The Red Head's Tech Blog: Creating and Attaching a Symantec SSL Cert to ELB

Posted by FALYCIA HANKINSON Sep 18, 2017
Have you or your loved ones been diagnosed with Mesothelioma?
Oh wait....

Are you sifting through Pulse pages right now trying to create a daisy-chain that will tell you how to create a CSR, build a cert, and upload the cert to AWS?
Are you trying to set up an interactive Slackbot but can't get Slack to talk to you because you have no VALID public SSL endpoint?

Then hooray for you, you made it!

This post only goes as far as uploading a cert to an AWS ELB. It doesn't walk you through creating a WAF or actually creating an internet-facing ELB. If you are wanting a Barracuda WAF, hop on over to their FAQ page. If you are using ModSec, I suggest moving to a Barracuda since ModSec is going away. If you are trying to use the AWS WAF, DON'T. Just stop right there because it's not approved for use due to a list of deficiencies.

Here is a quick TL;DR breakdown of what you are gonna need:
1. A catchy domain name such as myawesomeapp.capitalone.com. (Don't use that for real, we are in finance. Keep it straight and to the point.)
2. You'll need an external facing endpoint such as a public-facing ELB DNS Name or a Route53 DNS record from a public zone.
3. Take number 1 and number 2 and open an HPSM quote to get them connected so 1 translates to 2.
4. Generate a CSR with step 1 domain name as FQDN. Store the key somewhere safe. (Not Github, please no.)

5. Submit request to get a Symantec cert using the CSR generated in step 4 by following something like this. Requires supervisor approval.
6. Wait until they email you the cert. It doesn't take long, maybe a day or two at the latest.
7. Upload the cert, the parent chain cert (root), and the key from step 4 into your ELB SSL listener. The root cert is very important to validate your domain. (steps below)

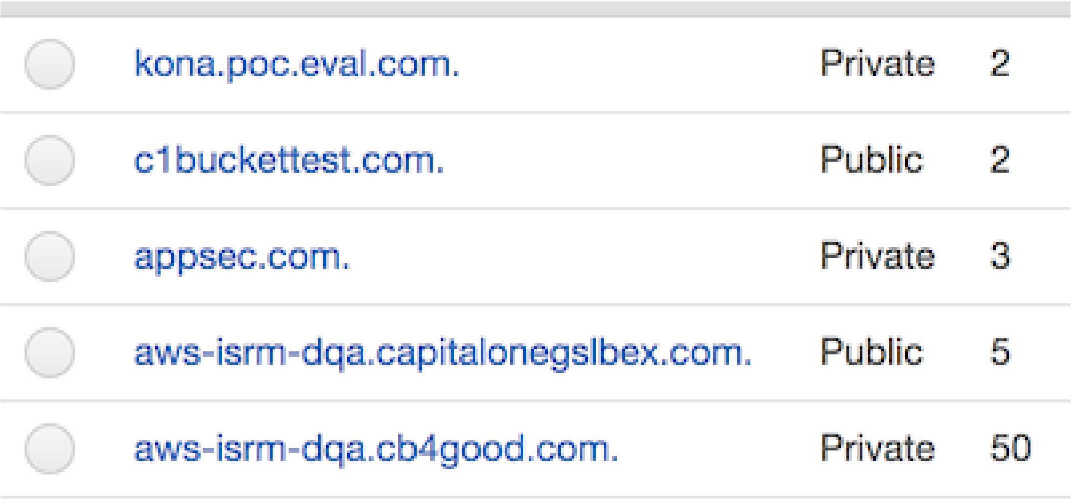
Voila! Try to navigate to https : // <myawesomeapp>.capitalone.com EXTERNALLY and it should present you a valid cert. (I shouldn't have to tell you that specific url is obviously not valid)

Ok, so would you like to go down the rabbit hole a little further?

Step 1 is self-explanatory. If you have an app named financeapp (how creative) it would be logical to use financeapp.capitalone.com but its a free country so don't let me tell you what to do. But there is also this thing called copyright infringement, so let's not name it something like disney.capitalone.com so some Daffy Duck can run across it on the WWW...

Step 2. In 9 out of 10 cases, this is an external-facing cloud resource such as a internet-facing Elastic Load Balancer that sits in front of a WAF (ex. Barracuda, etc.) or a Route 53 DNS record created in a PUBLIC zone. Making a Route53 DNS record that is in a private zone will be absolutely useless to you in this scenario. In this post, I am going to assume you know how to make these resources. If not and an AWS n00b is reading, hop on over to the #aws-questions slack channel or use Mr. Google.

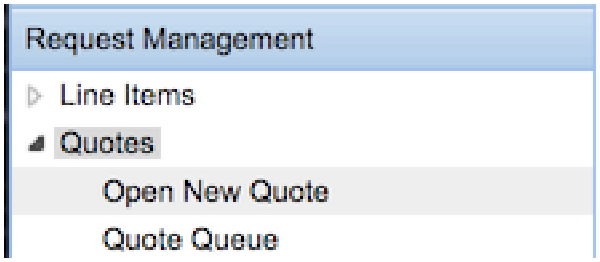
You can identify if the zone is public or private in the "Type" column in the R53 console page.



