网络配置： `ifconfig、 ip`

网络配置

1.Linux查看本机的ip地址

- `ifconfig -a`
    - 每个设备的IP地址位于“inet”字符后面
    -

```
gongna@gongna-Ubuntu:~$ ifconfig
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  (本地环回)
        RX packets 9418716  bytes 2201245263 (2.2 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9418716  bytes 2201245263 (2.2 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp0s20f3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.43.100  netmask 255.255.255.0  broadcast 192.168.43.255
        inet6 fe80::9308:d734:2e8:bf4e  prefixlen 64  scopeid 0x20<link>
        inet6 2409:894d:c17:2330:64de:3943:5701:9123  prefixlen 64  scopeid 0x0<global>
        inet6 2409:894d:c17:2330:82:511a:153e:7d5f  prefixlen 64  scopeid 0x0<global>
        ether f8:ac:65:ba:1c:33  txqueuelen 1000  (以太网)
        RX packets 2632010  bytes 1573473377 (1.5 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2355043  bytes 776695504 (776.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

gongna@gongna-Ubuntu:~$ ip
Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }
       ip [ -force ] -batch filename
where  OBJECT := { link | address | addrlabel | route | rule | neigh | ntable |
                   tunnel | tuntap | maddress | mroute | mrule | monitor | xfrm |
                   netns | l2tp | fou | macsec | tcp_metrics | token | netconf | ila |
                   vrf | sr | nexthop }
       OPTIONS := { -V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] |
                    -h[uman-readable] | -iec | -j[son] | -p[retty] |
                    -f[amily] { inet | inet6 | mpls | bridge | link } |
                    -4 | -6 | -I | -D | -M | -B | -0 |
                    -l[oops] { maximum-addr-flush-attempts } | -br[ief] |
                    -o[neline] | -t[imestamp] | -ts[hort] | -b[atch] [filename] |
                    -rc[vbuf] [size] | -n[etns] name | -N[umeric] | -a[ll] |
                    -c[olor]}
```

- `hostname -I`

```
gongna@gongna-Ubuntu:~$ hostname -I
192.168.43.100 2409:894d:c17:2330:82:511a:153e:7d5f 2409:894d:c17:2330:64de:3943:5701:9123
gongna@gongna-Ubuntu:~$
```

- `ip addr show`

```
gongna@gongna-Ubuntu:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: wlp0s20f3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether f8:ac:65:ba:1c:33 brd ff:ff:ff:ff:ff:ff
    inet 192.168.43.100/24 brd 192.168.43.255 scope global dynamic noprefixroute wlp0s20f3
       valid_lft 3384sec preferred_lft 3384sec
    inet6 2409:894d:c17:2330:82:511a:153e:7d5f/64 scope global temporary dynamic
       valid_lft 3213sec preferred_lft 3213sec
    inet6 2409:894d:c17:2330:64de:3943:5701:9123/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 3213sec preferred_lft 3213sec
    inet6 fe80::9308:d734:2e8:bf4e/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

2.ip

- `ip route`查看路由

