



cockpit
IT Service Manager

Load balancing configuration

Technical specification

Table of contents

Introduction.....	3
I. Overview.....	3
II. The Apache load balancer.....	3
III. Limitations.....	3
Prerequisites.....	4
Configuration.....	5
I. Portal configuration.....	6
II. Restart.....	6
III. Test.....	7

Introduction

This document explains how to setup two or more Cockpit IT Service Manager Portals in a load balancing setup.

I. Overview

The sample configuration was tested using a default installation on Ubuntu Server LTS and the load balancing module included in the default Apache 2 package provided with the OS. Other Linux flavours should work as well but the configuration file locations and format may vary from this documentation.

Load balancing is set up for the “V2” part of the software only: The legacy (PHP) portion is served by the main apache instance without load balancing.

II. The Apache load balancer

Documentation of the load balancer can be found here:

http://httpd.apache.org/docs/current/mod/mod_proxy_balancer.html

The load balancer supports three load balancing scheduler methods:

weighted request counting (default method)

weighted traffic byte count balancing

pending request balancing

Several methods for load balancer stickyness are supported. In our example, we use the JSESSIONID cookie to identify the user sessions and route each request to the corresponding node.

III. Limitations

The only known limitation is that the user's IP addresses will not be logged correctly in the connection log provided by the application. If IP logging is needed, it needs to be set up manually on the Apache server handling the load balancing.

Prerequisites

A default installation of the Cockpit IT Service Manager portal on a Linux server running Ubuntu Server. Other Linux flavours should work as well but the configuration file locations and format may vary from this documentation.

A second Cockpit IT Service Manager portal installed either locally or on a different machine. For our tests, one portal was installed locally and a second one on a second machine.

Note that these requirements are for this sample configuration only: The portal does not need to be set up on the same machine as the load balancer – any setup is possible with minor modifications to the configuration.

Configuration

Starting from default installation of Cockpit IT Service Manager and a secondary portal installed on a different machine, the following configuration changes will enable load balancing between the two:

I. Apache modules

First of all, make sure the following Apache modules are enabled. You can use the following commands to do so on Ubuntu Linux:

```
sudo ln -s /etc/apache2/mods-available/proxy_balancer.conf /etc/apache2/mods-enabled/
```

```
sudo ln -s /etc/apache2/mods-available/proxy_balancer.load /etc/apache2/mods-enabled/
```

II. Apache configuration

The default virtual host configuration created in a standard EXP installation is as follows:

```
<VirtualHost *:80>
  ServerName 127.0.0.1
  <Proxy *>
    AddDefaultCharset Off
    Order deny,allow
    Allow from all
  </Proxy>
  ProxyRequests Off
  ProxyPass /legacy http://127.0.0.1:8080
  ProxyPassReverse /legacy http://127.0.0.1:8080
  ProxyPass /ajp://127.0.0.1:8009/
  ProxyPassReverse /ajp://127.0.0.1:8009/
</VirtualHost>

<VirtualHost 127.0.0.1:8080>
  ServerName 127.0.0.1
  DocumentRoot "/home/koaly/exp/portal1/web"
  <Directory "/home/koaly/exp/portal1/web">
    Options -Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
  </Directory>
</VirtualHost>
```

To enable load balancing, we need to make the following changes:

```
<VirtualHost *:80>
  ServerName 127.0.0.1
  <Proxy *>
    AddDefaultCharset Off
    Order deny,allow
    Allow from all
  </Proxy>
  <Proxy balancer://expcluster>
    BalancerMember ajp://127.0.0.1:8009/ route=portal0
```

```
BalancerMember ajp://{IP address of secondary server}:8009/ route=portal1
ProxySet stickysession=JSESSIONID
</Proxy>
ProxyRequests Off
ProxyPass /legacy http://127.0.0.1:8080
ProxyPassReverse /legacy http://127.0.0.1:8080
ProxyPass / balancer://expcluster/
ProxyPassReverse / balancer://expcluster/
</VirtualHost>

<VirtualHost 127.0.0.1:8080>
  ServerName 127.0.0.1
  DocumentRoot "/home/koaly/exp/portal1/web"
  <Directory "/home/koaly/exp/portal1/web">
    Options -Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
  </Directory>
</VirtualHost>
```

I. Portal configuration

The following changes need to be applied to the file `/home/koaly/exp/portal/conf/server.xml` for each portal that is a member of the cluster.

The following lines:

```
<!-- You should set jvmRoute to support load-balancing via AJP ie :
<Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
-->
<Engine defaultHost="localhost" name="Catalina">
```

need to be modified as follows on the first node:

```
<!-- You should set jvmRoute to support load-balancing via AJP ie :
<Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
-->
<Engine defaultHost="localhost" name="Catalina" jvmRoute="portal0">
```

On the second node, “portal0” needs to be replaced by “portal1”, of course.

II. Restart

To apply the configuration, the application needs to be restarted, starting with the main node (portal0):

```
sudo service koaly-exp-portal stop
sudo service koaly-exp-portal start
sudo service apache2 restart
```

Once the main portal is up, the secondary portal(s) need to be restarted as well.

III. Test

To test the setup, we need to increase the log level of the portals to TRACE temporarily (for the duration of the test only). To do so, we need to edit the logging configuration file:

```
sudo vim /home/koaly/exp/portal/conf/log4j.xml
```

To change the log level, we need to change the lines

```
<!--      -->
<!-- Logger configuration -->
<!--      -->
<logger name="com.koaly">
  <level value="info"/>
</logger>
```

to

```
<!--      -->
<!-- Logger configuration -->
<!--      -->
<logger name="com.koaly">
  <level value="trace"/>
</logger>
```

To activate this configuration change, the portal needs to be restarted:

```
sudo service koaly-exp-portal stop
sudo service koaly-exp-portal start
```

Now, we can follow the logging output using the “tail” command:

```
sudo tail -f /home/koaly/exp/portal/logs/koaly_exp.log
```

After doing the same for the secondary portal(s), we need to connect to the portal. The following log should appear on one of the log files:

```
TRACE com.koaly.core.service.impl.SessionServiceImpl - Session opened for user operator2 connecting from {IP address}.
```

Now, connect to the portal from a different browser session. This is necessary as the load balancer will identify browser sessions by the session ID and redirect all requests belonging to the same session to the same node. Connecting from a different session should result in the same log on the secondary node.

If the connection notification appears in the log file of all nodes, your load balancing setup is working.

IMPORTANT: After completing the tests, the log level needs to be reset to “info” and the portals need to be restarted.

Document end