AWS News Blog

Application Load Balancers Now Support Multiple TLS Certificates With Smart Selection Using SNI

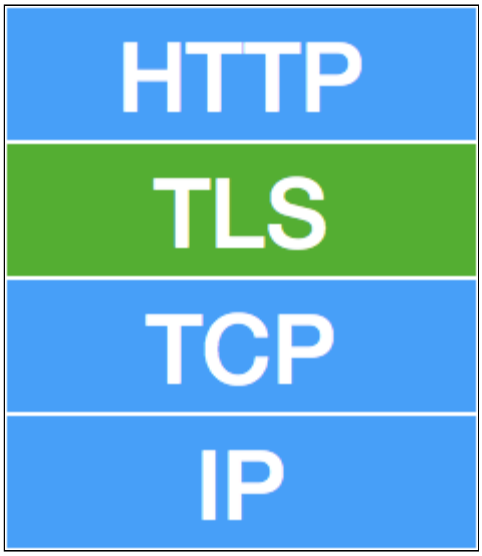
by Randall Hunt | on 10 OCT 2017 | in AWS Certificate Manager, Elastic Load Balanc :urity, Identity, & Compliance | Permalink | Share

Today we're launching support for multiple TLS/SSL certificates on Application Load Balancers (ALB) using Server Name Indication (SNI). You can now host multiple TLS secured applications, each with its own TLS certificate, behind a single load balancer. In order to use SNI, all you need to do is bind multiple certificates to the same secure listener on your load balancer. ALB will automatically choose the optimal TLS certificate for each client. These new features are provided at no additional charge.

If you're looking for a TL;DR on how to use this new feature just click here. If you're lik nd you're a little rusty on the specifics of Transport Layer Security (TLS) then keep reading.

TLS? SSL? SNI?

People tend to use the terms SSL and TLS interchangeably even though the two are technically different. SSL technically refers to a predecessor of the TLS protocol. To keep things simple I'll be using the term TLS for the rest of this post.



TLS is a protocol for securely transmitting data like passwords, cookies, and credit card numbers. It enables privacy, authentication, and integrity of the data being transmitted. TLS uses certificate based authentication where certificates are like ID cards for your websites. You trust the person that signed and issued the certificate, the certificate authority (CA), so you trust that the data in the certificate is correct. When a browser connects to your TLS-enabled ALB, ALB presents a certificate that contains your site's public key, which has been cryptographically signed by a CA. This way the client can be sure it's getting the 'real you' and that it's safe to use your site's public key to establish a secure connection.

With SNI support we're making it easy to use more than one certificate with the same ALB. The most common reason you might want to use multiple certificates is to handle different domains