○	kona.poc.eval.com.	Private	2
○	c1buckettest.com.	Public	2
○	appsec.com.	Private	3
○	aws-isrm-dqa.capitalonegslbex.com.	Public	5
○	aws-isrm-dqa.cb4good.com.	Private	50

Step 3 requires you open up an HPSM quote to get a public DNS record so myawesomeapp.com will point to myexternalelb.us-east-1.elb.amazonaws.com or whatever your public endpoint is from step 2. There are a lot of pulse pages on how to do this and I'm still not entirely sure which one the Ops guys prefer but they'll TYPICALLY reach out to you if they don't understand something you submitted. Besides, it's HPSM so it won't look pretty anyways.

Some rough steps:
a. Open new HPSM Quote.



b. Select Data Networking, then select Network Support OAS.
c. Select DNS from the dropdown. Click Save.
d. Under the quote description, follow the below format and paste it in.

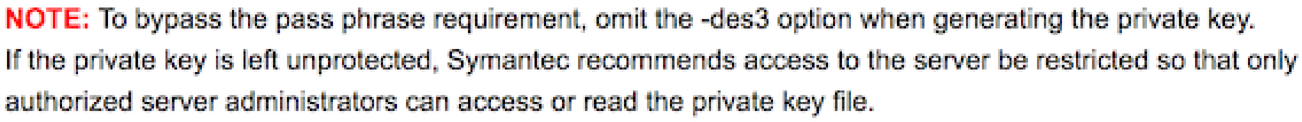
Type of Request: [Addition]
Line of Business Impacted by the Change: [<MYLOB>]
Record Type: [CNAME]
Hostname: <ELB public DNS name or R53 public zone record>
Fully Qualified Domain Name (FQDN): <changeme.capitalone.com>
IP Address for A records, or Host Record if Request is for an Alias/CNAME: <ELB public DNS name or R53 public zone record>
Please add any other pertinent information for your request:
<more info here if you have something super crazy>

----

e. Submit that beast and wait for them to contact you with the status.

Step 4 is creating your CSR (Certificate Signing Request). I had to google that. This is used to create your Private Key (keyword is PRIVATE) and a CSR file that has all the data about your certificate you want to create. There are variations of how to do this around Pulse like here and here. But don't get too click happy just yet, here are the steps that I know work:

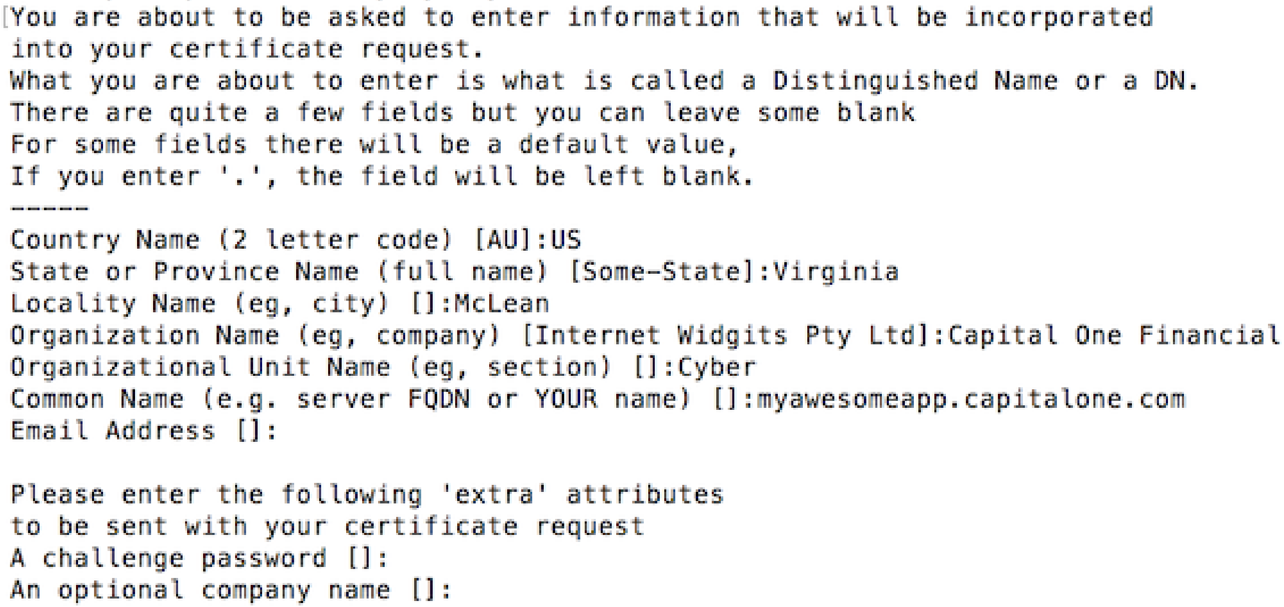
a. Run "openssl genrsa -des3 -out <private key file name>.key 2048" on an instance or local command line (source doesn't really matter).
b. Type in a passphrase twice. Save that passphrase somewhere in LastPass or other secret manager. Congrats you know have a Private Key.



