

# 虛擬主機

網頁伺服器支援多個網站的方式與人們回應其本名與綽號是相當類似的。在 Apache 的組態檔中，包括「主要」身分在內的每個不同的身分，都以 `<VirtualHost>` 區塊指令來辨識，並稱為虛擬主機（有時簡寫成 `vhost`）。視用來存取網頁伺服器之名稱的不同，Apache 會做相對應的回應，正如同當某人被稱呼為「陳小姐」或「小華」時，也會有不同的反應一樣。如果想讓單一的系統支援多個網站，你一定要適當地設定 Apache。

Apache 支援兩種不同的虛擬主機。第一種被稱為以 IP 定址的虛擬主機，與用來連結系統的數字格式的網路位址有關。Bruce Wayne 在其起居室接電話時，絕不會說「我是蝙蝠俠！」，而他在蝙蝠洞接緊急電話時，也不會說「我是 Bruce Wayne！」。【譯註】不過，接電話的仍然是同一個人，正如同接受請求的網頁伺服器也是同一部一樣。

另外一種虛擬主機是以名稱定址的，因為伺服器的回應會依照所稱呼的名稱而不同。現在繼續我們的接電話故事，假設一間有三位室友的公寓；不論你想與大牛、阿明、小華、或阿珍講電話，都撥相同的電話號碼。如同許多人會共用一個電話號碼一樣，許多網站也可以共用相同的 IP 位址。不過，你必須用 `NameVirtualHost` 指令來設定由多個 Apache 虛擬主機共用的 IP 位址。

在最簡單的 Apache 組態中，是沒有虛擬主機的。相反地，組態檔中的所有指令都套用至整部伺服器的運作。由在任何 `<VirtualHost>` 區塊之外的指令所定義的環境有時就稱為預設伺服器，或全域伺服器。它沒有正式的名稱，不過當在組態中加入虛擬主機時，卻是一個考慮因素。

●.....  
譯註 Bruce Wayne 是美國著名的漫畫，後來拍成卡通及電影的「Batman」（台灣翻做「蝙蝠俠」）中，男主角的真正姓名，他有另一個名字：Batman（蝙蝠俠）。蝙蝠洞是他的秘密基地。

然而，如果將 `<VirtualHost>` 區塊加入這樣的組態中會發生什麼事呢？Apache 會如何解譯那些在區塊之外的指令呢？它們對虛擬主機的作用又如何呢？

要回答這些問題並不容易：基本上，所產生的作用會視各個組態指令而定。有些會被虛擬主機繼承、有些會重置到預設值、而有些則會假裝成從未被用過。你需要查閱各個指令的文件才能確定。

虛擬主機有兩種主要的形式：對於以 IP 定址的虛擬主機，各個虛擬主機都有其唯一的 IP 位址；而對於以名稱定址的虛擬主機，多台虛擬主機會以不同的名稱在同一 IP 位址上執行。本章會說明如何設定這兩種虛擬主機，以及如何在同一伺服器上結合兩者。你也會學到如何解決虛擬主機的常見問題。

## 4.1 架設以名稱定址的虛擬主機

### 問題

你只有一個 IP 位址，但是想在系統上支援多個網站。

### 解答

請使用 `NameVirtualHost *` 指令，以及 `<VirtualHost>` 區塊：

```
ServerName 127.0.0.1
NameVirtualHost *

<VirtualHost *>
    ServerName TheSmiths.name
    DocumentRoot "C:/Apache/Sites/TheSmiths"
</VirtualHost>

<VirtualHost *>
    ServerName JohnSmith.name
    DocumentRoot "C:/Apache/Sites/JustJohnSmith"
</VirtualHost>
```

### 討論

隨著愈來愈難取得 IP 位址，以名稱定址的虛擬主機，已是在同一部 Apache 伺服器上執行多個網站的最常用方式。對於大多數的使用者，以上的解答可順利地在大多數的虛擬主機環境中運作。

在前面的指令中的 \* 表示所設定的主機可執行在所有的位址上。對於只有單一位址的機器，這表示會執行在該位址上，同時也會在 *loopback* (或 *localhost* 的位址) 上執行。因此，如果你是位於實體的伺服器系統上，就可以檢視網站。

<VirtualHost> 區塊指令的參數與 NameVirtualHost 指令的參數是相同的。在此處 (NameVirtualHost) 放置主機名稱可能會在啟動伺服器時，停用虛擬主機的機制，而意外地將對此虛擬主機的請求送至別處。如果在啟動 Apache 伺服器時，你的名稱伺服器掛了，或沒有回應，則 Apache 將無法比對特定的 <VirtualHost> 區塊，與其所屬的 NameVirtualHost 指令。

