을 참조하세요.

ACM에서 임시 자격 증명 사용

임시 자격 증명 지원: 예

임시 자격 증명은 AWS 리소스에 대한 단기 액세스를 제공하며 페더레이션 또는 전환 역할을 사용할 때 자동으로 생성됩니다. 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성하는 것이 AWS 좋습니다. 자세한 내용은 IAM 사용 설명서의 IAM의 임시 보안 자격 증명 및 IAM으로 작업하는 AWS 서비스 섹션을 참조하세요.

ACM의 서비스 간 보안 주체 권한

전달 액세스 세션(FAS) 지원: 예

전달 액세스 세션(FAS)은를 호출하는 보안 주체의 권한을 다운스트림 서비스에 대한 요청 AWS 서비스 과 AWS 서비스함께 사용합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.

ACM에 대한 서비스 역할

서비스 역할 지원: 아니요

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하는 것으로 가정하는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 AWS 서비스 AWS에 권한을 위임할 역할 생성을 참조하세요.

> **Warning**
>
> 서비스 역할에 대한 권한을 변경하면 ACM 기능이 중단될 수 있습니다. ACM에서 관련 지침을 제공하는 경우에만 서비스 역할을 편집하세요.

ACM에 대한 서비스 연결 역할

서비스 연결 역할 지원: 예

서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은에 나타나 AWS 계정 며 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

서비스 연결 역할 생성 또는 관리에 대한 자세한 내용은 IAM으로 작업하는AWS 서비스를 참조하세요. 서비스 연결 역할 열에서 Yes가 포함된 서비스를 테이블에서 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예(Yes) 링크를 선택합니다.

에 대한 자격 증명 기반 정책 예제 AWS Certificate Manager

기본적으로 사용자 및 역할에는 ACM 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성(콘솔)을 참조하세요.

각 리소스 유형에 대한 ARN 형식을 포함하여 ACM에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 인증 참조의 AWS Certificate Manager에 대한 작업, 리소스 및 조건 키를 참조하세요.

주제

- 정책 모범 사례
- ACM 콘솔 사용
- 사용자가 자신의 고유한 권한을 볼 수 있도록 허용
- 인증서 나열
- 인증서 요청
- 인증서 검색
- 인증서 가져오기
- 인증서 삭제

정책 모범 사례

ID 기반 정책에 따라 계정에서 사용자가 ACM 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따르세요.

- AWS 관리형 정책을 시작하고 최소 권한으로 전환 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반적인 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 에서 사용할 수 있습니다 AWS 계정. 사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 추가로 줄이는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책 또는 AWS 직무에 대한 관리형 정책을 참조하세요.
- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우, 작업을 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 IAM의 정책 및 권한을 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어, SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. AWS 서비스와 같은 특정를 통해 사용되는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다 CloudFormation. 자세한 내용은 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.
- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 새로운 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer에서 정책 검증을 참조하세요.
- 다중 인증(MFA) 필요 -에서 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 AWS 계정켭니다. API 작업을 직접적으로 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 MFA를 통한 보안 API 액세스를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하세요.

ACM 콘솔 사용

AWS Certificate Manager 콘솔에 액세스하려면 최소 권한 집합이 있어야 합니다. 이러한 권한은에서 ACM 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다 AWS 계정. 최소 필수 권한보다 더 제한적인 ID 기반 정책을 생성하는 경우, 콘솔이 해당 정책에 연결된 엔티티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요는 없습니다. 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자와 역할이 ACM 콘솔을 계속 사용할 수 있도록 하려면 ACM *AWSCertificateManagerReadOnly* AWS 관리형 정책도 엔터티에 연결합니다. 자세한 내용은 IAM 사용 설명서의 사용자에게 권한 추가를 참조하세요.

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로이 작업을 완료할 수 있는 권한이 포함됩니다.

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

인증서 나열

다음 정책은 사용자의 계정에 모든 ACM 인증서를 나열하도록 허용합니다.

JSON

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

> **Note**
>
> ACM 인증서를 Elastic Load Balancing 및 CloudFront 콘솔에 표시하려면 이 권한이 필요합니다.

인증서 요청

다음 정책은 사용자가 ACM 익스포터블 퍼블릭 인증서를 요청하는 것을 거부합니다.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DenyACMCertificateRequest",
      "Effect": "Deny",
      "Action": [
        "acm:RequestCertificate"
      ],
      "Resource": [
```

```
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "acm:Export": "ENABLED"
        }
      }
    }
  ]
}
```

인증서 검색

다음 정책은 사용자가 특정 ACM 인증서를 검색하도록 허용합니다.

JSON

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

인증서 가져오기

다음 정책은 사용자가 인증서를 가져오도록 허용합니다.

JSON

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

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

인증서 삭제

다음 정책은 사용자가 특정 ACM 인증서를 삭제하도록 허용합니다.

JSON

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

ACM API 권한: 작업 및 리소스 참조

액세스 제어를 설정하고 IAM 사용자 또는 역할에 연결할 수 있는 권한 정책을 작성할 때 다음 표를 참조로 사용할 수 있습니다. 표의 첫 번째 열에는 각 AWS Certificate Manager API 작업이 나열됩니다. 정책의 `Action` 요소에 작업을 지정합니다. 다음과 같이 남은 열에서 정보를 추가로 제공합니다.

IAM 정책에서 ACM 정책 요소를 사용하여 조건을 표시할 수 있습니다. 전체 목록은 IAM 사용 설명서에서 사용 가능한 키를 참조하세요.

> **Note**
>
> 작업을 지정하려면 `acm:` 접두사 다음에 API 작업 명칭을 사용합니다(예: `acm:RequestCertificate`).

ACM API 작업 및 권한

ACM API 작업	필요한 권한(API 작업)	리소스
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	*
GetCertificate	acm:GetCertificate	arn:aws:acm: *region*:*account*:certificate/ *certificate_ID*
ImportCertificate	acm:ImportCertificate	arn:aws:acm: *region*:*account*:certificate/* 또는 *
ListCertificates	acm:ListCertificates	*
ListTagsForCertificate	acm:ListTagsForCertificate	arn:aws:acm: *region*:*account*:certificate/ *certificate_ID*

ACM API 작업	필요한 권한(API 작업)	리소스
PutAccountConfiguration	acm:PutAccountConfiguration	*
RemoveTagsFromCertificate	acm:RemoveTagsFromCertificate	arn:aws:acm: *region*:*account*:certificate/ *certificate_ID*
RequestCertificate	acm:RequestCertificate	arn:aws:acm: *region*:*account*:certificate/* 또는 *
ResendValidationEmail	acm:ResendValidationEmail	arn:aws:acm: *region*:*account*:certificate/ *certificate_ID*
UpdateCertificateOptions	acm:UpdateCertificateOptions	arn:aws:acm: *region*:*account*:certificate/ *certificate_ID*

AWS 에 대한 관리형 정책 AWS Certificate Manager

AWS 관리형 정책은에서 생성하고 관리하는 독립 실행형 정책입니다 AWS. AWS 관리형 정책은 사용자, 그룹 및 역할에 권한 할당을 시작할 수 있도록 많은 일반적인 사용 사례에 대한 권한을 제공하도록 설계되었습니다.

AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있으므로 특정 사용 사례에 대해 최소 권한을 부여하지 않을 수 있습니다. 사용 사례에 고유한 고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다.

AWS 관리형 정책에 정의된 권한은 변경할 수 없습니다. 가 관리형 정책에 정의된 권한을 AWS 업데이트하는 AWS 경우 업데이트는 정책이 연결된 모든 보안 주체 자격 증명(사용자, 그룹 및 역할)에 영향을 줍니다. AWS AWS 서비스 는 새가 시작되거나 기존 서비스에 새 API 작업을 사용할 수 있게 될 때 AWS 관리형 정책을 업데이트할 가능성이 높습니다.

자세한 내용은 IAM 사용자 가이드의 AWS 관리형 정책을 참조하세요.

AWSCertificateManagerReadOnly

이 정책은 ACM 인증서에 대한 읽기 전용 액세스 권한을 제공하며, 사용자가 ACM 인증서를 설명, 나열, 검색할 수 있도록 허용합니다.

콘솔에서이 AWS 관리형 정책을 보려면 https://console.aws.amazon.com/iam/home#policies/arn:aws:iam::aws:policy/AWSCertificateManagerReadOnly 이동합니다.

정책 세부 정보의 JSON 목록은 AWSCertificateManagerReadOnly를 참조하세요.

AWSCertificateManagerFullAccess

이 정책은 모든 ACM 작업 및 리소스에 대한 모든 액세스를 허용합니다.

콘솔에서이 AWS 관리형 정책을 보려면 https://console.aws.amazon.com/iam/home#policies/arn:aws:iam::aws:policy/AWSCertificateManagerFullAccess 이동합니다.

정책 세부 정보의 JSON 목록은 AWSCertificateManagerFullAccess를 참조하세요.

AWS 관리형 정책에 대한 ACM 업데이트

이 서비스가 이러한 변경 사항을 추적하기 시작한 이후부터 ACM의 AWS 관리형 정책 업데이트에 대한 세부 정보를 봅니다. 이 페이지의 변경 사항에 대한 자동 알림을 받아보려면 ACM 문서 기록 페이지에서 RSS 피드를 구독하세요.

변경	설명	Date
AWSCertificateManagerReadOnly 정책에 `GetAccountConfiguration` 지원을 추가했습니다.	이 `AWSCertificateManagerReadOnly` 정책이 `GetAccountConfigur`	2021년 3월 3일

변경	설명	Date
	ation API 작업 호출 권한을 포함합니다.	
ACM이 변경 내용 추적 시작	ACM은 AWS 관리형 정책에 대한 변경 사항 추적을 시작합니다.	2021년 3월 3일

ACM에서 조건 키 사용

AWS Certificate Manager 는 AWS Identity and Access Management (IAM) 조건 키를 사용하여 인증서 요청에 대한 액세스를 제한합니다. IAM 정책 또는 서비스 제어 정책(SCP) 의 조건 키를 사용하여 조직의 지침을 준수하는 인증서 요청을 생성할 수 있습니다.

> **Note**
>
> ACM 조건 키를와 같은 AWS 글로벌 조건 키와 결합하여 특정 사용자 또는 역할로 작업을 추가로 제한aws:PrincipalArn합니다.

ACM에 지원되는 조건

ACM API 운영 및 지원되는 조건

조건 키	지원되는 ACM API 작업	Type	설명
acm:ValidationMethod	RequestCertificate	문자열(DNS, EMAIL, HTTP)	ACM 검증 방법을 기준으로 요청 필터링
acm:DomainNames	RequestCertificate	ArrayOfString	ACM 요청에서 도메인 이름을 기준으로 필터링
acm:KeyAlgorithm	RequestCertificate	문자열	ACM 키 알고리즘 및 크기를 기준으로 요청 필터링

조건 키	지원되는 ACM API 작업	Type	설명
acm:CertificateTransparencyLogging	RequestCertificate	문자열(ENABLED, DISABLED)	ACM 인증서 투명성 로깅 기본 설정을 기준으로 요청 필터링
acm:CertificateAuthority	RequestCertificate	ARN	ACM 요청의 인증 기관을 기준으로 요청 필터링

예 1: 검증 방법 제한

다음 정책은 이메일 검증 방법(arn:aws:iam::123456789012:role/AllowedEmailValidation 역할을 사용한 요청 제외)을 사용하여 새 인증서 요청을 거부합니다.

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"]
            }
        }
    }
}
```

예 2: 와일드카드 도메인 방지

다음 정책은 와일드카드 도메인을 사용하는 새 ACM 인증서 요청을 거부합니다.

JSON

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

예 3: 인증서 도메인 제한

다음 정책은 `*.amazonaws.com`(으)로 종료되지 않는 도메인에 대한 새 ACM 인증서 요청을 거부합니다

JSON

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

```
                "acm:DomainNames": ["*.amazonaws.com"]
            }
        }
    }
}
```

정책을 특정 하위 도메인으로 추가로 제한할 수 있습니다. 이 정책은 모든 도메인이 하나 이상의 조건부 도메인 이름과 일치하는 요청만 허용합니다.

JSON

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

예 4: 키 알고리즘 제한

다음 정책은 `StringNotLike` 조건 키를 사용하여 ECDSA 384비트(`EC_secp384r1`) 키 알고리즘으로 요청된 인증서만 허용합니다.

JSON

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

```
        "Statement":{
        "Effect":"Deny",
        "Action":"acm:RequestCertificate",
        "Resource":"*",
        "Condition":{
            "StringNotLike" : {
                "acm:KeyAlgorithm":"EC_secp384r1"
            }
        }
    }
}
```

다음 정책은 StringLike 조건 키 및 * 와일드카드를 사용하여 ACM에서 RSA 키 알고리즘으로 새 인증서를 요청하지 못하도록 매칭합니다.

JSON

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

예 5: 인증 기관 제한

다음 정책은 제공된 사설 인증 기관(PCA) ARN을 사용한 사설 인증서 요청만 허용합니다.