```
gongna@gongna-Ubuntu:~$ ip route
default via 192.168.43.1 dev wlp0s20f3 proto dhcp metric 600
169.254.0.0/16 dev wlp0s20f3 scope link metric 1000
192.168.43.0/24 dev wlp0s20f3 proto kernel scope link src 192.168.43.100 metric 600
```

连通性探测

1. `ping`判断网络的连通性和网速情况，偶尔用来查看域名的 IP，比如

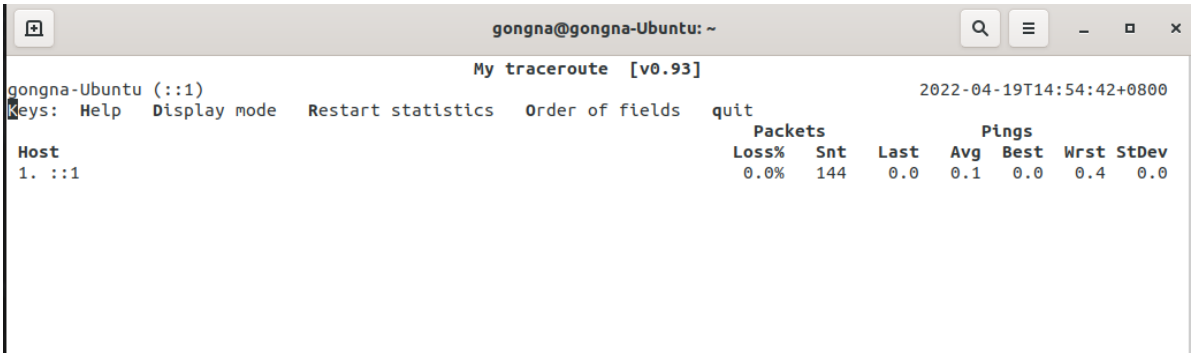
```
gongna@gongna-Ubuntu:~$ ping www.baidu.com
PING www.a.shifen.com (36.152.44.95) 56(84) bytes of data.
64 比特，来自 localhost (36.152.44.95): icmp_seq=1 ttl=52 时间=30.0 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=2 ttl=52 时间=69.7 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=3 ttl=52 时间=85.8 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=4 ttl=52 时间=82.2 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=5 ttl=52 时间=38.0 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=6 ttl=52 时间=77.8 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=7 ttl=52 时间=74.6 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=8 ttl=52 时间=49.9 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=9 ttl=52 时间=33.5 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=10 ttl=52 时间=46.1 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=11 ttl=52 时间=63.3 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=12 ttl=52 时间=192 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=13 ttl=52 时间=76.0 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=14 ttl=52 时间=40.4 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=15 ttl=52 时间=69.9 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=16 ttl=52 时间=56.6 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=17 ttl=52 时间=72.0 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=18 ttl=52 时间=222 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=19 ttl=52 时间=68.3 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=20 ttl=52 时间=47.2 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=21 ttl=52 时间=51.1 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=22 ttl=52 时间=90.8 毫秒
64 比特，来自 localhost (36.152.44.95): icmp_seq=23 ttl=52 时间=78.4 毫秒
```

2.`traceroute -I`探测源主机到目的主机之间的每一跳路由节点

```
gongna@gongna-Ubuntu:~$ traceroute -I  www.baidu.com
traceroute to www.a.shifen.com (36.152.44.96), 64 hops max
  1   192.168.43.1  32.006ms  3.016ms  2.721ms
  2   *  *  *
  3   172.21.0.97  26.210ms  13.536ms  14.854ms
  4   *  *  *
  5   *  *  *
  6   111.46.251.17  57.017ms  14.686ms  17.380ms
  7   211.137.61.25  17.818ms  18.225ms  17.460ms
  8   221.183.39.217  14.898ms  23.878ms  16.148ms
  9   221.183.41.194  56.323ms  41.429ms  39.241ms
 10   *  *  *
 11   *  *  *
 12   182.61.216.72  64.938ms  50.311ms  51.972ms
 13   *  *  *
 14   36.152.44.96  70.665ms  40.565ms  35.144ms
```

3.`mtr`从本地到源主机经过的所有路由，并显示每个路由间的丢包率、响应时间等。

```
mtr -n www.baidu.com
```



网络连接

1.`netstat`查看网络连接状况，主要是所有网络的连接。unix和socket,主要查看打开了哪些端口

