

Linux下常见服务器的搭建

- 操作系统选用ubuntu-15.10-desktop-amd64

Web服务器

下载Apache的http服务软件包httpd

```
sudo wget http://archive.apache.org/dist/httpd/httpd-2.2.31.tar.gz
```

```
liuyong@mylinux:/usr/local$ sudo wget http://archive.apache.org/dist/httpd/httpd-2.2.31.tar.gz
--2016-06-14 10:34:11-- http://archive.apache.org/dist/httpd/httpd-2.2.31.tar.gz
正在解析主机 archive.apache.org (archive.apache.org)... 163.172.17.199
正在连接 archive.apache.org (archive.apache.org)|163.172.17.199|:80... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度：7583841 (7.2M) [application/x-gzip]
正在保存至：“httpd-2.2.31.tar.gz”

httpd-2.2.31.ta 100%[=====>] 7.23M 11.9KB/s 用时 8m 3s

2016-06-14 10:42:15 (15.3 KB/s) - 已保存 “httpd-2.2.31.tar.gz”
[7583841/7583841])

liuyong@mylinux:/usr/local$ ls
bin  games  include  man  share
etc  httpd-2.2.31.tar.gz  lib  sbin  src
```

解压缩到当前目录下

```
sudo tar -zxvf httpd-2.2.31.tar.gz
```

```
liuyong@mylinux:/usr/local$ sudo tar -zxvf httpd-2.2.31.tar.gz

httpd-2.2.31/
httpd-2.2.31/CHANGES
httpd-2.2.31/ABOUT_APACHE
httpd-2.2.31/httpd.dsp
httpd-2.2.31/INSTALL
httpd-2.2.31/Makefile.win
httpd-2.2.31/build/
httpd-2.2.31/build/pkg/
httpd-2.2.31/build/pkg/pkginfo.in
httpd-2.2.31/build/pkg/buildpkg.sh
httpd-2.2.31/build/pkg/README
httpd-2.2.31/build/library.mk
```

进入到解压缩目录

```
cd httpd-2.2.31
```

```
liuyong@mylinux:/usr/local$ ls
bin  games  httpd-2.2.31.tar.gz  lib  sbin  src
etc  httpd-2.2.31  include  man  share
liuyong@mylinux:/usr/local$ cd httpd-2.2.31
liuyong@mylinux:/usr/local/httpd-2.2.31$
```

软件配置与检查

```
./configure --prefix=/usr/local/apache2
```

- 作用：
 - 定义需要的功能选项
 - 检测系统环境是否符合安装要求
 - 把定义好的功能选项和检测系统环境的信息写入Makefile文件，用于后续编辑

```
liuyong@mylinux:/usr/local/httpd-2.2.31$ ./configure --prefix=
/usr/local/apache2
checking for chosen layout... Apache
checking for working mkdir -p... yes
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu

Configuring Apache Portable Runtime library ...

checking for APR... reconfig
configuring package in srclib/apr now
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
```

- 正常结束，最后几行无error、warning或no的提示，即为正常安装

make编译

```
make
```

```
liuyong@mylinux:/usr/local/httpd-2.2.31$ make
Making all in srclib
make[1]: Entering directory '/usr/local/httpd-2.2.31/srclib'
Making all in apr
make[2]: Entering directory '/usr/local/httpd-2.2.31/srclib/apr'
make[3]: Entering directory '/usr/local/httpd-2.2.31/srclib/apr'
make[3]: Leaving directory '/usr/local/httpd-2.2.31/srclib/apr'
make[2]: Leaving directory '/usr/local/httpd-2.2.31/srclib/apr'
make[1]: Leaving directory '/usr/local/httpd-2.2.31/srclib'
make: Leaving directory '/usr/local/httpd-2.2.31'
```

