

# 常用的网络命令大盘点

菜鸟教程 今天

以下文章来源于小林coding，作者小林coding



**小林coding**

图解得了技术，谈吐得了烟火。



来源 | 小林coding

作者 | 小林coding

服务器一般都是命令页面，不像 windows 有图形页面，点点鼠标就好，所以掌握些基本的 Linux 命令是很有必要的，不然就无法操作 Linux，更体会不到 Linux 的精髓。

这次，我们就来看看网络相关的命令。

学习网络不应该只局限于理论，作为工程师的我们，掌握一些基本的网络命令对我们帮助会很大，因为平时在远程操作、开发、调试、排查线上问题的时候，会常常用到。

Linux 为我们提供了很多网络相关的命令，我们这次就来看看 Linux 系统里有哪些常用的网络命令。

## 远程连接命令

如果我们要想操作 Linux 服务器，不可能说拿个显示器、鼠标和键盘接到服务器上，服务器一般都是放在机房里的，只需让服务器把网络接通，我们在自己的电脑就可以使用 ssh 命令远程登录服务器，进而操作和管理服务器。

还有一个很常用的远程命令是 scp，它可以帮助我们传输文件到服务器上。

### ssh

在需要远程登录 Linux 系统，可以使用 ssh 命令，比如你想远程登录一台服务器，可以使用 `ssh user@ip` 的方式，如下图：

```
[xiaolin@MacBook-Pro-3 / % ssh root@121.43.173.240  
root@121.43.173.240's password: 
```

接着，会有输入密码的提示，输入正确的密码后，就进入到了服务器的终端页面，之后你操作的命令就是控制服务器的了。

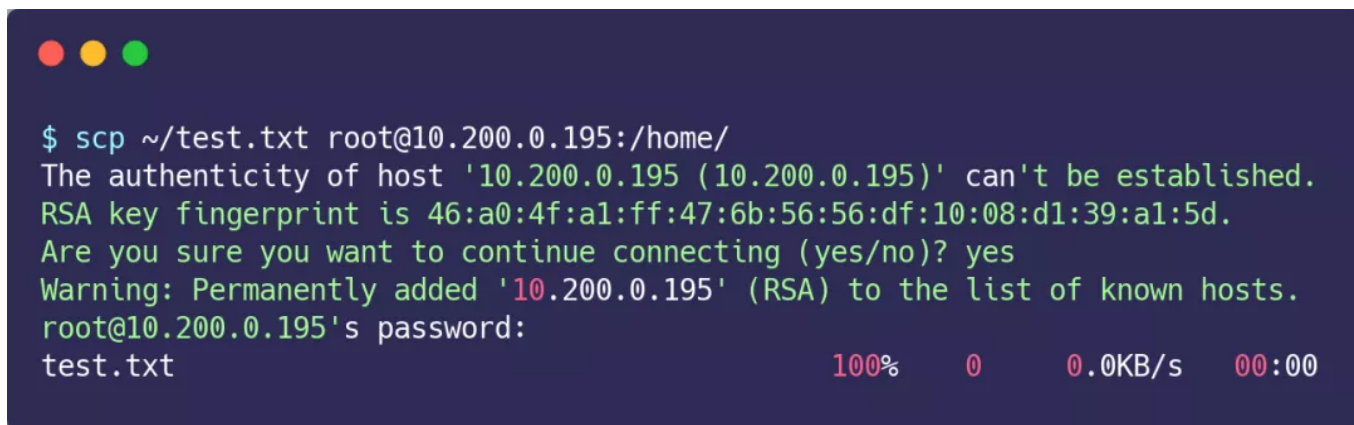
```
[xiaolin@MacBook-Pro-3 / % ssh root@121.43.173.240  
root@121.43.173.240's password:  
Last failed login: Sun Jul 18 14:36:43 CST 2021 from 47.106.250.53 on ssh:notty  
There were 3 failed login attempts since the last successful login.  
Last login: Sat Jul 17 14:28:32 2021 from 120.235.129.40  
  
Welcome to Alibaba Cloud Elastic Compute Service !  
  
[root@xiaolin ~]#
```

进入服务器成功!

