# Mulesoft SSL certificate renewal

# Mule certificate update (MANUAL)

### Download the cert from Venafi

* Login to [venafi.allegiantair.com](http://venafi.allegiantair.com)
* Go to 'Inventory/Certificated'
* In the search 'Certificate Name' type ex: 'stg.api.aws.allegiantair.com' (the CN of your cert)
* Complete the search and the cert will load on the right hand side, click on it
* Then select 'Actions' top right and select 'Download'
* On the popup select format: 'PEM OpenSSL' and selcet checkbox 'Extract PEM content into separate files (.crt, .key)' type in a new  password you will need to decrypt the key later

### Prepare the certificate details

* Move the dowloaded cert zip file in a folder with a decent name and explode the zip file there.  
    ex: unzip stg.api.aws.allegiantair.com.zip
* Decrypt the key ex:   
    openssl rsa -in stg.api.aws.allegiantair.com.key -out stg.api.aws.allegiantair.com.key\_decrypted  
    type in the password you set to download the cert from Venafy
* Prepare base64 one line encrypted value of the decrypted key ex: base64 -w0 stg.api.aws.allegiantair.com.key\_decrypted > stg.api.aws.allegiantair.com.key\_decrypted.base64.one.line
* You need to set up a BUNDLE
* cat ext.stg.api.aws.allegiantair.com.crt > ext.stg.api.aws.allegiantair.com-bundle.pem
* cat ext.stg.api.aws.allegiantair.com-chain.pem >> ext.stg.api.aws.allegiantair.com-bundle.pem
* This is how you build a bundle server cert on top and CA-chain afterwards
* Prepare base64 one line encrypted value of the bundle ex: base64 -w0 stg.api.aws.allegiantair.com-bundle.pem > stg.api.aws.allegiantair.com-bundle.pem.base64.one.line

### Actual renewal of SSL certificate

List all tls/secrets used in the cluster if you like to see them  
ex: kubectl get secrets --field-selector type=kubernetes.io/tls -A

Check what TLS/secrets are in rtf namespace  
ex: kubectl get secrets --field-selector type=[kubernetes.io/tls](http://kubernetes.io/tls) -n rtf  
NAME                       TYPE                DATA   AGE  
ext-mulesoft-tls           kubernetes.io/tls   3      356d  
ext-mulesoft-tls-godaddy   kubernetes.io/tls   2      355d  
mulesoft-tls               kubernetes.io/tls   2      355d  
mulesoft-tls-godaddy       kubernetes.io/tls   2      355d

List the ingress controllers in the rtf namespace  
ex: kubectl get ingress -n rtf  
NAME                       CLASS                HOSTS                              ADDRESS   PORTS     AGE  
ext-rtf-ingress-template   rtf-external-nginx   ext.stg.api.aws.allegiantair.com             80, 443   444d  
int-rtf-ingress-template   rtf-internal-nginx   stg.api.aws.allegiantair.com                 80, 443   444d

Do describe on each ingress template to see which tls/secret is in use  
ex: kubectl describe ingress -n rtf int-rtf-ingress-template

Name:             int-rtf-ingress-template  
Labels:           <none>  
Namespace:        rtf  
Address:            
Default backend:  default-http-backend:80 (<error: endpoints "default-http-backend" not found>)  
TLS:  
  mulesoft-tls-godaddy terminates stg.api.aws.allegiantair.com  
Rules:  
  Host                          Path  Backends  
  ----                          ----  --------  
  stg.api.aws.allegiantair.com    
                                /app-name   service:80 (<error: endpoints "service" not found>)  
Annotations:                    external-dns.alpha.kubernetes.io/hostname: stg.api.aws.allegiantair.com  
                                nginx.ingress.kubernetes.io/configuration-snippet: rewrite ^/app-name(/|$)(.\*) /$2 break;  
                                nginx.ingress.kubernetes.io/rewrite-target: /$2  
                                nginx.ingress.kubernetes.io/ssl-redirect: true  
Events:                         <none>

ex: kubectl describe ingress -n rtf ext-rtf-ingress-template

Name:             ext-rtf-ingress-template  
Labels:           <none>  
Namespace:        rtf  
Address:            
Default backend:  default-http-backend:80 (<error: endpoints "default-http-backend" not found>)  
TLS:  
  ext-mulesoft-tls-godaddy terminates ext.stg.api.aws.allegiantair.com  
Rules:  
  Host                              Path  Backends  
  ----                              ----  --------  
  ext.stg.api.aws.allegiantair.com    
                                    /app-name   service:80 (<error: endpoints "service" not found>)  
Annotations:                        external-dns.alpha.kubernetes.io/hostname: ext.stg.api.aws.allegiantair.com  
                                    nginx.ingress.kubernetes.io/configuration-snippet: rewrite ^/app-name(/|$)(.\*) /$2 break;  
                                    nginx.ingress.kubernetes.io/rewrite-target: /$2  
                                    nginx.ingress.kubernetes.io/ssl-redirect: true  
Events:                             <none>

Notice, in this case:

int-rtf-ingress-template is using 'mulesoft-tls-godaddy terminates stg.api.aws.allegiantair.com"

ext-rtf-ingress-template is using 'ext-mulesoft-tls-godaddy terminates ext.stg.api.aws.allegiantair.com'

**Backup the TLS secret you are going to update**

ex:

kubectl get secret mulesoft-tls-godaddy -n rtf -o yaml > mulesoft-tls-godaddy.backup.20230720

**Update the TLS secret**

To update mulesoft-tls-godaddy TLS secret you can just edit it in yaml format  
ex:  
kubectl edit secret -n rtf mulesoft-tls-godaddy -o yaml

You will need to update:

'tls.crt' value with the base64.one.line of the bundle you prepared earlier

'tls.key' value with the base64.one.line of the key you prepared earlier

suggested is to open up another terminal, cat each one and carefully copy the hash and paste it in the editor opened by the edit of the tls secret

Once you make sure you copy/paste correctly save the changes.

Then you can load in a web browser 'stg.api.aws.allegiantair.com' and make sure the new cert is loaded.

### Validation of the new SSL certificate

Check as an example (dependiong on the env adjust the link)  
<https://stg.api.aws.allegiantair.com/comms-exp-api/diag>