

Apache 及 Tomcat 的結合

網頁應用伺服器上，Apache 為眾多伺服器的首選，對於 JSP 的服務來說當然是以 Tomcat 為主要的選擇，在目前的主流上若要使用 JSP 開發 Web 網頁程式的話，Apache 和 Tomcat 當然是最佳選擇，在這裡就由小弟介紹如何使用 Apache 配合 connector 來連結 Tomcat。

首先在一個重要的 Web 服務平台上，若打算要把 Apache 和 JSP 分開以分化流量並保護 JSP 程式的話，一般都會把 JSP 放在另一台跑 Tomcat 的主機上，而靜態的 html 則是放在 Apache 上。JSP 程式主機若以安全原則來說是不應該放在別人看得到的公開網路上，最好是放在一個 Web Cluster 的內部環境裡面，以下是這次範列的 Web Server 加構圖。

我們的環境下，使用者連上我們的 http-fountain，而 java-garden 是放在內部的網路裡，Web 程式雖然是放在 java-garden 裡，但使用卻感覺不出來。

```
+-----+
| User   |
+-----+
      | (Port 80)
+-----+
| http   | (http-fountain)
+-----+
      | (Port 8009)
+-----+
| JSP    | (java-garden, 192.168.1.87)
+-----+
```

這樣的設計有幾點好處：程式和網頁分開減少檔案損壞的風險、JSP 應用伺服器較不會被外界攻擊、可單一維護程式或網頁、若日後流量大的話 Tomcat 可以做負載平衡。

以下我們就來看看如何建立這一個完整的環境。

設定 DNS：

DNS 應該不用說了吧，一定要先對完整才可以用得當，請使用 host 工具來看看是否設定正確。在這個 Web Cluster 上內部應該要查得到 http-fountain 和 java-garden。

```
root # host http-fountain.1-penguin.idv.tw
http-fountain.1-penguin.idv.tw has address 192.168.1.86
root # host java-garden.1-penguin.idv.tw
java-garden.1-penguin.idv.tw has address 192.168.1.87
```

而外部的使用者應該只能查到 http-fountain 的公開 IP

```
root # host http-fountain.1-penguin.idv.tw
http-fountain.1-penguin.idv.tw has address 11.22.33.44
```

因為有內外部，所以你有可能需要使用到 Bind 9 的 View 功能，至於詳細的使用方法請參考 [bind - DNS 設定 \(New window\)](#)。

建立一個美好泡咖啡的花園 java-garden.l-penguin.idv.tw :

設定 Java 環境

首先你的機器應該有有 Java 環境這樣 Tomcat 才可以順利執行。你可以到 Sun 的 下載 區 去 下載 最近 版本 的 Java JDK (<http://java.sun.com/javase/downloads/index.jsp>) 並且安裝它。在這裡我是下載可自解的二進制可執行檔 (Linux self-extracting file)。

最後你可以發現下載後會是一個 .bin 檔案,你可以附與它一個 x 可執行權限然後開始執行。

```
root # chmod +x jdk-1_5_0_07-linux-i586.bin
root # ./jdk-1_5_0_07-linux-i586.bin
```

解開之後會多出一個目錄,我習慣把應用程式搬到 /usr/local 之下,而這個 Java 也不例外。

```
root # mv jdk1.5.0_07/ /usr/local/java
```

當然為了 Tomcat 的安裝與執行,你並需設定 JAVA_HOME 這個環境變數,你可以設定在 /etc/profile 上。

```
root # echo 'JAVA_HOME=/usr/local/java' >> /etc/profile
```

如果你要馬上生效的話,執行 export 把這個變數轉成環境變數。

```
root # export JAVA_HOME=/usr/local/java
root # env | grep JAVA_HOME
JAVA_HOME=/usr/local/java
root #
```

設定 Tomcat 環境:

現在是重頭戲 Tomcat 的表演時刻了,你可以到 <http://tomcat.apache.org/> 下載最新版本的 Tomcat 來安裝,當然我是下載 Tomcat 5.5 版本的 Tomcat 二進制可執行檔。解開之後移到 /usr/local/tomcat5 目錄。

```
root # wget
http://apache.cdpa.nsysu.edu.tw/tomcat/tomcat-5/v5.5.17/bin/apache-to
mcat-5.5.17.tar.gz
root # tar -zxvf apache-tomcat-5.5.17.tar.gz
root # mv apache-tomcat-5.5.17 /usr/local/tomcat5
```

嚐試啟動 Tomcat 並看看它的樣子!