## scp

当我们需要把一台机器上的文件传输给另一台机器时，使用 scp 命令就可以。

如下图，使用 scp 命令将本地 test.txt 文件传输给了 IP 地址为 192.168.12.35 机器的 /home 目录。

A terminal window with a dark blue background and white text. It shows the execution of the scp command to transfer a file from the local machine to a remote host. The command is: \$ scp ~/test.txt root@10.200.0.195:/home/. The terminal output shows a warning about the host's authenticity, a confirmation to continue, and the successful completion of the file transfer.

```
$ scp ~/test.txt root@10.200.0.195:/home/
The authenticity of host '10.200.0.195 (10.200.0.195)' can't be established.
RSA key fingerprint is 46:a0:4f:a1:ff:47:6b:56:56:df:10:08:d1:39:a1:5d.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.200.0.195' (RSA) to the list of known hosts.
root@10.200.0.195's password:
test.txt                               100%   0   0.0KB/s   00:00
```

输入 scp 命令后，会弹出需要输入对方密码的提示，输入完成后，回车即可，如果密码验证通过后，就进行文件的传输。

---

### 查看本地网络状态

要想知道本地机器的网络状态，比较常用的网络命令是 ifconfig 和 netstat。

## ifconfig

当你想知道机器上有哪些网口，和网口对应的状态信息时，使用 `ifconfig` 就可以，状态信息包含 IP 地址、子网掩码、MAC 地址等。

如下图，是在我设备上的 `ifconfig` 信息。

```
$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:69:AA:B6
          inet addr:192.168.12.35  Bcast:192.168.12.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe69:aab6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:30660 errors:0 dropped:0 overruns:0 frame:0
          TX packets:254 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2130670 (2.0 MiB)  TX bytes:107166 (104.6 KiB)

eth1      Link encap:Ethernet  HWaddr 00:0C:29:69:AA:CA
          inet addr:10.200.1.60  Bcast:10.200.1.255  Mask:255.255.254.0
          inet6 addr: fe80::20c:29ff:fe69:aaca/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
          RX packets:69893 errors:0 dropped:0 overruns:0 frame:0
          TX packets:198 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:18513826 (17.6 MiB)  TX bytes:32604 (31.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1839 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1839 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:138256 (135.0 KiB)  TX bytes:138256 (135.0 KiB)
```

可以看到，这台机器一共有 3 个网口，分别是 eth0、eth1、lo。其中 lo 是本地回路，发送给 lo 就相当于发送给自己，eth0 和 eth1 都是真实的网口。