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"
            }
        }
    }
}
```

이 정책은 `acm:CertificateAuthority` 조건을 사용하여 Amazon Trust Service에서 발행한 공개적으로 신뢰할 수 있는 인증서에 대한 요청만 허용합니다. 인증 기관 ARN을 설정하면 `false`이(가) PCA의 사설 인증서 요청을 방지합니다.

JSON

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

ACM에서 서비스 연결 역할(SLR) 사용

AWS Certificate Manager 는 AWS Identity and Access Management (IAM) 서비스 연결 역할을 사용하여에서 공유하는 다른 계정에 대해 프라이빗 CA에서 발급된 프라이빗 인증서의 자동 갱신을 활성화합니다 AWS Resource Access Manager. 서비스 연결 역할(SLR)은 ACM 서비스에 직접 연결된 IAM 역할입니다. SLR은 ACM에 의해 사전 정의되며 서비스에서 다른 AWS 서비스를 자동으로 호출하기 위해 필요한 모든 권한을 포함합니다.

SLR은 무인 인증서 서명에 필요한 권한을 수동으로 추가할 필요가 없으므로 ACM을 더 쉽게 설정할 수 있습니다. ACM이 SLR의 권한을 정의하므로, 달리 정의되지 않은 한 만 해당 역할을 맡을 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며 이 권한 정책은 다른 IAM 엔티티에 연결할 수 없습니다.

SLR을 지원하는 기타 서비스에 대한 자세한 내용을 알아보려면 IAM으로 작업하는AWS 서비스를 참조하고, 서비스 연결 역할(Service-Linked Role) 열에 예(Yes)가 있는 서비스를 찾으세요. 해당 서비스에 대한 SLR 설명서를 보려면 [예(Yes)] 링크를 선택합니다.

ACM에 대한 SLR 권한

ACM은 Amazon Certificate Manager 서비스 역할 정책이라는 SLR을 사용합니다.

AWSServiceRoleForCertificateManager SLR은 역할을 수임하기 위해 다음 서비스를 신뢰합니다.

- `acm.amazonaws.com`

역할 권한 정책은 ACM이 지정된 리소스에서 다음 작업을 완료하도록 허용합니다.

- 작업: `acm-pca:IssueCertificate`, "*"에 대한 `acm-pca:GetCertificate`

IAM 객체(사용자, 그룹, 역할 등)가 SLR을 작성하고 편집하거나 삭제할 수 있도록 권한을 구성할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한 단원을 참조하세요.

> **Important**
>
> 이 경우 ACM은 계정에 SLR이 있는지 여부를 확인할 수 없다는 알림을 표시할 수 있습니다. 필요한 `iam:GetRole` 권한이 이미 계정의 ACM SLR에 부여된 경우, SLR이 생성된 후 알림이 다시 표시되지 않습니다. 알림이 다시 표시될 경우 사용자 또는 계정 관리자가 ACM에 `iam:GetRole` 권한을 부여하거나 계정을 ACM 관리형 정책 `AWSCertificateManagerFullAccess`에 연결해야 할 수 있습니다.

ACM에 대한 SLR 생성

ACM이 사용하는 SLR은 수동으로 생성할 필요가 없습니다. AWS Management Console AWS CLI, 또는 AWS API를 사용하여 ACM 인증서를 발급하면 ACM은에서 공유한 다른 계정의 프라이빗 CA가 처음 인증서에 서명 AWS RAM 할 때 SLR을 생성합니다.

ACM이 계정에 SLR이 존재하는지 여부를 확인할 수 없다는 메시지가 표시되는 경우 계정에에 AWS Private CA 필요한 읽기 권한이 부여되지 않았음을 의미할 수 있습니다. 이 경우 SLR이 설치되지 않으며, 인증서를 계속 발급할 수 있지만 문제를 해결할 때까지 ACM이 인증서를 자동으로 갱신할 수 없습니다. 자세한 내용은 ACM 서비스 연결 역할(SLR) 문제 단원을 참조하십시오.

> **Important**
>
> 이 SLR은 이 역할이 지원하는 기능을 사용하는 다른 서비스에서 작업을 완료했을 경우 계정에 나타날 수 있습니다. 또한 ACM 서비스가 SLR을 지원하기 시작한 2017년 1월 1일 이전에 이 서비스를 사용 중이었다면 ACM이 사용자 계정에 AWSServiceRoleForCertificateManager 역할을 이미 생성했습니다. 자세한 내용은 내 IAM 계정에 표시되는 새 역할을 참조하세요.

이 SLR을 삭제할 경우, 다시 생성해야 하는 경우 다음 방법 중 하나를 사용할 수 있습니다.

- IAM 콘솔에서 역할(Role), 역할 생성(Create role), Certificate Manager를 선택하여 CertificateManagerServiceRolePolicy 사용 사례로 새 역할을 생성합니다.
- IAM API CreateServiceLinkedRole 또는 해당 AWS CLI 명령 create-service-linked-role을 사용하여 `acm.amazonaws.com` 서비스 이름으로 SLR을 생성합니다.

자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 생성 섹션을 참조하세요.

ACM에 대한 SLR 편집

ACM은 AWSServiceRoleForCertificateManager 서비스 연결 역할을 편집하도록 허용하지 않습니다. 다양한 주체가 역할을 참조할 수 있기 때문에 SLR이 생성된 후에는 역할 이름을 편집할 수 없습니다. 하지만 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 편집을 참조하세요.

ACM에 대한 SLR 삭제

일반적으로 AWSServiceRoleForCertificateManager SLR은 수동으로 삭제하지 않아도 됩니다. 그러나 IAM 콘솔, AWS CLI 또는 AWS API를 사용하여 역할을 수동으로 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 삭제 섹션을 참조하십시오.

ACM SLR이 지원되는 리전

ACM은 ACM과를 모두 사용할 AWS Private CA 수 있는 모든 리전에서 SLRs 사용을 지원합니다. 자세한 내용은 AWS 리전 및 엔드포인트 섹션을 참조하세요.

리전 이름	리전 자격 증명	ACM의 지원
미국 동부(버지니아 북부)	us-east-1	예
미국 동부(오하이오)	us-east-2	예
미국 서부(캘리포니아 북부)	us-west-1	예
미국 서부(오리건)	us-west-2	예
아시아 태평양(뭄바이)	ap-south-1	예
아시아 태평양(오사카)	ap-northeast-3	예
아시아 태평양(서울)	ap-northeast-2	예
아시아 태평양(싱가포르)	ap-southeast-1	예
아시아 태평양(시드니)	ap-southeast-2	예
아시아 태평양(도쿄)	ap-northeast-1	예
캐나다(중부)	ca-central-1	예
유럽(프랑크푸르트)	eu-central-1	예
유럽(취리히)	eu-central-2	예
유럽(아일랜드)	eu-west-1	예
유럽(런던)	eu-west-2	예

리전 이름	리전 자격 증명	ACM의 지원
유럽(파리)	eu-west-3	예
남아메리카(상파울루)	sa-east-1	예
AWS GovCloud(미국 서부)	us-gov-west-1	예
AWS GovCloud(미국 동부) 동부	us-gov-east-1	예

AWS Certificate Manager 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 ACM 및 IAM에서 발생할 수 있는 공통적인 문제를 진단하고 수정할 수 있습니다.

주제

- ACM에서 작업을 수행할 권한이 없음
- ACM에서 인증서를 요청할 권한이 없음
- iam:PassRole을 수행하도록 인증되지 않음
- 내 외부의 사람이 내 ACM 리소스 AWS 계정 에 액세스하도록 허용하고 싶습니다.

ACM에서 작업을 수행할 권한이 없음

작업을 수행할 권한이 없다는 오류가 표시되면 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음의 예제 오류는 mateojackson IAM 사용자가 콘솔을 사용하여 가상 *my-example-widget* 리소스에 대한 세부 정보를 보려고 하지만 가상 acm:*GetWidget* 권한이 없을 때 발생합니다.

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

이 경우, acm:*GetWidget* 작업을 사용하여 *my-example-widget* 리소스에 액세스할 수 있도록 mateojackson 사용자 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

ACM에서 인증서를 요청할 권한이 없음

이 오류가 발생하면 ACM 또는 PKI 관리자가 현재 상태에서는 인증서를 요청하지 못하도록 규칙을 설정한 것입니다.

다음 예제 오류는 IAM 사용자가 콘솔을 사용하여 조직 관리자가 DENY(으)로 구성한 옵션을 통해 인증서를 요청하려고 하는 경우에 발생합니다.

