

HTTPS-terminating load balancers

In Cleura Cloud's load balancing service, [OpenStack Octavia](#), you can configure load balancers so that they manage HTTPS termination. That is to say that the load balancer encrypts and decrypts HTTPS traffic, and forwards HTTP to and from a backend web server.

To do so, the load balancer must have access to encryption credentials (such as certificates and private keys), which it stores in Barbican.

PKCS #12 Certificate Bundles

The [PKCS #12](#) archive format includes SSL certificates, certificate chains, and private keys all in one bundle. Most certificate providers give you the option of downloading certificate credentials using the PKCS #12 format.

In case your certificate provider has made your certificate chain and key available separately, using the PEM format, you can easily convert it to PKCS #12 using the following `openssl` command:

```
openssl pkcs12 -export -inkey key.pem -in fullchain.pem -out bundle.p12
```

When prompted for an export password, use a blank one.

Creating Barbican secrets from PKCS #12 bundles

To create a secret from a stored PKCS #12 bundle, you need pass in the contents of the bundle, *pre-encoded with [Base64](#)*, as the secret's payload.

```
$ openstack secret store \
  --name='tls_secret1' \
  -t 'application/octet-stream' \
  -e 'base64' \
  --payload="$(base64 < server.p12)"
```

Field	Value
Secret href	https://kna1.citycloud.com:9311/v1/secrets/69bd82f5-60c9-4764-99ec-7a3dff05d2aa
Name	tls_secret1
Created	None
Status	None
Content types	{'default': 'application/octet-stream'}
Algorithm	aes

Bit length	256	
Secret type	opaque	
Mode	cbc	
Expiration	None	
+-----+-----+-----+		

Creating HTTPS-enabled load balancer listeners

Once you have created your secret containing your certificate data, you can create a load balancer *listener* with the following properties:

- It uses the `TERMINATED_HTTPS` protocol,
- It sets its “default TLS container” to the Barbican secret containing the PKCS #12 bundle,
- It listens on the standard HTTPS port, 443.

You create such a listener with the following command:

```
$ openstack loadbalancer listener create \
  --protocol-port 443 \
  --protocol TERMINATED_HTTPS \
  --name listener1 \
  --default-tls-container-ref=https://kna1.citycloud.com:9311/v1/secrets/dacfbec1-fbed-403f-
a4dc-303e28942dae \
  <loadbalancer-name-or-id>
+-----+
+-----+
+
| Field          |
| Value          |
|               |
+-----+
+-----+
+
| admin_state_up |
| True           |
|               |
| connection_limit |
| -1             |
|               |
| created_at     |
| 2021-01-20T11:51:46 |
|               |
| default_pool_id |
| None           |
|               |
| default_tls_container_ref | https://kna1.citycloud.com:9311/v1/secrets/dacfbec1-fbed-403f-
a4dc-303e28942dae
|
| description    |
|               |
|               |
```

```
| id | 4ec6b23d-  
d08a-4de0-9e12-54ac690ee1ec  
|  
| insert_headers |  
None  
|  
| 17policies  
|  
|  
| loadbalancers | 2c2a0760-c3a8-48d2-  
bdd0-288c3d33a43f  
|  
| name |  
listener1  
|  
| operating_status |  
OFFLINE  
|  
| project_id |  
4a9484063d4c40d29301ad745c0e2c69  
|  
| protocol |  
TERMINATED_HTTPS  
|  
| protocol_port |  
443  
|  
| provisioning_status |  
PENDING_CREATE  
|  
| sni_container_refs |  
[]  
|  
| timeout_client_data |  
50000  
|  
| timeout_member_connect |  
5000  
|  
| timeout_member_data |  
50000  
|  
| timeout_tcp_inspect |  
0  
|  
| updated_at |  
None  
|  
| client_ca_tls_container_ref |  
None  
|  
| client_authentication |  
NONE  
|  
| client_crl_container_ref |  
None  
|
```

```
| allowed_cidrs      |
None
|
| tls_ciphers        |
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:
RSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-
SHA384:ECDHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-SHA256:DHE-RSA-AES128-
SHA256:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES128-SHA256 |
| tls_versions       |
|
+-----+
+-----+
+
```

Updating the TLS certificate for a HTTPS listener

When the certificate associated with a `TERMINATED_HTTPS` listener is about to expire, you will need to replace it. You can do this online, with no user-noticeable interruption to your service.

1. **Create a new PKCS#12 bundle** from the updated key, certificate, and CA certificate.
2. **Create a new Barbican secret** from the bundle.
3. List the listener(s) associated with your load balancer:

```
openstack loadbalancer listener list \
--loadbalancer <loadbalancer-name-or-id>
```

4. For all listeners using the `TERMINATED_HTTPS` protocol, run the following command:

```
openstack loadbalancer listener set \
--default-tls-container-ref=https://kna1.citycloud.com:9311/v1/secrets/e2d8acc1-
c6b9-4c01-9373-cc167b075c25 \
<listener-name-or-id>
```

Once all your load balancer listeners have completed the update, you may proceed to delete the old, now-unused secret:

```
openstack secret delete \
https://kna1.citycloud.com:9311/v1/secrets/dacfbec1-fbed-403f-a4dc-303e28942dae
```

Authors: Florian Haas