## netstat

netstat 命令主要用于查看目前本机的网络使用情况。

### 查看所有 socket

如果只是单纯执行 netstat 命令，则查询的是本地所有 socket，如下图：

```

$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:47564         localhost:mysql         TIME_WAIT
tcp        0      0 localhost:47568         localhost:mysql         TIME_WAIT

tcp        0      0 localhost:47571         localhost:mysql         TIME_WAIT
tcp        0      0 localhost:47570         localhost:mysql         TIME_WAIT
tcp        0      0 192.168.12.35:ssh      192.168.12.20:63770     ESTABLISHED
tcp        0      0 localhost:47569         localhost:mysql         TIME_WAIT
getnameinfo failed
getnameinfo failed
tcp        0      0 [UNKNOWN]:http         [UNKNOWN]:64404        FIN_WAIT2
getnameinfo failed
getnameinfo failed
tcp        0      0 [UNKNOWN]:http         [UNKNOWN]:64403        FIN_WAIT2
getnameinfo failed
getnameinfo failed
tcp        0      0 [UNKNOWN]:http         [UNKNOWN]:64405        FIN_WAIT2
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags   Type       State       I-Node Path
unix    2      [ ]     DGRAM          9596 @/org/kernel/udev/udev
unix   13      [ ]     DGRAM          13185 /dev/log
unix    2      [ ]     DGRAM          13834 @/org/freedesktop/hal/udev_event
unix    2      [ ]     DGRAM          17966
unix    3      [ ]     STREAM        CONNECTED    17844 /var/run/dbus/system_bus_socket
unix    3      [ ]     STREAM        CONNECTED    17843
unix    3      [ ]     STREAM        CONNECTED    17830 @/tmp/gdm-session-zG0BeZQk
unix    3      [ ]     STREAM        CONNECTED    17829
...

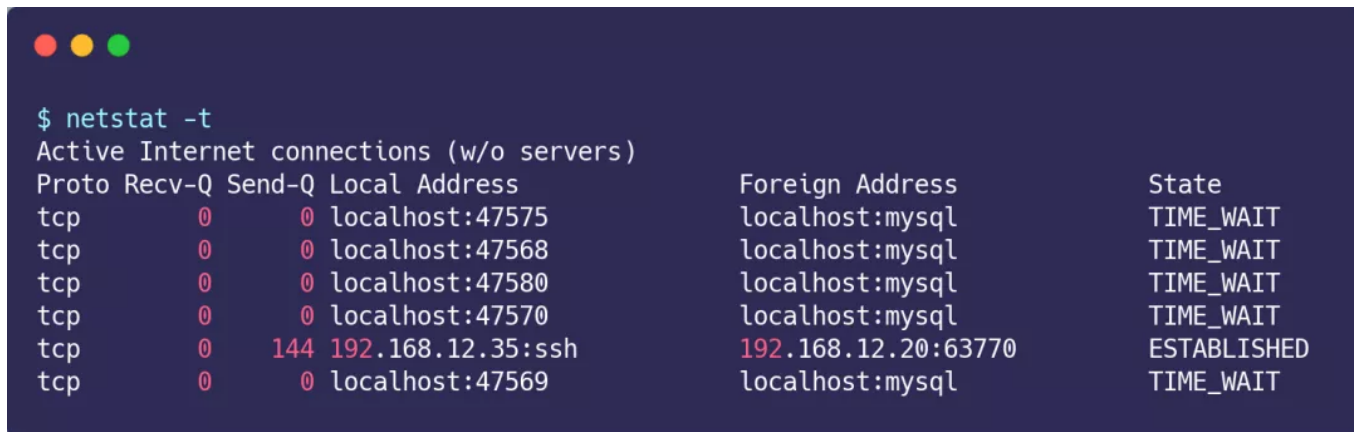
```

上图中，我们看到的都是 socket 文件，socket 负责在客户端与服务端之间收发数据，当客户端和服务端建立连接时，各自同时都会生成一个 socket 文件，用于管理这个连接。

## 查看 TCP/UDP 连接

如果只想看 TCP 连接的网络信息，可以使用 `netstat -t`。

比如下面我通过 `netstat -t` 看 tcp 协议的网络情况：

A terminal window with a dark blue background and light blue text. The command '\$ netstat -t' has been executed. The output shows 'Active Internet connections (w/o servers)' followed by a table of TCP connections. The table has columns for 'Proto', 'Recv-Q', 'Send-Q', 'Local Address', 'Foreign Address', and 'State'. There are six rows of data. The fifth row shows an established connection to 192.168.12.20:63770 via ssh, with a red '144' in the Send-Q column and a red '192.168.12.20:63770' in the Foreign Address column. The other rows show connections to localhost:mysql in a TIME\_WAIT state.

```
$ netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 localhost:47575         localhost:mysql        TIME_WAIT
tcp      0      0 localhost:47568         localhost:mysql        TIME_WAIT
tcp      0      0 localhost:47580         localhost:mysql        TIME_WAIT
tcp      0      0 localhost:47570         localhost:mysql        TIME_WAIT
tcp      0    144 192.168.12.35:ssh      192.168.12.20:63770    ESTABLISHED
tcp      0      0 localhost:47569         localhost:mysql        TIME_WAIT
```

上图末尾的 `state` 描述的是当前 TCP 连接处于的状态。

另外，如果要想看 UDP 的网络信息，可以使用 `netstat -u`。

## 查看端口占用

如果你想知道某个端口是哪个进程在占用，比如我想查 80 端口被哪个进程占用了，如下图：



A terminal window with a dark blue background and three colored window control buttons (red, yellow, green) in the top left corner. It displays the output of the command 'netstat -ntlp | grep 80'. The output shows a single line: 'tcp 0 0 :::80 :::\* LISTEN 2110/httpd'.

```
# -n 是将一些特殊的端口号用数字显示  
# -t 是指看 TCP 协议  
# -l 是只显示连接中的连接，  
# -p 是显示程序名称  
$ netstat -ntlp | grep 80  
tcp 0 0 :::80 :::* LISTEN 2110/httpd
```