```
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
```

이 경우 관리자가 설정한 정책에 따라 다시 요청해야 합니다. 아니면 인증서를 요청할 수 있도록 정책을 업데이트해야 합니다.

iam:PassRole을 수행하도록 인증되지 않음

`iam:PassRole` 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 ACM에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스 에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 역할을 서비스에 전달할 권한이 있어야 합니다.

다음 예제 오류는 `marymajor`(이)라는 IAM 사용자가 콘솔을 사용하여 ACM에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 권한이 없습니다.

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

이 경우, Mary가 `iam:PassRole`작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

내 외부의 사람이 내 ACM 리소스 AWS 계정 에 액세스하도록 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우, 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세한 내용은 다음을 참조하세요.

- ACM에서 이러한 기능을 지원하는지 여부를 알아보려면 AWS Certificate Manager 에서 IAM을 사용하는 방법을(를) 참조하세요.
- 소유 AWS 계정 한의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 소유한 다른의 IAM 사용자에게 액세스 권한 제공을 참조 AWS 계정 하세요.
- 타사에 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사가 AWS 계정 소유한에 대한 액세스 권한 제공을 AWS 계정참조하세요.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(ID 페더레이션)을 참조하세요.
- 크로스 계정 액세스에 대한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 크로스 계정 리소스 액세스를 참조하세요.

의 복원력 AWS Certificate Manager

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다.는 물리적으로 분리되고 격리된 여러 가용 영역을 AWS 리전 제공하며,이 가용 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워킹과 연결됩니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하세요.

의 인프라 보안 AWS Certificate Manager

관리형 서비스인는 AWS 글로벌 네트워크 보안으로 보호 AWS Certificate Manager 됩니다. AWS 보안 서비스 및가 인프라를 AWS 보호하는 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하세요. 인프라 보안 모범 사례를 사용하여 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 인프라 보호를 참조하세요 AWS .

AWS 에서 게시한 API 호출을 사용하여 네트워크를 통해 ACM에 액세스합니다. 클라이언트는 다음을 지원해야 합니다.

- Transport Layer Security(TLS). TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

ACM에 프로그래밍 가능 액세스 권한 부여

사용자는 AWS 외부에서와 상호 작용하려는 경우 프로그래밍 방식으로 액세스해야 합니다 AWS Management Console. 프로그래밍 방식 액세스를 부여하는 방법은에 액세스하는 사용자 유형에 따라 다릅니다 AWS.

사용자에게 프로그래밍 방식 액세스 권한을 부여하려면 다음 옵션 중 하나를 선택합니다.

프로그래밍 방식 액세스가 필요한 사용자	목적	방법
IAM	(권장) 콘솔 자격 증명을 임시 자격 증명으로 사용하여 AWS CLI, AWS SDKs 또는 AWS APIs.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. • 자세한 AWS CLI내용은 AWS Command Line Interface 사용 설명서의 AWS 로컬 개발을 위한 로그인을 참조하세요. • AWS SDKs 경우 SDK 및 도구 참조 안내서의 AWS 로컬 개발을 위한 로그인을 참조하세요. AWS SDKs
작업 인력 ID (IAM Identity Center에서 관리되는 사용자)	임시 자격 증명을 사용하여 AWS CLI, AWS SDKs 또는 AWS APIs.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. • 자세한 AWS CLI내용은 AWS Command Line Interface 사용 설명서의 AWS CLI 를 사용하도록 구성을 AWS IAM Identity Center 참조하세요. • AWS SDKs, 도구 및 AWS APIs의 경우 SDK 및 도구 참조 안내서의 IAM Identity

프로그래밍 방식 액세스가 필요한 사용자	목적	방법
		Center 인증을 참조하세요. AWS SDKs
IAM	임시 자격 증명을 사용하여 AWS CLI, AWS SDKs 또는 AWS APIs.	IAM 사용 설명서의 AWS 리소스에서 임시 자격 증명 사용의 지침을 따릅니다.
IAM	(권장되지 않음) 장기 자격 증명을 사용하여 AWS CLI, AWS SDKs 또는 AWS APIs.	사용하고자 하는 인터페이스에 대한 지침을 따릅니다. • 자세한 AWS CLI내용은 AWS Command Line Interface 사용 설명서의 IAM 사용자 자격 증명을 사용하여 인증을 참조하세요. • AWS SDKs 및 도구의 경우 SDK 및 도구 참조 안내서의 장기 자격 증명을 사용하여 인증을 참조하세요. AWS SDKs • AWS APIs 경우 IAM 사용 설명서의 IAM 사용자의 액세스 키 관리를 참조하세요.

모범 사례

모범 사례는 AWS Certificate Manager (AWS Certificate Manager)를 보다 효과적으로 사용하는 데 도움이 되는 권장 사항입니다. 다음 모범 사례는 현재 ACM 고객의 실제 경험을 기반으로 합니다.

주제

- 계정 수준 분리
- AWS CloudFormation
- 사용자 지정 트러스트 스토어

- 인증서 고정
- 도메인 검증
- 도메인 이름 추가 또는 삭제
- 인증서 투명성 로깅 옵트아웃
- 켜기 AWS CloudTrail

계정 수준 분리

정책에서 계정 수준 분리를 사용하여 계정 수준에서 인증서에 액세스할 수 있는 사용자를 제어합니다. 프로덕션 인증서를 테스트 및 개발 인증서와 별도의 계정에 보관합니다. 계정 수준 분리를 사용할 수 없는 경우 정책의 `kms:CreateGrant` 작업을 거부하여 특정 역할에 대한 액세스를 제한할 수 있습니다. 이렇게 하면 상위 수준에서 인증서에 서명할 수 있는 계정의 역할을 제한합니다. 권한 부여 용어를 비롯한 권한 부여에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 Grants in AWS KMS를 참조하세요.

계정별 `kms:CreateGrant` 사용을 제한하는 것보다 세분화된 제어를 원하는 경우 kms:EncryptionContext 조건 키를 사용하여 특정 인증서로 `kms:CreateGrant`를 제한할 수 있습니다. `arn:aws:acm`을 키로 지정하고 제한할 ARN의 값을 지정합니다. 다음 예제 정책에서는 특정 인증서의 사용을 금지하지만 다른 인증서는 허용합니다.

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

를 사용하여 사용하려는 AWS 리소스를 설명하는 템플릿을 생성할 AWS CloudFormation 수 있습니다. CloudFormation 그런 다음는 이러한 리소스를 프로비저닝하고 구성합니다. CloudFormation 는 Elastic Load Balancing, Amazon CloudFront, Amazon API Gateway와 같이 ACM에서 지원하는 리소스를 프로비저닝할 수 있습니다. 자세한 내용은 ACM에 통합된 서비스 단원을 참조하십시오.

CloudFormation 를 사용하여 여러 테스트 환경을 빠르게 생성하고 삭제하는 경우 각 환경에 대해 별도의 ACM 인증서를 생성하지 않는 것이 좋습니다. 그렇게 하면 인증서 할당량에 빠르게 도달하게 됩니다. 자세한 내용은 할당량 단원을 참조하십시오. 대신 테스트를 위해 사용중인 모든 도메인 이름을 포함하는 와일드카드 인증서를 생성합니다. 예를 들어 `<version>.service.example.com`과 같은 버전 번호에 따라서만 달라지는 도메인 이름에 대해 ACM 인증서를 반복적으로 생성할 경우 `<*>.service.example.com`에 대한 단일 와일드카드 인증서를 대신 생성하세요.

> **Important**
>
> Amazon CloudFront 배포를 사용하는 경우 HTTP 검증은 와일드카드 인증서를 지원하지 않습니다. Amazon CloudFront에서 사용할 CloudFormation 템플릿에 와일드카드 인증서를 포함할 때는 DNS 검증 또는 이메일 검증을 사용해야 합니다. 자동 갱신 기능을 사용하려면 DNS 검증을 사용하는 것이 좋습니다.

가 테스트 환경을 생성하는 데 CloudFormation 사용하는 템플릿에 와일드카드 인증서를 포함합니다.

사용자 지정 트러스트 스토어

ACM 인증서로 보호되는 엔드포인트에 대한 연결을 보장하려면 Amazon 루트를 사용자 지정 트러스트 스토어에 포함하는 것이 좋습니다. Amazon Root 인증 기관은 다양한 키 유형과 알고리즘을 나타낼 수 있습니다. Starfield Services Root Certificate Authority - G2는 업데이트할 수 없는 다른 이전 트러스트 스토어 및 클라이언트와 호환되는 이전 루트입니다. 모든 루트 CA를 포함하면 애플리케이션의 호환성을 극대화할 수 있습니다.

인증서 고정

인증서 고정(SSL 고정이라고도 함)은 애플리케이션에서 해당 호스트를 인증서 계층 대신 X.509 인증서 또는 퍼블릭 키와 직접 연결하여 원격 호스트의 유효성을 검증하는 데 사용할 수 있는 프로세스입니

다. 따라서 애플리케이션은 SSL/TLS 인증서 체인 유효성 검증을 피하기 위해 고정을 사용합니다. 일반적인 SSL 유효성 검증 프로세스는 루트 인증 기관(CA) 인증서에서 하위 CA 인증서(있는 경우)를 통해 인증서 체인 전체의 서명을 확인합니다. 또한 저계층부터 원격 호스트를 위해 인증서를 확인합니다. 대신 애플리케이션은 루트 인증서 또는 체인의 다른 인증서가 아닌 해당 인증서만 유일하게 신뢰할 수 있다는 원격 호스트에 대한 인증서를 고정할 수 있습니다. 과정 중에도 원격 호스트의 인증서 혹은 퍼블릭 키를 사용자 애플리케이션에 추가할 수 있습니다. 그렇지 않으면 호스트에 처음 연결할 때 원격 호스트의 인증서 혹은 퍼블릭 키를 사용자 애플리케이션에 추가할 수 있습니다.

> **Warning**
>
> 애플리케이션에서 ACM 인증서를 고정하지 않는 것이 좋습니다. ACM은 에서 관리형 인증서 갱신 AWS Certificate Manager을 수행하여 Amazon이 발급한 SSL/TLS 인증서를 만료되기 전에 자동으로 갱신합니다. 인증서를 갱신하기 위해 ACM은 새로운 퍼블릭-프라이빗 키 페어를 생성합니다. 애플리케이션에서 ACM 인증서를 고정했는데 그 인증서가 새 퍼블릭 키로 갱신되면 애플리케이션이 도메인에 연결하지 못하게 될 수 있습니다.

인증서를 고정하기로 결정한 경우 다음 옵션을 사용하면 애플리케이션이 도메인에 연결하는 데 지장을 주지 않습니다.

- 자체 인증서를 ACM으로 가져온 다음 애플리케이션을 가져온 인증서에 고정합니다. ACM은 가져온 인증서를 자동으로 갱신하지 않습니다.
- 퍼블릭 인증서를 사용하는 경우 애플리케이션을 Amazon 루트 인증서에 고정합니다. 프라이빗 인증서를 사용하는 경우 애플리케이션을 CA의 루트 인증서에 고정합니다.

도메인 검증

Amazon 인증 기관(CA)이 사이트에 대한 인증서를 발급하려면 먼저 AWS Certificate Manager (ACM)에서 요청에 지정한 모든 도메인을 소유 또는 제어하고 있는지 확인해야 합니다. 이메일 혹은 DNS를 사용하여 확인할 수 있습니다. 자세한 내용은 AWS Certificate Manager DNS 검증 및 AWS Certificate Manager 이메일 검증 섹션을 참조하세요.

도메인 이름 추가 또는 삭제

기존 ACM 인증서에서 도메인 이름을 추가하거나 제거할 수 없습니다. 대신 수정된 도메인 이름 목록을 사용하여 새 인증서를 요청해야 합니다. 예를 들어 인증서에 5개의 도메인 이름이 있고 4개를 더 추가하려면 9개의 도메인 이름을 모두 포함한 새 인증서를 요청해야 합니다. 새로운 인증서에 대해 이전에 검증한 도메인 이름을 비롯하여 요청 중인 모든 도메인 이름의 소유권을 확인해야 합니다.

이메일 검증을 사용하면 각 도메인 이름에 대해 최대 8개의 검증 이메일을 받습니다. 그 중 하나 이상의 이메일에 대해 72시간 이내에 조치해야 합니다. 예를 들어, 5개의 도메인 이름을 포함하는 인증서를 요청할 경우 최대 40개의 검증 메시지을 수신하고 그 중 5개 이상의 이메일에 대해 72시간 이내에 조치해야 합니다. 인증서 요청의 도메인 이름 수가 증가하면 이메일을 사용하여 도메인 이름에 대한 소유권을 확인하는 데 필요한 작업도 증가합니다.

대신 DNS 검증을 사용할 경우, 인증할 FQDN에 대해 1개의 새 DNS 레코드를 데이터베이스에 작성해야 합니다. ACM은 생성할 레코드를 사용자에게 전송하고, 나중에 데이터베이스에 쿼리하여 레코드가 추가되었는지 확인합니다. 기록 추가를 통해 사용자가 도메인을 소유하고 제어함을 증명합니다. 앞선 예를 보면 도메인 이름을 5개 지정하면 반드시 5개의 DNS 기록을 생성해야 합니다. 따라서 가능하다면 DNS 검증을 사용하는 것을 권장합니다.

인증서 투명성 로깅 옵트아웃

> **Important**
>
> 인증서 투명성 로깅에서 옵트아웃하기 위해 수행하는 작업과 상관없이, 인증서를 바인딩하는 퍼블릭 또는 프라이빗 엔드포인트에 액세스할 수 있는 모든 클라이언트나 개인은 여전히 인증서를 로깅할 수 있습니다. 하지만 인증서에는 서명된 인증서 타임스탬프(SCT)가 포함되지 않습니다. 발급 CA만 SCT를 인증서에 포함시킬 수 있습니다.

2018년 4월 30일부터 Google Chrome은 인증서 투명성 로그에 기록되지 않은 공인 SSL/TLS 인증서를 더 이상 신뢰하지 않습니다. 따라서 2018년 4월 24일부터 Amazon CA는 모든 새 인증서 및 갱신을 최소 2개 이상의 퍼블릭 로그에 게시하기 시작했습니다. 인증서가 로깅된 후에는 인증서를 제거할 수 없습니다. 자세한 내용은 인증서 투명성 로깅 단원을 참조하십시오.

로깅은 인증서를 요청하거나 인증서가 갱신될 때 자동으로 수행되지만 옵트아웃하도록 선택할 수 있습니다. 이렇게 하는 일반적인 이유 중에는 보안 및 개인 정보 보호에 대한 우려가 있습니다. 예를 들어, 내부 호스트 도메인 이름을 로깅하면 다른 방식으로는 공개되지 않는 내부 네트워크에 대한 정보가 잠재적 공격자에게 제공될 수 있습니다. 또한 로깅하면 신규 또는 미발표 제품과 웹 사이트의 이름이 유출될 수 있습니다.

인증서를 요청할 때 투명성 로깅을 옵트아웃하려면 request-certificate AWS CLI 명령의 `options` 파라미터 또는 RequestCertificate API 작업을 사용합니다. 인증서가 2018년 4월 24일 이전에 발급된 경우 갱신 중에 인증서가 로깅되지 않도록 하려면 update-certificate-options 명령 또는 UpdateCertificateOptions API를 사용하여 옵트아웃할 수 있습니다.

제한 사항

- 콘솔을 사용하여 투명성 로깅을 활성화 또는 비활성화할 수 없습니다.
- 인증서가 갱신 기간에 들어간 후에는 로깅 상태를 변경할 수 없으며, 일반적으로 퍼블릭 인증서의 인증서 만료 45일 전에는 변경할 수 없습니다. 상태 변경이 실패하면 오류 메시지가 생성되지 않습니다.

인증서가 로깅된 후에는 로그에서 인증서를 제거할 수 없습니다. 이 시점에는 옵트아웃해도 효과가 없습니다. 인증서를 요청할 때 로깅에서 옵트아웃한 다음 나중에 다시 옵트인하도록 선택하면 인증서를 갱신할 때가지 인증서가 로깅되지 않습니다. 인증서를 즉시 로깅하려면 새 인증서를 발급하는 것이 좋습니다.

다음 예제에서는 request-certificate 명령을 사용하여 새 인증서를 요청할 때 인증서 투명성을 비활성화하는 방법을 보여 줍니다.

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

위의 명령은 새 인증서의 ARN을 출력합니다.

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

인증서가 이미 있고 인증서를 갱신할 때 인증서가 로깅되지 않도록 하려는 경우 update-certificate-options 명령을 사용합니다. 이 명령은 값을 반환하지 않습니다.

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

켜기 AWS CloudTrail

ACM 사용을 시작하기 전에 CloudTrail 로깅을 활성화합니다. CloudTrail을 사용하면 AWS 관리 콘솔, AWS SDKs, AWS Command Line Interface및 상위 수준 Amazon Web Services를 통해 수행된 AWS

API 호출을 포함하여 계정에 대한 API 호출 기록을 검색하여 AWS 배포를 모니터링할 수 있습니다. 또한 어떤 사용자 및 계정이 ACM API를 호출했는지, 어떤 소스 IP 주소에 호출이 수행되었는지, 언제 호출이 발생했는지도 확인할 수 있습니다. API를 사용해 CloudTrail을 애플리케이션에 통합시킴으로써 조직에 대한 추적 생성을 자동화하고 추적 상태를 확인하며 관리자가 CloudTrail 로깅을 활성화하고 비활성화하는 방식을 제어할 수 있습니다. 자세한 내용은 추적 생성을 참조하세요. ACM 작업에 대한 예제 추적을 보려면 에서 CloudTrail 사용 AWS Certificate Manager 섹션으로 이동하세요.

모니터링 및 로그 AWS Certificate Manager

모니터링은 AWS Certificate Manager 및 AWS 솔루션의 안정성, 가용성 및 성능을 유지하는 데 중요한 부분입니다. 다중 지점 장애가 발생할 경우 보다 쉽게 디버깅할 수 있도록 AWS 솔루션의 모든 부분으로부터 모니터링 데이터를 수집해야 합니다.

다음 주제에서는 ACM과 함께 사용할 수 있는 AWS 클라우드 모니터링 도구에 대해 설명합니다.

주제

Amazon EventBridge 사용

Amazon EventBridge(이전 CloudWatch Events)를 사용하여 AWS 서비스를 자동화하고 애플리케이션 가용성 문제 또는 리소스 변경과 같은 시스템 이벤트에 자동으로 대응할 수 있습니다. ACM을 포함한 AWS 서비스의 이벤트는 거의 실시간으로 Amazon EventBridge로 전달됩니다. 이벤트를 사용하여 AWS Lambda 함수, AWS Batch 작업, Amazon SNS 주제 등 대상을 트리거할 수 있습니다. 자세한 내용은 Amazon EventBridge란 무엇입니까?를 참조하세요.

주제

ACM에 대한 Amazon EventBridge 지원

이 주제에서는 Amazon EventBridge에서 지원하는 ACM 관련 이벤트를 나열하고 설명합니다.

ACM 인증서 만료 임박 이벤트

ACM은 프라이빗/가져오는 인증서의 경우 만료 45일 전, 퍼블릭 인증서의 경우 만료 30일 전에 모든 활성 인증서(퍼블릭, 프라이빗 및 가져온 인증서)에 대한 일일 만료 이벤트를 전송합니다. 이 전송 일정은 ACM API의 PutAccountConfiguration 작업을 사용해 변경할 수 있습니다.

ACM에서 발급한 자격이 있는 인증서의 갱신을 자동으로 시작하지만 가져온 인증서는 중단을 방지하기 위해 만료 전에 재발급한 후 다시 가져와야 합니다. 자세한 내용은 인증서 다시 가져오기를 참조하

세요. 만료 이벤트를 사용하여 인증서를 ACM으로 다시 가져오도록 자동화를 설정할 수 있습니다. 를 사용한 자동화의 예는 단원을 AWS Lambda참조하십시오ACM에서 Amazon EventBridge를 사용하여 작업 시작.

ACM 인증서 만료 임박 이벤트는 다음과 같은 구조를 갖습니다.

```
{
  "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 인증서 만료 이벤트

> **Note**
>
> 가져온 인증서에 대해서는 인증서 만료 이벤트를 사용할 수 없습니다.

고객은 이 이벤트에 대해 수신 대기하여 계정에서 ACM에서 발급된 퍼블릭 또는 프라이빗 인증서가 만료되면 알림을 받을 수 있습니다.

ACM 인증서 만료 이벤트는 다음과 같은 구조를 갖습니다.

```
{
    "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 인증서 사용 가능 이벤트

고객은 이 이벤트에 대해 수신 대기하여 관리형 퍼블릭 또는 프라이빗 인증서를 사용할 준비가 되면 알림을 받을 수 있습니다. 이벤트는 발급, 갱신 및 가져오기를 수행하는 시점에 게시됩니다. 프라이빗 인증서의 경우 일단 사용 가능한 상태가 되면 이 인증서를 호스트에 배포하기 위해서는 여전히 고객의 조치가 필요합니다.

ACM 인증서 사용 가능 이벤트는 다음과 같은 구조를 갖습니다.

```
{
    "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 인증서 갱신 작업 필요 이벤트

> **Note**
>
> 가져온 인증서에 대해서는 인증서 갱신 작업 필수 이벤트를 사용할 수 없습니다.

고객은 이 이벤트에 대해 수신 대기하여 인증서를 갱신할 수 있는 상태가 되기 전에 고객이 조치를 취해야 할 때 알림을 받을 수 있습니다. 예를 들어 고객이 ACM이 인증서를 갱신하지 못하도록 하는 CAA 레코드를 추가하는 경우 ACM은 프라이빗 인증서의 경우 만료 45일 전, 퍼블릭 인증서의 경우 만료 30일 전에 자동 갱신이 실패할 때이 이벤트를 게시합니다. 고객 조치를 취하지 않으면 ACM은 30일(프라이빗만 해당), 15일, 3일 및 1일 또는 고객 조치를 취하거나 인증서가 만료되거나 인증서가 더 이상 갱신할 수 없을 때까지 추가 갱신을 시도합니다. 이러한 각 갱신 시도에 대해 이벤트가 게시됩니다.

ACM 인증서 갱신 작업 필요 이벤트의 구조는 다음과 같습니다.

```
{
   "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 Certificate Revoked 이벤트

고객은 이 이벤트에 대해 수신 대기하여 계정에서 ACM에서 발급된 퍼블릭 또는 프라이빗 인증서가 취소되면 알림을 받을 수 있습니다.

> **Note**
>
> 내보낸 인증서만 취소할 수 있습니다. 가져온 인증서는 revoke-certificate를 통해 취소할 수 없습니다.

ACM Certificate Revoked 이벤트는 다음과 같은 구조를 갖습니다.

```
{
  "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 상태 이벤트

AWS 상태 이벤트는 갱신할 수 있는 ACM 인증서에 대해 생성됩니다. 갱신 자격에 대한 자세한 내용은 에서 관리형 인증서 갱신 AWS Certificate Manager 섹션을 참조하세요.

상태 이벤트는 다음 두 가지 시나리오에서 생성됩니다.

- 퍼블릭 또는 프라이빗 인증서가 성공적으로 갱신된 경우.
- 갱신이 발생하려면 고객이 조치를 취해야 하는 경우. 이러한 조치는 이메일 메시지에 있는 링크를 클릭(이메일 검증 인증서의 경우)하거나 오류를 해결하는 것을 의미할 수 있습니다. 각 이벤트에는 다음 중 하나의 이벤트 코드가 포함됩니다. 코드는 필터링에 사용할 수 있는 변수로 노출됩니다.
  - AWS_ACM_RENEWAL_STATE_CHANGE(인증서가 갱신되었거나, 만료되었거나, 곧 만료되는 경우)
  - CAA_CHECK_FAILURE(CAA 검사 실패)
  - AWS_ACM_RENEWAL_FAILURE(프라이빗 CA에서 서명한 인증서의 경우)

상태 이벤트의 구조는 다음과 같습니다. 이 예에서는 AWS_ACM_RENEWAL_STATE_CHANGE 이벤트가 생성되었습니다.

```
{
   "source":[
      "aws.health"
   ],
   "detail-type":[
      "AWS Health Event"
   ],
   "detail":{
      "service":[
         "ACM"
      ],
      "eventTypeCategory":[
         "scheduledChange"
      ],
      "eventTypeCode":[
         "AWS_ACM_RENEWAL_STATE_CHANGE"
      ]
   }
}
```

ACM에서 Amazon EventBridge를 사용하여 작업 시작

이러한 이벤트를 기반으로 Amazon EventBridge 규칙을 생성하고 Amazon EventBridge 콘솔을 사용하여 이벤트가 감지될 때 수행되는 작업을 구성할 수 있습니다. 이 섹션에서는 Amazon EventBridge 규칙 및 결과 작업을 구성하는 샘플 절차를 제공합니다.

주제

Amazon SNS를 사용하여 이벤트에 응답

이 섹션에서는 ACM이 상태 이벤트를 생성할 때마다 텍스트 알림을 전송하도록 Amazon SNS를 구성하는 방법을 보여 줍니다.

응답을 구성하려면 다음 절차를 완료하세요.