對未列出之虛擬主機的請求，會送至在組態檔中的第一個虛擬主機。前面的例子中，TheSmiths.name 虛擬主機服務未在虛擬主機區塊指令中明確宣告的主機名稱。

執行 `httpd -S` 並觀察 Apache 解譯之虛擬主機組態，有助於檢視是否符合你的認定。`httpd -S` 會傳回虛擬主機的組態，顯示哪些主機是以名稱定址、哪些是以 IP 定址、以及哪些又是預設主機。

使用 `ServerAlias` 指令可以對特定的虛擬主機設定多個名稱，如下所示：

```
ServerName TheSmiths.name
ServerAlias www.TheSmiths.name Smith.Family.name
```

重點是，虛擬主機會使得使用者無法再存取組態檔中的主要伺服器 — 你必須對該主機明確地建立虛擬主機區塊。如果想讓此主要伺服器成為預設伺服器，則請先將它列出。

將以名稱定址的虛擬主機加入 Apache 組態中，並不會自動將對應的內容加入 DNS 伺服器。你還是要將紀錄加入 DNS 伺服器，以便能讓名稱解析至伺服器系統的 IP 位址。當使用者在其瀏覽器的網址列中輸入伺服器的名稱時，他們的電腦會先連上 DNS 伺服器來查詢該名稱，並解析成 IP 位址。如果沒有對應的 DNS 紀錄，瀏覽器將無法找到你的伺服器。

若要知道更多關於設定 DNS 伺服器的資訊，請參閱目前你所用的 DNS 軟體的文件，或洽詢 ISP (若未使用自己的 DNS 伺服器)。

## 參考資源

- <http://httpd.apache.org/docs/vhosts/>

## 4.2 指定一台以名稱定址的主機為預設伺服器

### 問題

你想要所有的請求（不論是否符合虛擬主機或使用 IP 位址）都導向預設主機，並處理可能的「找不到主機」錯誤訊息。

### 解答

請加入下列的 `<VirtualHost>` 區塊，並放在其它類似區塊的前面：

```
<VirtualHost *>
    ServerName default
    DocumentRoot /www/htdocs
    ErrorDocument 404 /site_list.html
</VirtualHost>
```

### 討論

請注意此秘訣只能用於以名稱定址的虛擬主機，所以會假定其它的虛擬主機也使用 `<VirtualHost *>` 指令，而且前面也有對應的 `NameVirtualHost *`。為了容易辨認之故，我們採用 `default` 的名稱；你可以隨意指定名稱。

將 `ErrorDocument 404` 指令設定成伺服器上可用的網站清單，可以引導使用者至有用的網頁內容，而非傳回無用的 404 錯誤訊息讓人不知所措。你也可以設定 `DirectoryIndex` 為網站清單，使直接存取網站首頁的使用者也會得到有用的資訊。

用 `ServerName` 或 `ServerAlias` 來明確地列出所有可用的主機名稱是不錯的點子，因為沒有人能再存取預設網站了。不過，如果直接用 IP 位址來存取網站，或在建立適當的虛擬主機之前，主機名稱就已加入 IP 位址，則使用者仍然會得到有用的網頁內容。

### 參閱

- 秘訣 4.4

## 4.3 架設以 IP 定址的虛擬主機

### 問題

你已在系統上設定多個 IP 位址，並想要在各個位址上架設一個網站。

## 解答

請替想列出的各個 IP 位址建立虛擬主機區塊：

```

ServerName 127.0.0.1

<VirtualHost 10.0.0.1>
    ServerName TheSmiths.name
    DocumentRoot "C:/Apache/Sites/TheSmiths"
</VirtualHost>

<VirtualHost 10.0.0.2>
    ServerName JohnSmith.name
    DocumentRoot "C:/Apache/Sites/JustJohnSmith"
</VirtualHost>

```

## 討論

在本例中的虛擬主機會接受所有對指定之 IP 位址的請求，而不論是用何種主機名稱來傳送。對未列出之任何其它 IP 位址的請求，則會送至組態檔中列在主設定範圍的虛擬主機。

必要時，`ServerName` 可用來設定虛擬主機的主要名稱，但並未用於將請求對應至正確主機的過程。Apache 只會使用 IP 位址來推算出哪台虛擬主機要提供請求的服務，而不是用 `Host` 標頭欄位。