```
gongna@gongna-Ubuntu:~$ netstat Inpt
激活Internet连接 (w/o 服务器)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:1089          localhost:39344         TIME_WAIT
tcp        0      0 localhost:8889          localhost:43680         ESTABLISHED
tcp        0      0 localhost:8889          localhost:43630         ESTABLISHED
tcp        0      0 gongna-Ubuntu:35692     120.232.208.195:19029   ESTABLISHED
tcp        0      0 localhost:8889          localhost:43632         ESTABLISHED
tcp        0      0 gongna-Ubuntu:35600     111.48.106.203:https    ESTABLISHED
tcp        0      0 localhost:1089          localhost:39352         TIME_WAIT
tcp        0      0 localhost:8889          localhost:43666         TIME_WAIT
tcp        0      0 localhost:43630         localhost:8889          ESTABLISHED
tcp        0      0 localhost:43682         localhost:8889          ESTABLISHED
tcp        0      0 localhost:43656         localhost:8889          ESTABLISHED
tcp        0      0 localhost:1089          localhost:39340         TIME_WAIT
tcp        0      0 localhost:1089          localhost:39280         ESTABLISHED
tcp        0      0 localhost:1089          localhost:39348         TIME_WAIT
tcp        0      0 localhost:43680         localhost:8889          ESTABLISHED
tcp        0      0 localhost:1089          localhost:39346         TIME_WAIT
tcp        0      0 localhost:1089          localhost:39342         TIME_WAIT
tcp        0      0 localhost:1089          localhost:39350         TIME_WAIT
tcp        0      0 localhost:8889          localhost:43656         ESTABLISHED
tcp        0      0 gongna-Ubuntu:35748     120.232.208.195:19029   ESTABLISHED
tcp        0      0 localhost:43632         localhost:8889          ESTABLISHED
tcp        0      0 localhost:15490         localhost:57260         ESTABLISHED
tcp        0      0 localhost:39280         localhost:1089          ESTABLISHED
tcp        0      0 gongna-Ubuntu:60856     112.14.22.104:19029     TIME_WAIT
tcp        0      0 localhost:8889          localhost:43682         ESTABLISHED
tcp        0      0 localhost:1089          localhost:39354         TIME_WAIT
tcp        0      0 gongna-Ubuntu:39408     120.232.214.210:19029   ESTABLISHED
tcp6       0      0 gongna-Ubuntu:37422     2409:8c4c:c00:40::https ESTABLISHED
tcp6       0      0 127.0.0.1:57260         127.0.0.1:15490         ESTABLISHED
tcp6       0      0 gongna-Ubuntu:37424     2409:8c4c:c00:40::https ESTABLISHED
udp        0      0 gongna-Ubuntu:bootpc    _gateway:bootps         ESTABLISHED
udp        0      0 gongna-Ubuntu:42684     _gateway:domain         ESTABLISHED
活跃的UNIX域套接字 (w/o 服务器)
Proto RefCnt Flags       Type       State         I-Node   路径
unix  2      [ ]         数据报                   47139    /run/user/1000/systemd/notify
unix  5      [ ]         数据报                   22196    /run/systemd/notify
unix  2      [ ]         数据报                   22210    /run/systemd/journal/syslog
unix  19     [ ]         数据报                   22220    /run/systemd/journal/dev-log
unix  8      [ ]         数据报                   22224    /run/systemd/journal/socket
```

2.`ss`也可以用来查看打开了哪些端口

```
gongna@gongna-Ubuntu:~$ ss
Netid  State   Recv-Q  Send-Q                                              Local Address:Port
                               Peer Address:Port                     Process
u_seq  ESTAB   0       0                                                   @0003a 4773945
                                          * 4773946
u_seq  ESTAB   0       0                                                   @0000d 54445
                                          * 54444
u_seq  ESTAB   0       0                                                   @0000e 54450
                                          * 54451
u_seq  ESTAB   0       0                                                   @0000f 54452
                                          * 54453
u_seq  ESTAB   0       0                                                   @0003b 4773947
                                          * 4773948
u_seq  ESTAB   0       0                                                   @0000a 54444
                                          * 54445
u_seq  ESTAB   0       0                                                   @0000b 49841
                                          * 49842
u_seq  ESTAB   0       0                                                   @0000c 49842
                                          * 49841
u_seq  ESTAB   0       0                                                   @00011 50158
                                          * 50159
u_seq  ESTAB   0       0                                                   @00010 51166
                                          * 51167
u_str  ESTAB   0       0                                                        * 4793061
                                          * 4793062
u_str  ESTAB   0       0                                                        * 4783944
                                          * 4783945
u_str  ESTAB   0       0                                                        * 4789583
                                          * 4789582
u_str  ESTAB   0       0                       @/dbus-vfs-daemon/socket-S7SNa5bU 313416
                                          * 313415
u_str  ESTAB   0       0                                                        * 53431
                                          * 55381
u_str  ESTAB   0       0                                     /run/user/1000/bus 48031
                                          * 47041
u_str  ESTAB   0       0                                     @/tmp/.X11-unix/X0 4762430
                                          * 4773955
u_str  ESTAB   0       0                                                        * 4754952
                                          * 4750944
```

3. `lsof` 用来列出当前系统打开的文件，进程的端口，可以用来查看与我自己主机某个端口的ipv4连接