Amazon EventBridge 규칙을 생성하고 작업을 트리거하려면

1. Amazon EventBridge 규칙을 생성합니다. 자세한 내용은 이벤트에 응답하는 Amazon EventBridge 규칙 생성을 참조하세요.

   a. Amazon EventBridge 콘솔(https://console.aws.amazon.com/events/)에서 Events(이벤트) > Rules(규칙) 페이지로 이동하여 Create rule(규칙 생성)을 선택합니다.
   b. [규칙 생성(Create rule)] 페이지에서 [이벤트 패턴(Event Pattern)]을 선택합니다.
   c. [서비스 이름(Service Name)]의 메뉴에서 [상태(Health)]를 선택합니다.
   d. [이벤트 유형(Event Type)에서 [특정 상태 이벤트(Specific Health events)]를 선택합니다.
   e. [특정 서비스(Specific service(s))]를 선택하고 메뉴에서 [ACM]을 선택합니다.
   f. [특정 이벤트 유형 범주(Specific event type category(s))]를 선택하고 [accountNotification]을 선택합니다.
   g. [이벤트 유형 코드(Any event type code)]를 선택합니다.
   h. [모든 리소스(Any resource)]를 선택합니다.
   i. [이벤트 패턴 미리보기(Event pattern preview)] 편집기에서 이벤트에서 생성 된 JSON 패턴을 붙여 넣습니다. 이 예에서는 AWS 상태 이벤트 섹션의 패턴을 사용합니다.

```
{
```

```
   "source":[
      "aws.health"
   ],
   "detail-type":[
      "AWS Health Event"
   ],
   "detail":{
      "service":[
         "ACM"
      ],
      "eventTypeCategory":[
         "scheduledChange"
      ],
      "eventTypeCode":[
         "AWS_ACM_RENEWAL_STATE_CHANGE"
      ]
   }
}
```

2. 작업을 구성합니다.

   [대상(Targets)] 섹션에서 Amazon Simple Notification Service(SNS)와 같이 이벤트를 즉시 사용할 수 있는 다양한 서비스 중에서 선택하거나, Lambda 함수를 사용하여 이벤트를 사용자 지정된 실행 코드로 전달할 수 있습니다. AWS Lambda 의 구현 예는 Lambda 함수를 사용하여 이벤트에 응답 섹션을 참조하세요.

Lambda 함수를 사용하여 이벤트에 응답

이 절차에서는를 사용하여 Amazon EventBridge에서 수신 AWS Lambda 대기하고, Amazon Simple Notification Service(SNS)를 사용하여 알림을 생성하고 AWS Security Hub CSPM, 결과를 게시하여 관리자와 보안 팀에 가시성을 제공하는 방법을 보여줍니다.

Lambda 함수 및 IAM 역할을 설정하려면

1. 먼저 AWS Identity and Access Management (IAM) 역할을 구성하고 Lambda 함수에 필요한 권한을 정의합니다. 이 보안 모범 사례를 통해 유연하게 함수를 호출할 권한이 있는 사용자를 지정하고 해당 사용자에게 부여된 권한을 제한할 수 있습니다. 대부분의 AWS 작업은 사용자 계정에서 직접 실행하지 않는 것이 좋습니다. 특히 관리자 계정에서는 실행하지 않는 것이 좋습니다.

   IAM 콘솔(https://console.aws.amazon.com/iam/)을 엽니다.

2. JSON 정책 편집기를 사용하여 아래 템플릿에 정의된 정책을 생성합니다. 자체 리전 및 AWS 계정 세부 정보를 제공합니다. 자세한 내용은 JSON 탭에서 정책 생성을 참조하세요.

   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 역할을 생성하여 여기에 새 정책을 연결합니다. IAM 역할 생성 및 정책 연결에 대한 자세한 내용은 AWS 서비스에 대한 역할 생성(콘솔)을 참조하세요.
4. https://console.aws.amazon.com/lambda/ AWS Lambda 콘솔을 엽니다.
5. Lambda 함수를 생성합니다. 자세한 내용은 콘솔로 Lambda 함수 생성을 참조하세요. 다음 단계를 완료합니다.

   a. [함수 생성(Create function)] 페이지에서 [새로 작성(Author from scratch)] 옵션을 선택하여 함수를 생성합니다.

   b. ‘handle-expiring-certificates’와 같은 이름을 [함수 이름(Function name)] 필드에 지정합니다.

   c. [런타임(Runtime)] 목록에서 Python 3.8을 선택합니다.

   d. [기본 실행 역할 변경(Change default execution role)]을 선택하고 [기존 역할 사용(Use an existing role)]을 선택합니다.

   e. [기존 역할(Existing role)] 목록에서 앞서 생성한 역할을 선택합니다.

   f. 함수 생성을 선택합니다.

   g. [함수 코드(Function code)] 아래에 다음 코드를 삽입합니다.

```
# 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. [환경 변수(Environment variables)]에서 [편집(Edit)]을 선택하고 선택적으로 다음 변수를 추가합니다.

- (선택 사항) EXPIRY_DAYS

  인증서 만료 알림을 보내기 전의 리드 타임(일)을 지정합니다. 이 함수의 기본값은 45일이지만 사용자 지정 값을 지정할 수 있습니다.

- (선택 사항) SNS_TOPIC_ARN

  Amazon SNS의 ARN을 지정합니다. arn:aws:sns:*<region>*:*<account-number>*:*<topic-name>* 형식으로 전체 ARN을 제공합니다.

- (선택 사항) SECURITY_HUB_REGION

  다른 리전 AWS Security Hub CSPM 에서를 지정합니다. 이 값을 지정하지 않으면 실행 중인 Lambda 함수의 리전이 사용됩니다. 함수가 여러 리전에서 실행되는 경우 모든 인증서 메시지가 단일 리전의 Security Hub CSPM으로 전달되도록 하는 것이 좋습니다.

i. [기본 설정(Basic settings)]에서 [제한 시간(Timeout)]을 30초로 설정합니다.

j. 페이지 상단에서 [배포(Deploy)]를 선택합니다.

다음 절차의 작업을 완료하여 이 솔루션의 사용을 시작합니다.

만료 이메일 알림을 자동화하려면

이 예에서는 Amazon EventBridge를 통해 이벤트가 발생하는 순간, 만료되는 각 인증서마다 하나의 이메일을 제공합니다. 기본적으로 ACM은 만료일이 45일 이하로 남은 인증서에 대해 매일 이벤트를 발생시킵니다. (이 기간은 ACM API의 PutAccountConfiguration 작업을 사용하여 사용자 지정할 수 있습니다.) 이러한 각 이벤트는 다음과 같은 일련의 자동화된 작업을 트리거합니다.

```
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 함수를 생성하고 권한을 구성합니다. (이미 완료됨 - Lambda 함수 및 IAM 역할을 설정하려면 참조).
2. 알림을 보내는 데 사용할 Lambda 함수의 표준 SNS 주제입니다. 자세한 내용은 Amazon SNS 주제 생성을 참조하세요.
3. 관심있는 당사자가 새로운 SNS 주제를 구독하도록 설정합니다. 자세한 내용은 Amazon SNS 주제에 구독 설정을 참조하세요.
4. Lambda 함수를 트리거하는 Amazon EventBridge 규칙을 생성합니다. 자세한 내용은 이벤트에 응답하는 Amazon EventBridge 규칙 생성을 참조하세요.

   Amazon EventBridge 콘솔(https://console.aws.amazon.com/events/)에서 Events(이벤트) > Rules(규칙) 페이지로 이동하여 Create rule(규칙 생성)을 선택합니다. [서비스 이름(Service Name)], [이벤트 유형(Event Type)] 및 [Lambda 함수(Lambda function)]를 지정합니다. [이벤트 패턴 미리 보기(Event Pattern preview)] 편집기에서 다음 코드를 붙여 넣습니다.

   ```
   {
     "source": [
       "aws.acm"
     ],
     "detail-type": [
       "ACM Certificate Approaching Expiration"
     ]
   ```

```
}
```

Lambda 수신과 같은 이벤트가 [샘플 이벤트 표시(Show sample event(s))] 아래에 표시됩니다.

```
{
  "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"
  }
}
```

정리하려면

구성 예 또는 다른 어떤 구성이 더 이상 필요하지 않은 경우, 보안 문제 및 예기치 않은 향후 요금 발생을 방지하기 위해 모든 트레이스를 제거하는 것이 좋습니다.

- IAM 정책 및 역할
- Lambda 함수
- CloudWatch Events 규칙
- Lambda와 연결된 CloudWatch Logs
- SNS 주제

에서 CloudTrail 사용 AWS Certificate Manager

AWS Certificate Manager 는 ACM에서 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스와 통합됩니다. CloudTrail은 AWS 계정에서 기본적으로 활성화되어 있습니다. CloudTrail은 ACM 콘솔의 호출과 API 작업에 대한 코드 호출을 비롯하여 ACM에 대한

API 호출을 이벤트로 캡처합니다. 트레일을 구성하면 ACM에 대한 이벤트를 포함한 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. 추적을 구성하지 않은 경우에도 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수 있습니다.

CloudTrail에서 수집한 정보를 사용하여 ACM에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다. 자세한 설명은 CloudTrail 이벤트 기록으로 이벤트 보기를 참조하세요. 지원되는 이벤트 활동이 ACM에서 발생하면 해당 활동은 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다.

또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다.

CloudTrail에 대한 자세한 내용은 다음 설명서를 참조하세요.

- AWS CloudTrail 사용 설명서.
- 트레일 생성 개요
- CloudTrail 지원 서비스 및 통합
- CloudTrail에서 Amazon SNS 알림 구성
- 여러 리전으로부터 CloudTrail 로그 파일 받기 및 여러 계정으로부터 CloudTrail 로그 파일 받기

주제

- CloudTrail 로깅에서 지원되는 ACM API 작업
- 통합 서비스에 대한 API 호출 로깅

CloudTrail 로깅에서 지원되는 ACM API 작업

ACM은 CloudTrail 로그 파일에 다음 작업을 이벤트로 로깅합니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 관한 정보가 포함됩니다. ID 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 AWS 계정 루트 사용자 또는 AWS Identity and Access Management (IAM) 사용자 자격 증명으로 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자의 임시 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 요청이 다른 AWS 서비스에서 이루어졌는지 여부

자세한 설명은 CloudTrail userIdentity 요소를 참조하세요.

다음 섹션에는 지원되는 API 작업에 대한 예제 로그가 나와 있습니다.

- 인증서에 태그 추가(AddTagsToCertificate)
- 인증서 삭제(DeleteCertificate)
- 인증서 설명(DescribeCertificate)
- 인증서 내보내기(ExportCertificate)
- 인증서 가져오기(ImportCertificate)
- 인증서 나열(ListCertificates)
- 인증서의 태그 나열(ListTagsForCertificate)
- 인증서에서 태그 제거(RemoveTagsFromCertificate)
- 인증서 요청(RequestCertificate)
- 검증 이메일 재전송(ResendValidationEmail)
- 인증서 검색(GetCertificate)

인증서에 태그 추가(AddTagsToCertificate)

다음 CloudTrail 예에서는 AddTagsToCertificate API에 대한 호출의 결과를 보여 줍니다.


```
{

   "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"
      }
   ]
}
```

인증서 삭제(DeleteCertificate)

다음 CloudTrail 예에서는 DeleteCertificate API에 대한 호출의 결과를 보여 줍니다

```
{

   "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"
      }
   ]
}
```

인증서 설명(DescribeCertificate)

다음 CloudTrail 예에서는 DescribeCertificate API에 대한 호출의 결과를 보여 줍니다.

> **Note**
>
> 지정한 ACM 인증서에 대한 정보가 `DescribeCertificate` 작업에 대한 CloudTrail 로그에 표시되지 않습니다. 콘솔 AWS Command Line Interface, 또는 DescribeCertificate API를 사용하여 인증서에 대한 정보를 볼 수 있습니다.

```
{
   "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"
      }
   ]
}
```

인증서 내보내기(ExportCertificate)

다음 CloudTrail 예에서는 ExportCertificate API에 대한 호출의 결과를 보여 줍니다.

```
{
   "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"
}
```

인증서 가져오기(ImportCertificate)

다음 예에서는 ACM ImportCertificate API 작업에 대한 호출을 기록하는 CloudTrail 로그 항목을 보여 줍니다.

```
{
   "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"
}
```


인증서 나열(ListCertificates)

다음 CloudTrail 예에서는 ListCertificates API에 대한 호출의 결과를 보여 줍니다.

> **Note**
>
> ACM 인증서가 `ListCertificates` 작업에 대한 CloudTrail 로그에 표시되지 않습니다. 콘솔 AWS Command Line Interface, 또는 ListCertificates API를 사용하여 인증서 목록을 볼 수 있습니다.


```
{
   "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"
      }
   ]
}
```

인증서의 태그 나열(ListTagsForCertificate)

다음 CloudTrail 예에서는 ListTagsForCertificate API에 대한 호출의 결과를 보여 줍니다.

> **Note**
>
> `ListTagsForCertificate` 작업에 대한 CloudTrail 로그에 태그가 표시되지 않습니다. 콘솔, AWS Command Line Interface또는 ListTagsForCertificate API를 사용하여 태그 목록을 볼 수 있습니다.


```
{
   "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"
      }
   ]
}
```

인증서에서 태그 제거(RemoveTagsFromCertificate)

다음 CloudTrail 예에서는 RemoveTagsFromCertificate API에 대한 호출의 결과를 보여 줍니다.


```
{
   "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"
      }
   ]
}
```

인증서 요청(RequestCertificate)

다음 CloudTrail 예에서는 RequestCertificate API에 대한 호출의 결과를 보여 줍니다.


```
{
   "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"
```

```
      }
   ]
}
```

인증서 취소(RevokeCertificate)

다음 CloudTrail 예제에서는 RevokeCertificate API를 호출한 결과를 보여 줍니다.

```
{
    "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"
}
```

검증 이메일 재전송(ResendValidationEmail)

다음 CloudTrail 예에서는 ResendValidationEmail API에 대한 호출의 결과를 보여 줍니다.

```
{
   "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"
      }
   ]
}
```

인증서 검색(GetCertificate)

다음 CloudTrail 예에서는 GetCertificate API에 대한 호출의 결과를 보여 줍니다.

```
{

   "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"
      }
   ]
}
```

통합 서비스에 대한 API 호출 로깅

CloudTrail을 사용하여 ACM과 통합된 서비스에서 생성하는 API 호출을 감사할 수 있습니다. CloudTrail 사용에 대한 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하세요. 다음 예에서는 ACM 인증서를 프로비저닝하는 AWS 리소스에 따라 생성 가능한 로그 유형을 보여줍니다.

주제

- 로드 밸런서 생성

로드 밸런서 생성

CloudTrail을 사용하여 ACM과 통합된 서비스에서 생성하는 API 호출을 감사할 수 있습니다. CloudTrail 사용에 대한 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하세요. 다음 예제에서는 ACM 인증서를 프로비저닝하는 AWS 리소스에 따라 생성할 수 있는 로그 유형을 보여줍니다.