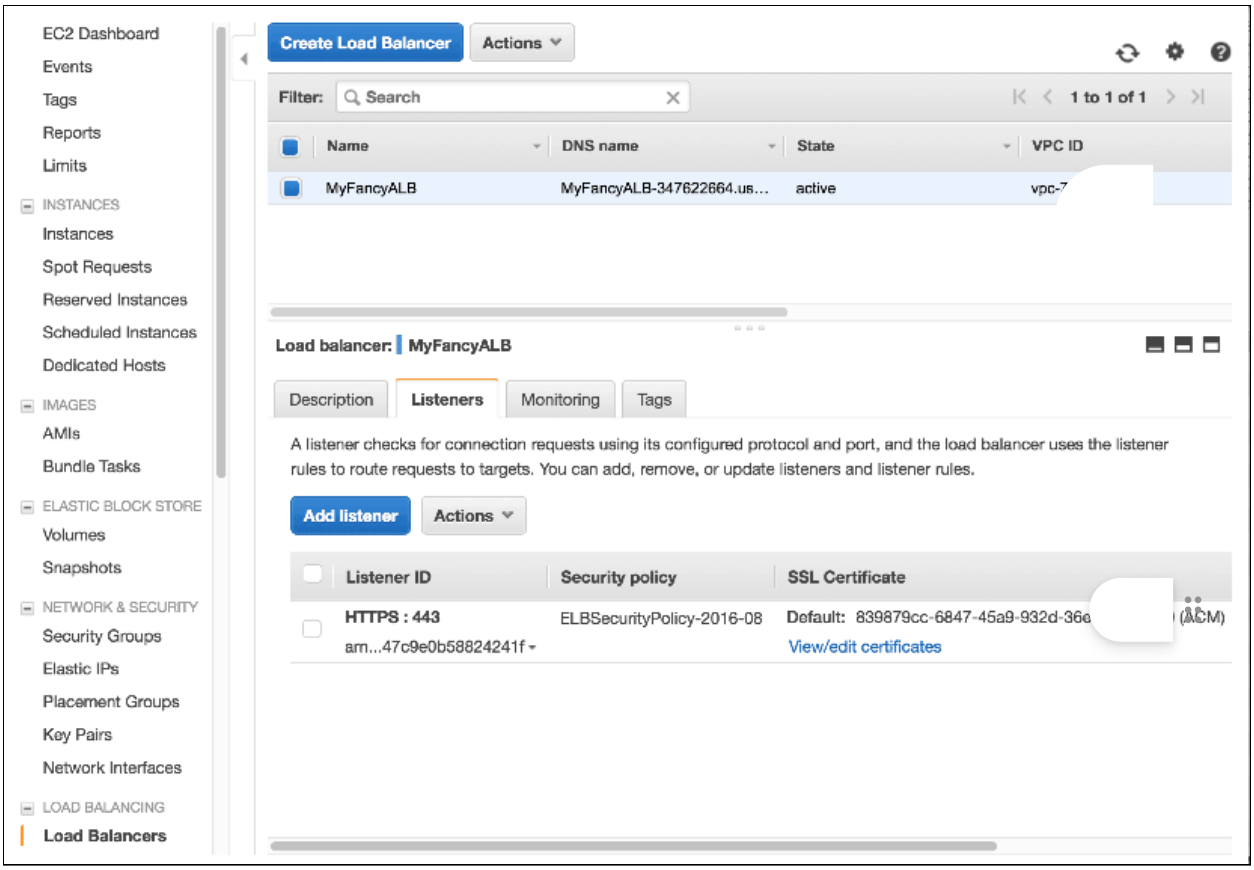
with the same load balancer. It's always been possible to use wildcard and subject-alternate-name (SAN) certificates with ALB, but these come with limitations. Wildcard certificates only work for related subdomains that match a simple pattern and while SAN certificates can support many different domains, the same certificate authority has to authenticate each one. That means you have reauthenticate and reprovision your certificate everytime you add a new doma

One of our most frequent requests on forums, reddit, and in my e-mail inbox has been to use the Server Name Indication (SNI) extension of TLS to choose a certificate for a client. Since TLS operates at the transport layer, below HTTP, it doesn't see the hostname requested by a client. SNI works by having the client tell the server "This is the domain I expect to get a certificate for" when it first connects. The server can then choose the correct certificate to respond to the client. All modern web browsers and a large majority of other clients support SNI. In fact, today we see SNI supported by over 99.5% of clients connecting to CloudFront.

