09 Linux 中的网络指令：如何查看一个域名有哪些 NS 记录?

我看到过一道关于 Linux 指令的面试题：如何查看一个域名有哪些 NS 记录?

这类题目是根据一个场景，考察一件具体的事情如何处理。虽然你可以通过查资料找到解决方案，但是，这类问题在面试中还是有必要穿插一下，用于确定求职者技能是否熟练、经验是否丰富。特别是计算机网络相关的指令，平时在远程操作、开发、联调、Debug 线上问题的时候，会经常用到。

Linux 中提供了不少网络相关的指令，因为网络指令比较分散，本课时会从下面几个维度给你介绍，帮助你梳理常用的网络指令：

- 远程操作；
- 查看本地网络状态；
- 网络测试；
- DNS 查询；
- HTTP。

这块知识从体系上属于 Linux 指令，同时也关联了很多计算机网络的知识，比如说 TCP/IP 协议、UDP 协议，我会在“**模块七**”为你简要介绍。

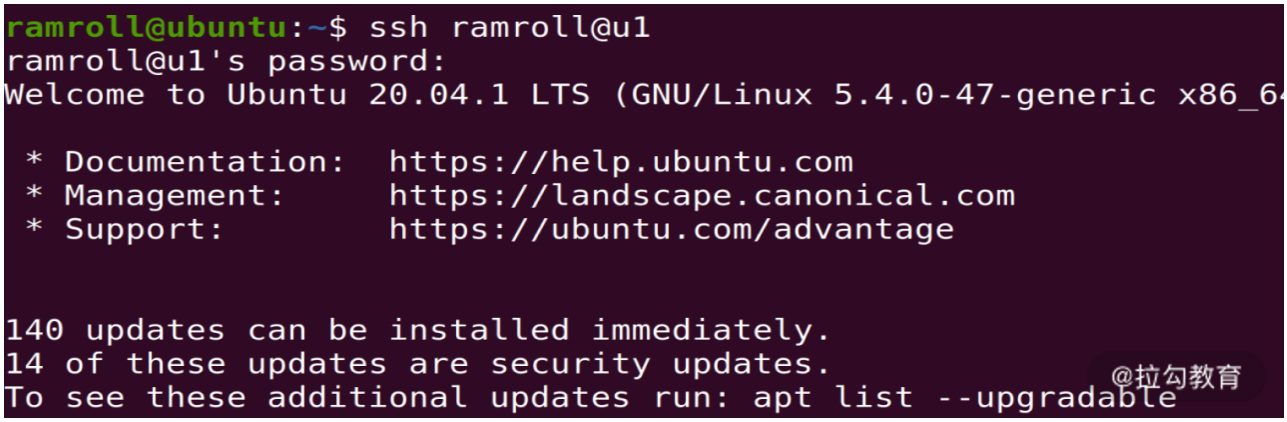
如果你对这部分指令背后的网络原理有什么困惑，可以在评论区提问。另外，你也可以关注我将在拉勾教育推出的《**计算机网络**》课程。下面我们开始学习今天的内容。

远程操作指令

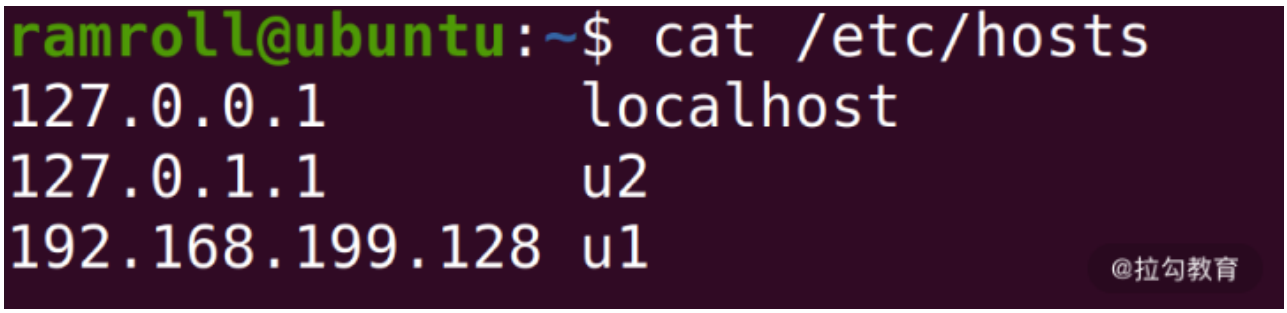
远程操作指令用得最多的是 `ssh`， `ssh` 指令允许远程登录到目标计算机并进行远程操作和管理。还有一个比较常用的远程指令是 `scp`， `scp` 帮助我们远程传送文件。

ssh（Secure Shell）

有一种场景需要远程登录一个 Linux 系统，这时我们会用到 `ssh` 指令。比如你想远程登录一台机器，可以使用 `ssh user@ip` 的方式，如下图所示：



