

10 条条大路通罗马，形形色色的虚拟主机

更新时间：2019-12-26 09:09:39



机会不会上门来找人，只有人去找机会。——狄更斯

虚拟主机

说起虚拟主机，想必对于程序猿来说，这个东西并不陌生。阿里云，腾讯云，百度云等云服务器厂商都提供虚拟主机服务，使得普通人可以以很低的价格拥有自己的主机，我们可以搭建自己的网站，博客等。

那么究竟是什么虚拟主机呢？不知道大家是不是真正的理解这个概念。

我们看一下维基百科对 [虚拟主机](#) 的解释：

虚拟主机（英语：[virtual hosting](#)）或称 **共享主机**（[shared web hosting](#)），又称**虚拟服务器**，是一种在单一主机或主机群上，实现多网域服务的方法，可以运行多个网站或服务的技术。虚拟主机之间完全独立，并可由用户自行管理，虚拟并非指不存在，而是指空间是由实体的服务器延伸而来，其硬件系统可以是基于服务器群，或者单个服务器。

其实简单的来说，虚拟主机就是多个网站使用同一个物理服务器，每一个服务器都可以单独的向外界提供服务，互不影响。虚拟主机之间完全独立，在外界看来，每一台虚拟主机都和一台独立的主机表现的完全一致。

虚拟主机的出现，对互联网来说是一个非常重要的贡献，由于可以让多个虚拟主机共享一台物理主机的所有资源，所以让每个用户承担的成本也大大的降低了。

[Nginx](#) 作为一个广泛应用的 [Web](#) 服务器，支持三种形式的虚拟主机配置：

- 基于端口号的主机配置

- 基于 ip 的主机配置
- 基于域名的主机配置

我们逐个介绍这几种配置方式。

基于端口号的主机配置

所谓基于端口的主机配置就是多个虚拟主机的 ip 地址相同，但是他们的端口不同，我们可以通过端口号来访问不同的虚拟主机。

我们以 8081, 8082, 8083 三个端口为例来了解这种配置。

为每一个端口号创建目录。

1. 在 /usr/local/nginx/html/ 目录下面创建 port 子目录
2. 在 port 下面创建 8081, 8082, 8083 三个子目录、
3. 每个子目录分别创建各个端口的 index 文件

```
[root@a691d858ef77 ~]#
[root@a691d858ef77 ~]# cd /usr/local/nginx/html/
[root@a691d858ef77 html]# mkdir port
[root@a691d858ef77 html]# cd port/
[root@a691d858ef77 port]# mkdir 8081 8082 8083
[root@a691d858ef77 port]# tree /usr/local/nginx/html/port/
/usr/local/nginx/html/port/
|-- 8081
|-- 8082
`-- 8083

3 directories, 0 files
[root@a691d858ef77 port]# echo "8081 port" > 8081/index.html
[root@a691d858ef77 port]# echo "8082 port" > 8082/index.html
[root@a691d858ef77 port]# echo "8083 port" > 8083/index.html
[root@a691d858ef77 port]# tree /usr/local/nginx/html/port/
/usr/local/nginx/html/port/
|-- 8081
|   '-- index.html
|-- 8082
|   '-- index.html
`-- 8083
    '-- index.html

3 directories, 3 files
[root@a691d858ef77 port]#
```

修改配置文件

修改 `/usr/local/nginx/conf` 目录下面的 `nginx.conf` 文件，如下：

```
server {
    listen      127.0.0.1:8081;
    server_name localhost;
    location / {
        root  html/port/8081/;
        index index.html index.htm;
    }
}

server {
    listen      127.0.0.1:8082;
    server_name localhost;
    location / {
        root  html/port/8082/;
        index index.html index.htm;
    }
}

server {
    listen      127.0.0.1:8083;
    server_name localhost;
    location / {
        root  html/port/8083/;
        index index.html index.htm;
    }
}
```

8081端口

8082端口

8083端口

启动 `nginx` 并进行请求

我们启动 `nginx` 之后，查看系统的端口情况如下：可以看到，三个端口都已经被监听了。

```
[root@a691d858ef77 conf]# netstat -lnt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 127.0.0.1:8081          0.0.0.0:*
tcp      0      0 127.0.0.1:8082          0.0.0.0:*
tcp      0      0 127.0.0.1:8083          0.0.0.0:*
[root@a691d858ef77 conf]#
[root@a691d858ef77 conf]#
```

测试端口

我们可以通过 `curl` 命令测试各个端口，如下：

```
[root@a691d858ef77 ~]#
[root@a691d858ef77 ~]# curl http://localhost:8081/index.html
8081 port
[root@a691d858ef77 ~]# curl http://localhost:8082/index.html
8082 port
[root@a691d858ef77 ~]# curl http://localhost:8083/index.html
8083 port
[root@a691d858ef77 ~]#
```

上面的几个步骤非常简单明了的说明了如何基于端口配置不同的 **server**，大家可以和后面的基于 **ip** 和 **域名** 的端口进行比较。

基于 **IP** 的虚拟主机

所谓基于 **IP** 的虚拟主机就是将虚拟主机绑定到不同的 **IP** 地址。我们可以通过不同的 **IP** 地址访问不同的虚拟主机。

配置 **IP**

一般情况下，我们自己的电脑都只会有最多两个网卡，这个时候我们可以使用 **ifconfig** 命令来增加几个虚拟网卡来完成本次实验。如下：

```
root:~# ifconfig lo:2 127.0.0.4/24 up
root:~# ifconfig lo:3 127.0.0.5/24 up
root:~# ifconfig lo:4 127.0.0.6/24 up
root:~#
root:~#
root:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 172.17.189.39 netmask 255.255.240.0 broadcast 172.17.191.255
          ether 00:16:3e:0e:ba:38 txqueuelen 1000 (Ethernet)
          RX packets 14321162 bytes 1188550511 (1.1 GiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 12396982 bytes 1517679655 (1.4 GiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
          loop txqueuelen 1 (Local Loopback)
          RX packets 1483249197 bytes 184714750129 (172.0 GiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 1483249197 bytes 184714750129 (172.0 GiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo:1: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.2 netmask 255.255.255.0
          loop txqueuelen 1 (Local Loopback)

lo:2: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.4 netmask 255.255.255.0
          loop txqueuelen 1 (Local Loopback)

lo:3: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.5 netmask 255.255.255.0
          loop txqueuelen 1 (Local Loopback)

lo:4: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.6 netmask 255.255.255.0
          loop txqueuelen 1 (Local Loopback)
```