NOTE: To bypass the pass phrase requirement, omit the -des3 option when generating the private key. If the private key is left unprotected, Symantec recommends access to the server be restricted so that only authorized server administrators can access or read the private key file.

c. Run "openssl req -new -key <private key file name>.key -out <csr file name>.csr" on an instance or local command line. The only thing you should change from the below screenshot is the Organizational UNIT Name and the Common Name.

Make sure you spell out the state, don't use "VA".
Don't change the Organization Name. It's for your own good.
No need to enter in an email address, challenge password, or optional company name.



```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:Virginia
Locality Name (eg, city) []:McLean
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Capital One Financial
Organizational Unit Name (eg, section) []:Cyber
Common Name (e.g. server FQDN or YOUR name) []:myawesomeapp.capitalone.com
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

You can also create a CSR in non-interactive mode by replacing step a, b, and c by formatting a one-liner.

Congratulations, you now have a CSR and a Private Key.

d. Upload the CSR to S3 or just open the file (cat in Linux), copy the full contents, and paste into Symantec Cert request in next steps.

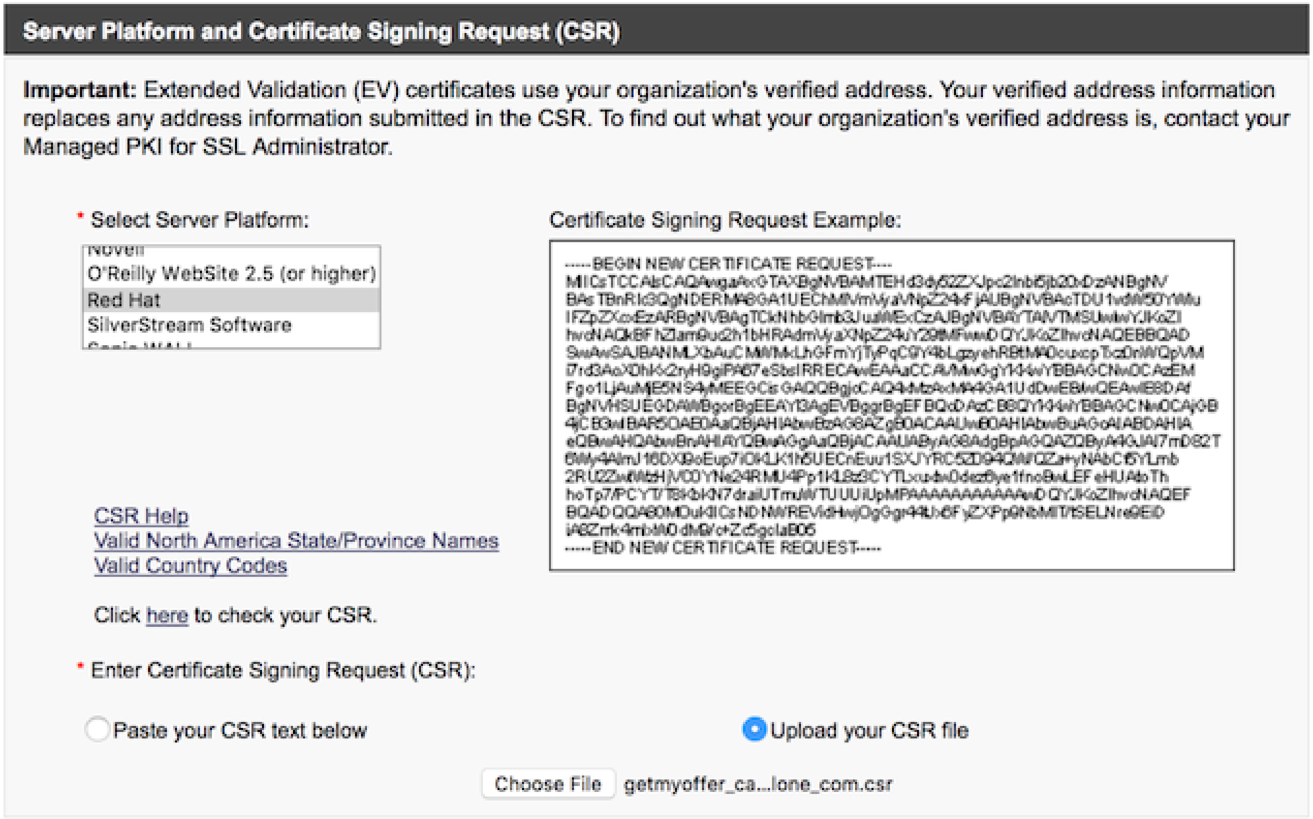
Don't forget to save your key somewhere else!! Again, not in GitHub or a wide open S3 bucket.

Step 5 is when you FINALLY get to submit your request to Symantec to spit out a certificate.

a. First, go here and type in/select everything they ask for.
Note: You will have to provide the Cost Center, ENV for your app, LOB, and App Owner among other things. Use the ENV that matches what App Environment you select.

b. Next section, you'll be required to select the platform where the cert will live. If the new cert is going to live in AWS and hosted on an ELB or CloudFront, I have personally set the platform to Red Hat or Apache and had the cert work just fine.

c. Then, you'll paste in the actual contents of the CSR file you created in Step 4 or just upload that puppy. I prefer pasting but below is an example of uploading. Stole screenshot from here.



Server Platform and Certificate Signing Request (CSR)

Important: Extended Validation (EV) certificates use your organization's verified address. Your verified address information replaces any address information submitted in the CSR. To find out what your organization's verified address is, contact your Managed PKI for SSL Administrator.

* Select Server Platform:

Certificate Signing Request Example:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIICsTCCAbCAQAwgaAxGTAXBgNVBAMTEHd3dy52ZXJpc2lnbi5jb20xDzANBgNV
BAsTBnRlc3QgNDERMA8GA1UEChMIVmVyaVNpZ24xFjAUBgNVBAcTDU1vdW50YWlu
IFZpZXcxEzARBgNVBAgTCkNhbGlmb3JuaWExCzAJBgNVBAYTAlVTMSUwIwYJKoZI
hvcNAQkBFhZlam9uZ2h1bHRAdmVyaXNpZ24uY29tMFwwDQYJKoZIhvcNAQEBBQAD
SwAwSAJBANMLXbAuCMiWMkLhGFmYjTyPqC9Y4bLgzyehRBtM40cuxcpTxzDnWQpVM
I7rd3AoXDhKx2nyH9gIPA67eSbsIRRECAwEAAaCCAWAwGgYKKwYBBAGCNwDCAzEM
Fgo1LjAuMjE5NS4yMEEGCisGAQQBgjcCAQ4xMzAxMA4GA1UdDwEB/wQEAwIE8DAf
BgNVHSUEGDAWBgorBgEEAYI3AgEVBggrBgEFBQcDAzCB8QYKKwYBBAGCNwOCAjGB
4jCB3wIBAR5OAEDAaQBjAHIAbwBzAG8AZgB0ACAAUwB0AHIAbwBuAGcAIABDAHIA
eQBwAHQAbwBnAHIAYQBwAGgAaQBjACAAUAByAG8AdgBpAGQAZQByA4GJAI7mD82T
6Wy4AImJ16DX9oEup7iOKLK1h5UECnEuu1SXJYRC5ZD94QMfQZa+yNAbCf6YLmb
2RU2ZwMbHjVCDYNe24RMU4Pp1KL8z3CYTLxubw0dez6ye1fnoBwLEFeHUAtoTh
hoTp7/PCYT/T8KbKN7dralUTmuWTUUUiUpMPAAAAAAAAAAAwDQYJKoZIhvcNAQEF
BQADQQA80MDuKlICsNDNWREVldHwjOgGgr44Ux6FyZXPp9NbMlT/tSELNre9EiD
iA8Zmk4mbWOdMB/c+Zc5gclaBO6
-----END NEW CERTIFICATE REQUEST-----
```

