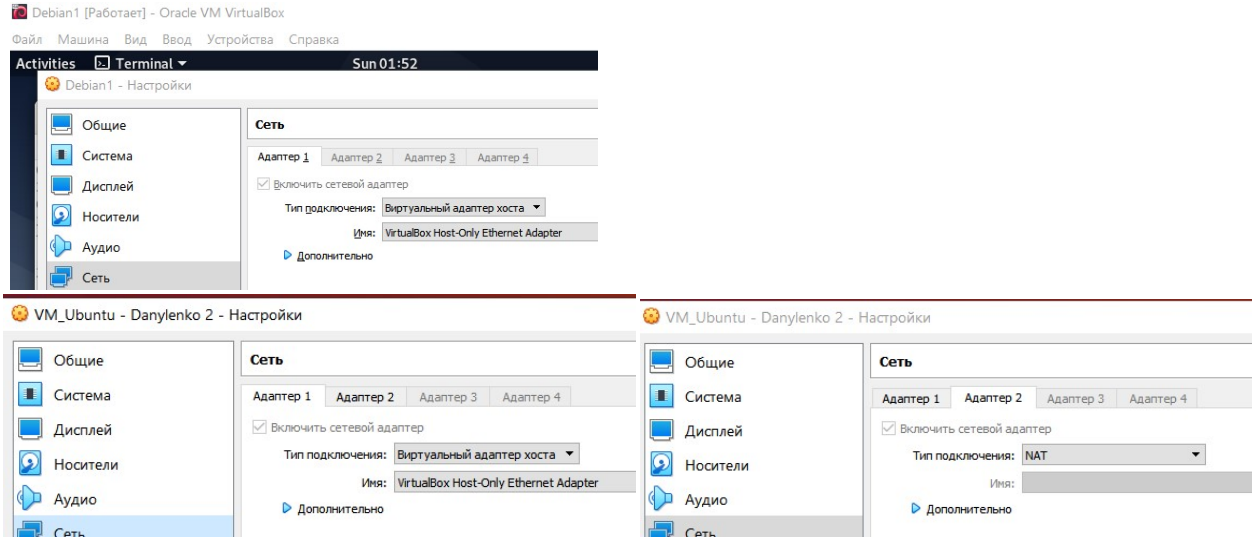
EPAM University Programs
DevOps external course
Module 5 Networking
TASK 5

Danylenko Homework

1. создать VM 1 на локальных ресурсах Debian OS
2. создать VM 2 на локальных ресурсах Ubuntu Os
3. создать VM 3 - EC2 линукс интанс on AWS.

сеть между VM 1 и VM 2 - хост онли нетворк. вторая сеть для VM 2 к хосту с гипервизором - NAT сеть.
настроить роутинг: VM 2 - дефаулт гейтвей для VM 1, для VM 2 - дефаулт роутер - хост с гипервизором.
настроить IPSEC VPN с VM 2 до VM3
продемонстировать трейс с VM 1 до google.com
добавить на VM 1-3 правила фаервола, которые запретят все, но позволять работать ссш и трейсроуту.

сеть между VM 1 и VM 2 - хост онли нетворк. вторая сеть для VM 2 к хосту с гипервизором - NAT сеть.



настроить роуминг: VM 2 - дефаулт гейтвей для VM 1, для VM 2 - дефаулт роутер - хост с гипервизором.

VM1:

```
danylenko@debian1:~$ cat /etc/network/interfaces | grep -ve "^$\|#"
source /etc/network/interfaces.d/*
auto lo
iface lo inet loopback
auto enp0s3
iface enp0s3 inet static
address 192.168.56.3
netmask 255.255.255.0
gateway 192.168.56.2
danylenko@debian1:~$ ip route show
default via 192.168.56.2 dev enp0s3
192.168.56.0/24 dev enp0s3 proto kernel scope link src 192.168.56.3
danylenko@debian1:~$ sudo traceroute google.com -T
traceroute to google.com (172.217.22.78), 30 hops max, 60 byte packets
 1  _gateway (192.168.56.2)  1.515 ms  1.566 ms  1.556 ms
 2  10.0.3.2 (10.0.3.2)  1.480 ms  1.470 ms  1.279 ms
 3  fra15s17-in-f78.1e100.net (172.217.22.78)  40.721 ms  37.308 ms  38.969 ms
```