上图中，我在使用 `ssh` 指令从机器 `u1` 登录我的另一台虚拟机 `u2` 。这里 `u1` 和 `u2` 对应着 IP 地址，是我在 `/etc/hosts` 中设置的，如下图所示：

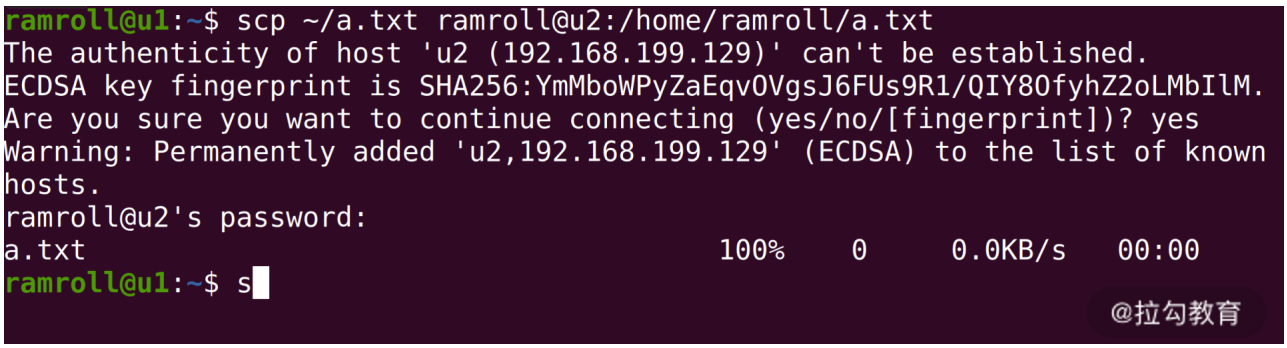


`/etc/hosts` 这个文件可以设置 IP 地址对应的域名。我这里是一个小集群，总共有两台机器，因此我设置了方便记忆和操作的名字。

scp

另一种场景是我需要拷贝一个文件到远程，这时可以使用 `scp` 指令，如下图，我使用 `scp` 指令将本地计算机的一个文件拷贝到了 ubuntu 虚拟机用户的家目录中。

比如从 `u1` 拷贝家目录下的文件 `a.txt` 到 `u2` 。家目录有一个简写，就是用 `~` 。具体指令见下图：



输入 scp 指令之后会弹出一个提示，要求输入密码，系统验证通过后文件会被成功拷贝。

查看本地网络状态

如果你想要了解本地的网络状态，比较常用的网络指令是 `ifconfig` 和 `netstat` 。

ifconfig

当你想知道本地 `ip` 以及本地有哪些网络接口时，就可以使用 `ifconfig` 指令。你可以把一个网络接口理解成一个网卡，有时候虚拟机会装虚拟网卡，虚拟网卡是用软件模拟的网卡。

比如：VMware 为每个虚拟机创造一个虚拟网卡，通过虚拟网卡接入虚拟网络。当然物理机也可以接入虚拟网络，它可以通过虚拟网络向虚拟机的虚拟网卡上发送信息。

下图是我的 ubuntu 虚拟机用 ifconfig 查看网络接口信息。

```
ramroll@u1:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.199.128  netmask 255.255.255.0  broadcast 192.168.199.25
5
        inet6 fe80::620a:8bfd:ff3:71b6  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:e1:a1:64  txqueuelen 1000  (Ethernet)
        RX packets 1500  bytes 825908 (825.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 488  bytes 50464 (50.4 KB)
        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
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 226  bytes 19279 (19.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 226  bytes 19279 (19.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0 @拉勾教育
```