주제

- 로드 밸런서 만들기
- 로드 밸런서를 사용하여 Amazon EC2 인스턴스 등록
- 프라이빗 키 암호화
- 프라이빗 키 해독

로드 밸런서 만들기

다음 예에서는 Alice라는 IAM 사용자의 CreateLoadBalancer 함수 호출을 보여 줍니다. 로드 밸런서의 이름이 TestLinuxDefault이고 ACM 인증서를 사용하여 리스너를 생성합니다.


```
{

   "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"
}
```

로드 밸런서를 사용하여 Amazon EC2 인스턴스 등록

Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 웹 사이트 또는 애플리케이션을 프로비저닝하는 경우 로드 밸런서에서 해당 인스턴스를 인식해야 합니다. 이는 Elastic Load Balancing 콘솔 또는 AWS Command Line Interface를 통해 수행할 수 있습니다. 다음 예제는 AWS 계정 123456789012의 LinuxTest라는 로드 밸런서에 `RegisterInstancesWithLoadBalancer` 대한 호출을 보여줍니다.

```
{
   "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"
}
```

프라이빗 키 암호화

다음 예에서는 ACM 인증서와 연결된 프라이빗 키를 암호화하는 Encrypt 호출을 보여 줍니다. 암호화는 AWS내에서 수행됩니다.

```
{
   "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"
      }
   ]
}
```

프라이빗 키 해독

다음 예에서는 ACM 인증서와 연결된 프라이빗 키를 복호화하는 Decrypt 호출을 보여 줍니다. 복호화는 내에서 수행되며 AWS복호화된 키는 절대 나가지 않습니다 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"
}
```


지원되는 CloudWatch 지표

Amazon CloudWatch는 AWS 리소스에 대한 모니터링 서비스입니다. CloudWatch를 사용하여 지표를 수집 및 추적하고, 경보를 설정하고, AWS 리소스의 변경 사항에 자동으로 대응할 수 있습니다. ACM은 만료 시점까지 계정의 모든 인증서에 대해 매일 두 번씩 지표를 게시합니다.

`AWS/CertificateManager` 네임스페이스에는 다음 지표가 포함되어 있습니다.

지표	설명	단위	측정 기준
`DaysToExpiry`	인증서 만료까지 남은 일수입니다. 인증서가	Integer	CertificateArn

지표	설명	단위	측정 기준
	만료되고 나면 ACM이 이 지표의 게시를 중지합니다.		• 값: 인증서의 ARN

CloudWatch 지표에 대한 자세한 내용은 다음 주제를 참조하세요.

- Amazon CloudWatch 지표 사용
- Amazon CloudWatch 경보 생성

SDK for Java와 AWS Certificate Manager 함께 사용

AWS Certificate Manager API를 사용하여 HTTP 요청을 전송하여 프로그래밍 방식으로 서비스와 상호 작용할 수 있습니다. 자세한 내용은 AWS Certificate Manager API 참조를 참조하세요.

웹 API(또는 HTTP API) 외에도 AWS SDKs 및 명령줄 도구를 사용하여 ACM 및 기타 서비스와 상호 작용할 수 있습니다. 자세한 내용은 Amazon Web Services용 도구를 참조하세요.

다음 주제에서는 AWS SDKs 중 하나인를 사용하여 AWS Certificate Manager API에서 사용 가능한 일부 작업을 AWS SDK for Java수행하는 방법을 보여줍니다.

주제

인증서에 태그 추가

다음 예제에서는 AddTagsToCertificate 함수의 사용법을 보여줍니다.

```
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);
   }
}
```

인증서 삭제

다음 예제에서는 DeleteCertificate 함수를 사용하는 방법을 보여 줍니다. 성공할 경우 함수는 빈 세트인 {}를 반환합니다.

```
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);

   }
}
```

인증서 설명

다음 예제에서는 DescribeCertificate 함수를 사용하는 방법을 보여줍니다.

```
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);

   }
}
```

이 명령이 제대로 실행되면 위 예제에는 다음과 비슷한 정보가 출력에 표시됩니다.

```
{
    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,
    }
}
```

인증서 내보내기

다음 예제에서는 ExportCertificate 함수의 사용법을 보여줍니다. 이 함수는 PKCS #8 형식에 있는 Private Certificate Authority(CA)에서 발급한 사설 인증서를 내보냅니다. (공인 인증서는 ACM에서 발급되었거나 가져왔는지 여부에 관계없이 내보낼 수 없습니다.) 또한 인증서 체인 및 프라이빗 키를 내보냅니다. 이 예제에서는 키에 대한 암호가 로컬 파일에 저장됩니다.

```
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();
   }
}
```

인증서 및 인증서 체인 검색

다음 예제에서는 GetCertificate 함수를 사용하는 방법을 보여 줍니다.

```
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);
   }
}
```

위 예제에서는 다음과 비슷한 내용을 출력합니다.

```
{Certificate: -----BEGIN CERTIFICATE-----
    base64-encoded certificate
-----END CERTIFICATE-----,
CertificateChain: -----BEGIN CERTIFICATE-----
    base64-encoded certificate chain
-----END CERTIFICATE-----
}
```

인증서 가져오기

다음 예제에서는 ImportCertificate 함수의 사용법을 보여줍니다.

```
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);
    }
}
```

인증서 나열

다음 예제에서는 ListCertificates 함수를 사용하는 방법을 보여 줍니다.

```
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);
   }
```

```
}
```

위 예제에서는 다음과 비슷한 내용을 출력합니다.

```
{
    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
    }]
}
```

인증서 갱신

다음 예제에서는 RenewCertificate 함수를 사용하는 방법을 보여 줍니다. 이 함수는 Private Certificate Authority(CA)에서 발급하고 ExportCertificate 함수로 내보낸 사설 인증서를 갱신합니다. 이때 내보낸 사설 인증서만을 이 함수로 갱신할 수 있습니다. ACM으로 AWS Private CA 인증서를 갱신하려면 먼저 ACM 서비스 보안 주체에게 권한을 부여해야 합니다. 자세한 내용은 ACM에 인증서 갱신 권한 할당을 참조하세요.

```
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);
   }
}
```

인증서 태그 나열

다음 예제에서는 ListTagsForCertificate 함수의 사용법을 보여줍니다.

```
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);

   }
}
```

위 예제에서는 다음과 비슷한 내용을 출력합니다.

```
{Tags: [{Key: Purpose,Value: Test}, {Key: Short_Name,Value: My_Cert}]}
```

인증서에서 태그 제거

다음 예제에서는 RemoveTagsFromCertificate 함수의 사용법을 보여줍니다.

```
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);
   }
}
```

인증서 요청

다음 예제에서는 RequestCertificate 함수를 사용하는 방법을 보여 줍니다.

```
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);

    }

}
```

위 예제에서는 다음과 비슷한 내용을 출력합니다.

```
{CertificateArn:
 arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012}
```

검증 이메일 재전송

다음 예에서는 ResendValidationEmail 함수를 사용하는 방법을 보여줍니다.

```
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());

    }
}
```

위 예제에서는 검증 이메일을 다시 전송하고 빈 세트를 표시합니다.

의 문제 해결 AWS Certificate Manager

AWS Certificate Manager을 사용하는 동안 문제가 발생할 경우 다음 주제를 참조하세요.

> **Note**
>
> 이 섹션에서 처리된 문제가 보이지 않을 경우, AWS 지식 센터 방문을 권장합니다.

주제

인증서 요청 문제 해결

ACM 인증서 요청 시 문제가 발생하면 다음 주제를 참조하세요.

주제

인증서 요청 시간 초과

ACM 인증서에 대한 요청은 72시간 이내에 확인되지 않는 경우 시간 초과됩니다. 이 상태를 해결하려면 콘솔을 열고 인증서에 대한 레코드를 찾은 다음 해당 확인란을 클릭하고 [작업(Actions)]을 선택한 후 [삭제(Delete)]를 선택합니다. 그런 다음 [작업(Actions)], [인증서 요청(Request a certificate)]을 선택하여 다시 시작합니다. 자세한 내용은 AWS Certificate Manager DNS 검증 또는 AWS Certificate Manager 이메일 검증을 참조하세요. 따라서 가능하다면 DNS 검증을 사용하는 것을 권장합니다.

인증서 요청 실패

ACM에 대한 요청이 실패하고 다음 오류 메시지 중 하나가 수신되면 제안된 단계를 따라 문제를 해결하세요. 실패한 인증서 요청은 다시 제출할 수 없습니다. 문제를 해결한 후 새 요청을 제출하세요.

주제

오류 메시지: 가용 연락처 없음

인증서를 요청할 때 이메일 검증을 선택하면 ACM은 요청에서 하나 이상의 도메인 이름을 검증하는 데 사용할 이메일 주소를 찾을 수 없습니다. 이 문제를 수정하려면 다음 중 하나를 수행할 수 있습니다.

- 도메인이 이메일을 수신하도록 구성되어 있는지 확인합니다. ACM의 이메일 서버가 도메인 검증 이메일을 보낼 위치를 알 수 있으려면 도메인 이름 서버에 MX(Mail Exchanger) 레코드가 있어야 합니다.

위에서 설명한 작업 중 하나만 완료하면 이 문제를 수정할 수 있습니다. 두 작업을 모두 수행할 필요는 없습니다. 문제를 수정한 후 새 인증서를 요청합니다.

ACM에서 도메인 검증 이메일을 수신하는 방법에 대한 자세한 내용은 AWS Certificate Manager 이메일 검증 또는 검증 이메일이 수신되지 않음 섹션을 참조하세요. 다음 단계를 따라 계속해서 No Available Contacts 메시지를 받으면 문제에 대해 조사할 수 있도록 이 문제가 AWS에 보고됩니다.

오류 메시지: 추가 검증 필요

ACM에서 이 인증서 요청을 처리하기 위해 추가 정보를 요구합니다. 이는 예를 들어 도메인이 Alexa Top 1000 웹 사이트 순위에 든 경우 사기 방지 조치로 이루어질 수 있습니다. 필요한 정보를 제공하려면 지원 센터를 사용하여 지원에 문의하세요. 지원 계획이 없는 경우, ACM 토론 포럼에서 새 스레드를 게시할 수 있습니다.

> **Note**
>
> amazonaws.com, cloudfront.net 또는 elasticbeanstalk.com으로 끝나는 이름과 같이 Amazon 소유 도메인 이름에 대한 인증서를 요청할 수 없습니다.

오류 메시지: 잘못된 퍼블릭 도메인

인증서 요청에서 하나 이상의 도메인 이름이 올바르지 않습니다. 일반적으로 이 문제는 요청의 도메인 이름이 올바른 최상위 도메인이 아니기 때문에 발생합니다. 실패한 요청에 있는 철자 오류 또는 오타를 수정하고 요청에 있는 모든 도메인 이름이 올바른 최상위 도메인 이름인지 확인한 후 인증서를 다시 요청해 보세요. 예를 들어 `example.invalidpublicdomain`의 경우 "invalidpublicdomain"이 올바른 최상위 도메인이 아니기 때문에 이 이름에 대한 ACM 인증서를 요청할 수 없습니다. 이 실패 이유가 계속 표시되는 경우 지원 센터에 문의하세요. 지원 계획이 없는 경우, ACM 토론 포럼에서 새 스레드를 게시할 수 있습니다.

오류 메시지: 기타

일반적으로 인증서 요청에서 하나 이상의 도메인 이름에 오타가 있는 경우에 이 실패가 발생합니다. 실패한 요청에 있는 철자 오류 또는 오타를 수정한 후 인증서를 다시 요청해 보세요. 이 실패 메시지가 계속 수신되는 경우 지원 센터를 통해 지원에 문의하세요. 지원 계획이 없는 경우, ACM 토론 포럼에서 새 스레드를 게시할 수 있습니다.

인증서 검증 문제 해결

ACM 인증서 요청 상태가 검증 보류 중(Pending validation)인 경우 사용자의 조치를 기다리는 중입니다. 요청 후에 이메일 검증을 선택했다면 사용자 혹은 담당자가 인증 이메일 메시지에 응답해야 합니다. 요청된 도메인에 대해 공통된 이메일 주소로 이러한 메시지를 전송합니다. 자세한 내용은 AWS Certificate Manager 이메일 검증 단원을 참조하십시오. DNS 검증을 선택했다면 사용자는 ACM에 의해 사용자 DNS 데이터베이스에 생성된 CNAME 기록을 작성해야 합니다. 자세한 내용은 AWS Certificate Manager DNS 검증 단원을 참조하십시오.

> **Important**
>
> 사용자는 인증서 요청에 따라 추가된 각각의 도메인에 대해 사용자가 소유하거나 관리 권한을 보유하고 있는지 검증해야 합니다. 이메일 검증을 선택했다면 각각 도메인 하나당 인증 이메일 메시지를 수신합니다. 계정이 없을 경우 검증 이메일이 수신되지 않음 단원을 참조하세요. DNS 검증을 선택했다면 각각 도메인 하나당 하나의 CNAME을 생성해야 합니다.

> **Note**
>
> 퍼블릭 ACM 인증서는 Nitro Enclave에 연결된 Amazon EC2 인스턴스에 설치할 수 있습니다. 모든 Amazon EC2 인스턴스에서 사용할 퍼블릭 인증서를 내보낼 수도 있습니다. Nitro Enclave에 연결되지 않은 Amazon EC2 인스턴스에서 독립형 웹 서버를 설정하는 방법에 대한 자세한 내용은 자습서: Amazon Linux 2에 LAMP 웹 서버 설치 또는 자습서: Amazon Linux AMI를 사용하여 LAMP 웹 서버 설치를 참조하세요.

따라서 가능하다면 이메일 검증보다 DNS 검증을 사용하는 것을 권장합니다.

DNS 검증 문제가 있는 경우 다음 주제를 참조하세요.

주제

- DNS 검증 문제 해결
- 이메일 검증 문제 해결
- HTTP 검증 문제 해결

DNS 검증 문제 해결

DNS로 인증서를 검증하는 경우 다음 지침을 참조하세요.

DNS 문제 해결의 첫 번째 단계는 다음과 같은 도구를 사용하여 도메인의 현재 상태를 확인하는 것입니다.

- dig — Linux, Windows
- nslookup — Linux, Windows

주제

- DNS 공급자가 금지하는 밑줄
- DNS 공급자가 추가한 기본 후행 기간
- GoDaddy의 DNS 검증 실패
- ACM 콘솔에서 'Route 53에 레코드 생성' 버튼이 나타나지 않음
- 프라이빗(신뢰할 수 없는) 도메인에서 Route 53 검증 실패
- DNS 검증이 성공했지만 발급 또는 갱신이 실패한 경우

- VPN에서 DNS 서버 검증 실패

DNS 공급자가 금지하는 밑줄

DNS 공급자가 CNAME 값의 앞에 붙는 밑줄을 금지하는 경우, ACM에서 제공하는 값에서 밑줄을 삭제한 상태에서 도메인을 검증할 수 있습니다. 예를 들어, 검증을 위해 CNAME 값 `_x2.acm-validations.aws`를 `x2.acm-validations.aws`로 변경할 수 있습니다. 단, CNAME 이름 파라미터는 항상 밑줄로 시작해야 합니다.

도메인 검증 시 아래의 표에서 오른쪽에 있는 값 중 하나를 사용할 수 있습니다.

이름	Type	값
`_<random value>.example.com.`	CNAME	`_<random value>.acm-validations.aws.`
`_<random value>.example.com.`	CNAME	`<random value>.acm-validations.aws.`

DNS 공급자가 추가한 기본 후행 기간

일부 DNS 공급자는 사용자가 제공하는 CNAME 값에 후행 기간을 기본적으로 추가합니다. 따라서 기간을 직접 추가하면 오류가 발생합니다. 예를 들어 "`<random_value>.acm-validations.aws.`"는 거부되지만 "`<random_value>.acm-validations.aws`"는 허용됩니다.

GoDaddy의 DNS 검증 실패

ACM에서 제공하는 CNAME 값을 수정하지 않는 한 Godaddy 및 기타 레지스트리에 등록된 도메인에 대한 DNS 검증이 실패할 수 있습니다. example.com을 도메인 이름으로 간주하면 발급된 CNAME 레코드의 형식이 다음과 같습니다.

```
NAME: _ho9hv39800vb3examplew3vnewoib3u.example.com. VALUE:
 _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws.
```

다음과 같이 NAME 필드의 끝에서 apex 도메인(마침표 포함)을 잘라내서 GoDaddy와 호환되는 CNAME 레코드를 만들 수 있습니다.