可以看到，80 端口被 http 进程占用了，最末尾的信息也能看到这个进程对应的 pid。


---

## 网络测试

当我们想确认网络的延时情况，以及与服务器网络是否畅通，则可以使用 ping 和 telnet 命令。

### ping

想知道本机到目标网页的网络延时，可以使用 ping 命令，如下图所示：



```
# -c 5 通信 5 次
$ ping 192.168.12.20 -c 5
PING 192.168.12.20 (192.168.12.20) 56(84) bytes of data.
64 bytes from 192.168.12.20: icmp_seq=1 ttl=128 time=0.756 ms
64 bytes from 192.168.12.20: icmp_seq=2 ttl=128 time=0.383 ms
64 bytes from 192.168.12.20: icmp_seq=3 ttl=128 time=0.492 ms
64 bytes from 192.168.12.20: icmp_seq=4 ttl=128 time=0.390 ms
64 bytes from 192.168.12.20: icmp_seq=5 ttl=128 time=0.479 ms

--- 192.168.12.20 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4002ms
rtt min/avg/max/mdev = 0.383/0.500/0.756/0.135 ms
```

ping 是基于 ICMP 协议的，所以对方防火墙如果屏蔽了 ICMP 协议，那么我们就无法与它 ping 通，但这并不代表网络是不通的。

每一个 ICMP 包都有序号，所以你可以看到上图中 icmp 序号，如果序号是断断续续的，那么可能出现了丢包现象。

time 显示了网络包到达远程主机后返回的时间，单位是毫秒。time 的时间越小，说明网络延迟越低，如果你看到 time 的时间变化很大，这种现象叫做网络抖动，这说明客户端与服务器之间的网络状态不佳。

ttl 全称叫 time to live，指定网络包被路由器丢弃之前允许通过的网段数量，说白了就是定义了网络包最大经过路由器的数量，这个目的是防止网络包在网络中被无限转发，永不停止。当网络包在网络中被传输时，ttl 的值通过一个路由器时会

递减1，当 ttl 递减到 0 时，网络包就会被路由器抛弃。

另外，ping 不单单能输入 ip 地址，也能输入域名地址，如果输入的是域名地址，会先通过 DNS 查询该域名的 ip 地址，再进行通信。

## telnet

有时候，我们想知道本机到某个 IP + 端口的网络是否通畅，也就是想知道对方服务器是否有对应该端口的进程，于是就可以使用 telnet 命令，如下所示：

```
telnet 192.168.0.5
```

telnet 执行后会进入一个交互式的页面，这时就可以填写你将要发送给对方的信息，比如你想发 HTTP 请求给服务器，那么你就可以写出 HTTP 请求的格式信息。

---

## DNS 查询

如果想知道 DNS 解析域名的过程，可以使用 host 和 dig 命令。

## host

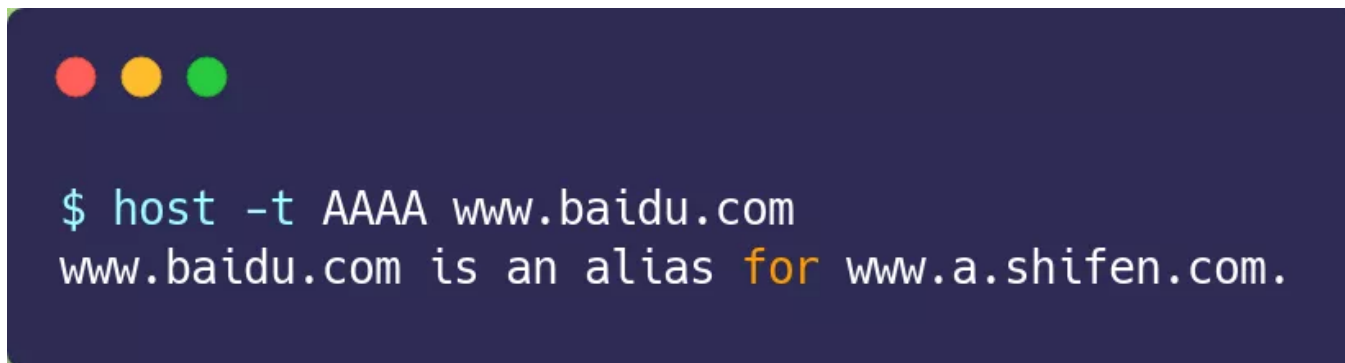
host 就是一个 DNS 查询命令，比如我们要查百度的 DNS，如下图所示：



```
$ host www.baidu.com
www.baidu.com is an alias for www.a.shifen.com.
www.a.shifen.com has address 14.215.177.39
www.a.shifen.com has address 14.215.177.38
```