Click here to check your CSR.

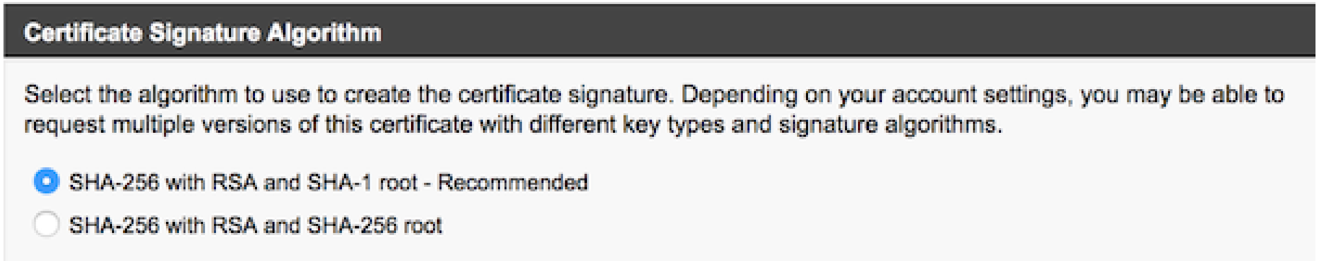
* Enter Certificate Signing Request (CSR):

○ Paste your CSR text below ● Upload your CSR file

getmyoffer_ca...lone_com.csr

d. Next screen asks for the domain name you chose in Step 1. Since it is a SAN cert, you can add both www.myawesomeapp.capitalone.com and myawesomeapp.capitalone.com but www isn't really necessary since no one has really used it in the past 8-10 years or so. You're showing your age... Here is an interesting tirade about using "www" for your site.

e. Select the recommended SHA config (duh).



Certificate Signature Algorithm

Select the algorithm to use to create the certificate signature. Depending on your account settings, you may be able to request multiple versions of this certificate with different key types and signature algorithms.

● SHA-256 with RSA and SHA-1 root - Recommended

○ SHA-256 with RSA and SHA-256 root

f. Select how many servers will host the cert. If you are hosting this on one ELB listener, put 1. If you have a DR site that will have the same cert on both ELB's, put 2. If you are hosting it server side for some reason, put the max number that your ASG can grow to just to be safe. Select 1 or 2 years for validity period. We all know you're going to select 2 years.

