

Running Jenkins behind HAProxy

In situations where you want a user friendly URL, different public ports, or to terminate SSL connections before they reach Jenkins, you may find it useful to run Jenkins (or the servlet container that Jenkins runs in) behind HAProxy. This document discusses some of the approaches for doing this.

Plain HTTP

Using HAProxy 1.7.9, here is an example HAProxy.cfg to proxy over plain HTTP:

```
# If you already have an haproxy.cfg file, you can probably leave the
# global and defaults section as-is, but you might need to increase the
# timeouts so that long-running CLI commands will work.
global
    maxconn 4096
    log 127.0.0.1 local0 debug

defaults
    log global
    option httplog
    option dontlognull
    option forwardfor
    maxconn 20
    timeout connect 5s
    timeout client 60s
    timeout server 60s

frontend http-in
    bind *:80
    mode http
    acl prefixed-with-jenkins path_beg /jenkins/
    acl host-is-jenkins-example hdr(host) eq jenkins.example.com
    use_backend jenkins if host-is-jenkins-example prefixed-with-jenkins

backend jenkins
    server jenkins1 127.0.0.1:8080
    mode http
    reqrep ^([\ :]*)\ /(.*?) \1\ /\2
    acl response-is-redirect res.hdr(Location) -m found
    rspirep ^Location:\ (http|https)://127.0.0.1:8080/jenkins/(.*) Location:\ \1://jenkins.example.com/jenkins
    /\2 if response-is-redirect
```



This assumes Jenkins is running locally on port 8080.



This assumes that you are using the /jenkins/ context path for both the site exposed from HAProxy, and Jenkins itself. If this is not the case, you will need to adjust the configuration.



If you are experiencing the following error when attempting to run long CLI commands in Jenkins > 2.80, and Jenkins is running behind HAProxy, it is probably due to HAProxy timing out the CLI connection. You can increase the `timeout client` and `timeout server` settings as necessary so the command will complete successfully. Versions of Jenkins before 2.80 might not respect the `proxy_read_timeout` setting.

```
WARNING: null
hudson.cli.DiagnosedStreamCorruptionException
Read back: 0x00 0x00 0x00 0x1e 0x07 'Started reverse-proxy-test #68' 0x00 0x00 0x00 0x01 0x07 0x0a
Read ahead:
Diagnosis problem:
  java.io.IOException: Premature EOF
    at sun.net.www.http.ChunkedInputStream.readAheadBlocking(ChunkedInputStream.java:565)
    ...
    at hudson.cli.FlightRecorderInputStream.analyzeCrash(FlightRecorderInputStream.java:82)
    at hudson.cli.PlainCLIProtocol$EitherSide$Reader.run(PlainCLIProtocol.java:153)
Caused by: java.io.IOException: Premature EOF
    at sun.net.www.http.ChunkedInputStream.readAheadBlocking(ChunkedInputStream.java:565)
    ...
    at java.io.DataInputStream.readInt(DataInputStream.java:387)
    at hudson.cli.PlainCLIProtocol$EitherSide$Reader.run(PlainCLIProtocol.java:111)
```

With SSL

Using HAProxy 1.7.9, here is an example HAProxy.cfg to connect to the proxy using SSL, terminate the SSL connection, and then talk to Jenkins using plain HTTP:

```
# If you already have an haproxy.cfg file, you can probably leave the
# global and defaults section as-is, but you might need to increase the
# timeouts so that long-running CLI commands will work.
global
    maxconn 4096
    log 127.0.0.1 local0 debug

defaults
    log global
    option httplog
    option dontlognull
    option forwardfor
    maxconn 20
    timeout connect 5s
    timeout client 5min
    timeout server 5min

frontend http-in
    bind *:80
    bind *:443 ssl crt /usr/local/etc/haproxy/ssl/server.pem
    mode http
    redirect scheme https if !{ ssl_fc } # Redirect http requests to https
    use_backend jenkins if { path_beg /jenkins/ }

backend jenkins
    server jenkins1 127.0.0.1:8080
    mode http
    http-request set-header X-Forwarded-Port %[dst_port]
    http-request add-header X-Forwarded-Proto https if { ssl_fc }
    reqrep ^([\^ :]*)\ /(.*?) \1\ /\2
    acl response-is-redirect res.hdr(Location) -m found
    rspirep ^Location:\ (http)://127.0.0.1:8080/(.*?) Location:\ https://jenkins.example.com:443/\2 if
    response-is-redirect
```