make安装

```
sudo make install
```

```
liuyong@mylinux:/usr/local/httpd-2.2.31$ sudo make install
[sudo] liuyong 的密码：
Making install in srclib
make[1]: Entering directory '/usr/local/httpd-2.2.31/srclib'
Making install in apr
make[2]: Entering directory '/usr/local/httpd-2.2.31/srclib/apr'
make[3]: Entering directory '/usr/local/httpd-2.2.31/srclib/apr'
make[3]: Nothing to be done for 'local-all'.
make[3]: Leaving directory '/usr/local/httpd-2.2.31/srclib/apr'
make[2]: Leaving directory '/usr/local/httpd-2.2.31/srclib/apr'
make[1]: Leaving directory '/usr/local/httpd-2.2.31/srclib'
make: Leaving directory '/usr/local/httpd-2.2.31'
```

启动Apache服务

```
./bin/apachectl start
```

```
liuyong@mylinux:/usr/local/apache2$ sudo ./bin/apachectl start
httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName
liuyong@mylinux:/usr/local/apache2$
```

访问


```
liuyong@mylinux:~$ sudo apt-get install vsftpd
[sudo] liuyong 的密码：
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列【新】软件包将被安装：
  vsftpd
升级了 0 个软件包，新安装了 1 个软件包，要卸载 0 个软件包，有 154 个软件包未被升级。
需要下载 116 kB 的软件包。
解压后会消耗掉 331 kB 的额外空间。
获取：1 http://cn.archive.ubuntu.com/ubuntu/ wily/main vsftpd amd64 3.0.2-20ubuntu1 [116 kB]
下载 116 kB，耗时 0秒 (232 kB/s)
正在预设软件包 ...
正在选中未选择的软件包 vsftpd。
(正在读取数据库 ... 系统当前共安装有 177484 个文件和目录。)
正准备解包 .../vsftpd_3.0.2-20ubuntu1_amd64.deb ...
正在解包 vsftpd (3.0.2-20ubuntu1) ...
正在处理用于 man-db (2.7.4-1) 的触发器 ...
正在处理用于 ureadahead (0.100.0-19) 的触发器 ...
正在处理用于 systemd (225-1ubuntu9) 的触发器 ...
正在设置 vsftpd (3.0.2-20ubuntu1) ...
```

查看ftp端口是否打开

```
netstat -tl
```

```
liuyong@mylinux:~$ netstat -tl
激活Internet连接（仅服务器）
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 mylinux:domain          *:                        LISTEN
tcp        0      0 localhost:ipp            *:                        LISTEN
tcp6       0      0 [::]:ftp                 [::]:*                  LISTEN
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN
```

配置vsftpd.conf

```
sudo nano /etc/vsftpd.conf
```

```
#禁止匿名访问
anonymous_enable=NO
#接受本地用户
local_enable=YES
#允许上传
write_enable=YES
#用户只能访问限制的目录
chroot_local_user=YES
#设置固定目录，在结尾添加。如果不添加这一行，各用户对应自己的目录，当然这个文件夹自己建
local_root=/home/ftp
```

添加ftp用户

```
sudo mkdir /home/ftp

sudo useradd -d /home/ftp -M ftpuser

sudo passwd ftpuser
```

调整文件夹权限

- 这个是避免“500 OOPS: vsftpd: refusing to run with writable root inside chroot()”

```
sudo chmod a-w /home/ftp

sudo mkdir /home/ftp/data
```