## 參閱

- <http://httpd.apache.org/docs/vhosts/>

## 4.4 建立以 IP 定址的預設虛擬主機

### 問題

你想要建立虛擬主機，來接受所有對未設定之以 IP 定址主機的請求。

### 解答

請使用 `_default_` 的關鍵字來指定預設的主機：

```

<VirtualHost _default_>
    DocumentRoot /www/htdocs
</VirtualHost>

```

## 討論

`_default_` 關鍵字會建立一虛擬主機，可處理所有對未設定虛擬主機的「位址:埠號」之請求。

`_default_` 指令可以與特定的埠號一起使用，如下所示：

```
<VirtualHost _default_:443>
```

這表示所設定的虛擬主機會接受所有對 443 埠，而且未明確設定虛擬主機的位址上的請求。這也是一般在 SSL 的預設組態檔中看到的設定 SSL 的方式。

對於以名稱定址的虛擬主機，`_default_` 的一般運作方式則不如大眾所期待的。它不會符合沒有虛擬主機區塊的主機名稱，而只會符合沒有設定虛擬主機的「位址:埠號」組合。如果想要建立以名稱定址的預設主機，請參閱秘訣 4.2。

## 參閱

- 秘訣 4.2

## 4.5 混用以 IP 定址及以名稱定址的虛擬主機

### 問題

你已在系統上設定多個 IP 位址，並想要在各個位址上支援多個網站。

### 解答

請替各個 IP 位址設定 `NameVirtualHost` 指令，然後依照對單一 IP 位址的處理方式進行設定：

```
ServerName 127.0.0.1
NameVirtualHost 10.0.0.1
NameVirtualHost 10.0.0.2

<VirtualHost 10.0.0.1>
    ServerName TheSmiths.name
    DocumentRoot "C:/Apache/Sites/TheSmiths"
</VirtualHost>

<VirtualHost 10.0.0.1>
    ServerName JohnSmith.name
    DocumentRoot "C:/Apache/Sites/JustJohnSmith"
</VirtualHost>
```

```
<VirtualHost 10.0.0.2>
    ServerName TheFergusons.name
    DocumentRoot "C:/Apache/Sites/TheFergusons"
</VirtualHost>

<VirtualHost 10.0.0.2>
    ServerName DoriFergusons.name
    DocumentRoot "C:/Apache/Sites/JustDoriFerguson"
</VirtualHost>
```

## 討論

使用伺服器的位址，而非 \* 的萬用字元，可以讓虛擬主機只傾聽該 IP 位址。不過，你應該注意 `<VirtualHost>` 的參數仍然要符合其所屬的 `NameVirtualHost` 的參數。

## 參閱

- <http://httpd.apache.org/docs/vhosts/>

## 4.6 以 `mod_vhost_alias` 建立大量的虛擬主機

### 問題

你想要設定許多虛擬主機，而且每個主機的組態都完全相同。

### 解答

請使用由 `mod_vhost_alias` 提供的 `VirtualDocumentRoot` 與 `VirtualScriptAlias` 指令。

```
VirtualDocumentRoot /www/vhosts/%-1/%-2.1/%-2/htdocs
VirtualScriptAlias /www/vhosts/%-1/%-2.1/%-2/cgi-bin
```

### 討論

本段使用 `mod_vhost_alias` 提供的指令。不過當在組建 Apache 時，你可能還未安裝此模組，因為在預設狀況下 `mod_vhost_alias` 是不啟用的。

這兩個指令會將請求對應至從主機名稱片段組成之目錄。每個變數代表主機名稱的一部分，讓每個主機名稱對應至不同的目錄。

在此例中，對 *www.example.com* 的請求會對應至 */www/vhosts/com/e/example/htdocs* 目錄，或 */www/vhosts/com/e/example/cgi-bin*（對 CGI 的請求）。表 4-1 列出變數的完整範圍。

表 4-1：mod\_vhost\_alias 變數

變數	意義
%%	插入 %
%p	插入虛擬主機的埠號
%M.N	插入名稱（或其一部分）

M 與 N 可為正或負整數，其意義如表 4-2 所列。

表 4-2：變數值的意義

值	意義
0	整個名稱
1	名稱的第一部分
-1	名稱的最後一部分
2	名稱的第二部分
-2	名稱的倒數第二部分
2+	第二及之後的所有部分
-2+	倒數第二及之前的所有部分