可以看到，www.baidu.com 只是个别名，原名是 www.a.shifen.com，且对应了 2 条 IPv4 地址。

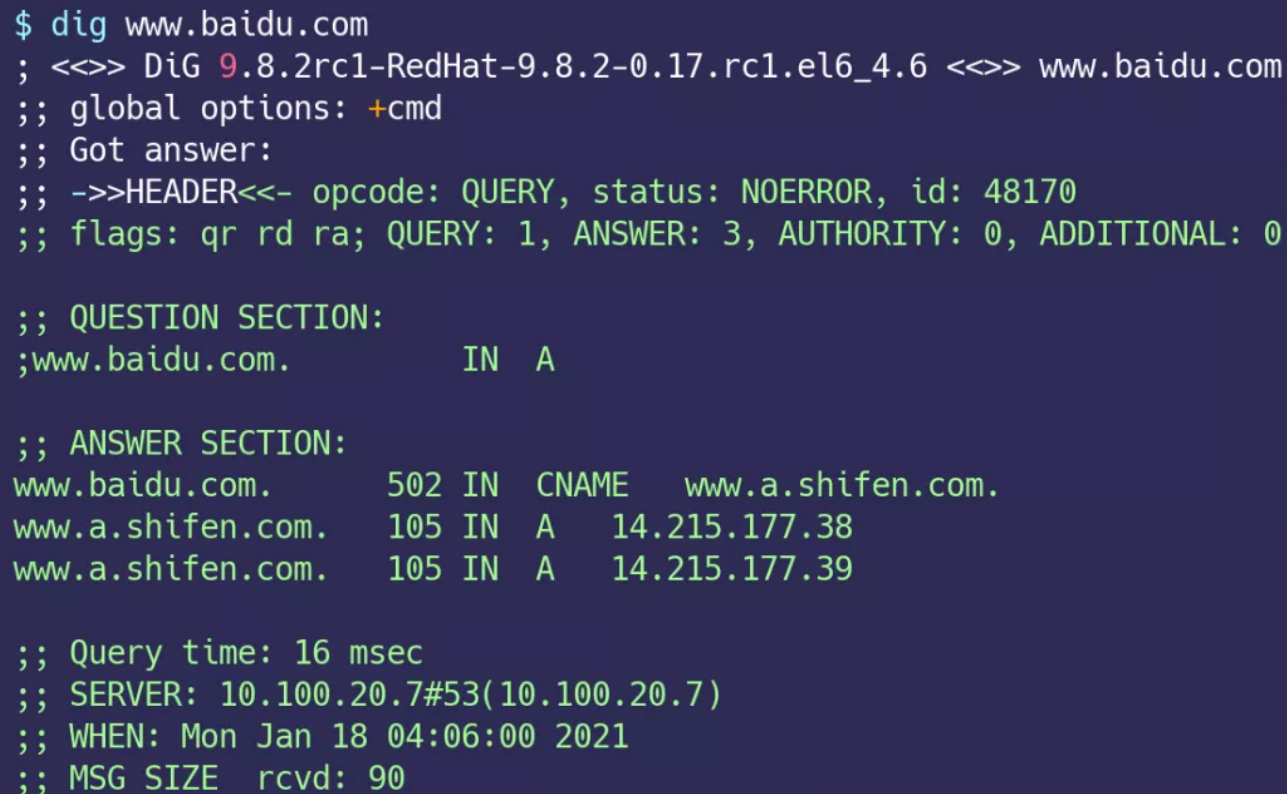
如果想追查某种类型的记录，可以加个 -t 参数，比如下图我们追查百度的 AAAA 记录，也就是查询域名对应的 IPv6 地址，由于百度还没部署 IPv6 地址，所以没有查询到。



```
$ host -t AAAA www.baidu.com
www.baidu.com is an alias for www.a.shifen.com.
```