改pam.d/vsftpd

- 这时候直接用useradd的帐号登录ftp会530 login incorrect

```
sudo nano /etc/pam.d/vsftpd
```

- 注释掉

```
#auth    required pam_shells.so
```

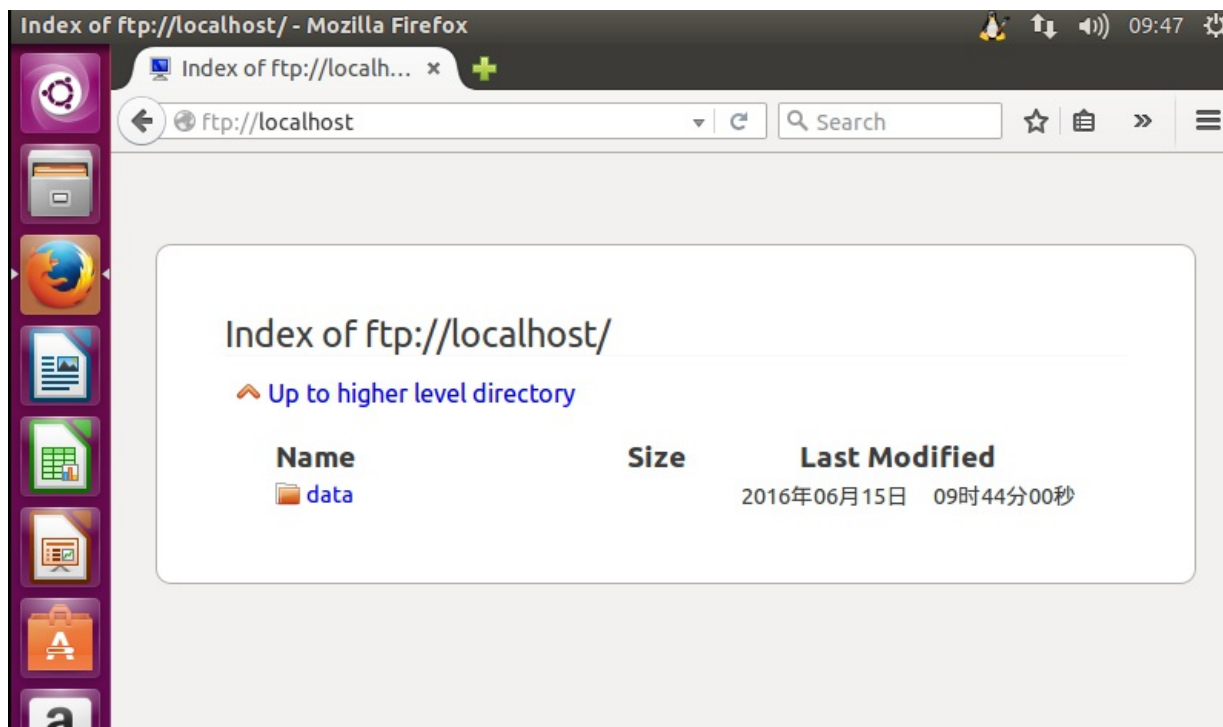
重启vsftpd

```
sudo service vsftpd restart
```

- 这时就可以用刚才建的ftpuser这个用户登录ftp了，看到的是local_root设置的/home/ftp，并且限制在该目录。
- 可以在浏览器用ftp://xxx.xxx.xxx.xxx访问，也可以用ftp软件，密码就是ftpuser的密码。

```
liuyong@mylinux:~$ sudo nano /etc/vsftpd.conf
[sudo] liuyong 的密码：
liuyong@mylinux:~$ sudo useradd -d /home/ftp -M ftpuser
liuyong@mylinux:~$ sudo chmod a-w /home/ftp
chmod: 无法访问"/home/ftp": 没有那个文件或目录
liuyong@mylinux:~$ mkdir /home/ftp
mkdir: 无法创建目录"/home/ftp": 权限不够
liuyong@mylinux:~$ sudo mkdir /home/ftp
liuyong@mylinux:~$ sudo chmod a-w /home/ftp
liuyong@mylinux:~$ sudo mkdir /home/ftp/data
liuyong@mylinux:~$ sudo nano /etc/pam.d/vsftpd
liuyong@mylinux:~$ sudo service vsftpd restart
liuyong@mylinux:~$ sudo passwd ftpuser
输入新的 UNIX 密码：
重新输入新的 UNIX 密码：
passwd：已成功更新密码
liuyong@mylinux:~$
```