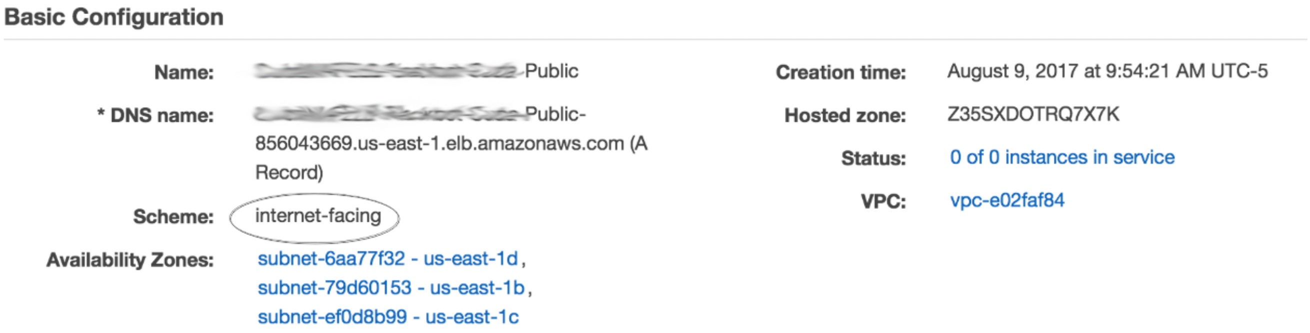
g. Next, put in your challenge phrase. No, its not the same as your private key passphrase. This is used to revoke or renew your cert. Save this somewhere too.

h. Click Accept on the Subscriber Agreement. You can pretend to read it.

Guess what? You just submitted a request for a cert. Feel proud of yourself?

Step 6 is where a lot of people get tripped up. Most of the time it has to do with the Parent Chain Cert, other times its just an extra space, or missing character when pasting in the 3 parts of the cert in the ELB listener.

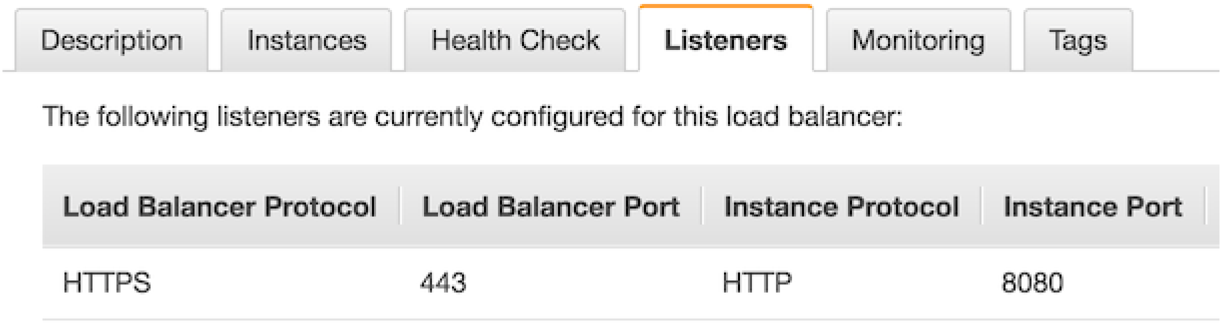
a. First things first, make sure you are attaching the cert to an internet-facing ELB.



Basic Configuration

Name:	[redacted]-Public	**Creation time:**	August 9, 2017 at 9:54:21 AM UTC-5
*** DNS name:**	[redacted]-Public-856043669.us-east-1.elb.amazonaws.com (A Record)	**Hosted zone:**	Z35SXDOTRQ7X7K
		Status:	0 of 0 instances in service
		VPC:	vpc-e02faf84
Scheme:	internet-facing		
Availability Zones:	subnet-6aa77f32 - us-east-1d , subnet-79d60153 - us-east-1b , subnet-ef0d8b99 - us-east-1c		

b. Go to the listener tab of your ELB and change your Load Balancer port and protocol to HTTPS/443. Change the Instance port/protocol to match whatever port your instances are listening on.



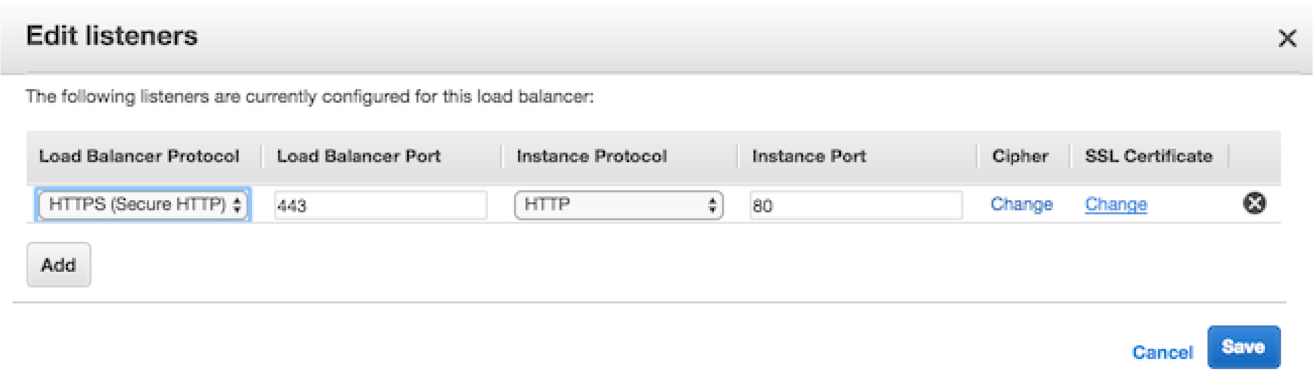
The following listeners are currently configured for this load balancer:

