Runtime Infrastructure - Feature #5170

add support for cloud-based load balancing and WAF

03/02/2021 10:11 AM - Greg Shah

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**vendor_id:**	GCD

Description

Related issues:

Related to Database - Feature #4369: implement stateless FWD server clustering	**New**
Related to User Interface - Feature #2683: reverse proxy implementation	**Closed**
Related to Runtime Infrastructure - Feature #4406: server-side REST execution...	**New**
Related to Runtime Infrastructure - Feature #5184: enable non-SSL mode for Jetty	**New**

History

#1 - 03/02/2021 10:11 AM - Greg Shah

- Related to Feature #4369: implement stateless FWD server clustering added

#2 - 03/02/2021 10:12 AM - Greg Shah

- Related to Feature #2683: reverse proxy implementation added

#3 - 03/02/2021 10:17 AM - Greg Shah

We need to enable FWD to be used behind a load balancer and WAF. The initial target is AWS but we really should implement this generically.

This is highly related to our reverse proxy support and the success of using this approach will also be dependent upon completing #4369. Our objective is to ensure that server-based applications (such as web services like REST or SOAP) can be supported in this manner.

I think load balancing is fundamentally incompatible with stateful UI code so that is not in scope.

#4 - 03/02/2021 10:18 AM - Greg Shah

- Related to Feature #4406: server-side REST execution without appserver agents added

#5 - 03/06/2021 01:29 PM - Greg Shah

- Related to Feature #5184: enable non-SSL mode for Jetty added

#6 - 03/06/2021 01:35 PM - Greg Shah

Sergey: From your perspective (as the person that added reverse proxy support for the web client), what is needed for this task?

#7 - 03/08/2021 02:45 AM - Sergey Ivanovskiy

#2683 only implemented the custom reverse proxy support for the web clients with help of the Apache web server running at the frontend of the FWD java web clients. This solution was due to known issues http://jetty.4.x6.nabble.com/Reverse-proxy-with-WebSocket-support-td4966317.html, https://www.eclipse.org/lists/jetty-users/msg07621.html. It seems that we have no environment on which we can test this approach fully except I just tested this configuration with my local environment. #2683 requires the environment with the Apache web server. I didn't test this solution when there are two or more identical Apache web servers running in AWS. Is #4406 ready for implementation? The web java client has the special configuration settings that defines internal range of IP4 addresses for new launched web clients. It looks that this solution is not ready for REST or SOAP services.

#8 - 03/08/2021 07:34 AM - Greg Shah

> I didn't test this solution when there are two or more identical Apache web servers running in AWS.

I don't know that we care about multiple proxy/WAF in front of us. I think that any redundancy there will be transparent (it won't be visible to us). This task is really about what we need to do at a protocol level to enable this to be possible.

As noted in our Reverse Proxy documentation:

> FWD supports reverse proxy usage with its web clients. At this time it does not support this usage with appserver clients.

Is this in reference only to callers accessing the FWD server via RemoteObject? Or are there things that we need to do to resolve access for web services?

> Is #4406 ready for implementation?

We have not worked it yet. But this task is not strictly needing it. #4406 is about eliminating the need to switch contexts into an appserver agent before executing the web service. It doesn't change how Jetty handles the protocol flows with the external caller.

#9 - 03/08/2021 11:11 AM - Sergey Ivanovskiy

If WebHandler handles and delegates all web requests, then it needs to add the corresponding server host to the Apache reverse proxy configuration by mapping its web root to the corresponding server host. In the following example it is supposed that the FWD server is running on the same host as the Apache reverse proxy server and this web root /gui is mapped to the https://localhost:7443/gui

```
ProxyPass /gui  https://localhost:7443/gui
ProxyPassReverse /gui https://localhost:7443/gui
```