浏览器验证



DNS服务器

前言

- 其实在我们没有安装DNS服务之前，可以将/etc/hosts文件比作一个DNS服务配置文件，因为它实现和DNS类似。
- 之所以会独立出DNS服务，是因为因特网主机多，如果每个主机都靠/etc/hosts文件来维护主机名到ip的映射，那么工作量非常大，对本地更新、网络资源占用都很浪费，所以出现了DNS。

安装bind

```
sudo apt-get install bind9
```

相关文件

- /etc/host 本地的一个小"DNS"文件。
- /etc/resolv.conf 用来指定DNS服务的地址，在没有自定义DNS地址时，发现其指向本机。如下：

```
liuyong@mylinux:/etc/bind$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#      DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 127.0.1.1
search localdomain
```

- 所以，此时主机查找域名时是根据本地/etc/hosts。
- /etc/host.conf 指定主机找哪个DNS解析的顺序.如下：

```
liuyong@mylinux:/etc/bind$ cat /etc/host.conf
# The "order" line is only used by old versions of the C library.
order hosts,bind
multi on
```

- 所以，是先在本地搜索（hosts），然后再用bind指定的DNS区找（相关的查找信息在/etc/bind目录中）。
- 先来介绍一下/etc/bind目录下的文件：

```
liuyong@mylinux:~$ cd /etc/bind
liuyong@mylinux:/etc/bind$ ls -la
总用量 68
drwxr-sr-x  2 root bind  4096 6月 15 10:26 .
drwxr-xr-x 131 root root 12288 6月 15 10:26 ..
-rw-r--r--  1 root root  2389 3月  8 22:59 bind.keys
-rw-r--r--  1 root root   237 3月  8 22:59 db.0
-rw-r--r--  1 root root   271 3月  8 22:59 db.127
-rw-r--r--  1 root root   237 3月  8 22:59 db.255
-rw-r--r--  1 root root   353 3月  8 22:59 db.empty
-rw-r--r--  1 root root   270 3月  8 22:59 db.local
-rw-r--r--  1 root root  3048 3月  8 22:59 db.root
-rw-r--r--  1 root bind   463 3月  8 22:59 named.conf
-rw-r--r--  1 root bind   490 3月  8 22:59 named.conf.default-zones
-rw-r--r--  1 root bind   165 3月  8 22:59 named.conf.local
-rw-r--r--  1 root bind   890 6月 15 10:26 named.conf.options
-rw-r-----  1 bind bind    77 6月 15 10:26 rndc.key
-rw-r--r--  1 root root  1317 3月  8 22:59 zones.rfc1918
```

- 其中，主要的是named.conf文件，它包括了DNS的重要配置信息，它有下面三个文件组成：

```
named.conf.default-zones
named.conf.local
named.conf.options
```

- named.conf.default-zones 包含了反解文件：db.0 db.127 正解文件：db.root db.local. 指定了DNS查找的配置信息，如db.local:

```
liuyong@mylinux:/etc/bind$ cat db.local
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      localhost. root.localhost. (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       localhost.
@         IN      A        127.0.0.1
@         IN      AAAA     ::1
```

- @代表域名，在这里是local.(有个点)。
- 另外，db.root 包含了最顶层的域名和对应的地址，所有有需要的DNS都可以从这些地址得到其它域名的地址信息。
- named.conf.local暂时没用到
- named.conf.options 包含了一些设置信息，如设置为cache-only DNS，添加forwarding功能等。

配置使用

- 这里我以 liuyong666.com 域名为例

```
cd /etc/bind
```

- 域名添加

```
sudo vim /etc/bind/named.conf.default-zones
```

```
zone "liuyong666.com" {
    type master;
    file "/etc/bind/liuyong666.local";
};

zone "1.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192";
};
```



```

zone "localhost" {
    type master;
    file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
    type master;
    file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
    type master;
    file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
    type master;
    file "/etc/bind/db.255";
};

zone "liuyong666.com" {
    type master;
    file "/etc/bind/liuyong666.local";
};

zone "1.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192";
};
-- 插入 --