```
NAME: _ho9hv39800vb3examplew3vnewoib3u VALUE:
 _cjhwou20vhu2exampleuw20vuyb2ovb9.j9s73ucn9vy.acm-validations.aws.
```

ACM 콘솔에서 'Route 53에 레코드 생성' 버튼이 나타나지 않음

Amazon Route 53를 DNS 공급자로 선택하면가 해당 공급자와 직접 상호 작용하여 도메인 소유권을 검증할 AWS Certificate Manager 수 있습니다. 경우에 따라 콘솔의 Route 53에 레코드 생성 버튼을 정상적으로 사용하지 못할 수도 있습니다. 이러한 일이 발생할 경우, 다음 발생 가능 원인을 확인합니다.

- Route 53을 DNS 공급자로 사용하지 않고 있습니다.
- 다른 계정들을 통해 ACM 및 Route 53으로 로그인했습니다.
- Route 53에서 호스팅한 영역에 레코드를 생성하기 위한 IAM 권한이 부족합니다.
- 사용자 또는 누군가가 이미 도메인을 검증했습니다.
- 도메인은 공개적으로 주소를 지정할 수 없습니다.

프라이빗(신뢰할 수 없는) 도메인에서 Route 53 검증 실패

DNS 검증 중에 ACM은 공개적으로 호스팅되는 영역에서 CNAME를 검색합니다. 검색되지 않으면 72시간 후에 시간이 초과되어 검증 시간 초과 상태가 됩니다. Amazon VPC 프라이빗 호스팅 영역, 프라이빗 PKI에서 신뢰할 수 없는 도메인, 자체 서명된 인증서 내의 리소스를 포함하는 프라이빗 도메인에 대한 DNS 레코드를 호스팅하는 데 이를 사용할 수 없습니다.

AWS 는 AWS Private CA 서비스를 통해 공개적으로 신뢰할 수 없는 도메인에 대한 지원을 제공합니다.

DNS 검증이 성공했지만 발급 또는 갱신이 실패한 경우

DNS가 올바른데도 "검증 보류 중"과 함께 인증서 발급이 실패하는 경우 인증 기관 권한 부여(CAA) 레코드에 의해 발급이 차단되고 있지 않은지 확인하세요. 자세한 내용은 (선택 사항) CAA 레코드 구성 단원을 참조하십시오.

VPN에서 DNS 서버 검증 실패

VPN에서 DNS 서버를 찾았는데 ACM이 해당 DNS 서버에 대해 인증서를 검증하지 못할 경우, 서버에 공개적으로 액세스할 수 있는지 확인합니다. ACM DNS 검증을 사용하여 퍼블릭 인증서를 발급하려면 퍼블릭 인터넷을 통해 도메인 레코드를 확인할 수 있어야 합니다.

이메일 검증 문제 해결

이메일로 인증서 도메인을 검증하는 경우 다음 지침을 참조하세요.

주제

검증 이메일이 수신되지 않음

ACM으로부터 인증서를 요청하고 이메일 검증을 선택하면 5개의 공통 관리 주소로 도메인 검증 이메일이 전송됩니다. 자세한 내용은 AWS Certificate Manager 이메일 검증 단원을 참조하십시오. 검증 이메일을 수신하는 동안 문제가 발생할 경우 다음과 같은 제안 사항을 검토하세요.

이메일을 찾을 위치

ACM은 요청된 도메인 이름으로 검증 이메일 메시지를 전송합니다. 대신 해당 도메인에서 이러한 이메일을 수신하려는 경우 슈퍼 도메인을 검증 도메인으로 지정할 수도 있습니다. 최소 웹 사이트 주소까지의 모든 하위 도메인이 유효하며 이메일 주소의 도메인으로 @ 뒤에 접미사로 사용됩니다. 예를 들어, example.com을 subdomain.example.com의 검증 도메인으로 지정하는 경우 admin@example.com으로 이메일을 받을 수 있습니다. ACM 콘솔에 표시되는(또는 CLI나 API에서 반환되는) 이메일 주소의 목록을 검토하여 검증 이메일을 찾을 위치를 결정합니다. 목록을 보려면 [Validation not complete]라는 상자에서 도메인 이름 옆의 아이콘을 클릭합니다.

이메일이 스팸으로 표시됨

스팸 폴더에 검증 이메일이 있는지 확인합니다.

GMail에서 이메일을 자동으로 분류함

GMail을 사용하는 경우 검증 이메일이 [Updates] 또는 [Promotions] 탭으로 자동으로 분류되었을 수 있습니다.

지원 센터에 문의

앞에서 설명한 지침을 검토한 후에도 도메인 검증 이메일을 받지 못하는 경우 지원 센터에 방문하여 사례를 생성하세요. 지원 계약이 없는 경우 ACM 토론 포럼에 메시지를 게시합니다.

이메일 검증을 위한 영구적인 초기 타임스탬프

인증서의 첫 번째 이메일 검증 요청의 타임스탬프는 이후 검증 갱신 요청을 거치더라도 유지됩니다. 이는 ACM 작업에서 오류가 발생한 증거가 아닙니다.

DNS 검증으로 전환할 수 없음

이메일 검증을 통해 인증서를 생성한 후에는 해당 인증서를 DNS를 통한 인증으로 전환할 수 없습니다. DNS 검증을 사용하려면 인증서를 삭제한 다음, DNS 검증을 사용하는 새 인증서를 생성합니다.

HTTP 검증 문제 해결

HTTP로 인증서를 검증하는 데 문제가 있는 경우 다음 지침을 참조하세요.

HTTP 문제 해결의 첫 번째 단계는 다음과 같은 도구를 사용하여 도메인의 현재 상태를 확인하는 것입니다.

- curl — Linux 및 Windows
- wget — Linux 및 Windows

주제

- RedirectFrom 및 RedirectTo 위치 간의 콘텐츠 불일치
- 잘못된 CloudFront 구성
- HTTP 리디렉션 문제
- 검증 제한 시간

RedirectFrom 및 RedirectTo 위치 간의 콘텐츠 불일치

`RedirectFrom` 위치의 콘텐츠가 `RedirectTo` 위치의 콘텐츠와 일치하지 않으면 검증이 실패합니다. 인증서의 각 도메인에 대해 콘텐츠가 동일한지 확인하세요.

잘못된 CloudFront 구성

CloudFront 배포가 검증 콘텐츠를 제공하도록 올바르게 구성되어 있는지 확인합니다. 오리진 및 동작 설정이 올바르고 배포가 완료되었는지 확인합니다.

HTTP 리디렉션 문제

콘텐츠를 직접 제공하는 대신 리디렉션을 사용하는 경우 다음 단계에 따라 구성을 확인합니다.

리디렉션 구성을 확인하려면

1. `RedirectFrom` URL을 복사하여 브라우저의 주소 표시줄에 붙여 넣습니다.
2. 새 브라우저 탭에서 `RedirectTo` URL을 붙여 넣습니다.
3. 두 URL의 콘텐츠를 비교하여 정확히 일치하는지 확인합니다.
4. 리디렉션이 302 상태 코드를 반환하는지 확인합니다.

검증 제한 시간

예상 기간 내에 콘텐츠를 사용할 수 없는 경우 HTTP 검증 시간이 초과될 수 있습니다. 검증 문제를 해결하려면 다음 단계를 수행하세요.

검증 제한 시간 초과 문제를 해결하려면

1. 다음 중 하나를 수행하여 어느 도메인이 검증 보류 중인지 확인합니다.
   a. ACM 콘솔을 열고 인증서 세부 정보 페이지를 봅니다. 검증 보류 중으로 표시된 도메인을 찾습니다.
   b. `DescribeCertificate` API 작업을 호출하여 각 도메인의 검증 상태를 확인합니다.
2. 보류 중인 각 도메인에 대해 인터넷에서 검증 콘텐츠에 액세스할 수 있는지 확인합니다.

관리형 인증서 갱신 문제 해결

ACM은 ACM 인증서를 만료되기 이전에 자동으로 갱신하므로 별도의 조치가 필요하지 않습니다. 에서 관리형 인증서 갱신 AWS Certificate Manager에 문제가 있는 경우 다음 주제를 참조하세요.

자동 도메인 검증 준비

ACM 전에 사용자가 인증서를 자동으로 갱신하려면 다음 조건이 충족되어야 합니다.

- 인증서는 ACM과 통합된 AWS 서비스와 연결되어야 합니다. ACM이 지원하는 리소스에 대한 자세한 내용은 ACM에 통합된 서비스 섹션을 참조하세요.
- 이메일 검증 인증서의 경우 ACM에서 인증서에 나열된 각 도메인의 관리자 이메일 주소로 연락할 수 있어야 합니다. 연락을 시도하는 이메일 주소는 AWS Certificate Manager 이메일 검증에 나열되어 있습니다.
- DNS 검증 인증서의 경우 AWS Certificate Manager DNS 검증에 설명된 대로 DNS 구성에 올바른 CNAME 레코드가 포함되어 있는지 확인해야 합니다.

- HTTP 검증 인증서의 경우 AWS Certificate Manager HTTP 검증에 설명된 대로 리디렉션이 구성되어 있는지 확인합니다.

관리형 인증서 갱신 실패 처리

인증서 만료가 가까워지면(DNS의 경우 45일, 이메일의 경우 45일, 프라이빗의 경우 60일) ACM은 자격 기준을 충족하는 경우 인증서 갱신을 시도합니다. 성공적으로 갱신하려면 추가 조치를 취해야 할 수도 있습니다. 자세한 내용은 에서 관리형 인증서 갱신 AWS Certificate Manager 단원을 참조하십시오.

이메일 검증 인증서에 대한 관리형 인증서 갱신

ACM 인증서는 198일 동안 유효합니다. 인증서를 갱신하려면 도메인 소유자의 작업이 필요합니다. ACM은 만료 45일 전에 도메인과 연결된 이메일 주소로 갱신 알림을 전송하기 시작합니다. 이 알림에는 도메인 소유자가 갱신을 위해 클릭할 수 있는 링크가 포함되어 있습니다. 나열된 모든 도메인이 검증되면 ACM은 동일한 ARN 사용하여 갱신된 인증서를 발급합니다.

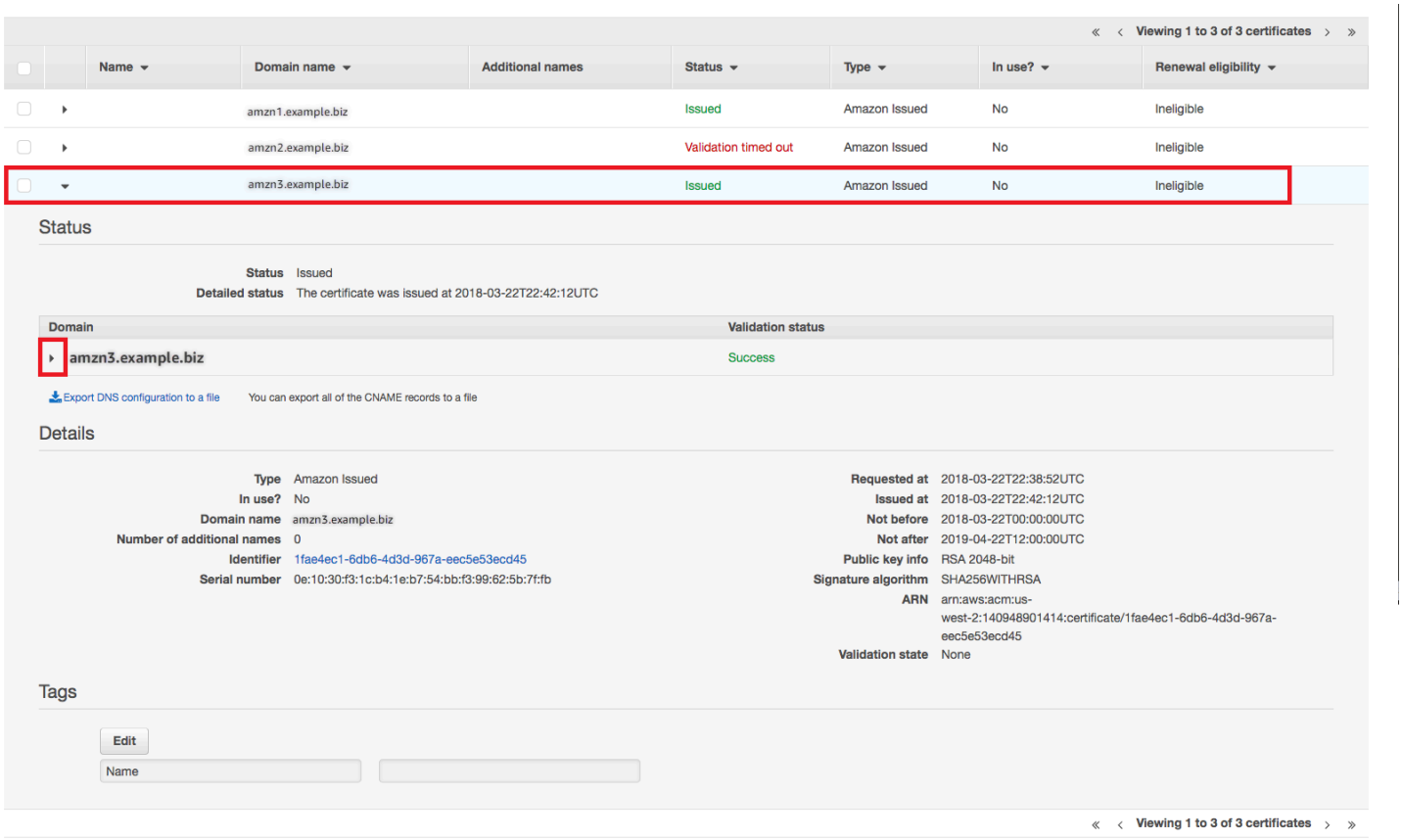
`PENDING_VALIDATION` 상태의 도메인을 식별하고 해당 도메인에 대한 검증 프로세스를 반복하는 방법에 대한 지침은 이메일을 통한 확인을 참조하세요.

DNS 검증 인증서에 대한 관리형 인증서 갱신

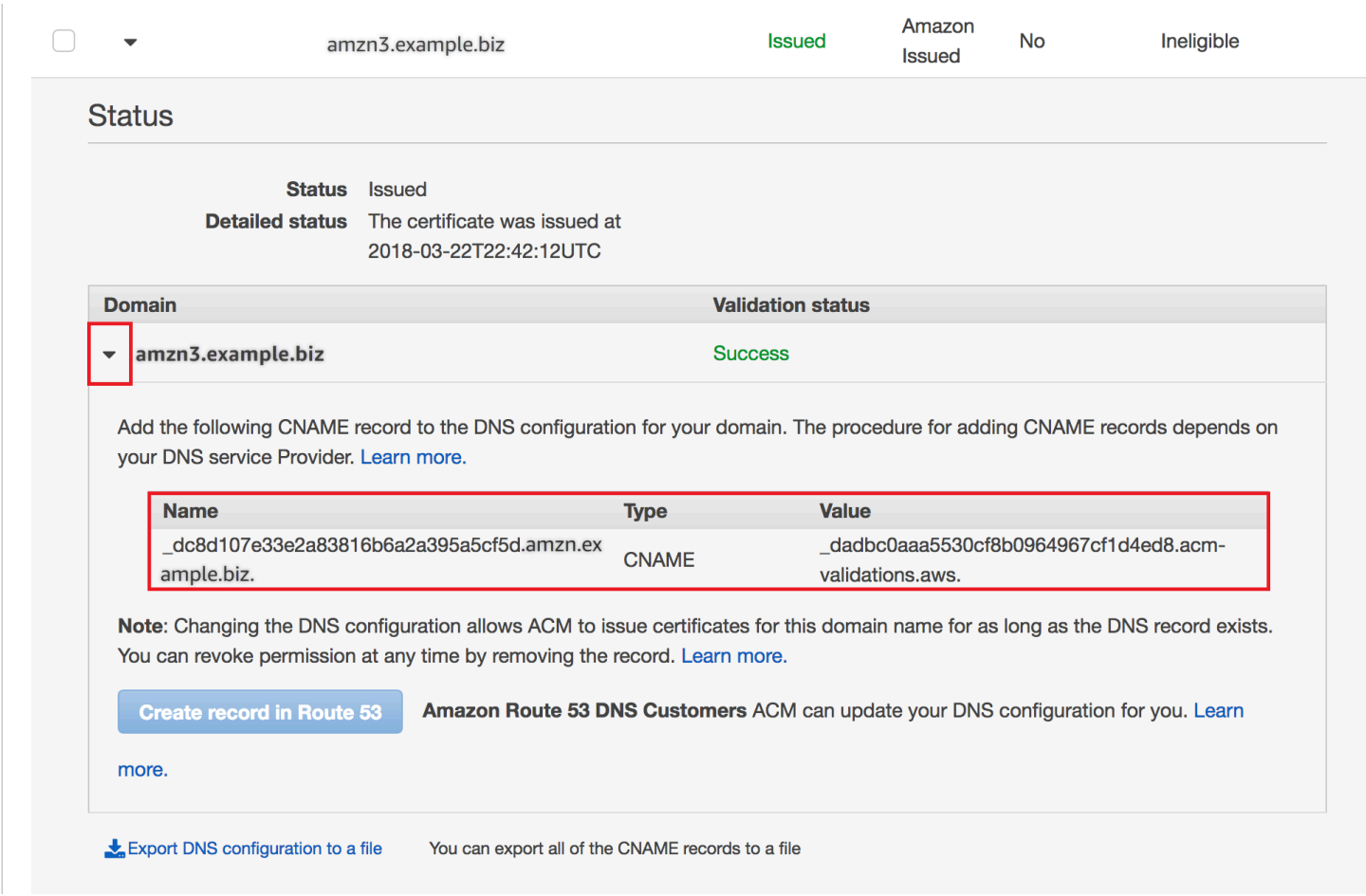
ACM은 DNS 검증 인증서에 대한 TLS 검증을 시도하지 않습니다. ACM이 사용자가 DNS 검증으로 검증한 인증서를 갱신하지 못한다면, 가장 가능성이 높은 원인은 DNS 구성에 CNAME 레코드가 없거나 정확하지 않기 때문입니다. 이런 상황이라면, ACM은 사용자에게 인증서를 자동으로 갱신할 수 없다고 알립니다.

> **Important**
>
> 올바른 CNAME 레코드를 DNS 데이터베이스에 삽입해야 합니다. 이를 수행하는 방법은 도메인 등록 기관에 문의하세요.

ACM 콘솔에서 인증서와 도메인 항목을 확장하면 도메인에 대한 CNAME 레코드를 찾을 수 있습니다. 자세한 내용은 아래 그림을 참조하세요. ACM API의 DescribeCertificate 작업 또는 ACM CLI의 describe-certificate 명령을 사용하여 CNAME 레코드를 검색할 수도 있습니다. 자세한 내용은 AWS Certificate Manager DNS 검증 단원을 참조하십시오.



콘솔에서 대상 인증서를 선택합니다.



인증 창을 확장해 인증서의 CNAME 정보를 찾습니다.

문제가 지속된다면 지원 센터에 문의하세요.

HTTP 검증 인증서에 대한 관리형 인증서 갱신

ACM은 HTTP 검증 인증서의 갱신을 자동으로 시도합니다. 갱신에 실패한다면 HTTP 검증 레코드 관련 문제로 인한 것일 수 있습니다. 이런 경우, ACM은 사용자에게 인증서를 자동으로 갱신할 수 없다고 알립니다.

> **Important**
>
> `RedirectFrom` 위치의 콘텐츠가 인증서의 각 도메인에 대한 `RedirectTo` 위치의 콘텐츠와 일치하는지 확인해야 합니다.

ACM 콘솔에서 인증서와 도메인 항목을 확장하면 도메인에 대한 HTTP 검증 정보를 찾을 수 있습니다. ACM API의 DescribeCertificate 작업 또는 ACM CLI의 describe-certificate 명령을 사용하여 이 정보를 검색할 수도 있습니다. 자세한 내용은 AWS Certificate Manager HTTP 검증 단원을 참조하십시오.

문제가 지속된다면 지원 센터에 문의하세요.

갱신 타이밍 이해

에서 관리형 인증서 갱신 AWS Certificate Manager은 비동기식 프로세스입니다. 즉, 단계가 연속으로 수행되지 않습니다. ACM 인증서의 모든 도메인 이름이 검증된 후 ACM이 새 인증서를 가져오기 전에 지연이 발생할 수 있습니다. ACM이 갱신된 인증서를 가져오는 시간과 인증서가 사용될 AWS 리소스에 배포되는 시간 사이에 추가 지연이 발생할 수도 있습니다. 따라서 인증서 상태 변경이 콘솔에 표시되는 데 최대 몇 시간 정도 걸릴 수 있습니다.

기타 문제 해결

이 섹션에는 ACM 인증서 발급 또는 검증과 관련이 없는 문제에 대한 지침이 포함되어 있습니다.

주제

- 인증 기관 권한 부여(CAA) 문제
- 인증서 가져오기 문제
- 인증서 고정 문제
- API Gateway 문제
- 작동 중인 인증서가 예기치 않게 실패할 경우 해결 방법
- ACM 서비스 연결 역할(SLR) 문제

인증 기관 권한 부여(CAA) 문제