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
HTTPS	443	HTTP	8080

For the cipher, there isn't a pre-configured policy that won't sound the alarms on an insecure ELB listener so just use the below to custom configure it (the ones with an x next to it):

```
-----------
SSL Protocols
-----------
  Protocol-TLSv1
  Protocol-SSLv3
x Protocol-TLSv1.1
x Protocol-TLSv1.2
-----------
SSL Options
-----------
x Server Order Preference
-----------
SSL Ciphers
-----------
x ECDHE-ECDSA-AES128-GCM-SHA256
x ECDHE-RSA-AES128-GCM-SHA256
x ECDHE-ECDSA-AES128-SHA256
x ECDHE-RSA-AES128-SHA256
x ECDHE-ECDSA-AES128-SHA
x ECDHE-RSA-AES128-SHA
  DHE-RSA-AES128-SHA
x ECDHE-ECDSA-AES256-GCM-SHA384
x ECDHE-RSA-AES256-GCM-SHA384
  ECDHE-ECDSA-AES256-SHA384
x ECDHE-RSA-AES256-SHA384
x ECDHE-RSA-AES256-SHA
x ECDHE-ECDSA-AES256-SHA
x AES128-GCM-SHA256
x AES128-SHA256
x AES128-SHA
x AES256-GCM-SHA384
x AES256-SHA256
x AES256-SHA
  DHE-DSS-AES128-SHA
  CAMELLIA128-SHA
  EDH-RSA-DES-CBC3-SHA
x DES-CBC3-SHA
  ECDHE-RSA-RC4-SHA
  <ignore everything below this>
```

c. Now to actually upload the cert into the ELB. Note, you can upload certs via AWS CLI but it is finicky and a pain in the &$#. I highly recommend using it as a last resort.
Go to your ELB listener and under SSL Certificate, click Change.

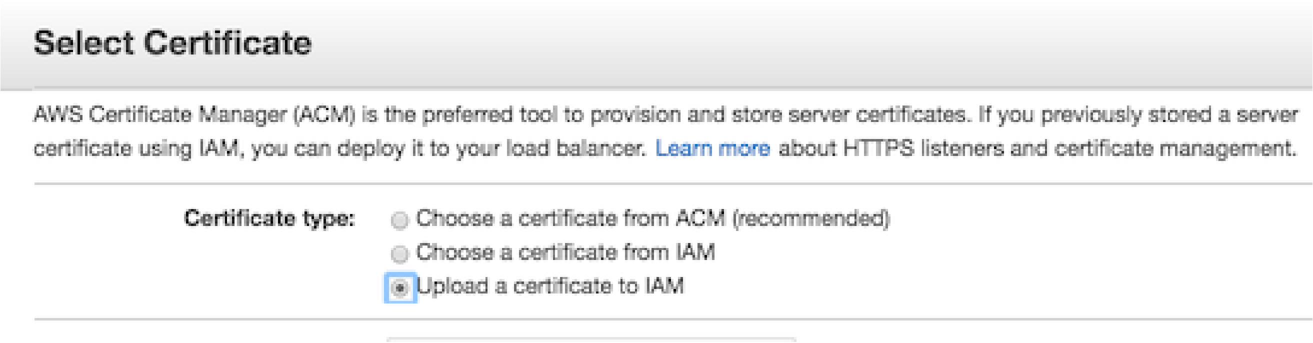


Edit listeners

The following listeners are currently configured for this load balancer:

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port	Cipher	SSL Certificate
HTTPS (Secure HTTP)	443	HTTP	80	Change	Change

Select "upload a cert to IAM".



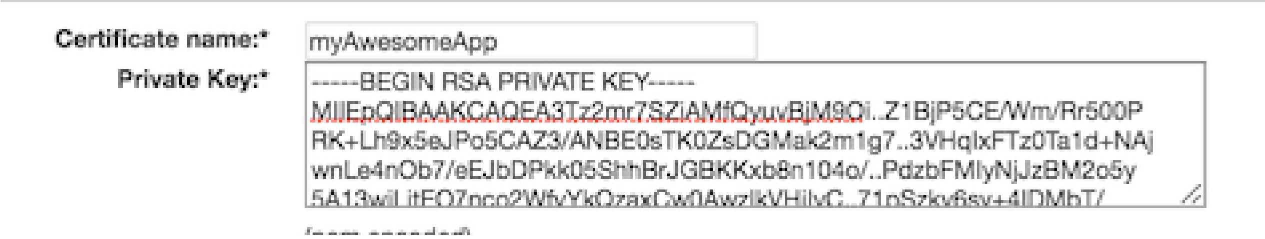
Select Certificate

AWS Certificate Manager (ACM) is the preferred tool to provision and store server certificates. If you previously stored a server certificate using IAM, you can deploy it to your load balancer. Learn more about HTTPS listeners and certificate management.