當變數值放置在參數的第一部分 — 在 %M.N 的 M 部分 — 時，它表示主機名稱本身的部分。而當用於參數的第二部分 — N 的部分 — 時，則表示該部分中的特定字母。例如，在主機名稱 *www.example.com* 中，變數的意義如表 4-3 所列。

表 4-3：主機名稱 *www.example.com* 的範例值

值	意義
%0	www.example.com
%1	www
%2	example
%3	com
%-1	com
%-2	example
%-3	www

表 4-3：主機名稱 `www.example.com` 的範例值（續）

值	意義
%-2.1	e
%-2.2	x
%-2.3+	ample

依虛擬主機的數量而定，你可能會想建立以網域名稱、最上層網域、或只是主機名稱的字母順序來區分的目錄結構。

### 參閱

- [http://httpd.apache.org/docs/mod/mod\\_vhost\\_alias.html](http://httpd.apache.org/docs/mod/mod_vhost_alias.html)
- <http://httpd.apache.org/docs/vhosts/>

## 4.7 以 Rewrite 規則建立大量的虛擬主機

### 問題

雖然 `mod_vhost_alias` 模組可明確地用來支援大量的虛擬主機，它卻有許多限制，而且所有的虛擬主機都需要以完全相同的方式來設定。你想要支援大量的動態設定的 `vhost`，但同時卻不想使用 `mod_vhost_alias`。

### 解答

請使用 `mod_rewrite` 的指令，依據主機名稱將請求對應至目錄：

```

RewriteEngine on
RewriteCond %{HTTP_HOST} ^(\www\.)?([\^.]+)\.com$
RewriteRule ^(.*)$ /home/%2$1
    
```

### 討論

`mod_vhost_alias` 很有用，但只適用於除了主機名稱外，所有的虛擬主機的設定都相同的情形。使用 `mod_vhost_alias` 會讓其它對應 URL 的模組（例如 `mod_userdir`、`mod_rewrite`、以及 `mod_alias`）無法作用，而且也會有很多限制。`mod_rewrite` 的效率會比較差，但是卻有較大的彈性。

例如，當使用 `mod_vhost_alias` 時，所有的主機都必須用 `mod_vhost_alias`；而使用這個替代方案時，一些主機可以用重寫（rewrite）規則，而其它的則可以用傳統方法來設定。

在解答中的指令會將對 `www.something.com` 的請求（不論有沒有 `www`）對應至 `/home/something` 的目錄。

## 參閱

- 秘訣 5.16
- <http://httpd.apache.org/docs/vhosts/>
- [http://httpd.apache.org/docs/mod/mod\\_rewrite.html](http://httpd.apache.org/docs/mod/mod_rewrite.html)

## 4.8 SSL 及以名稱定址的虛擬主機

### 問題

你想要在同一台伺服器上設定多個 SSL 網站。

### 解答

在大多數常見的 SSL 實作中，只能在每個位址及埠號上使用一個 SSL 主機。因此，對每個 SSL 主機需要有唯一的 IP 位址，或者在不同的埠上執行，以便在特定的 IP 位址上設定多個 SSL 主機：

```
Listen 443
Listen 444

<VirtualHost 10.0.1.2:443>
    ServerName secure1.example.com
    DocumentRoot /www/vhosts/secure1

    SSLEngine On
    SSLCertificateFile /www/conf/ssl/secure1.crt
    SSLCertificateKeyFile /www/conf/ssl/secure1.key
</VirtualHost>

<VirtualHost 10.0.1.3:443>
    ServerName secure2.example.com
    DocumentRoot /www/vhosts/secure2
```

```

        SSLEngineOn
        SSLCertificateFile    /www/conf/ssl/secure2.crt
        SSLCertificateKeyFile /www/conf/ssl/secure2.key
    </VirtualHost>

<VirtualHost 10.0.1.3:444>
    ServerName secure3.example.com
    DocumentRoot /www/vhosts/secure3

    SSLEngineOn
    SSLCertificateFile    /www/conf/ssl/secure3.crt
    SSLCertificateKeyFile /www/conf/ssl/secure3.key
</VirtualHost>

```

## 討論

每個 IP 位址只能使用一個 SSL 主機的，限制不是 Apache 要求的，而是基於 SSL 的運作方式。當瀏覽器連接伺服器時，第一件事是協商出安全的連線。在這個過程中，伺服器會將其憑證送給用戶端，這表示其餘的交易過程會被加密。