CAA DNS 레코드를 사용하여 해당 Amazon 인증 기관(CA)이 도메인 또는 하위 도메인에 대한 ACM 인증서를 발급하도록 지정할 수 있습니다. 인증서 발급 도중 “Certificate Authority Authorization(CAA, 인증서 발급 기관 인증) 오류로 인해 하나 이상의 도메인 이름을 검증하는 데 실패했습니다.”라는 오류 메시지를 받으면 해당 CAA DNS 레코드를 확인하세요. ACM 인증서 요청이 검증된 후에 이 오류 메시지를 받는 경우, CAA 레코드를 업데이트하고 인증서를 다시 요청해야 합니다. CAA 레코드에 있는 value 필드에 다음 도메인 이름 중 하나가 포함되어야 합니다.

- amazon.com
- amazontrust.com

- awstrust.com
- amazonaws.com을 위한 CNAME 별칭으로 구성해야 합니다

CAA 레코드 생성에 대한 자세한 내용은 (선택 사항) CAA 레코드 구성을 참조하세요.

> **Note**
>
> CAA 확인 기능을 활성화하고 싶지 않다면 도메인에 CAA 레코드를 구성하지 않도록 선택할 수 있습니다.

인증서 가져오기 문제

서드 파티 인증서를 ACM으로 가져와서 통합 서비스와 연결할 수 있습니다. 문제가 발생할 경우 사전 요구 사항 및 인증서 형식 주제를 검토하세요. 특히 다음 사항에 유의하세요.

- X.509 버전 3 SSL/TLS 인증서만 가져올 수 있습니다.
- 인증서는 자체 서명하거나 인증 기관(CA)에서 서명할 수 있습니다.
- CA에서 인증서에 서명하는 경우 기관의 루트에 대한 경로를 제공하는 중간 인증서 체인을 포함시켜야 합니다.
- 인증서가 자체 서명된 경우 프라이빗 키를 일반 텍스트에 포함시켜야 합니다.
- 체인의 각 인증서는 앞에 지정된 인증서를 직접 인증해야 합니다.
- 최종 엔터티 인증서를 중간 인증서 체인에 포함시키지 마세요.
- 인증서, 인증서 체인 및 프라이빗 키(있는 경우)는 PEM 인코딩되어야 합니다. 일반적으로 PEM 인코딩은 Base64로 인코딩된 ASCII 텍스트 블록으로 구성되며 일반 텍스트 머리글 줄로 시작하고 일반 텍스트 바닥글 줄로 끝납니다. PEM 파일을 복사하거나 업로드하는 동안 줄 또는 공백을 추가하거나 PEM 파일을 변경할 수 없습니다. OpenSSL 검증 유틸리티를 사용하여 인증서 체인을 검증할 수 있습니다.
- 프라이빗 키(있는 경우)는 암호화되지 않아야 합니다. (팁: 암호가 있는 경우 암호화됨)
- ACM과 통합된 서비스는 ACM이 지원하는 알고리즘 및 키 크기를 사용해야 합니다. 인증서가 작동하는지 확인하려면 AWS Certificate Manager 사용 설명서와 각 서비스에 대한 설명서를 참조하세요.
- 통합 서비스에서 지원되는 인증서는 인증서를 IAM으로 가져올지 ACM으로 가져올지 여부에 따라 달라질 수 있습니다.
- 인증서는 가져오는 시점에 유효해야 합니다.

- 모든 인증서의 세부 정보가 콘솔에 표시됩니다. 그러나 기본적으로 keyTypes 필터를 지정하지 않고 ListCertificates API 또는 list-certificates AWS CLI 명령을 호출하면 RSA_1024 또는 RSA_2048 인증서만 표시됩니다.

인증서 고정 문제

인증서를 갱신하기 위해 ACM은 새로운 퍼블릭-프라이빗 키 페어를 생성합니다. 애플리케이션에서 SSL 고정이라고도 인증서 고정하는를 사용하여 ACM 인증서를 고정하면가 인증서를 AWS 갱신한 후 애플리케이션이 도메인에 연결하지 못할 수 있습니다. 따라서 ACM 인증서를 고정하지 않는 것이 좋습니다. 애플리케이션이 인증서를 고정해야 하는 경우 다음 단계를 따릅니다.

- 자체 인증서를 ACM으로 가져온 다음 애플리케이션을 가져온 인증서에 고정합니다. ACM은 가져온 인증서에 대한 관리형 갱신을 제공하지 않습니다.
- 퍼블릭 인증서를 사용하는 경우 애플리케이션을 Amazon 루트 인증서에 고정합니다. 프라이빗 인증서를 사용하는 경우 애플리케이션을 CA의 루트 인증서에 고정합니다.

API Gateway 문제

edge-optimized API 엔드포인트를 배포할 때 API Gateway는 CloudFront 배포를 자동으로 설정합니다. CloudFront 배포는 사용자의 계정이 아닌 API Gateway의 소유입니다. 배포는 API를 배포할 때 사용한 ACM 인증서에 바인딩됩니다. 바인딩을 제거하고 ACM이 인증서를 삭제하도록 허용하려면 인증서와 연결된 API Gateway 사용자 지정 도메인을 제거해야 합니다.

리전별 API 엔드포인트를 배포할 때, API Gateway는 사용자를 대신하여 Application Load Balancer(ALB)를 생성합니다. 이 로드 밸런서는 API Gateway의 소유이고, 사용자에게 보이지 않습니다. ALB는 API를 배포할 때 사용한 ACM 인증서에 바인딩됩니다. 바인딩을 제거하고 ACM이 인증서를 삭제하도록 허용하려면 인증서와 연결된 API Gateway 사용자 지정 도메인을 제거해야 합니다.

작동 중인 인증서가 예기치 않게 실패할 경우 해결 방법

ACM 인증서를 통합 서비스에 성공적으로 연결했지만 인증서가 작동을 멈추고 통합 서비스가 오류를 반환하기 시작하는 경우, ACM 인증서를 사용하기 위해 서비스에 필요한 사용 권한이 변경되기 때문일 수 있습니다.

예를 들어 Elastic Load Balancing(ELB)은 인증서의 프라이빗 키를 해독 AWS KMS key 하는를 해독할 수 있는 권한이 필요합니다. 이 권한은 인증서를 ELB와 연결할 때 ACM이 적용하는 리소스 기반 정책에 의해 부여됩니다. ELB에 해당 권한이 더 이상 부여되지 않게 되면 다음에 인증서 키의 암호를 복호화하려고 할 때 실패합니다.

문제를 조사하려면에서 AWS KMS 콘솔을 사용하여 권한 부여 상태를 확인합니다https://console.aws.amazon.com/kms. 그런 다음 다음 작업 중 하나를 수행합니다.

- 통합 서비스에 부여된 권한이 취소되었다고 판단되면 통합 서비스의 콘솔을 방문하여 서비스에서 인증서를 분리한 다음 다시 연결합니다. 그러면 리소스 기반 정책이 다시 적용되고 새로운 권한이 적용됩니다.
- ACM에 부여된 권한이 취소되었다고 생각되면 https://console.aws.amazon.com/support/home 지원에 문의하십시오.

ACM 서비스 연결 역할(SLR) 문제

다른 계정에서 공유한 프라이빗 CA에서 서명한 인증서를 발급하면 ACM은 먼저를 사용하여 AWS Private CA 리소스 기반 액세스 정책과 보안 주체로 상호 작용하도록 서비스 연결 역할(SLR)을 설정하려고 시도합니다. 공유 CA에서 프라이빗 인증서를 발급했는데 SLR이 설정되어 있지 않으면 ACM이 해당 인증서를 자동으로 갱신할 수 없습니다.

이 경우 ACM은 계정에 SLR이 있는지 여부를 확인할 수 없다는 알림을 표시할 수 있습니다. 필요한 `iam:GetRole` 권한이 이미 계정의 ACM SLR에 부여된 경우, SLR이 생성된 후 알림이 다시 표시되지 않습니다. 알림이 다시 표시될 경우 사용자 또는 계정 관리자가 ACM에 `iam:GetRole` 권한을 부여하거나 계정을 ACM 관리형 정책 `AWSCertificateManagerFullAccess`에 연결해야 할 수 있습니다.

자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한 섹션을 참조하세요.

예외 처리

AWS Certificate Manager 명령은 여러 가지 이유로 실패할 수 있습니다. 각 예외에 대한 자세한 내용은 아래 표를 참조하세요.

프라이빗 인증서 예외 처리

에서 발급한 프라이빗 PKI 인증서를 갱신하려고 할 때 다음과 같은 예외가 발생할 수 있습니다 AWS Private CA.

> **Note**
>
> AWS Private CA 중국(베이징) 리전 및 중국(닝샤) 리전에서는가 지원되지 않습니다.

ACM 오류 코드	설명
PCA_ACCESS_DENIED	프라이빗 CA에 ACM 권한이 부여되지 않았습니다. 이렇게 하면 AWS Private CA AccessDeniedException 실패 코드가 트리거됩니다. 문제를 해결하려면 AWS Private CA CreatePermission 작업을 사용하여 ACM 서비스 보안 주체에 필요한 권한을 부여합니다.
PCA_INVALID_DURATION	요청된 인증서의 유효 기간이 발급한 프라이빗 CA의 유효 기간을 초과합니다. 이렇게 하면 AWS Private CA ValidationException 실패 코드가 트리거됩니다. 이 문제를 해결하려면 적절한 유효 기간이 있는 새 CA 인증서를 설치하세요.
PCA_INVALID_STATE	호출 중인 프라이빗 CA가 요청된 ACM 작업을 수행할 수 있는 올바른 상태가 아닙니다. 이렇게 하면 AWS Private CA InvalidStateException 실패 코드가 트리거됩니다. 다음과 같이 문제를 해결하세요. • CA 상태가 CREATING인 경우, 생성이 완료될 때까지 기다린 다음 CA 인증서를 설치합니다. • CA 상태가 PENDING_CERTIFICATE 인 경우 CA 인증서를 설치합니다. • CA 상태가 DISABLED인 경우 ACTIVE 상태로 업데이트합니다. • CA 상태가 DELETED인 경우 복원하세요. • CA 상태가 EXPIRED인 경우 새 인증서를 설치합니다. • CA 상태가 FAILED이고 문제를 해결할 수 없는 경우 지원에 문의하세요.

ACM 오류 코드	설명
PCA_LIMIT_EXCEEDED	사설 CA가 발급 할당량에 도달했습니다. 이렇게 하면 AWS Private CA LimitExceededException 실패 코드가 트리거됩니다. 이 도움말을 계속 진행하기 전에 요청을 반복해 보세요. 오류가 지속되면 지원에 문의하여 할당량 증가를 요청하세요.
PCA_REQUEST_FAILED	네트워크 또는 시스템 오류가 발생했습니다. 이렇게 하면 AWS Private CA RequestFailedException 실패 코드가 트리거됩니다. 이 도움말을 계속 진행하기 전에 요청을 반복해 보세요. 오류가 지속될 경우 지원에 문의하세요.
PCA_RESOURCE_NOT_FOUND	사설 CA가 영구적으로 삭제되었습니다. 이렇게 하면 AWS Private CA ResourceNotFoundException 실패 코드가 트리거됩니다. 올바른 ARN을 사용했는지 확인합니다. 실패하면 이 CA를 사용할 수 없습니다. 문제를 해결하려면 새 CA를 만듭니다.
SLR_NOT_FOUND	ACM이 다른 계정에 상주하는 프라이빗 CA로 서명한 인증서를 갱신하려면, 인증서가 상주하는 계정에 대한 서비스 연결 역할(SLR)이 필요합니다. 삭제된 SLR을 다시 생성해야 하는 경우 ACM에 대한 SLR 생성 섹션을 참조하세요.

할당량

다음 AWS Certificate Manager (ACM) 서비스 할당량은 각 계정당 각 AWS AWS 리전에 적용됩니다.

조정할 수 있는 할당량을 확인하려면 AWS 일반 참조 가이드의 ACM 할당량 표를 참조하세요. 할당량 증가를 요청하려면 지원 센터에서 사례를 생성합니다.

일반 할당량

항목	기본 할당량
ACM 인증서 수 만료 및 해지된 인증서도 이 합계에 계속 포함됩니다. 에서 CA가 서명한 인증서는이 합계에 포함되지 AWS Private CA 않습니다.	2500
연간(지난 365일) ACM 인증서 수 연간, 리전당, 계정당 ACM 인증서 할당량의 최대 두 배까지 요청할 수 있습니다. 예를 들어 할당량이 2,500개인 경우, 지정된 리전 및 계정에서 연간 ACM 인증서를 최대 5,000개 요청할 수 있습니다. 특정 시점에 2,500개의 인증서만 보유할 수 있습니다. 1년에 인증서를 5,000개 요청하려면 연중에 2,500개를 삭제하여 할당량을 유지해야 합니다. 특정 시점에 2,500개 이상의 인증서가 필요한 경우, 지원 센터에 문의하세요. 에서 CA가 서명한 인증서는이 합계에 포함되지 AWS Private CA 않습니다.	5,000
가져온 인증서 수	2,500
연간(지난 365일) 인증서를 가져온 개수	5,000

항목	기본 할당량