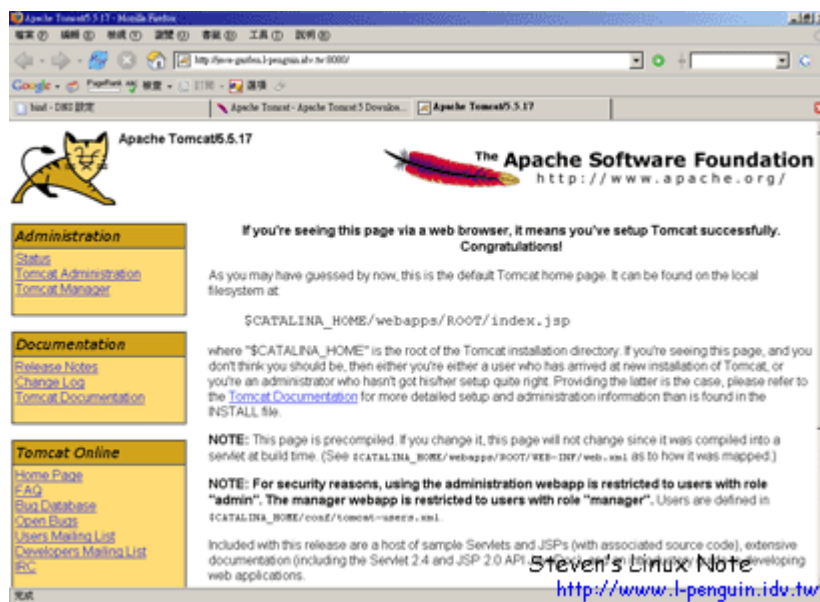
現在你已經把 Tomcat 裝好,並且擁有 JAVA 環境了,你可以試著啟動 Tomcat 來看看。

```
root # /usr/local/tomcat5/bin/startup.sh
Using CATALINA_BASE: /usr/local/tomcat5
Using CATALINA_HOME: /usr/local/tomcat5
Using CATALINA_TMPDIR: /usr/local/tomcat5/temp
Using JRE_HOME: /usr/local/java
root #
```

好的，你可以看看 Tomcat 是使用那些 Port。

```
root # netstat -ntulp | grep java
tcp 0 0 :::ffff:127.0.0.1:8005 :::* LISTEN 2157/java
tcp 0 0 :::8009 :::* LISTEN 2157/java
tcp 0 0 :::8080 :::* LISTEN 2157/java
root #
```

由 netstat 可得知它是使用了 8080 和 8009，這個 8009 等一下是要由 mod_jk2 來使用的，而 8080 是 Tomcat 的 Web Service 服務，你可以在 Web Browser 輸入 <http://java-garden.l-penguin.idv.tw:8080/> 看看。



當然這個 java-garden 只有在內部才可看到，使用者是看不到的，若使用者看得到的話就失去了我故意安排 java-garden 在內部網的安全心意了。以上就完成了 Tomcat 的設置，下面再來看看如何設定 Apache 並使用 mod_jk2 這個 connector 來連結 Apache 和 Tomcat。

建設 http 噴水池 <http://fountain.l-penguin.idv.tw>：

建立基本的 Apache

首先你要設定安裝基本的 Apache，安裝 Apache 你可以參考 LAMP 文章有詳細的安裝過程，並不需要有額外的安裝選項。

編譯 mod_jk2 模組

當你安裝完成 Apache 之後，還需要編譯 mod_jk2 這個連結器才可以讓 Apache 和 Tomcat 相互溝通，以下我示範使用 mod_jk2 原始檔來編譯，你可以到 <http://tomcat.apache.org/download-connectors.cgi> 下載得到。

```
root # wget
http://apache.cdpa.nsysu.edu.tw/tomcat/tomcat-connectors/jk2/jakarta-
tomcat-connectors-jk2-src-current.tar.gz
root # tar -zxvf jakarta-tomcat-connectors-jk2-src-current.tar.gz
```