## dig

dig 同样也是做 DNS 查询的，区别在于，dig 显示的内容更加详细，比如下图是 dig 百度的结果：



```
$ dig www.baidu.com
; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.17.rc1.el6_4.6 <<>> www.baidu.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48170
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.baidu.com.      IN  A

;; ANSWER SECTION:
www.baidu.com.      502 IN  CNAME  www.a.shifen.com.
www.a.shifen.com.   105 IN  A      14.215.177.38
www.a.shifen.com.   105 IN  A      14.215.177.39

;; Query time: 16 msec
;; SERVER: 10.100.20.7#53(10.100.20.7)
;; WHEN: Mon Jan 18 04:06:00 2021
;; MSG SIZE  rcvd: 90
```

也可以看到 `www.baidu.com` 的别名（CNAME）为 `www.a.shifen.com`，然后共有 2 条 A 记录，也就是 IPv4 地址的记录，通常对应多个是为了负载均衡或分发内容。

## HTTP

在电脑桌面我们常使用浏览器去请求网页，而在服务器一般是没有可视化页面的，也就没有浏览器，这时如果想要 HTTP 访问，就需要网络相关的命令。

### curl

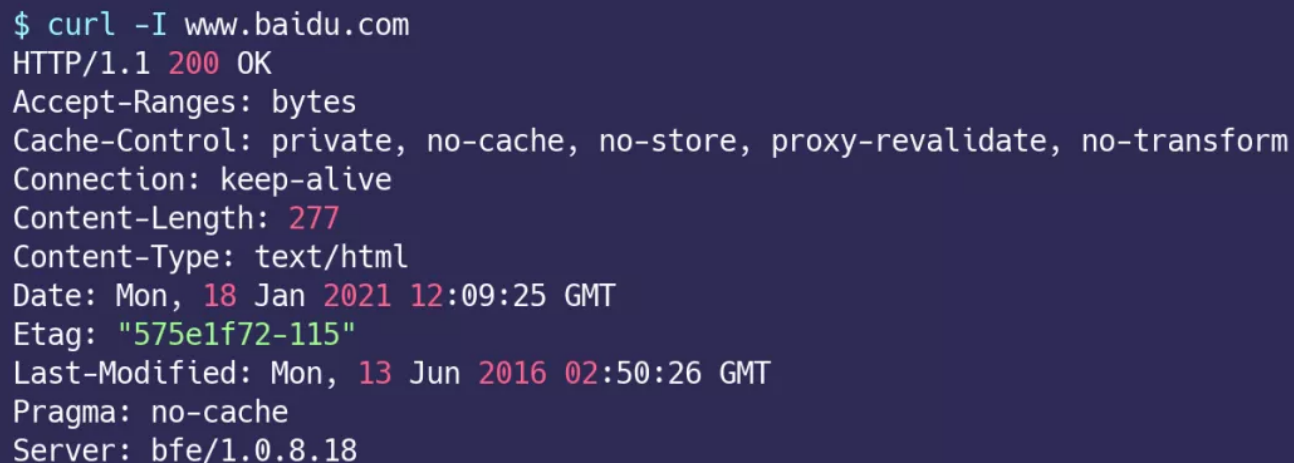
如果要在命令行请求网页或者接口，可以使用 curl 命令，curl 支持很多应用协议，比如 HTTP、FTP、SMTP 等，实际运用中最常用还是 HTTP。