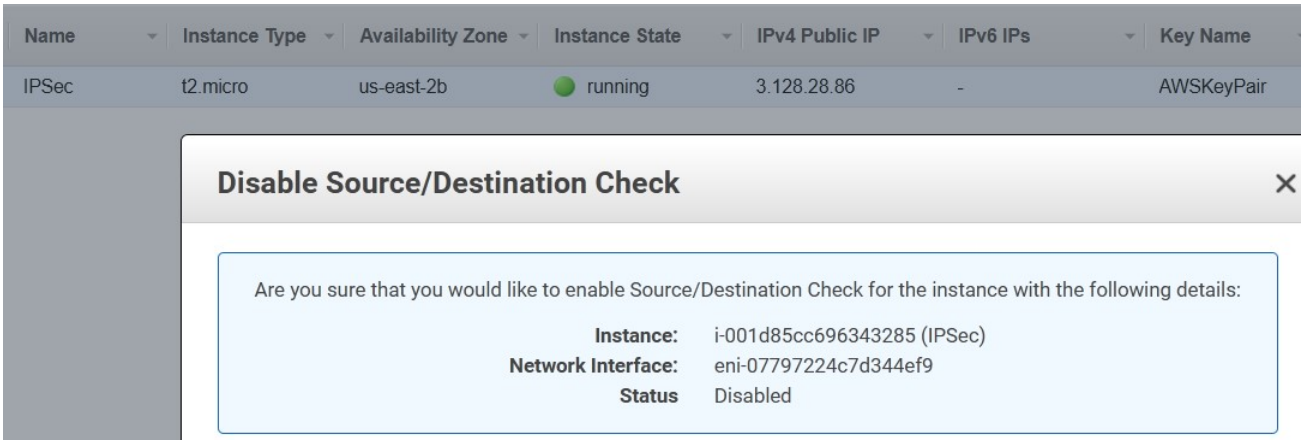
VM2:

```
danylenko@VM2:~/router$ sudo cat /etc/sysctl.conf | grep -e 'ipv4.*forward\|ipv4.*redirect'
net.ipv4.ip_forward=1
net.ipv4.conf.all.accept_redirects = 0
# net.ipv4.conf.all.secure_redirects = 1
#net.ipv4.conf.all.send_redirects = 0
danylenko@VM2:~/router$ ip route show
default via 10.0.3.2 dev enp0s8 proto dhcp metric 101
10.0.3.0/24 dev enp0s8 proto kernel scope link src 10.0.3.15 metric 101
169.254.0.0/16 dev enp0s3 scope link metric 1000
192.168.56.0/24 dev enp0s3 proto kernel scope link src 192.168.56.2 metric 102
danylenko@VM2:~/router$ sudo iptables -t nat -S
-P PREROUTING ACCEPT
-P INPUT ACCEPT
-P OUTPUT ACCEPT
-P POSTROUTING ACCEPT
-A POSTROUTING -o enp0s8 -m policy --dir out --pol ipsec -j ACCEPT
-A POSTROUTING -o enp0s8 -j MASQUERADE
danylenko@VM2:~/router$ sudo traceroute google.com -T
traceroute to google.com (216.58.192.238), 30 hops max, 60 byte packets
 1  _gateway (10.0.3.2)  0.212 ms  0.180 ms  0.163 ms
 2  ord30s26-in-f238.1e100.net (216.58.192.238)  136.511 ms  133.956 ms  135.924 ms
danylenko@VM2:~/router$
```