增加三个虚拟ip

新增加的虚拟ip

为每个 **IP** 设置目录

```
root:/usr/local/nginx/html# 
root:/usr/local/nginx/html# mkdir ip
root:/usr/local/nginx/html# cd ip/
root:/usr/local/nginx/html/ip# mkdir 4 5 6
root:/usr/local/nginx/html/ip# echo "hello 127.0.0.4" > 4/index.html
root:/usr/local/nginx/html/ip# echo "hello 127.0.0.5" > 5/index.html
root:/usr/local/nginx/html/ip# echo "hello 127.0.0.6" > 6/index.html
root:/usr/local/nginx/html/ip# 
root:/usr/local/nginx/html/ip# tree ../ip/
```

修改配置文件

针对每一个 IP，设置配置文件：

```
server {
    listen      127.0.0.4:8081;

    location / {
        root  html/ip/4/;
        index index.html index.htm;
    }

}

server {
    listen      127.0.0.5:8081;

    location / {
        root  html/ip/5/;
        index index.html index.htm;
    }

}

server {
    listen      127.0.0.6:8081;

    location / {
        root  html/ip/6/;
        index index.html index.htm;
    }

}
```

测试各个 IP

启动 Nginx，并使用不同的 IP 进行访问，得到的结果如下：

```
root:/usr/local/nginx/html/ip# curl http://127.0.0.4:8081/index.html
hello 127.0.0.4
root:/usr/local/nginx/html/ip# curl http://127.0.0.5:8081/index.html
hello 127.0.0.5
root:/usr/local/nginx/html/ip# curl http://127.0.0.6:8081/index.html
hello 127.0.0.6
```

基于域名的虚拟主机

所谓基于域名的虚拟主机是非常广泛的一种使用方式，绝大多数公司都是使用该种方式配置的虚拟主机。

配置域名

在 `/etc/hosts` 文件中配置三个域名，分别如下：

```
127.0.0.1      www.hr.com
127.0.0.1      www.platmis.com
127.0.0.1      www.napi.com
```

为域名创建目录

我们为每个域名创建一个单独的目录，创建 `index` 文件，如下：

```
[root@a691d858ef77 html]#
[root@a691d858ef77 html]# mkdir domain
[root@a691d858ef77 html]# cd domain/
[root@a691d858ef77 domain]# mkdir hr platmis napi
[root@a691d858ef77 domain]# echo "hello www.hr.com" > hr/index.html
[root@a691d858ef77 domain]# echo "hello www.platmis.com" > platmis/index.html
[root@a691d858ef77 domain]# echo "hello www.napi.com" > napi/index.html
[root@a691d858ef77 domain]# tree
.
|-- hr
|   '-- index.html
|-- napi
|   '-- index.html
`-- platmis
    '-- index.html

3 directories, 3 files
[root@a691d858ef77 domain]#
```

修改配置文件

对上面三个虚拟主机进行配置，如下：

```
http {
    include      mime.types;
    default_type application/octet-stream;

    server {
        listen      127.0.0.1:8080;
        server_name www.hr.com;
        location / {
            root      html/domain/hr/;
            index   index.html index.htm;
        }
    }

    server {
        listen      127.0.0.1:8080;
        server_name www.platmis.com;
        location / {
            root      html/domain/platmis/;
            index   index.html index.htm;
        }
    }

    server {
        listen      127.0.0.1:8080;
        server_name www.napi.com;
        location / {
            root      html/domain/napi/;
            index   index.html index.htm;
        }
    }
}
```

hr

platmis

napi

我们对三个域名都监听了相同的 IP 地址和端口号。

测试域名

重启 `nginx` 之后，查看 `8080` 端口，如下：

```
[root@a691d858ef77 ~]# netstat -lnt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 127.0.0.1:8080          0.0.0.0:*
[root@a691d858ef77 ~]#
```

可以看到，`nginx` 监听了 `8080` 端口，我们分别测试三个域名：

```
[root@a691d858ef77 ~]# curl http://www.hr.com:8080/index.html
hello www.hr.com
[root@a691d858ef77 ~]# curl http://www.platmis.com:8080/index.html
hello www.platmis.com
[root@a691d858ef77 ~]# curl http://www.napi.com:8080/index.html
hello www.napi.com
[root@a691d858ef77 ~]#
```

从结果中可以看到，虽然三个域名监听了相同的 `IP` 地址和端口号，但是通过域名进行访问的时候，`nginx` 可以完美的区分不同的域名。

在本文中，我们介绍了 `Nginx` 支持的三种虚拟主机形式，根据我个人的经验来看，几乎所有的公司使用的都是最后一种，也就是基于域名的主机配置方式，这些东西并不难，大家注重理解就可以了。

如果大家在测试的时候出现问题，不要着急，请打开 `Nginx` 的日志文件，对于我们解决问题非常有帮助。

}

← 09 必要的全局配置选项

11 让正则飞起来：PCRE 库的相识相知 →