

## Одновременная балансировка PAM Core и PAM Gateway

При одновременной балансировке сервер HAProxy будет принимать запросы и как веб-сервер на адреса core, и как RDS сервер.

Необходимо указать адрес HAProxy в файлах:

```
C:\inetpub\wwwroot\pam\mc\assets\config\config.prod.json
```

```
"core": {  
    "url": "https://haproxy.domain.local/pam/core"  
},  
"idp": {  
    "url": "https://haproxy.domain.local/pam/idp",  
    "requireHttps": true  
},
```

```
C:\inetpub\wwwroot\pam\uc\assets\config\config.prod.json
```

```
"core": {  
    "url": "https://haproxy.domain.local/pam/core"  
},  
"idp": {  
    "url": "https://haproxy.domain.local/pam/idp",  
    "requireHttps": true  
},
```

```
C:\Program Files\Indeed\Indeed PAM\Gateway\ProxyApp\appsettings.json
```

```
"Core": {  
    "Url": "https://haproxy.domain.local/pam/core"  
},  
"Auth": {  
    "IdpUrl": "https://haproxy.domain.local/pam/idp",  
},
```

```
C:\Program Files\Indeed\Indeed PAM\SSH Proxy\appsettings.json
```

```
"Settings": {  
    "CoreUrl": "https://haproxy.domain.local/pam/core",  
    "IdpUrl": "https://haproxy.domain.local/pam/idp",  
    ...  
}
```

Таким образом, в конфигурации HAProxy должны быть настроены отдельные frontend и backend для каждого сервиса.

Для двух PAM Core и двух PAM Gateway конфигурация HAProxy представлена ниже:

```
global
  log      /dev/log local2 debug
  chroot   /var/lib/haproxy
  pidfile  /var/run/haproxy.pid
  maxconn  4000
  user     haproxy
  group    haproxy
  daemon
  stats socket /var/lib/haproxy/stats
  stats timeout 30s
  ssl-default-bind-ciphers PROFILE=SYSTEM
  ssl-default-server-ciphers PROFILE=SYSTEM
  ssl-dh-param-file /etc/haproxy/dhparams.pem

defaults
  mode      tcp
  log       global
  option    httplog
  option    dontlognull
  option    redispatch
  balance   roundrobin
  retries   3
  timeout connect 10s
  timeout client 1h
  timeout server 1h

listen stats
  mode http
  bind *:8888 ssl crt /etc/haproxy/haproxydomainlocal.pem
  stats enable
  timeout client 5m
  timeout server 5m
  stats realm Haproxy\ Statistics
  stats uri /haproxy
  stats auth stat:stat
  stats hide-version
  stats refresh 3s

#-----
# main frontend which proxys to the backends
#-----
frontend frontend_pam
  mode http
  bind *:443 ssl crt /etc/haproxy/haproxydomainlocal.pem
  option forwardfor
  acl url_core path_beg /pam/core
  use_backend backend_api if url_core
  acl url_idp path_beg /pam/idp
  use_backend backend_idp if url_idp
  acl url_mc path_beg /pam/mc
  use_backend backend_mc if url_mc
  acl url_uc path_beg /pam/uc
  use_backend backend_uc if url_uc
```

```

frontend frontend_ssdp
    mode tcp
    bind *:22 # порт не должен совпадать с изначальным SSH портом данной Linux машины
    log global
    option tcplog
    default_backend backend_ssdp

frontend frontend_gw
    mode tcp
    bind *:3389
    log global
    option tcplog
    tcp-request inspect-delay 2000
    tcp-request content accept if RDP_COOKIE
    default_backend backend_gw

#-----
# balancing between the various backends
#-----

backend backend_ssdp
    mode tcp
    balance leastconn
    option tcp-check
    log global
    tcp-check connect port 22
    timeout server 30m
    timeout connect 5000
    server gw1 gw1.domain.local:22 weight 10 check verify required ca-file /path/ca-cert.crt
    server gw2 gw2.domain.local:22 weight 10 check verify required ca-file /path/ca-cert.crt

backend backend_api
    mode http
    balance source
    option prefer-last-server
    option httpchk GET /pam/core/health
    server srv1 srv1.domain.local:443 ssl verify required ca-file /path/ca-cert.crt check inter 3000 fall 3
    server srv2 srv2.domain.local:443 ssl verify required ca-file /path/ca-cert.crt check inter 3000 fall 3

backend backend_idp
    mode http
    balance source
    option prefer-last-server
    option httpchk GET /pam/idp
    server srv1 srv1.domain.local:443 ssl verify required ca-file /path/ca-cert.crt check inter 3000 fall 3
    server srv2 srv2.domain.local:443 ssl verify required ca-file /path/ca-cert.crt check inter 3000 fall 3

backend backend_mc
    mode http
    balance source
    option prefer-last-server
    option httpchk GET /pam/mc
    server srv1 srv1.domain.local:443 ssl verify required ca-file /path/ca-cert.crt check inter 3000 fall 3
    server srv2 srv2.domain.local:443 ssl verify required ca-file /path/ca-cert.crt check inter 3000 fall 3

```

```
backend backend_uc
  mode http
    balance source
  option prefer-last-server
  option httpchk GET /pam/uc
  server srv1 srv1.domain.local:443 ssl verify required ca-file /path/ca-cert.crt check inter 3000 fall 3
  server srv2 srv2.domain.local:443 ssl verify required ca-file /path/ca-cert.crt check inter 3000 fall 3

backend backend_gw
  mode tcp
  balance leastconn
  option tcp-check
  log global
  tcp-check connect port 3389
  stick-table type ip size 1m expire 12h
  stick on src
  default-server inter 3000 rise 2 fall 3
  server gw1 gw1.domain.local:3389 weight 10 check verify required ca-file /path/ca-cert.crt
  server gw2 gw2.domain.local:3389 weight 10 check verify required ca-file /path/ca-cert.crt
```