

Distil Networks & HAProxy Integration Guide

Table of Contents

INTRODUCTION	3
HAPROXY	3
Testing Rule - HTTP Header-Based	3
Onboarding Rule - Percentage Allocation	5
Production Rule - Full Allocation	6
Health Checking	7
SSL Offloading	8

Introduction

The objective for implementing the following rules is to send traffic to Distil if it hasn't already done so. Placement before any other logic is important so as to avoid a situation where some domain traffic doesn't pass through the Distil instances.

For more information on the load balancer (LB) integration methodology, please see the [Distil Networks Load Balancer Integration Guide](#).

HAProxy

HAProxy uses access control lists (ACLs) to control traffic flow. The following configurations show how to send test traffic through Distil as well as perform a percentage-based onboarding. This guide also provides a full production ruleset.

The provided configuration examples are for HAProxy version 1.5. Additional configuration might be required for a different version.

TESTING RULE - HTTP HEADER-BASED

This rule is very similar to the production rule, with the exception that the `acl distil_qa_header` is added. Its presence routes traffic only to Distil instances and allows for testing site functionality.

Normal users should not supply this header and will not be proxied through Distil.

```
#HAProxy Frontend
frontend http-frontend
  bind *:80
  mode http

  #Distil ACL
  acl distil_src      src      192.168.181.4
  acl distil_qa_header      hdr(distilqa) -m found
  acl distil_not_enough_capacity nbsrv(distil_cluster) gt 0

  use_backend      distil_cluster  if  !distil_src distil_qa_header distil_not_enough_
capacity
  use_backend      nginx_cluster

#HAProxy Backend
backend nginx_cluster
  server nginx01      192.168.181.28:80

backend distil_cluster
  option httpchk GET /distil_ok.html
  http-check expect status 200
  server distilserver1 192.168.181.4:80 check
```

ONBOARDING RULE - PERCENTAGE ALLOCATION

Use this rule if it's necessary to only route a percentage of traffic through Distil when initially onboarding.

```
#HAProxy Frontend
frontend http-frontend
  bind *:80
  mode http

  #Distil ACL
  acl distil_src src 192.168.181.4
  acl distil_rand_number rand(100) lt 25
  acl distil_js_prefix path_beg /dstl /distil
  acl distil_not_enough_capacity nbsrv(distil_cluster) gt 0

  use_backend distil_cluster if !distil_src distil_rand_number distil_not_enough_
  capacity || distil_js_prefix
  use_backend nginx_cluster

#HAProxy Backend
backend nginx_cluster
  server nginx01 192.168.181.28:80

backend distil_cluster
  option httpchk GET /distil_ok.html
  http-check expect status 200
  server distilserver1 192.168.181.4:80 check
```

Three ACLs are evaluated:

- distil_src
- distil_rand_number
- distil_js_prefix

The ACL `distil_src` header routes traffic to Distil only when traffic has not already passed through our service.

ACL `distil_rand_number` generates a random number between 0 and 100; it evaluates as TRUE when the value is less than 25.

ACL `distil_js_prefix` is required for percentage-based onboarding. HA Proxy will send traffic to Distil when requests are made which match the configured Distil JavaScript prefix.

PRODUCTION RULE - FULL ALLOCATION

Use the following configuration on the existing VIP that load balances traffic to the origin server. This ACL looks for the IP of the Distil instances. If the IP is not from Distil, then the request is sent to our service. If the request is from Distil, then logic flows as normal.

```
#HAProxy Frontend
frontend http-frontend
  bind *:80
  mode http

  #Distil ACL
  acl distil_src src 192.168.181.4 10.50.5.5
  acl distil_not_enough_capacity nbsrv(distil_cluster) gt 0

  use_backend distil_cluster if !distil_src distil_not_enough_capacity
  use_backend nginx_cluster

#HAProxy Backend
backend nginx_cluster
  server nginx01 192.168.181.28:80

backend distil_cluster
  option httpchk GET /distil_ok.html
  http-check expect status 200
  server distilserver1 192.168.181.4:80 check
```

HEALTH CHECKING

Use the following configuration snippet for health checking your Distil instances. The health check looks for the `/distil_ok.html` page and expects HTTP status code 200 in response.

```
option httpchk GET /distil_ok.html
http-check expect status 200
```

SSL OFFLOADING

When offloading SSL at the HAProxy, the latter can be configured to add a specific HTTP header/value combination to indicate that the request was received encrypted. Use the **X-Forwarded-Proto** header or customize it in HAProxy.

Use the following rule to set the **X-Forwarded-Proto** header value to **HTTPS** at an SSL-offloading VIP.

```
reqadd X-Forwarded-Proto:\ https
```

When Distil receives inbound SSL traffic and proxies it to the origin as SSL, by default the headers will also be proxied. If present, Distil removes the offload header to avoid a redirection loop or HTTP header spoofing.