解開之後你會在同目錄之下得到一個 jakarta-tomcat-connectors-jk2-2.0.4-src 的目錄，接下來請依如下的順序來編譯。

```
root # cd jakarta-tomcat-connectors-jk2-2.0.4-src/
root # cd jk/native2/
root # ./configure --with-apxs2=/usr/local/httpd/bin/apxs
--with-apache2=/usr/local/httpd/
root # make
root # cd ../build/jk2/apache2/
```

經過以上煩雜的動作之後，會得到一個叫做 mod_jk2.so 的 module，請把它複製到 apache 的 module 目錄裡。

```
root # cp mod_jk2.so /usr/local/httpd/modules/
```

設定 httpd.conf 及 mod_jk2 需要的設定檔 (workers2.properties)，同時建立 java-garden 的 JSP 連結

現在請編輯 httpd.conf 檔案以便在 apache 啟動時載入 mod_jk2 模組。

```
root # vi /usr/local/httpd/conf/httpd.conf
-----
LoadModule jk2_module modules/mod_jk2.so
-----
```

編輯 workers2.properties

```
root # vi /usr/local/httpd/conf/workers2.properties
-----

[logger]
level=DEBUG

[config:]
file=/usr/local/httpd/conf/workers2.properties
debug=0
debugEnv=0

[shm:]
info=Scoreboard. Required for reconfiguration and status with
multiprocess servers
```

```
file=/usr/local/httpd/logs/jk2.shm
size=1000000
debug=0
disabled=0
[workerEnv:]
info=Global server options
timing=1
debug=0
[channel.socket:java-garden.1-penguin.idv.tw:8009]
info=Ajp13 forwarding over socket
debug=0
tomcatId=java-garden.1-penguin.idv.tw:8009
[channel.jni:jni]
info=The jni channel, used if tomcat is started inprocess
[status:]
info=Status worker, displays runtime informations
[uri:/jkstatus/*]
info=Display status information and checks the config file for changes.
group=status:
[uri:/java-garden/*.jsp]
info=java-garden.
context=/java-garden
worker=ajp13:java-garden.1-penguin.idv.tw:8009
debug=0
-----
```

以上在 `workers2.properties` 檔裡粗體字表示需要注意的地方，是要輸入 Tomcat 主機的名稱或是 IP。編輯好了以上的設定之後還不能完全啟動，必需再為 Tomcat 設立 `java-garden` 的虛擬目錄才行。

上面 `[uri:/java-garden/*.jsp]` 表示只有 `.jsp` 檔案才會經由 `mod_jk2` 向 Tomcat 請求需要，一般的 `html` 或圖片影像檔就由本機的 `http` 主機負責傳輸就可以了。

在 Tomcat 上為 `java-garden` 建立虛擬目錄

要編輯 Tomcat 的虛擬目錄你需要看得懂 XML 語法，其實也跟 `httpd.conf` 差不多一樣，而它的編輯檔是在 `${TOMCAT_HOME}/conf/server.xml` 之下，而目錄是建在 `<host>` 和 `</host>` 之間。

現在請到你的 Tomcat 主機裡執行編輯動作。

```
root # vi /usr/local/tomcat5/conf/server.xml
-----
<Host>
~略~
<Context path="/java-garden"
docBase="/data/web/java-garden"
debug="0"
reloadable="true"
crossContext="true">
</Context>
</Host>
-----
```

