**Domain: 10. Security, Identity, & Compliance**

**Topic: AWS Certificate Manager (ACM)**

## Service Overview

AWS Certificate Manager is a service that lets you easily provision, manage, and deploy public and private Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates for use with AWS services and your internal connected resources. SSL/TLS certificates are used to secure network communications and establish the identity of websites over the Internet as well as resources on private networks. AWS Certificate Manager removes the time-consuming manual process of purchasing, uploading, and renewing SSL/TLS certificates.

## Use cases / Considerations

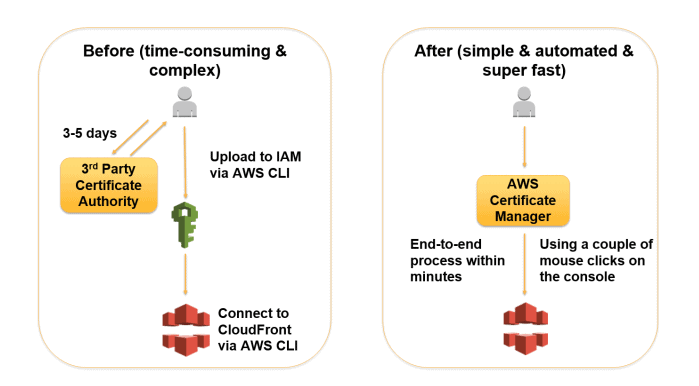
With AWS Certificate Manager, you can quickly request a certificate, deploy it on ACM-integrated AWS resources, such as Elastic Load Balancers, Amazon CloudFront distributions, and APIs on API Gateway, and let AWS Certificate Manager handle certificate renewals. It also enables you to create private certificates for your internal resources and manage the certificate lifecycle centrally. **Public and private certificates provisioned through AWS Certificate Manager for use with ACM-integrated services are free**. You pay only for the AWS resources you create to run your application. With [AWS Certificate Manager Private Certificate Authority](https://aws.amazon.com/certificate-manager/private-certificate-authority/), you pay monthly for the operation of the private CA and for the private certificates you issue.

ACM is integrated with the following services:

* Elastic Load Balancing
* Amazon CloudFront – To use an ACM certificate with CloudFront, you must request or import the certificate in the US East (N. Virginia) region.
* AWS Elastic Beanstalk
* Amazon API Gateway
* AWS Nitro Enclaves
* AWS CloudFormation

AWS Certificate Manager manages the renewal process for the certificates managed in ACM and used with ACM-integrated services.

You can **import your own certificates** into ACM, however you have to **renew these yourself**.



**Governance**

* ACM Certificate are X.509 version 3 certificates. Each is valid for **13 months**.
* When you request an ACM certificate, you must validate that you own or control all of the domains that you specify in your request.
* **Each ACM Certificate must include at least one fully qualified domain name (FQDN)**. You can add additional names if you want to.
* You can create an ACM Certificate containing a wildcard name (\*.example.com) that can protect several sites in the same domain (subdomains).
* You cannot download the private key for an ACM Certificate.
* The first time you request or import a certificate in an AWS region, ACM creates an AWS-managed customer master key (CMK) in AWS KMS with the alias aws/acm. This CMK is unique in each AWS account and each AWS region. ACM uses this CMK to encrypt the certificate’s private key.
* You cannot add or remove domain names from an existing ACM Certificate. Instead you must request a new certificate with the revised list of domain names.
* You cannot delete an ACM Certificate that is being used by another AWS service. To delete a certificate that is in use, you must first remove the certificate association.
* Applications and browsers trust public certificates automatically by default, whereas an administrator must explicitly configure applications to trust private certificates.

## Cautions

ACM certificates must be requested or imported in the same AWS Region as your Classic Load Balancer or Application Load Balancer.

To use the ACM certificates with Amazon CloudFront, the certificates must be imported or requested in the US East (N. Virginia) Region. For more information, see [AWS Region that you request a certificate in (for AWS Certificate Manager)](https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/cnames-and-https-requirements.html#https-requirements-aws-region).