```

```
sudo cp db.local liuyong666.local
```

```
sudo cp db.127 db.192
```

```

liuyong@mylinux:/etc/bind$ sudo cp db.local liuyong666.local
liuyong@mylinux:/etc/bind$ sudo cp db.127 db.192
liuyong@mylinux:/etc/bind$ ls
bind.keys  db.255    liuyong666.local    named.conf.options
db.0       db.empty  named.conf           rndc.key
db.127     db.local  named.conf.default-zones  zones.rfc1918
db.192     db.root   named.conf.local
liuyong@mylinux:/etc/bind$

```

- 编辑 liuyong666.local，末尾两行添加域名

```
sudo vim liuyong666.local
```

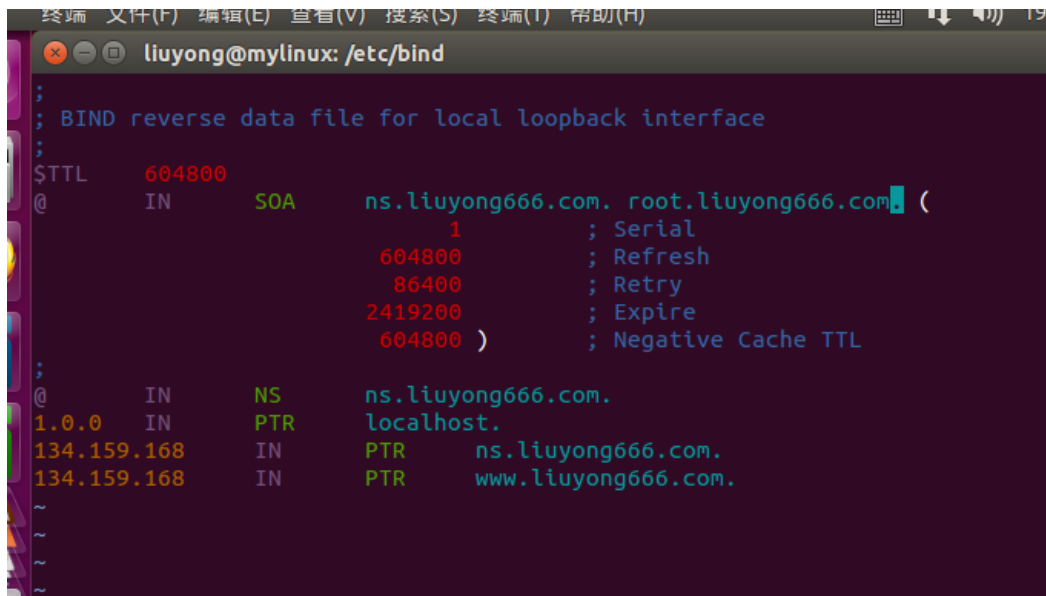
```

liuyong@mylinux: /etc/bind
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      ns.liuyong666.com. root.liuyong666.com. (
                                2      ; Serial
                                604800 ; Refresh
                                86400  ; Retry
                                2419200 ; Expire
                                604800 ) ; Negative Cache TTL
;
@         IN      NS       ns.liuyong666.com.
@         IN      A        192.168.159.134
@         IN      AAAA     ::1
ns        IN      A        192.168.159.134
www       IN      A        192.168.159.134
~
~
~