在 server.xml 文件文打粗體的表示要注意的地方，path 表示為 http 網址後的連結，docBase 表示真正所在的目錄位置。

最終一回－實際驗收

終於到了可以實際驗收的階段了，請記得只要你收改過設定檔，就要重新啟動 httpd 和 Tomcat 服務以重新讀取設定檔。

Tomcat 主機

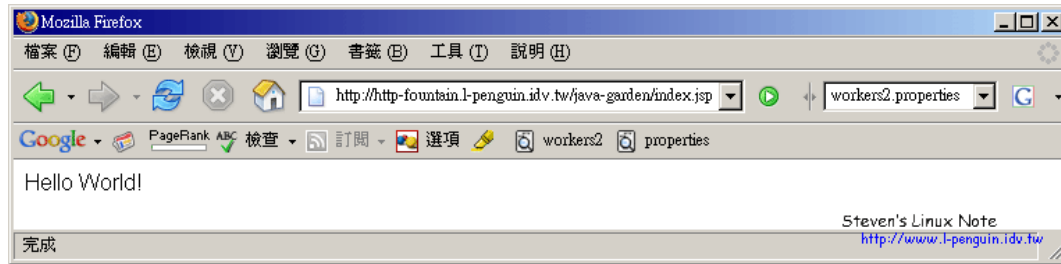
```
root # /usr/local/tomcat5/bin/shutdown.sh
Using CATALINA_BASE: /usr/local/tomcat5
Using CATALINA_HOME: /usr/local/tomcat5
Using CATALINA_TMPDIR: /usr/local/tomcat5/temp
Using JRE_HOME: /usr/local/java
root # /usr/local/tomcat5/bin/startup.sh
Using CATALINA_BASE: /usr/local/tomcat5
Using CATALINA_HOME: /usr/local/tomcat5
Using CATALINA_TMPDIR: /usr/local/tomcat5/temp
Using JRE_HOME: /usr/local/java
root #
```

Http 主機

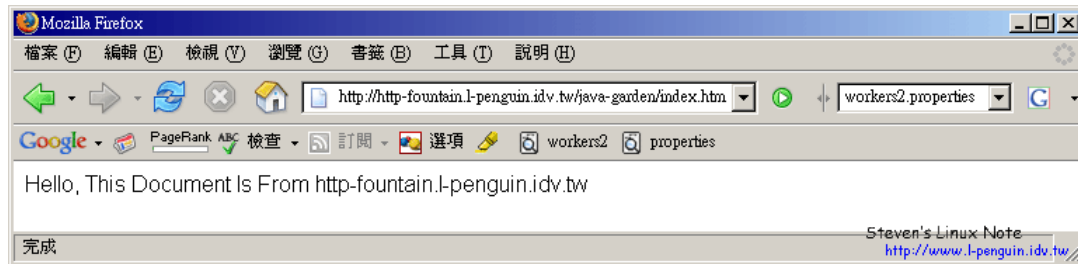
```
root # /usr/local/httpd/bin/apachectl stop
root # /usr/local/httpd/bin/apachectl start
```

我在 Tomcat 主機上的 /data/web/java-garden 放了幾隻 jsp 文件，包含一個 index.html 檔；而在 HTTP 主機上放了幾個 html 文件，也包含了一個 index.html 檔，現在來看看是否可以正常的執行。

若你要測試的話請在 Web Browser 輸入 <http://http-fountain.l-penguin.idv.tw/java-garden/index.jsp> 看看有什麼反應。



再來看看 index.html 檔。



好了，經過以上的實驗，你可以發現其實 Apache 只有在要求 jsp 時才會向 Tomcat 主機要求回應，否則的話都是由 Web 主機本身回應。

參考文件

- Tomcat 技術手冊 (O'Reilly, ISBN: 986-7794-26-5) :
http://www.oreilly.com.tw/product_java.php?id=a137
- [The Apache Tomcat 5.5 Servlet/JSP Container - Documentation Index](#)
- [The Apache Tomcat Connector Documentation Index](#)

For more articles, please visit <http://www.l-penguin.idv.tw/>

作者：廖子儀 (Tzu-Yi Liao)
Certified：LPIC Level I、LPIC Level II、RHCE
E-mail：steven@l-penguin.idv.tw
Web site：http://www.l-penguin.idv.tw/