可以看到我的这台 ubuntu 虚拟机一共有 2 个网卡，ens33 和 lo。 `lo` 是本地回路（local lookback），发送给 `lo` 就相当于发送给本机。 `ens33` 是一块连接着真实网络的虚拟网卡。

netstat

另一个查看网络状态的场景是想看目前本机的网络使用情况，这个时候可以用 `netstat` 。

默认行为

不传任何参数的 `netstat` 帮助查询所有的本地 socket，下图是 `netstat | less` 的结果。

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 u1:bootpc              192.168.199.254:bootps  ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ]         DGRAM                    29187    /run/systemd/journal/syslog
unix  16     [ ]         DGRAM                    29197    /run/systemd/journal/dev-log
unix  8      [ ]         DGRAM                    29201    /run/systemd/journal/socket
unix  2      [ ]         DGRAM                    46127    /run/user/1000/systemd/notify
unix  3      [ ]         DGRAM                    29173    /run/systemd/notify
unix  3      [ ]         STREAM     CONNECTED     46941
unix  3      [ ]         STREAM     CONNECTED     46762
unix  3      [ ]         STREAM     CONNECTED     43684
unix  3      [ ]         STREAM     CONNECTED     61231    /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     49927
unix  3      [ ]         STREAM     CONNECTED     23030    /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     45886
unix  3      [ ]         STREAM     CONNECTED     47467
unix  3      [ ]         STREAM     CONNECTED     47296
unix  3      [ ]         STREAM     CONNECTED     48225
unix  3      [ ]         STREAM     CONNECTED     40631    /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     47128    /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     58858
unix  3      [ ]         STREAM     CONNECTED     43702    @/tmp/dbus-tGDpM2Le8X
unix  3      [ ]         STREAM     CONNECTED     43522
unix  3      [ ]         STREAM     CONNECTED     49220    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     46551
unix  3      [ ]         STREAM     CONNECTED     45894
unix  3      [ ]         STREAM     CONNECTED     47510
```

@拉勾教育

如上图，我们看到的是 socket 文件。socket 是网络插槽被抽象成了文件，负责在客户端、服务器之间收发数据。当客户端和服务端发生连接时，客户端和服务端会同时各自生成一个 socket 文件，用于管理这个连接。这里，可以用 `wc -l` 数一下有多少个 `socket` 。

```
ramroll@u1:~$ netstat |wc -l
615
```

@拉勾教育

你可以看到一共有 615 个 socket 文件，因为有很多 socket 在解决进程间的通信。就是将两个进程一个想象成客户端，一个想象成服务端。并不是真的有 600 多个连接着互联网的请求。

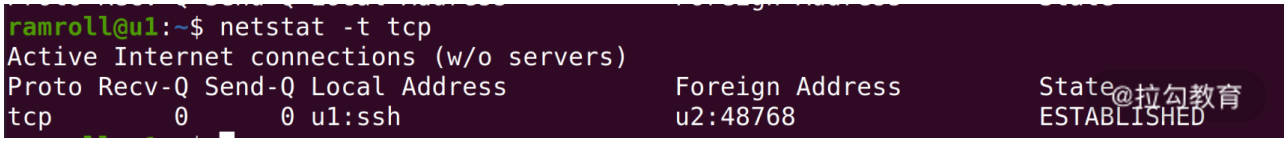
查看 TCP 连接

如果想看有哪些 TCP 连接，可以使用 `netstat -t` 。比如下面我通过 `netstat -t` 看 `tcp` 协议的网络情况：