因為這發生在瀏覽器告知伺服器需要什麼資源之前，此一階段只能依據用戶端連接的 IP 位址。在伺服器接受到 Host 標頭欄位時，這已經太遲了 — 因為憑證早已經送出了。

如果在 URL 中有明確地指定埠號，是可以在 443 之外的其它埠上執行 SSL 主機。你可以藉此避過這個限制，但是使用者就必須輸入正確的 URL 及埠號。

## 參閱

- 秘訣 4.11

## 4.9 替每個虛擬主機建立日誌紀錄

### 問題

你想要每個虛擬主機都有自己的日誌檔。

### 解答

請在每個虛擬主機的設定區塊中指定 ErrorLog 及 CustomLog：

```

<VirtualHost *>
    ServerName    waldo.example.com
    DocumentRoot /home/waldo/www/htdocs

```

```
ErrorLog      /home/waldo/www/logs/error_log
CustomLog    /home/waldo/www/logs/access_log combined
</VirtualHost>
```

## 討論

各種不同的日誌指令可以放置在組態檔中的主要設定區，或 `<VirtualHost>` 區塊中。當放入虛擬主機之內時，該主機的日誌內容就會寫入所設定的日誌檔中，而非定義在主伺服器組態內的日誌檔。



每一個日誌檔都會算在伺服器允許擁有的檔案及網路連線總數之內。假設有 100 個虛擬主機，而且每個都有自己的錯誤及活動日誌，就表示有 200 個開啟的通道，因此如果伺服器的容量限制是 256，你每次只能處理 56 個同步請求。

在本節中，日誌檔是放在特定使用者的主目錄，而非主要的日誌目錄內。如此，可以較方便處理這些檔案，但是你仍然需要採取適當的防範措施來設定目錄的權限。請參閱第六章對檔案權限的討論。

## 參閱

- 第三章
- 第六章

## 4.10 分割日誌檔

### 問題

由於擁有大量的虛擬主機，因此你想只保有一份日誌檔，事後再加以分割。

### 解答

```
LogFormat "%v %h %l %u %t \"%r\" %>s %b" vhost
CustomLog logs/vhost_log vhost
```

接著，在更替日誌檔之後：

```
split-logfile < logs/vhost_log
```

## 討論

本秘訣中的 `LogFormat` 指令會建立類似於 `common` 日誌格式的日誌檔，不過此檔還另外包括被存取之虛擬主機名稱。`split-logfile` 程式會將此日誌檔分割到所構成的虛擬主機中。

## 參閱

- 秘訣 3.11

## 4.11 以埠定址的虛擬主機

### 問題

你想要在不同的埠上顯現不同的 HTTP 網頁連線內容。

### 解答

請在 `<VirtualHost>` 區塊中明確地設定埠號：

```
Listen 8080

<VirtualHost 10.0.1.2:8080>
    DocumentRoot /www/vhosts/port8080
</VirtualHost>

Listen 9090

<VirtualHost 10.0.1.2:9090>
    DocumentRoot /www/vhosts/port9090
</VirtualHost>
```

### 討論

相較於本章中其它的技術，以埠定址的虛擬主機比較少見。不過，它還是可用於各種不同的狀況。如果只有一個 IP 位址、無法將主機名稱加入 DNS、或者 ISP 阻斷在 80 埠上向內的連線，就可用它在其它埠上執行虛擬主機。

網站的訪客必須將所用的埠號加到 URL 中。例如，要造訪前面所列的第二個虛擬主機，會需要使用下列的 URL：

```
http://server.example.com:9090/
```

## 參閱

- <http://httpd.apache.org/docs/vhosts/>

## 4.12 在數個位址上顯示相同的網頁內容

### 問題

你想要在兩個位址上顯現相同的網頁內容。

### 解答

請在 `<VirtualHost>` 指令中設定這兩個位址：

```
NameVirtualHost 192.168.1.1
NameVirtualHost 172.20.30.40

<VirtualHost 192.168.1.1 172.20.30.40>
    DocumentRoot /www/vhosts/server
    ServerName server.example.com
    ServerAlias server
</VirtualHost>
```

### 討論

對於有兩個位址（其中之一屬於內部網路，另一則只能從網路外部存取）的機器，此種設定最有用。如果只有這兩個位址，你可以使用秘訣 4.1 所介紹的 \* 表示法。不過，如果有更多的位址時，此種方式可以讓你設定哪些內容要顯現在哪些位址上。

## 參閱

- <http://httpd.apache.org/docs/vhosts/>