Docker & mod_proxy

Rui Couto António Nestor Ribeiro

May 14, 2023

Contents

1 Apache mod_proxy

The `mod_proxy` is a module for the Apache web server. Thus, the first step is to install apache in the host machine. By default, the macOS distributions already have apache. In unix machines, apache exists in the repositories so it can be installed with the usual package manager (e.g. `apt-get install apache2`). For windows, packages as WAMP[1] ease the process of installing apache.

1.1 Enable mod_proxy

The module `mod_proxy` is included by default with Apache. However, it needs to be loaded prior to its usage. Such is done in the apache configuration file. In the configuration file:

- macOs: `/etc/apache2/httpd.conf`
- Ubuntu: `/etc/apache2/apache2.conf`
- Windows: `c:wampconfhttpd.conf`

Uncomment the following lines:

```
1 LoadModule proxy_module modules/mod_proxy.so
2 LoadModule proxy_http_module modules/mod_proxy_http.so
3 LoadModule proxy_balancer_module libexec/apache2/mod_proxy_balancer.so
4 LoadModule lbmethod_byrequests_module libexec/apache2/mod_lbmethod_byrequests.so
```

[1] `http://www.wampserver.com/en/`

After restarting apache, it will load the new configurations, and `mod_proxy` will be enabled.

- macOs: `sudo apachectl restart`
- Ubuntu: `sudo ./etc/init.d/apache2 restart`
- Windows: Via GUI.

Tasks

1. Setup apache.
2. Enable `mod_proxy`, and restart apache.

1.2 Balancer setup

With `mod_proxy` enabled, it is possible to configure the load balancer. The first step, is to forward the requests to the appropriate local url. That can be achieved with the `Proxy` directives:

```
ProxyPass / balancer://mycluster/ stickysession=JSESSIONID|jsessionid
ProxyPassReverse / balancer://tutcluster/
ProxyPassReverse / http://localhost:8080/
ProxyPassReverse / http://localhost:8000/
ProxyPreserveHost On
ProxyRequests Off
```

These settings will forward requests on `/` to be redirected to the balancer `mycluster`. The JSESSIONID will be preserved. Also, it is assumed that two instances of the application are running, in ports `8080` and `8000`.

Next, the permissions for `/` need to be set.

```
<Location / >
  Order deny,allow
  Allow from All
</Location>
```

These settings will allow incoming and outcoming connections for `/`. Finally, the balancer can be configured.

```
<Proxy balancer://mycluster/>
  BalancerMember http://localhost:8080 route=node1 retry=60
  BalancerMember http://localhost:8000 route=node2 retry=60
</Proxy>
```

These configurations will define two balancer members on the `mycluster` balancer. The final set of configurations is as follows:

```
ProxyPass / balancer://mycluster/ stickysession=JSESSIONID|jsessionid
ProxyPassReverse / balancer://tutcluster/
ProxyPassReverse / http://localhost:8080/
ProxyPassReverse / http://localhost:8000/
ProxyPreserveHost On
ProxyRequests Off

<Location / >
  Order deny,allow
  Allow from All
</Location>

<Proxy balancer://mycluster/>
  BalancerMember http://localhost:8080 route=node1 retry=60
  BalancerMember http://localhost:8000 route=node2 retry=60
</Proxy>
```

These configurations should be put in the configuration configuration file, before `Include /private/etc/apache2/other/*.conf`.

Tasks

1. Setup the load balancer according to your settings.
2. Open the `http://localhost/` in order to verify that the configuration is correct.

1.3 Setup wildfly

The final step is to define in the application server to which node it corresponds. This must be done for each deployment. In the wildfly installation folder, locate the `standalone.xml` file, under `standalone/configuration`. Add the following settings, between `</extensions>` and `<management>`:

```
<system-properties>
  <property name="jvmRoute" value="node1"/>
</system-properties>
```

Adjust the node value (e.g. `node1`), according to your configurations.

Tasks

1. Perform the presented steps to configure `mod_proxy` with wildfly.
2. Create two docker images with the Java applications, and configure Apache to distribute load between them (e.g. 70%, 30%).