```
ramroll@u1:~$ netstat -t tcp
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
```

@拉勾教育

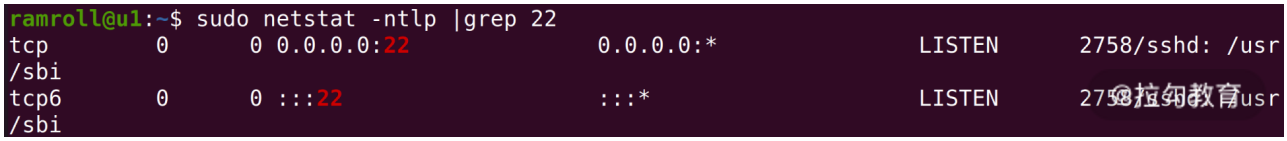
这里没有找到连接中的 tcp，因为我们这台虚拟机当时没有发生任何的网络连接。因此我们尝试从机器 u2 （另一台机器）ssh 登录进 u1，再看一次：



如上图所示，可以看到有一个 TCP 连接了。

查看端口占用

还有一种非常常见的情形，我们想知道某个端口是哪个应用在占用。如下图所示：



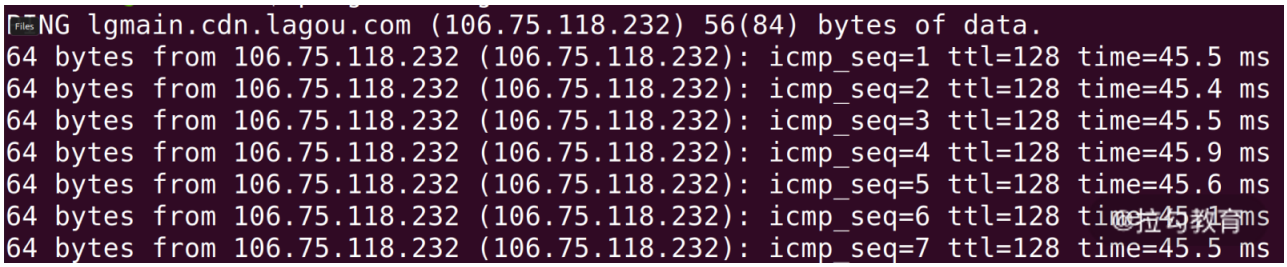
这里我们看到 22 端口被 sshd，也就是远程登录模块被占用了。 -n 是将一些特殊的端口号用数字显示， -t 是指看 TCP 协议， -l 是只显示连接中的连接， -p 是显示程序名称。

网络测试

当我们需要测试网络延迟、测试服务是否可用时，可能会用到 ping 和 telnet 指令。

ping

想知道本机到某个网站的网络延迟，就可以使用 ping 指令。如下图所示：



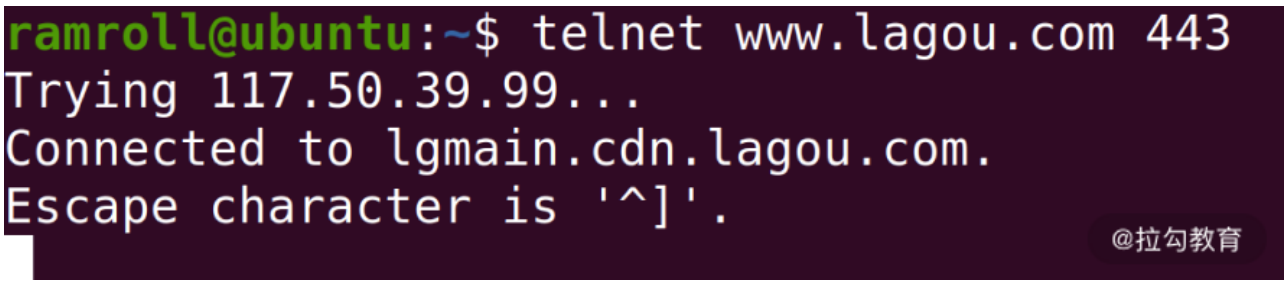
ping 一个网站需要使用 ICMP 协议。因此你可以在上图中看到 icmp 序号。 这里的时间 time 是往返一次的时间。 ttl 叫作 time to live，是封包的生存时间。就是说，一个封包从发出就开始倒计时，如果途中超过 128ms，这个包就会被丢弃。如果包被丢弃，就会被算进丢包率。

另外 ping 还可以帮助我们看到一个网址的 IP 地址。 通过网址获得 IP 地址的过程叫作 DNS Lookup（DNS 查询）。 ping 利用了 DNS 查询，但是没有显示全部的 DNS 查询结

果。

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在这个端口上提供了服务。这个时候可以用 `telnet` 指令。 如下图所示：



telnet 执行后会进入一个交互式的界面，比如这个时候，我们输入下图中的文字就可以发送 HTTP 请求了。如果你对 HTTP 协议还不太了解，建议自学一下 HTTP 协议。如果希望和林老师一起学习，可以等待下我之后的《**计算机网络**》专栏。

如上图所示，第 5 行的 `GET` 和第 6 行的 `HOST` 是我输入的。 拉勾网返回了一个 301 永久跳转。这是因为拉勾网尝试把 `http` 协议链接重定向到 `https` 。

DNS 查询

我们排查网络故障时想要进行一次 DNS Lookup，想知道一个网址 DNS 的解析过程。这个时候有多个指令可以用。

host

host 就是一个 DNS 查询工具。比如我们查询拉勾网的 DNS，如下图所示：