Smart Certificate Selection on ALB

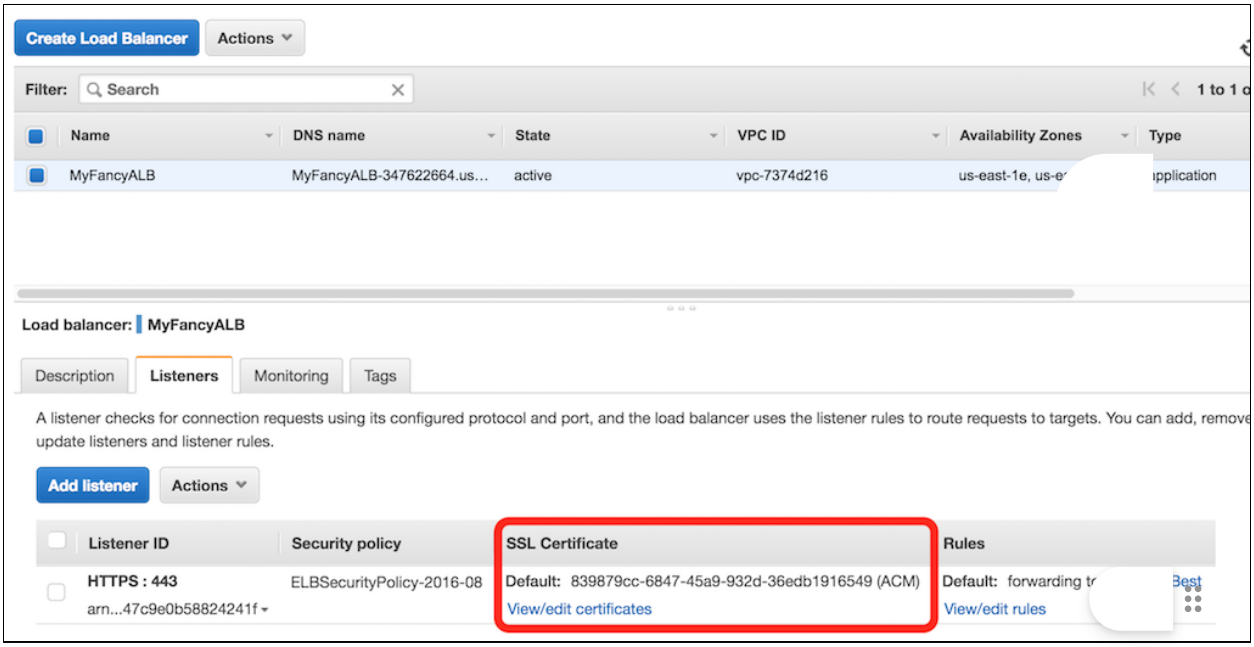
ALB's smart certificate selection goes beyond SNI. In addition to containing a list of valid domain names, certificates also describe the type of key exchange and cryptography that the server supports, as well as the signature algorithm (SHA2, SHA1, MD5) used to sign the certificate. To establish a TLS connection, a client starts a TLS handshake by sending a "ClientHello" message that outlines the capabilities of the client: the protocol versions, extensions, cipher suites, and compression methods. Based on what an individual client supports, ALB's smart selection algorithm chooses a certificate for the connection and sends it to the client. ALB supports both the classic RSA algorithm and the newer, hipper, and faster Elliptic-curve based ECDSA algorithm. ECDSA support among clients isn't as prevalent as SNI, but it is supported by all modern web browsers. Since it's faster and requires less CPU, it can be particularly useful for ultra-low latency applications and for conserving the amount of battery used by mobile applications. Since ALB can see what each client supports from the TLS handshake, you can upload both RSA and ECDSA certificates for the same domains and ALB will automatically choose the best one for each client.