To import a self–signed SSL/TLS certificate into ACM, you must provide the certificate and its private key. To import a signed certificate, you must also include the certificate chain. Your certificate must satisfy the following criteria:

* The certificate must specify a cryptographic algorithm and a key size. The following algorithms are supported by ACM:
  + 1024-bit RSA (RSA\_1024)
  + 2048-bit RSA (RSA\_2048)
  + 4096-bit RSA (RSA\_4096)
  + Elliptic Prime Curve 256 bit (EC\_prime256v1)
  + Elliptic Prime Curve 384 bit (EC\_secp384r1)
  + Elliptic Prime Curve 521 bit (EC\_secp521r1)
* The certificate must be an SSL/TLS X.509 version 3 certificate. It must contain a public key, the fully qualified domain name (FQDN) or IP address for your website, and information about the issuer. The certificate can be self-signed by your private key or by the private key of an issuing CA. If your certificate is signed by a CA, you must include the certificate chain when you import your certificate.
* The certificate must be valid at the time of import. You cannot import a certificate before its validity period begins or after it expires. The NotBefore certificate field contains the validity start date, and the NotAfter field contains the end date.
* The private key must be unencrypted. You cannot import a private key that is protected by a password or passphrase.
* The certificate, private key, and certificate chain must be PEM–encoded. For more information and examples, see [Certificate and key format for importing](https://docs.aws.amazon.com/acm/latest/userguide/import-certificate-format.html).
* The cryptographic algorithm of an imported certificate must match the algorithm of the signing CA. For example, if the signing CA key type is RSA, then the certificate key type must also be RSA.

[*https://docs.aws.amazon.com/acm/latest/userguide/import-certificate-prerequisites.html*](https://docs.aws.amazon.com/acm/latest/userguide/import-certificate-prerequisites.html)

**Pricing considerations**

Public SSL/TLS certificates provisioned through AWS Certificate Manager are free. You pay only for the AWS resources you create to run your application.

Please pay attention to pricing examples [here](https://aws.amazon.com/certificate-manager/pricing/).

Be careful with Private CA, since it costs much. Private Certificate Authority Operation

$400.00 per month for each ACM private CA until you delete the CA.

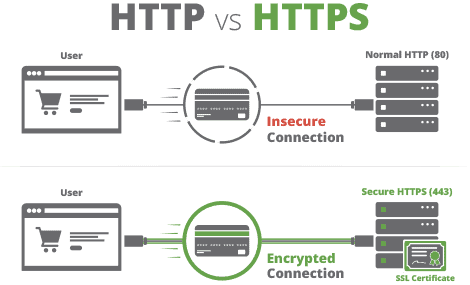
ACM Private CA operation is pro-rated for partial months based on when you create and delete the CA. You are not charged for a private CA after you delete it. However, **if you restore a deleted CA, you are charged for the time between deleting it and restoring it**.

**More details**

## ****Overview of SSL/TLS Certificates****

An SSL certificate is like an ID card or a badge that proves someone is who they say they are. SSL certificates are stored and displayed on the Web by a website’s or application’s server.

**SSL (Secure Socket Layer)** is the standard security technology for establishing an encrypted link between a web server and a browser. This link ensures that all data passed between the web server and browsers remain private and integral. Secure Sockets Layer/Transport Layer Security (SSL/TLS) is a must-have whenever sensitive data is moved to and from a website. For instance, sites that require to fulfil compliance requirements such as PCI-DSS, FedRAMP, and HIPAA make extensive use of SSL/TLS. Unfortunately, provisioning and managing SSL/TLS certificates can entail a lot of work that is usually manual and not easily automated.

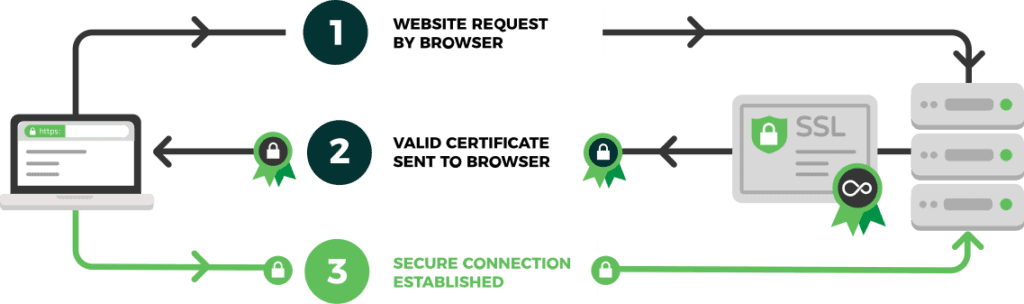


### **Are SSL and TLS identical?**

Transport Layer Security (TLS) is the successor protocol to SSL. TLS is an improved version of SSL. It works in much the same way as the SSL, using encryption to protect the transfer of data and information. The two terms are often used interchangeably in the industry although SSL is still widely used.

