Enabling CORS in HAProxy

- Login to HAProxy VM
- set http-headers in frontend configurations part of `/etc/haproxy/haproxy.cfg` config file

```
http-response set-header Access-Control-Allow-Origin "*"
capture request header origin len 128
http-request use-service lua.cors-response if METH_OPTIONS { capture.
req.hdr(0) -m reg -f /usr/local/etc/haproxy/cors-origins.lst }
```

- Create a new `/etc/haproxy/cors.lua` file, add the below mentioned content into this file

```
core.register_service("cors-response", "http", function(applet)
    applet:set_status(200)
    applet:add_header("Content-Length", "0")
    applet:add_header("Access-Control-Allow-Credentials", "true")
    applet:add_header("Access-Control-Allow-Origin", applet.headers
["origin"][0])
    applet:add_header("Access-Control-Allow-Headers", "*")
    applet:add_header("Access-Control-Allow-Methods", "GET, HEAD,
POST, PUT, DELETE, PATCH, OPTIONS")
    applet:add_header("Access-Control-Max-Age", "1728000")
    applet:start_response()
end)
```

- Create a new cors-origin list file `/etc/haproxy/cors-origins.lst` and add the cors origin sites in this file

```
[manjunath@argoid-saas-prod1-host-072 ~]$ cat /etc/haproxy/cors-
origins.lst
odette.in
[manjunath@argoid-saas-prod1-host-072 ~]$
```

- Whole `haproxy.cfg` file should similar to below mentioned file
  only line-number 25,26,27 are with CORs configuration

```
global
   maxconn 50000
   tune.ssl.default-dh-param 2048
   user haproxy
   group haproxy
   daemon
   log /var/log/haproxy/log  local0 info
   lua-load /usr/local/etc/haproxy/cors.lua
```

```
defaults
   log global
   mode http
   option httplog
   option dontlognull
   timeout connect 5000
   timeout client 50000
   timeout server 50000


frontend https_front
   bind *:443 ssl crt /etc/ssl/prod.odette.saas.argoid.com.pem
   mode http
   stats enable
   stats uri /haproxy?stats
   http-response set-header Access-Control-Allow-Origin "*"
   capture request header origin len 128
   http-request use-service lua.cors-response if METH_OPTIONS {
capture.req.hdr(0) -m reg -f /usr/local/etc/haproxy/cors-origins.lst }


   acl nifi_events_api    path_beg /v1/events
   use_backend nifi_events if nifi_events_api

   acl v2_recommendations_api path_beg /odette/v2/recommendations
   use_backend v2_recommendations if v2_recommendations_api

   acl v2_is_reco_served_by_argoid_api path_beg /odette/v2/IS-RECO-
SERVED-BY-ARGOID
   use_backend  v2_is_reco_served_by_argoid if
v2_is_reco_served_by_argoid_api


backend nifi_events
   server  nifi_1   10.0.0.9:6201   check

backend v2_recommendations
   balance leastconn
   server server_01 10.0.0.71:8088  check
   server server_02 10.0.0.73:8088  check

backend v2_is_reco_served_by_argoid
   balance leastconn
   server server_01 10.0.0.71:8088  check
   server server_02 10.0.0.73:8088  check

frontend http2_front
   bind *:9093
   mode http
   stats enable
   stats uri /haproxy?stats
   default_backend http2_back

backend http2_back
   balance leastconn
   server server_02 192.0.1.38:8082  check
```

- Restart HAProxy Service