Certificate type:
○ Choose a certificate from ACM (recommended)
○ Choose a certificate from IAM
● Upload a certificate to IAM

Name your certificate. It'll show the name on the dropdown for "Choose a certificate from IAM". So nothing generic like "cert".

For the private key field, copy and paste the contents of your .key file created in step 4. Include the "-----BEGIN RSA PRIVATE KEY-----" and "-----END RSA PRIVATE KEY-----".



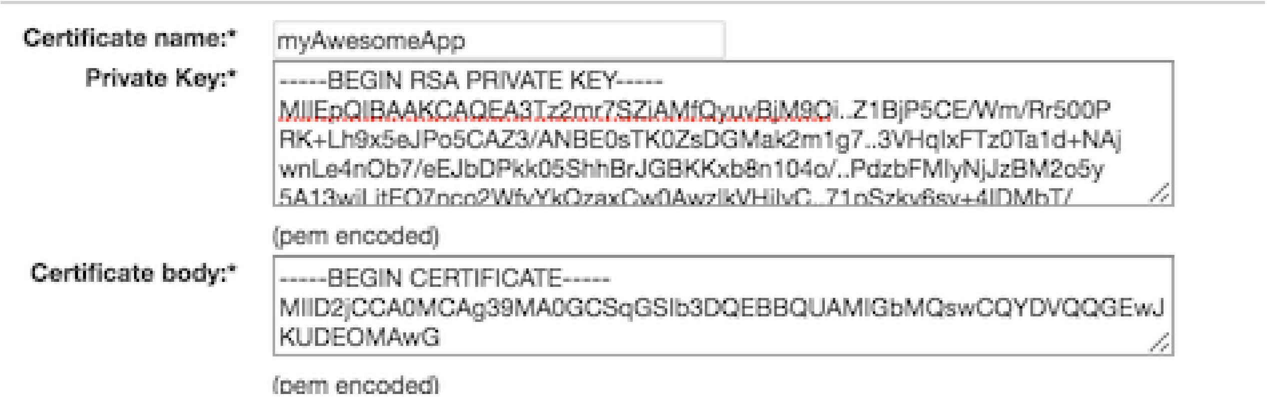
Certificate name:* myAwesomeApp

Private Key:*

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEA3Tz2mr7SZiAMfQyuvBjM9Oi..Z1BjP5CE/Wm/Rr500P
RK+Lh9x5eJPo5CAZ3/ANBE0sTK0ZsDGMak2m1g7..3VHqlxFTz0Ta1d+NAj
wnLe4nOb7/eEJbDPkk05ShhBrJGBKKxb8n104o/..PdzbFMlyNjJzBM2o5y
5A13wiLitEO7nco2WfvYkQzaxCw0AwzlkVHilvC..71nSzkv6sv+4IDMbT/
```

(pem encoded)

For the Certificate Body, copy and paste the contents of the public cert that Symantec emailed you the link for. (THIS IS NOT YOUR CSR CONTENTS) Include the "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----". It should look something like this now:



Certificate name:* myAwesomeApp

Private Key:*

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEA3Tz2mr7SZiAMfQyuvBjM9Oi..Z1BjP5CE/Wm/Rr500P
RK+Lh9x5eJPo5CAZ3/ANBE0sTK0ZsDGMak2m1g7..3VHqlxFTz0Ta1d+NAj
wnLe4nOb7/eEJbDPkk05ShhBrJGBKKxb8n104o/..PdzbFMlyNjJzBM2o5y
5A13wiLitEO7nco2WfvYkOzaxCw0AwzlkVHilvC..71oSzkv6sv+4IDMbT/
```

(pem encoded)