```
gongna@gongna-Ubuntu:~/go/src/github.com$ lsof -i 6@192.168.43.100:80
lsof: IPv4 addresses are prohibited: -i 6@192.168.43.100:80
lsof 4.93.2
 latest revision: https://github.com/lsof-org/lsof
 latest FAQ: https://github.com/lsof-org/lsof/blob/master/00FAQ
 latest (non-formatted) man page: https://github.com/lsof-org/lsof/blob/master/L
sof.8
 usage: [-?abhKlnNoOPRtUvVX] [+|-c c] [+|-d s] [+D D] [+|-E] [+|-e s] [+|-f[gG]]
 [-F [f]] [-g [s]] [-i [i]] [+|-L [l]] [+m [m]] [+|-M] [-o [o]] [-p s]
 [+|-r [t]] [-s [p:s]] [-S [t]] [-T [t]] [-u s] [+|-w] [-x [fl]] [--] [names]
Use the ``-h'' option to get more help information.
```

4. `netcat(nc)`

- nc 被称为瑞士军刀，非常轻巧但功能强大，能够创建各种不同类型的网络连接
- 能够实现简单的聊天工具
- 远程传输文件
- debug 分析
- 扫描端口等。

```
nc -zv 192.168.43.100 1-1024 |grep 'succeeded'*
```

```
gongna@gongna-Ubuntu:~/go/src/github.com$ nc -zv 192.168.43.100 1-1024 |grep 'succeeded'
nc: connect to 192.168.43.100 port 1 (tcp) failed: Connection refused
nc: connect to 192.168.43.100 port 2 (tcp) failed: Connection refused
nc: connect to 192.168.43.100 port 3 (tcp) failed: Connection refused
nc: connect to 192.168.43.100 port 4 (tcp) failed: Connection refused
nc: connect to 192.168.43.100 port 5 (tcp) failed: Connection refused
nc: connect to 192.168.43.100 port 6 (tcp) failed: Connection refused
nc: connect to 192.168.43.100 port 7 (tcp) failed: Connection refused
nc: connect to 192.168.43.100 port 8 (tcp) failed: Connection refused
nc: connect to 192.168.43.100 port 9 (tcp) failed: Connection refused
nc: connect to 192.168.43.100 port 10 (tcp) failed: Connection refused
nc: connect to 192.168.43.100 port 11 (tcp) failed: Connection refused
nc: connect to 192.168.43.100 port 12 (tcp) failed: Connection refused
nc: connect to 192.168.43.100 port 13 (tcp) failed: Connection refused
nc: connect to 192.168.43.100 port 14 (tcp) failed: Connection refused
nc: connect to 192.168.43.100 port 15 (tcp) failed: Connection refused
nc: connect to 192.168.43.100 port 16 (tcp) failed: Connection refused
nc: connect to 192.168.43.100 port 17 (tcp) failed: Connection refused
nc: connect to 192.168.43.100 port 18 (tcp) failed: Connection refused
```

流量统计

1.ifstat

ifstat 主要用来监测主机网口的网络流量