Using SNI with ALB

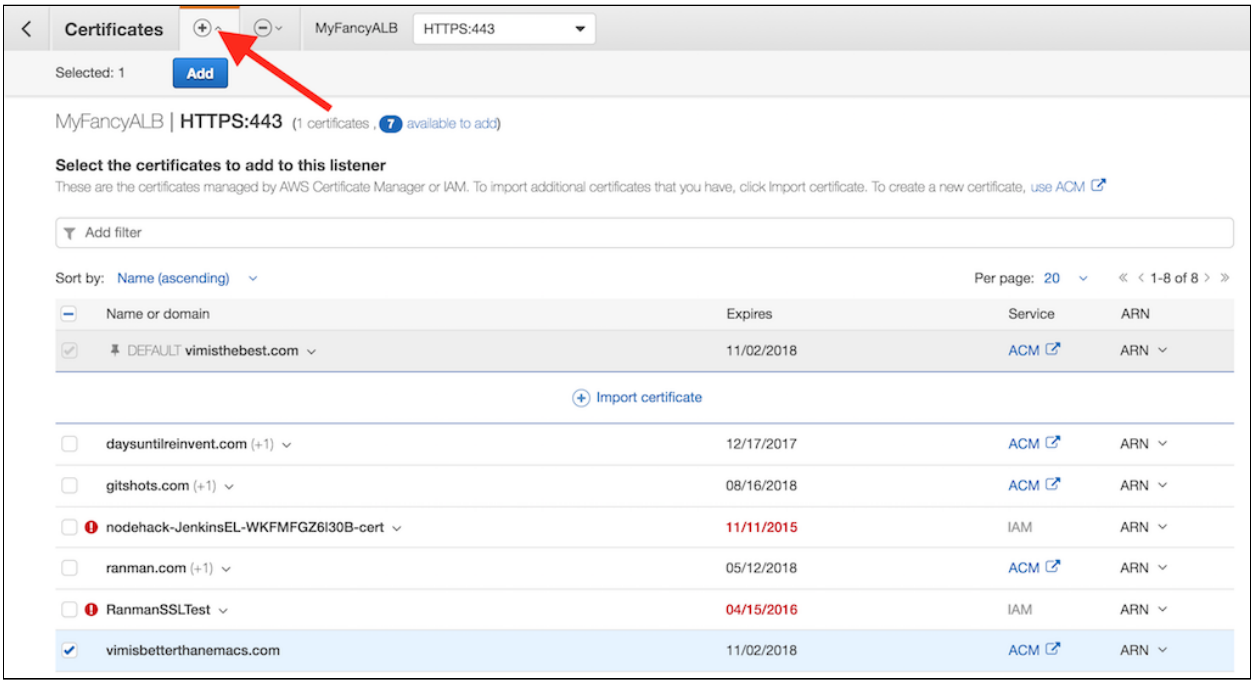


I'll use a few example websites like VimIsBetterThanEmacs.com and VimIsTheBest.com. I've purchased and hosted these domains on Amazon Route 53, and provisioned two separate certificates for them in AWS Certificate Manager (ACM). If I want to securely serve both of these sites through a single ALB, I can quickly add both certificates in the console.

First, I'll select my load balancer in the console, go to the listeners tab, and select "view/edit certificates".



Next, I'll use the "+" button in the top left corner to select some certificates then I'll click the "Add" button.



There are no more steps. If you're not really a GUI kind of person you'll be pleased to know that it's also simple to add new certificates via the AWS Command Line Interface (AWS CLI) (or SDKs).

```
Bash
aws elbv2 add-listener-certificates --listener-arn <listener-arn> --certi
```

Things to know

- ALB Access Logs now include the client's requested hostname and the certificate ARN used. If the "hostname" field is empty (represented by a "-") the client did not use the SNI extension in their request.
- You can use any of your certificates in ACM or IAM.
- You can bind multiple certificates for the same domain(s) to a secure listener. Your ALB will choose the optimal certificate based on multiple factors including the capabilities of the client.
- If the client does not support SNI your ALB will use the default certificate (the one you specified when you created the listener).
- There are three new ELB API calls: AddListenerCertificates, RemoveListenerCertificates, and DescribeListenerCertificates.
- You can bind up to 25 certificates per load balancer (not counting the default certificate).
- These new features are supported by AWS CloudFormation at launch.

You can see an example of these new features in action with a set of websites created by my colleague Jon Zobrist: https://www.exampleloadbalancer.com/.

Overall, I will personally use this feature and I'm sure a ton of AWS users will benefit from it as well. I want to thank the Elastic Load Balancing team for all their hard work in getting this into the hands of our users.

– Randall