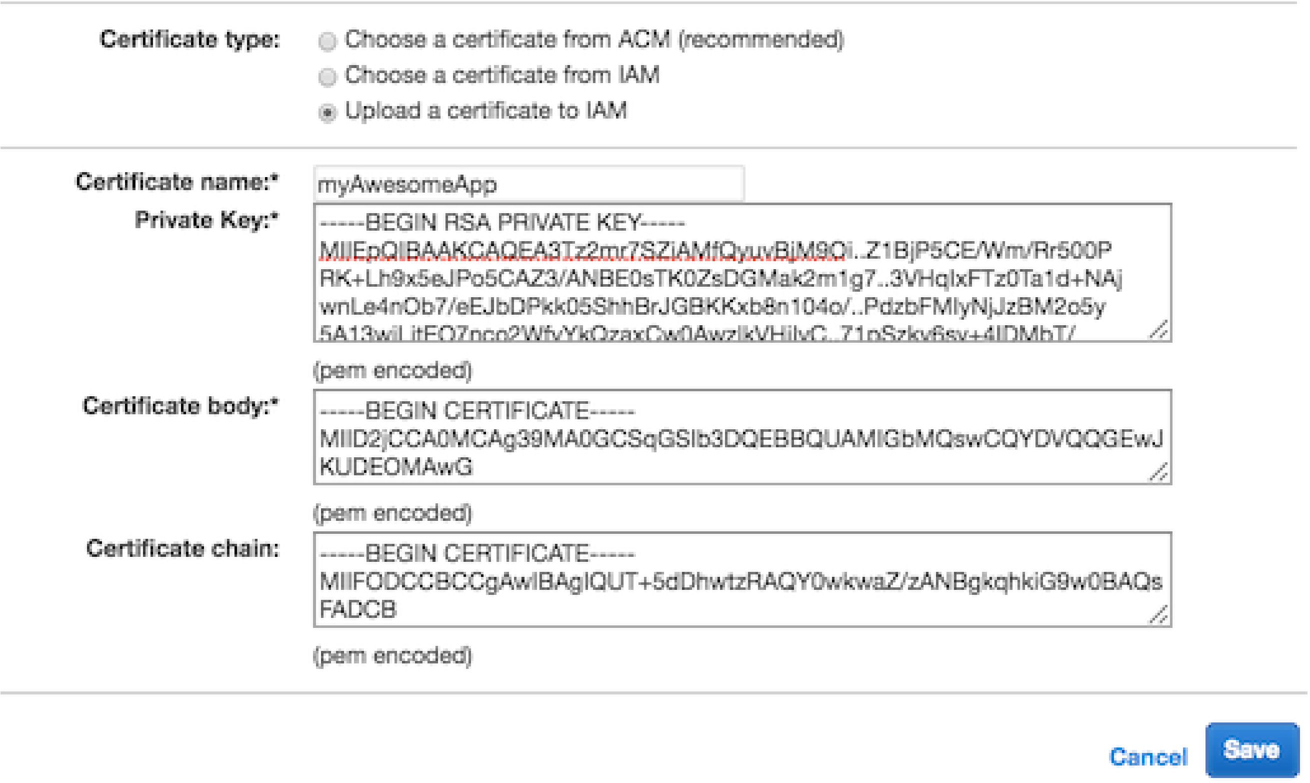
Certificate body:*

```
-----BEGIN CERTIFICATE-----
MIID2jCCA0MCAg39MA0GCSqGSIb3DQEBBQUAMIGbMQswCQYDVQQGEwJ
KUDEOMAwG
```

(pem encoded)

The Certificate Chain says its optional but for certs that are being used on internet-facing endpoints, it is not optional. If you don't have that puppy in there, your browser will most likely yell at you for not having a valid cert and third-party will not trust you.

If you googled your way to this page for Symantec Root CA's and couldn't figure out which one to download, you are in luck. The one you want is for RSA SHA-2 tab, SHA-2 Intermediate CAs (under SHA-1 Root) section on the top, Secure Site Certificate Type, Intermediate CA. Orrrr just click the direct download link here. Copy the file contents of the downloaded .cer file into the Certificate Chain section.



Certificate type:

○ Choose a certificate from ACM (recommended)
○ Choose a certificate from IAM
● Upload a certificate to IAM

Certificate name:* myAwesomeApp

Private Key:*

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEA3Tz2mr7SZiAMfQyuvBjM9Oi..Z1BjP5CE/Wm/Rr500P
RK+Lh9x5eJPo5CAZ3/ANBE0sTK0ZsDGMak2m1g7..3VHqlxFTz0Ta1d+NAj
wnLe4nOb7/eEJbDPkk05ShhBrJGBKKxb8n104o/..PdzbFMlyNjJzBM2o5y
5A13wiLitEO7nco2WfvYkOzaxCw0AwzlkVHilvC..71oSzkv6sv+4IDMbT/
```

(pem encoded)

Certificate body:*

```
-----BEGIN CERTIFICATE-----
MIID2jCCA0MCAg39MA0GCSqGSIb3DQEBBQUAMIGbMQswCQYDVQQGEwJ
KUDEOMAwG
```

(pem encoded)

Certificate chain:

```
-----BEGIN CERTIFICATE-----
MIIFODCCBCCgAwIBAgIQUT+5dDhwtzRAQY0wkwaZ/zANBgkqhkiG9w0BAQs
FADCB
```

(pem encoded)

Now click SAVE!!!!

Words of Wisdom

1. If you are going to want to browse to the site internally, you'll have to get your new domain whitelisted through the proxy.
2. If the domain is for an interactive slackbot, the parent chain cert MUST be valid and correctly added to the ELB for Slack to send data to it.
3. If you are like me, you probably have a "vendor" junk filter. Make sure you add Symantec as an exception because your cert may drop into your junk folder.

775 Views Tags: elb, aws elb, waf elb, #techitout, web certificate, aws-cert, aws cert, slack interactive, public endpoint, internet facing

There are no comments on this post