```


- 编辑 db.192，末尾两行添加

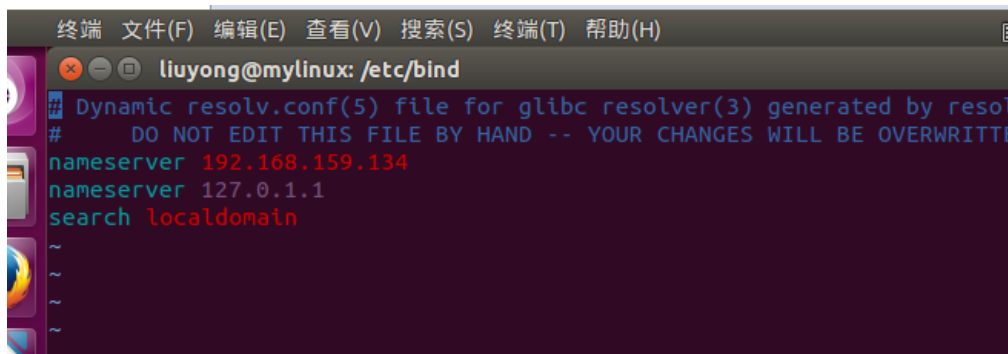
```
sudo vim db.192
```



```
终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
liuyong@mylinux: /etc/bind
;
; BIND reverse data file for local loopback interface
;
$TTL      604800
@         IN      SOA      ns.liuyong666.com. root.liuyong666.com. (
                                1              ; Serial
                                604800         ; Refresh
                                86400          ; Retry
                                2419200        ; Expire
                                604800         ; Negative Cache TTL
                                )
;
@         IN      NS       ns.liuyong666.com.
1.0.0     IN      PTR      localhost.
134.159.168 IN      PTR      ns.liuyong666.com.
134.159.168 IN      PTR      www.liuyong666.com.
~
~
~
```

- 配置解析文件,添加nameserver

```
sudo vim /etc/resolv.conf
```



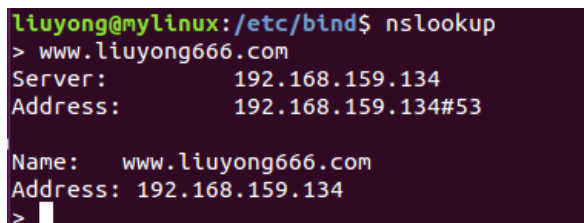
```
终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
liuyong@mylinux: /etc/bind
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 192.168.159.134
nameserver 127.0.1.1
search localdomain
~
~
~
```

重启bind服务

```
/etc/init.d/bind9 restart
```

```
nslookup
```

```
> www.liuyong666.com
```



```
liuyong@mylinux: /etc/bind$ nslookup
> www.liuyong666.com
Server:      192.168.159.134
Address:     192.168.159.134#53

Name:   www.liuyong666.com
Address: 192.168.159.134
>
```

- 说明能够正常解析

DHCP服务器

安装DHCP服务

```
sudo apt-get install dhcp3-server
```

配置DHCP网卡

```
emacs /etc/default/isc-dhcp-server
```

改为对应网卡

```
INTERFACES="eth0"
```

编辑当前配置

```
sudo vi /etc/dhcp/dhcpd.conf #删掉所有内容
```

```
emacs /etc/dhcp/dhcpd.conf
```

```
ddns-update-style none;
option domain-name "tagpt.mtn";
default-lease-time 14400;
#最小租约14400秒=4小时
max-lease-time 36000;
#最大租约36000秒=10小时
subnet 192.168.2.0 netmask 255.255.255.0 {
#IP地址起止范围
range 192.168.2.77 192.168.2.240;
option subnet-mask 255.255.255.0;
#子网掩码 255.255.255.0
option routers 192.168.2.10;
#默认网关 192.168.2.10
option broadcast-address 192.168.2.255;
#广播地址 192.168.2.255
}
```

- 不过现在还不能启动服务器，现在要先去Ubuntu的网络设置那里，把eth0的ipv4 setting修改一下，其中Method设置为Manual。Address添加一个192.168.2.0，Newmask:255.255.255.0然后Apply，并连接eth0就行。

重启服务

```
sudo /etc/init.d/isc-dhcp-server restart
```

或

```
service isc-dhcp-server restart
```

客户端设置为dhcp获取IP即可