比如，我用 curl 访问了百度网页，如下图：



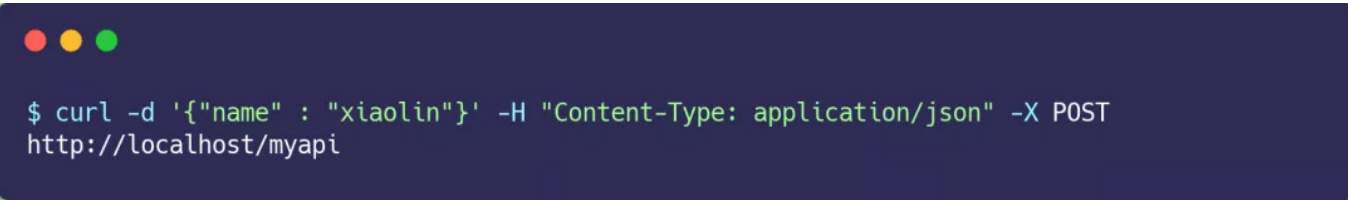
```
$ curl www.baidu.com
<!DOCTYPE html>
<!--STATUS OK--><html> <head><meta http-equiv=content-type content=text/html;charset=utf-
8><meta http-equiv=X-UA-Compatible content=IE=Edge><meta content=always name=referrer>
<link rel=stylesheet type=text/css
href=http://s1.bdstatic.com/r/www/cache/bdorz/baidu.min.css><title>百度一下，你就知道
</title></head> <body link=#0000cc> <div id=wrapper> <div id=head> <div
class=head_wrapper> <div class=s_form> <div class=s_form_wrapper> <div id=lg> <img
hidefocus=true src=//www.baidu.com/img/bd_logo1.png width=270 height=129> </div> <form
id=form
....
```

如果不想看 HTTP 数据部分，只想看 HTTP GET 返回头，可以再加个 `-I` 参数，如 `curl -I`，如下图所示：

A terminal window with a dark blue background and three colored window control buttons (red, yellow, green) in the top left corner. The terminal displays the output of the command `$ curl -I www.baidu.com`. The output shows HTTP headers for a 200 OK response from www.baidu.com, including Accept-Ranges, Cache-Control, Connection, Content-Length, Content-Type, Date, Etag, Last-Modified, Pragma, and Server information.

```
$ curl -I www.baidu.com
HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: private, no-cache, no-store, proxy-revalidate, no-transform
Connection: keep-alive
Content-Length: 277
Content-Type: text/html
Date: Mon, 18 Jan 2021 12:09:25 GMT
Etag: "575e1f72-115"
Last-Modified: Mon, 13 Jun 2016 02:50:26 GMT
Pragma: no-cache
Server: bfe/1.0.8.18
```

上面演示了 HTTP GET 请求，如果想使用 POST 请求，命令如下：

A terminal window with a dark blue background and three colored window control buttons (red, yellow, green) in the top left corner. The terminal displays the command `$ curl -d '{"name" : "xiaolin"}' -H "Content-Type: application/json" -X POST http://localhost/myapi`.

```
$ curl -d '{"name" : "xiaolin"}' -H "Content-Type: application/json" -X POST
http://localhost/myapi
```

`curl` 向 `http://localhost/myapi` 接口发送 POST 请求，各参数的说明：

- `-d` 后面是要发送的数据，例子中发送的是 JSON 格式的数据；
- `-X` 后面是指定 HTTP 的方法，例子中指定的是 POST 方法；

- -H 是指定自定义的请求头，例子中由于发送的是 JSON 数据，所以内容类型指定了 JSON。

---

## 总结

---

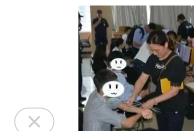
最后，列一下本文提到的 Linux 下常用的网络命令：

- 远程登录的 ssh 指令；
- 远程传输文件的 scp 指令；
- 查看网络接口的 ifconfig 指令；
- 查看网络状态的 netstat 指令；
- 测试网络延迟的 ping 指令；
- 可以和服务端进行交互式调试的 telnet 指令；
- 两个 DNS 查询指令 host 和 dig；
- 可以发送各种请求包括 HTTPS 的 curl 指令。

喜欢此内容的人还喜欢

一打卡作弊软件CEO被判5年6个月，网友：这也太...

漫话编程





程序员的九阳真经！

小林coding



小夕0.2秒居然复制了100G文件？

小夕学算法