```
ramroll@ubuntu:~$ host www.lagou.com
www.lagou.com is an alias for lgmain.cdn.lagou.com.
lgmain.cdn.lagou.com has address 106.75.118.232
lgmain.cdn.lagou.com has address 117.50.39.99
lgmain.cdn.lagou.com has address 117.50.36.103
```

@拉勾教育

我们看到拉勾网 www.lagou.com 是一个别名，它的原名是 lgmain 开头的一个域名，这说明拉勾网有可能在用 CDN 分发主页（关于 CDN，我们《计算机网络》专栏见）。

上图中，可以找到 3 个域名对应的 IP 地址。

如果想追查某种类型的记录，可以使用 `host -t` 。比如下图我们追查拉勾的 AAAA 记录，因为拉勾网还没有部署 IPv6，所以没有找到。

```
ramroll@ubuntu:~$ host -t AAAA www.lagou.com
www.lagou.com is an alias for lgmain.cdn.lagou.com.
```

@拉勾教育

dig

`dig` 指令也是一个做 DNS 查询的。不过 `dig` 指令显示的内容更详细。下图是 `dig` 拉勾网的结果。

```
ramroll@ubuntu:~$ dig www.lagou.com

; <<>> DiG 9.16.1-Ubuntu <<>> www.lagou.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28133
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.lagou.com.                 IN      A

;; ANSWER SECTION:
www.lagou.com.          5       IN      CNAME   lgmain.cdn.lagou.com.
lgmain.cdn.lagou.com.   4       IN      A       106.75.118.232
lgmain.cdn.lagou.com.   4       IN      A       117.50.39.99
lgmain.cdn.lagou.com.   4       IN      A       117.50.36.103

;; Query time: 12 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
```

```
;; WHEN: Sun Sep 27 01:14:13 PDT 2020
;; MSG SIZE  rcvd: 115
```

@拉勾教育

从结果可以看到www.lagou.com 有一个别名，用 CNAME 记录定义 lgmain 开头的一个域名，然后有 3 条 A 记录，通常这种情况是为了均衡负载或者分发内容。

HTTP 相关

最后我们来说说 `http` 协议相关的指令。

curl

如果要在命令行请求一个网页，或者请求一个接口，可以用 `curl` 指令。 `curl` 支持很多种协议，比如 LDAP、SMTP、FTP、HTTP 等。

我们可以直接使用 curl 请求一个网址，获取资源，比如我用 curl 直接获取了拉勾网的主页，如下图所示：

```
ramroll@ubuntu:~$ curl https://www.lagou.com | head -n 10
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 11979    0 11979      <!DOCTYPE html>   0 --:--:-- --:--:-- --:--:--     0
<html>
<head>
        <!-- meta -->
        <meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
<meta name="renderer" content="webkit">
<meta property="qc:admins" content="23635710066417756375" />
<meta name="baidu-site-verification" content="QIQ6KC1oZ6" />
```

@拉勾教育

如果只想看 HTTP 返回头，可以使用 `curl -I` 。

另外 `curl` 还可以执行 POST 请求，比如下面这个语句：

```
curl -d '{"x" : 1}' -H "Content-Type: application/json" -X POST http://localhost:30
```

curl在向 `localhost:3000` 发送 POST 请求。 `-d` 后面跟着要发送的数据， `-X` 后面是用到的 HTTP 方法， `-H` 是指定自定义的请求头。

总结

这节课我们学习了不少网络相关的 Linux 指令，这些指令是将来开发和调试的强大工具。这里再给你复习一下这些指令：

- 远程登录的 ssh 指令；
- 远程拷贝文件的 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；
- 可以交互式调试和服务端的 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。

那么通过这节课的学习，你现在可以来回答本节关联的面试题目：如何查看一个域名有哪些 NS 记录了吗？

老规矩，请你先在脑海里构思下给面试官的表述，并把你的思考写在留言区，然后再来看我接下来的分析。

【解析】 host 指令提供了一个 `-t` 参数指定需要查找的记录类型。我们可以使用 `host -t ns {网址}` 。另外 dig 也提供了同样的能力。如果你感兴趣，还可以使用 `man` 对系统进行操作。

上一页 下一页