```
ifstat -at 25
```

```
gongna@gongna-Ubuntu:~/go/src/github.com$ ifstat -a
        lo                wlp0s20f3
 KB/s in  KB/s out   KB/s in  KB/s out
    2.34      2.34      0.00      0.00
    2.42      2.42      0.00      0.00
    2.15      2.15      0.00      0.17
    2.53      2.53      0.06      0.08
    2.51      2.51      0.00      0.00
    2.33      2.33      0.00      0.00
    2.42      2.42      0.06      0.08
    2.43      2.43      0.00      0.00
    2.69      2.69      0.34      0.99
    4.34      4.34      0.53      1.72
    4.21      4.21      0.86      1.26
    2.69      2.69      0.57      1.38
    2.52      2.52      0.05      0.34
    2.31      2.31      0.29      0.29
    2.54      2.54      0.15      0.27
    2.09      2.09      0.00      0.18
    2.52      2.52      0.06      0.27
    2.41      2.41      0.80      0.00
    2.55      2.55      1.42      0.62
    2.74      2.74      1.30      0.70
    2.73      2.73      0.89      0.11
    2.28      2.28      0.80      0.00
    4.10      4.10      0.91      1.50
    4.00      4.00      0.27      0.59
    2.54      2.54      0.73      1.34
    2.48      2.48      0.30      0.34
    4.69      4.69      1.03      1.60
    5.28      5.28      0.71      2.17
    2.55      2.55      0.21      0.27
    2.22      2.22      0.06      0.08
^C
gongna@gongna-Ubuntu:~/go/src/github.com$ ifstat -at 25
  Time            lo                wlp0s20f3
HH:MM:SS   KB/s in  KB/s out   KB/s in  KB/s out
15:15:01      3.32      3.32      0.40      0.88
```

2.sar

sar 是一个系统历史数据统计工具

sar 是一个系统历史数据统计工具。统计的信息非常全，包括 CPU、内存、磁盘 I/O、网络、进程、系统调用等等信息。网络信息通常使用 -n参数来统计

交换与路由

1.arp

用来管理主机的 ARP 缓存，增删查改等。

```
gongna@gongna-Ubuntu:~/go/src/github.com$ arp
地址                     类型    硬件地址            标志  Mask            接口
_gateway                 ether   e2:4d:60:b5:9e:82   C                     wlp0s20f3
gongna@gongna-Ubuntu:~/go/src/github.com$
```

2.`arping -I wlp0s20f3 36.152.44.95 -c 1`指定从某个接口向某台主机发送 ARP 包，来获得 MAC 地址。

```
gongna@gongna-Ubuntu:~/go/src/github.com$ arping -I wlp0s20f3 36.152.44.95 -c 1
ARPING 36.152.44.95 from 192.168.43.100 wlp0s20f3
Sent 1 probes (1 broadcast(s))
Received 0 response(s)
```

3.vconfig

Linux vlan 配置命令，比如给某个接口增加两个 vlan

```
gongna@gongna-Ubuntu:~/go/src/github.com$ vconfig

Warning: vconfig is deprecated and might be removed in the future, please migrate to ip(route2) as soon as possi
le!

Expecting argc to be 3-5, inclusive.  Was: 1

Usage: add             [interface-name] [vlan_id]
       rem             [vlan-name]
       set_flag        [interface-name] [flag-num]       [0 | 1]
       set_egress_map  [vlan-name]      [skb_priority]   [vlan_qos]
       set_ingress_map [vlan-name]      [skb_priority]   [vlan_qos]
       set_name_type   [name-type]

* The [interface-name] is the name of the ethernet card that hosts
  the VLAN you are talking about.
* The vlan_id is the identifier (0-4095) of the VLAN you are operating on.
* skb_priority is the priority in the socket buffer (sk_buff).
* vlan_qos is the 3 bit priority in the VLAN header
* name-type:  VLAN_PLUS_VID (vlan0005), VLAN_PLUS_VID_NO_PAD (vlan5),
              DEV_PLUS_VID (eth0.0005), DEV_PLUS_VID_NO_PAD (eth0.5)
* FLAGS:  1 REORDER_HDR  When this is set, the VLAN device will move the
            ethernet header around to make it look exactly like a real
            ethernet device.  This may help programs such as DHCPd which
            read the raw ethernet packet and make assumptions about the
            location of bytes.  If you don't need it, don't turn it on, because
            there will be at least a small performance degradation.  Default
            is OFF.
```

域名相关

1.host

host 命令是域名分析查询工具，用来测试域名系统工作是否正常。

```
host www.baidu.com
```

```
gongna@gongna-Ubuntu:~/go/src/github.com$ host www.baidu.com
www.baidu.com is an alias for www.a.shifen.com.
www.a.shifen.com has address 36.152.44.95
www.a.shifen.com has address 36.152.44.96
www.a.shifen.com has IPv6 address ::1
gongna@gongna-Ubuntu:~/go/src/github.com$
```