## ****How SSL/TLS works****

1. A server attempts to connect to a website (i.e. a web-server) secured with SSL. The server requests the web-server to identify itself.
2. The web-server sends the server a copy of its SSL certificate.
3. The server checks to see whether or not it trusts the SSL certificate. If so, it sends a message to the web-server.
4. The web-server sends back a digitally signed acknowledgement to start an SSL encrypted session.
5. Encrypted data is shared between the server and the web-server.



**Task: Import self-signed certificate to ACM and use it for application load balancer**

**Problem to Be Solved**

*In case if you or customer want to use https connection between some of ACM-integrated AWS resources.*

## Explanation of the Solution

*Create own Root Certification Authority (rootCA)*

*Create a new certificate request (\*.csr) using IP address or hostname of desired resource.*

*Issue a certificate using csr and rootCA*

[*https://docs.aws.amazon.com/acm/latest/userguide/import-certificate-format.html*](https://docs.aws.amazon.com/acm/latest/userguide/import-certificate-format.html)

***Note:*** *Use the following steps to get appropriate values (X.509 PEM format) for ACM import in case if customer provides you required certificate in \*.pfx format and password for it.*

# Get Certificate body:

openssl pkcs12 -in cert.pfx -nodes -out certificate.pem

cat certificate.pem

# Get Certificate private key:

openssl pkcs12 -in cert.pfx -nocerts -out encrypted.key

openssl rsa -in encrypted.key -out unercrypted.key

# Get Certificate chain:

openssl pkcs12 -in cert.pfx -cacerts -nokeys -chain -out cacerts.crt

sed -nie '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' cacerts.crt

cat cacerts.crt

## Implementation Details

*Use Linux (laptop, WSL, VM, EC2 instance)*

***Note:*** *Skip first two steps, if you need only 1 certificate.*

* 1. *Generate Root Certification Authority Key (RSA Private Key, take it in secret!)*

openssl genrsa -out rootca.key 2048

* 1. *Generate RootCA certificate for 5 years (1825 days) – public part (could be shared with others)*

openssl req -x509 -new -key rootca.key -days 1825 -out rootca.crt

You will be asked about providing some info, so put anything as you wish, f.i.

*-----*

*Country Name (2 letter code) [AU]:BY*

*State or Province Name (full name) [Some-State]:Minsk*

*Locality Name (eg, city) []:Minsk*

*Organization Name (eg, company) [Internet Widgits Pty Ltd]:AWS Mentoring Program*

*Organizational Unit Name (eg, section) []:DevOps*

*Common Name (e.g. server FQDN or YOUR name) []:Root CA for ACM*

*Email Address []:we@we.we*

*Now you are ready for certificates issuing for your server/service.*

* 1. *Generate key for your host/IP*

openssl genrsa -out mysite.domain.key 2048

* 1. *Create new certificate request*

openssl req -new -key mysite.domain.key -out mysite.domain.csr

*Give near the same values as in step 2, except “Common Name (e.g. server FQDN or YOUR name)”, set it to appropriate domain name or IP-address. Leave password blank*

*-----*

*Country Name (2 letter code) [AU]:BY*

*State or Province Name (full name) [Some-State]:Minsk*

*Locality Name (eg, city) []:Minsk*

*Organization Name (eg, company) [Internet Widgits Pty Ltd]:AWS Mentoring Program*

*Organizational Unit Name (eg, section) []:DevOps*

*Common Name (e.g. server FQDN or YOUR name) []:mysite.domain*

*Email Address []:we@we.we*

*Please enter the following 'extra' attributes*

*to be sent with your certificate request*

*A challenge password []:*

*An optional company name []:*

* 1. *Sign our request from the previous step by our RootCA certificate:*

openssl x509 -req -in mysite.domain.csr \

-CA rootca.crt -CAkey rootca.key \

-CAcreateserial -out mysite.domain.crt -days 375

* 1. *Now RootCA certificate could be installed in your browser.*

*IE and Chrome uses Windows Certificates Storage, make import into Trusted Root Certificate Authorities.*

*Firefox for Windows has own Certificates Storage.*

*For Ubuntu run next commands:*

sudo mkdir /usr/share/ca-certificates/extra

sudo cp rootCA.crt /usr/share/ca-certificates/extra/rootca.crt

sudo dpkg-reconfigure ca-certificates

sudo update-ca-certificates

* 1. *Assign your certificate with appropriate service and check by browser.*

## Benefits / Outcomes / Pros and Cons / Summary

*Now https connection could be established without additional charge for issuing certificate from ACM.*

## Tearing down

*Remove certificate from resource, remove certificate from ACM.*