ACM 인증서당 도메인 이름 수 기본 할당량은 ACM 인증서당 도메인 이름 10개입니다. 할당량이 더 클 수 있습니다. 제출하는 첫 번째 도메인 이름은 인증서의 일반 이름(CN)으로 포함됩니다. 모든 이름은 주체 대체 이름(SAN) 확장자에 포함됩니다. 최대 100개의 도메인 이름을 요청할 수 있습니다. 할당량 증가를 요청하려면 ACM 서비스에 대한 Service Quotas 콘솔에서 요청을 생성합니다. 하지만 사례를 생성하기 전에 이메일 검증을 사용하려면 더 많은 도메인 이름을 추가하여 더 많은 관리 작업을 만들 수 있는 방법을 이해하십시오. 자세한 내용은 도메인 검증 단원을 참조하십시오. ACM 인증서당 도메인 이름 수 할당량은 ACM에서 제공한 인증서에만 적용됩니다. 이 할당량은 ACM으로 가져오는 인증서에는 적용되지 않습니다. 다음 섹션은 ACM 인증서에만 해당합니다.	10

항목	기본 할당량
사설 CA 수 ACM은 AWS Private Certificate Authority ()와 통합됩니다AWS Private CA. ACM 콘솔 AWS CLI또는 ACM API를 사용하여에서 호스팅하는 기존 사설 인증 기관(CA)에서 사설 인증서를 요청할 수 있습니다 AWS Private CA. 인증서는 ACM 환경 내에서 관리되며 ACM에서 발급된 공인 인증서와 동일한 제한 사항이 있습니다. 자세한 내용은 에서 프라이빗 인증서 요청 AWS Certificate Manager 단원을 참조하십시오. 독립 실행형 AWS Private CA 서비스를 사용하여 프라이빗 인증서를 발급할 수도 있습니다. 자세한 내용은 사설 인증서 발급을 참조하십시오. 삭제된 사설 CA는 그 복원 기간이 종료될 때까지 할당량에 포함됩니다. 자세한 내용은 사설 CA 삭제를 참조하십시오.	200
CA별 사설 인증서 수(수명)	1,000,000

API 비율 할당량

다음 할당량은 각 리전 및 계정별로 ACM API에 적용됩니다. ACM은 API 작업에 따라 서로 다른 할당량으로 API 작업을 제한합니다. 제한이란 요청이 초당 요청 수에 대한 해당 작업의 할당량을 초과하기 때문에 다른 경우라면 유효한 요청을 ACM이 거부하는 것을 의미합니다. 요청이 제한되면 ACM이 `ThrottlingException` 오류를 반환합니다. 다음 표에는 각각의 API 작업과 ACM이 해당 작업에 대한 요청을 제한하는 할당량이 나와 있습니다.

> **Note**
>
> 아래 표에 나열된 API 작업 외에도 ACM은 AWS Private CA에서 외부 `IssueCertificate` 작업을 호출할 수도 있습니다. `IssueCertificate`에 대한 최신 서비스 할당량 정보는 AWS Private CA에 대한 엔드포인트 및 할당량을 참조하세요.

각 ACM API 작업에 대한 초당 요청 할당량

API 직접 호출	초당 요청
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

자세한 내용은 AWS Certificate Manager API 참조를 참조하세요.

문서 기록

다음 표에서는 2018 AWS Certificate Manager 년부터의 설명서 릴리스 기록을 설명합니다.

변경 사항	설명	날짜
업데이트된 퍼블릭 인증서 유효 기간	이제 퍼블릭 ACM 인증서는 13개월(395일)에서 줄어든 198일 동안 유효합니다. 이 업데이트를 통해 ACM에서 발급한 퍼블릭 인증서는 CA/B 포럼의 향후 인증서 수명 요구 사항을 준수합니다. 새 요구 사항에 따라 2026년 3월 15일 이후에 발급된 퍼블릭 인증서의 최대 유효 기간은 200일이어야 합니다. 퍼블릭 인증서의 갱신 기간이 만료 45일 전으로 업데이트되었습니다. 프라이빗 인증서는 13개월(395일) 동안 유효합니다.	2026년 2월 18일
인증서 다시 가져오기에 대한 변경	ACM은 ClientAuth EKU가 이전 인증서에서 누락된 경우에만 인증서를 동일한 ARN으로 다시 가져오도록 허용합니다. 이는 인증 기관이 더 이상 Chrome의 루트 프로그램 요구 사항을 준수하기 위해 ClientAuth EKU로 인증서를 발급하지 않는 업계 변화를 수용합니다.	2025년 10월 22일
인증서 발급에 대한 참고 사항 추가	ACM 인증서 개념 주제에 TLS Web Client Authentication 확장을 사용한 ACM 인증서 발급	2025년 7월 23일

	에 대한 변경 사항을 자세히 설명하는 참고 사항이 추가되었습니다.	
인증 확장에 대한 참조 제거	예제 인증서에서 TLS Web Client Authentication 확장에 대한 참조가 제거되었습니다.	2025년 7월 3일
AWS Certificate Manager 내보내기 가능한 퍼블릭 인증서	ACM 퍼블릭 인증서를 내보낼 수 있습니다.	2025년 6월 17일
ACM은 CloudFront의 HTTP 검증을 지원합니다	이제 ACM은 CloudFront 배포용 인증서를 발급할 때 도메인 소유권 확인을 위한 HTTP 검증을 지원합니다.	2025년 4월 24일
메일 교환기(MX) 이메일 검증 사용 중단	ACM 콘솔은 더 이상 메일 교환기(MX)를 지원하지 않습니다.	2024년 7월 11일
계정 수준 분리에 대한 모범 사례 추가	가능하면 정책에서 계정 수준 분리를 사용합니다. 불가능한 경우 계정 수준에서 또는 정책의 암호화 컨텍스트 조건 키를 통해 권한을 제한할 수 있습니다.	2024년 6월 11일
향후 WHOIS 이메일 검증 사용 중단	2024년 6월부터 WHOIS 이메일 검증 사용 중단에 대한 참고 사항이 추가되었습니다.	2024년 2월 5일
조건 키 지원 추가	ACM 인증서를 요청할 때 IAM 조건 키에 대한 지원이 추가되었습니다. 지원되는 조건 목록은 https://docs.aws.amazon.com/acm/latest/userguide/acm-conditions.html#acm-conditions-supported을(를) 참조하세요.	2023년 8월 24일

ECDSA 지원이 추가됨	퍼블릭 ACM 인증서를 요청할 때 ECDSA(Elliptic Curve Digital Signature Algorithm)에 대한 지원을 추가했습니다. 지원되는 키 알고리즘 목록은 https://docs.aws.amazon.com/acm/latest/userguide/acm-certificate.html#algorithms 섹션을 참조하세요.	2022년 11월 8일
새로운 CloudWatch 이벤트	ACM 인증서 만료, ACM 인증서 사용 가능 및 ACM 인증서 갱신 작업 필요 이벤트를 추가했습니다. 지원되는 CloudWatch 이벤트 목록은 https://docs.aws.amazon.com/acm/latest/userguide/cloudwatch-events.html 섹션을 참조하세요.	2022년 10월 27일
가져오기를 위한 키 알고리즘 유형 업데이트	이제 ACM으로 가져온 인증서에는 RSA 및 타원 곡선 알고리즘이 추가로 포함된 키가 있을 수 있습니다. 현재 지원되는 키 알고리즘 목록은 https://docs.aws.amazon.com/acm/latest/userguide/import-certificate-prerequisites.html 섹션을 참조하세요.	2021년 7월 14일

"모니터링 및 로깅"을 별도의 장으로 승격합니다.	모니터링 및 로깅 설명서를 별도의 장으로 옮겼습니다. 이 변경 사항은 CloudWatch 지표, CloudWatch Events/Eventbridge 및 CloudTrail에 적용됩니다. 자세한 내용은 https://docs.aws.amazon.com/acm/latest/userguide/monitoring-and-logging.html 단원을 참조하십시오.	2021년 3월 23일
CloudWatch 지표 및 이벤트 지원 추가	DaysToExpiry 지표와 이벤트 및 지원 API가 추가되었습니다. 자세한 내용은 https://docs.aws.amazon.com/acm/latest/userguide/cloudwatch-metrics.html 및 https://docs.aws.amazon.com/acm/latest/userguide/cloudwatch-events.html 섹션을 참조하세요.	2021년 3월 3일
계정 간 지원 추가	에서 프라이빗 CAs를 사용하기 위한 교차 계정 지원이 추가되었습니다 AWS Private CA. 자세한 내용은 https://docs.aws.amazon.com/acm/latest/userguide/ca-access.html 단원을 참조하십시오.	2020년 8월 17일
리전 지원 추가	AWS 중국(베이징 및 닝샤) 리전에 대한 리전 지원이 추가되었습니다. 지원되는 리전 목록은 https://docs.aws.amazon.com/general/latest/gr/rande.html#acm-pca_region 섹션을 참조하세요.	2020년 3월 4일

갱신 워크플로 테스트 추가	이제 고객은 ACM 관리형 갱신 워크플로의 구성을 수동으로 테스트할 수 있습니다. 자세한 내용은 ACM의 관리형 갱신 구성 테스트를 참조하세요.	2019년 3월 14일
이제 인증서 투명성 로깅 기본 적용	ACM 퍼블릭 인증서를 인증서 투명성 로그에 기본적으로 게시하는 기능이 추가되었습니다.	2018년 24월 4일
시작 AWS Private CA	ACM Private Certificate Manager(CM)를 시작하고 사용자가 프라이빗 디지털 인증서를 발급하고 취소하기 위한 안전한 관리형 인프라를 설정할 수 AWS Certificate Manager 있도록를 확장했습니다. 자세한 내용은 AWS Private Certificate Authority를 참조하세요.	2018년 4월 4일
인증서 투명성 로깅	모범 사례에 인증서 투명성 로깅을 추가했습니다.	2018년 3월 27일

다음 표에서는 2018 AWS Certificate Manager 년 이전의 설명서 릴리스 기록을 설명합니다.

변경	설명	릴리스 날짜
새로운 내용	DNS 검증을 추가하려면 AWS Certificate Manager DNS 검증	2017년 11월 21일
새로운 내용	SDK for Java와 AWS Certificate Manager 함께 사용에 새로운 Java 코드 예제를 추가했습니다.	2017년 10월 12일

변경	설명	릴리스 날짜
새로운 내용	(선택 사항) CAA 레코드 구성에 CAA 레코드에 대한 정보를 추가했습니다.	2017년 9월 21일
새로운 내용	.IO 도메인에 대한 정보를 의 문제 해결 AWS Certificate Manager에 추가했습니다.	2017년 7월 7일
새로운 내용	인증서 다시 가져오기에 대한 정보를 인증서 다시 가져오기에 추가했습니다.	2017년 7월 7일
새로운 내용	인증서 고정에 대한 정보를 모범 사례 및 의 문제 해결 AWS Certificate Manager에 추가했습니다.	2017년 7월 7일
새로운 내용	CloudFormation 에가 추가되었습니다ACM에 통합된 서비스.	2017년 5월 27일
업데이트	자세한 정보를 할당량에 추가했습니다.	2017년 5월 27일
새로운 내용	에 대한 자격 증명 및 액세스 관리 AWS Certificate Manager에 대한 문서를 추가했습니다.	2017년 4월 28일
업데이트	검증 이메일을 전송하는 위치를 보여 주는 그래픽을 추가했습니다. AWS Certificate Manager 이메일 검증을(를) 참조하세요.	2017년 21월 4일
업데이트	도메인에 대한 이메일 설정 관련 정보를 추가했습니다. AWS Certificate Manager 이메일 검증을(를) 참조하세요.	2017년 4월 6일

변경	설명	릴리스 날짜
업데이트	콘솔에서 인증서 갱신 상태를 확인하는 방법에 대한 정보를 추가했습니다. 인증서의 갱신 상태 확인을(를) 참조하세요.	2017년 3월 28일
업데이트	Elastic Load Balancing 사용에 대해 설명서가 업데이트되었습니다.	2017년 3월 21일
새로운 내용	AWS Elastic Beanstalk 및 Amazon API Gateway에 대한 지원이 추가되었습니다. ACM에 통합된 서비스을(를) 참조하세요.	2017년 3월 21일
업데이트	관리형 인증서 갱신에 대한 문서를 업데이트했습니다.	2017년 20월 2일
새로운 내용	가져온 인증서에 대한 문서를 추가했습니다.	2016년 10월 13일
새로운 내용	ACM 작업에 대한 AWS CloudTrail 지원이 추가되었습니다. 에서 CloudTrail 사용 AWS Certificate Manager을(를) 참조하세요.	2016년 3월 25일
새 안내서	이 릴리스는 AWS Certificate Manager을 도입했습니다.	2016년 1월 21일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.