

Cloud for Education LBaaS cli

SSL terminated HTTPS LoadBalancer

Példa leírása:

- Adott egy publikus (esetünkben Smart1) hálózat, amelyből lebegő publikus IP címet allokalunk a loadbalancer-nek, valamint egy privát subnet.
- Adott két backend server 10.10.1.13 és 10.10.1.18 privát IP címekkel. A gépeken az apache2 hallgat a TCP 80-as porton.
- Szeretnénk egy SSL terminált loadbalancer-t, amely elérhető a külvilág felől és segítségével eloszthatók a webes kérések a backend (nem SSL) szerverek között.
- Adott egy előre igényelt certificate, valamint egy projekthez allokalált lebegő IP cím.

Példa megoldása:

Első lépésben hozzuk létre a terheléselosztót az alábbi módon, ahol a vip-subnet-id az a subnet, amelyben a backend gépek elhelyezkednek:

```
openstack loadbalancer create --name s3_edu_lb --vip-subnet-id b9bb709a-7249-42ad-96b7-c517204871af
```

A következő két parancs segítségével létrehozuk a szükséges pkcs12 tanúsítványt, valamint feltöltjük a secret store-ba (Barbican).

```
openssl pkcs12 -export -inkey s3_edu_hu.key -in s3_edu_hu_cert.cer -certfile sectigo_ca_ov.pem -passout pass: -out server.p12  
openstack secret store --name='s3_edu_hu_cert' -t 'application/octet-stream' -e 'base64' --payload="$(base64 < server.p12)"
```

Miután sikeresen feltöltöttük a tanúsítványt, következő lépésben látható parancs segítségével adjunk meg néhány nagyon fontos információt, többek között, hogy milyen protokollt szeretnénk használni, illetve adjuk meg az előzőleg feltöltött tanúsítványt.

```
openstack loadbalancer listener create --protocol-port 443 --protocol TERMINATED_HTTPS --name s3_edu_listener_https --default-tls-container=$(openstack secret list | awk '/ s3_edu_hu_cert
```

```
/ {print $2}') s3_edu_lb
```

Következő lépésként hozzuk létre a pool-t amelyben a backend szerverek fognak elhelyezkedni.

```
openstack loadbalancer pool create --name s3_edu_pool_https --lb-algorithm ROUND_ROBIN --  
listener s3_edu_listener_https --protocol HTTP
```

Sikeres pool létrehozás után, rendeljük a két backend szerveret az előző lépésben létrehozott pool-hoz.

```
openstack loadbalancer member create --subnet-id b9bb709a-7249-42ad-96b7-c517204871af --  
address 10.10.1.13 --protocol-port 80 s3_edu_pool_https  
openstack loadbalancer member create --subnet-id b9bb709a-7249-42ad-96b7-c517204871af --  
address 10.10.1.18 --protocol-port 80 s3_edu_pool_https
```

Végül rendeljük hozzá a korábban létrehozott publikus lebegő címet a terheléelosztóhoz.

```
openstack floating ip set --port f85476bd-dceb-44a0-b10b-652f5ca45ee9 ee72cb6c-aaac-442d-b062-  
bd575983f994
```

Probléma:

Szeretnénk a C4E-ben létrehozott Kubernetes cluster-ben futó konténerekben azonosítani a kliensek IP címét, a korábbi megoldás nem adta vissza megfelelően az IP címet.

A leírás a C4E-ben található LBaaS szolgáltatást részletezi.

Probléma megoldása:

Ennek beállításához cli használata szükséges. Az alábbi példában egy loadbalancer létrehozást következik:

Loadbalancer létrehozása

```
openstack loadbalancer create --name <name> --vip-subnet-id <subnet_id>
```

Floating IP csatolása

```
openstack floating ip set --port <load_balancer_vip_port_id> <floating_ip_id>
```

Listener létrehozása

```
openstack loadbalancer listener create --name <name> --protocol <proto> --protocol-port <port>
loadbalancer_id
```

Pool/poolok létrehozása

```
openstack loadbalancer pool create --name <name> --protocol PROXY --listener e4cca438-3fb7-
4134-a135-65830175e724 --lb-algorithm <algorithm>
```

Health monitor létrehozása

```
openstack loadbalancer healthmonitor create --type <type> --delay <delay> --timeout <timeout>
--max-retries <max-retries> --name <name> pool_id
```

#d# Members (nodeok) hozzáadása

```
openstack loadbalancer member create --subnet-id private-subnet --name <nodename> --address
address --protocol-port port pool_id
```

Változat #1

document-uploader hozta létre 2025-08-07 12:10:23 CEST

document-uploader frissítette 2025-08-07 12:10:23 CEST