VM3 AWS:

```
[ec2-user@ip-172-31-22-24 ~]$ cat /etc/sysctl.conf | grep ipv4
net.ipv4.ip_forward = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
[ec2-user@ip-172-31-22-24 ~]$ ip route show
default via 172.31.16.1 dev eth0
169.254.169.254 dev eth0
172.31.16.0/20 dev eth0 proto kernel scope link src 172.31.22.24
[ec2-user@ip-172-31-22-24 ~]$ sudo iptables -t nat -S
-P PREROUTING ACCEPT
-P INPUT ACCEPT
-P OUTPUT ACCEPT
-P POSTROUTING ACCEPT
-A POSTROUTING -o eth0 -m policy --dir out --pol ipsec -j ACCEPT
-A POSTROUTING -o eth0 -j MASQUERADE
[ec2-user@ip-172-31-22-24 ~]$ traceroute google.com
traceroute to google.com (172.217.8.174), 30 hops max, 60 byte packets
 1  ec2-52-15-0-95.us-east-2.compute.amazonaws.com (52.15.0.95)  1.989 ms ec2-52-15-0-105
.us-east-2.compute.amazonaws.com (52.15.0.105)  8.236 ms ec2-52-15-0-103.us-east-2.comput
e.amazonaws.com (52.15.0.103)  7.616 ms
 2  100.65.26.32 (100.65.26.32)  1.002 ms 100.65.26.0 (100.65.26.0)  0.992 ms 100.65.24.0
 (100.65.24.0)  6.752 ms
 3  100.66.12.192 (100.66.12.192)  6.087 ms 100.66.12.216 (100.66.12.216)  3.637 ms 100.6
```

Disabled Source/Destinations Checks for NAT to work:



настроить IPSEC VPN с VM 2 до VM3

For IPSEC I used strongSwan on both sides with private key,

VM2 VM3 AWS:

With enabled policy routing:

```
danylenko@VM2:~/router$ cat /etc/ipsec.conf
config setup
        uniqueids = yes
        charondebug="all"

conn vm2-to-ec2
        type=tunnel
        auto=start
        keyexchange=ikev2
        authby=secret
        left=192.168.56.2
        leftid=213.110.102.151
        leftsubnet=0.0.0.0/0
        right=3.128.28.86
        rightsubnet=0.0.0.0/0
        ike=aes256-sha1-modp1024!
        esp=aes256-sha1!
        aggressive=no
        keyingtries=%forever
        ikelifetime=28800s
        lifetime=3600s
        dpddelay=30s
        dpdtimeout=120s
        dpdaction=restart
#       mark=4

conn ignorelan
        left=127.0.0.1
        leftsubnet=192.168.56.0/24
        rightsubnet=192.168.56.0/24
        authby=never
        type=passthrough
        auto=route
```

```
[ec2-user@ip-172-31-22-24 ~]$ sudo cat /etc/str
config setup
        charondebug="all"
        uniqueids=yes

conn ec2-to-vm2
        type=tunnel
        auto=start
        keyexchange=ikev2
        authby=secret
        left=172.31.22.24
        leftid=3.128.28.86
        leftsubnet=0.0.0.0/0
        right=%any
        rightsubnet=10.0.3.0/24
#,192.168.56.0/24
#       rightsourceip=10.0.4.15
        ike=aes256-sha1-modp1024!
        esp=aes256-sha1!
        aggressive=no
        keyingtries=%forever
        ikelifetime=28800s
        lifetime=3600s
        dpddelay=30s
        dpdtimeout=120s
        dpdaction=restart
[ec2-user@ip-172-31-22-24 ~]$
```

```
danylenko@VM2:~/router$ sudo ipsec status
Shunted Connections:
   ignorelan:  192.168.56.0/24 === 192.168.56.0/24 PASS
Security Associations (1 up, 0 connecting):
  vm2-to-ec2[1]: ESTABLISHED 4 seconds ago, 192.168.56.2[213.110.102.151]...3.128.28.86[3.128.28.86]
  vm2-to-ec2{1}:  INSTALLED, TUNNEL, reqid 1, ESP in UDP SPIs: c8d74a97_i c9313782_o
  vm2-to-ec2{1}:   10.0.3.0/24 === 0.0.0.0/0
danylenko@VM2:~/router$
```

```
[ec2-user@ip-172-31-22-24 ~]$ sudo strongswan status
Security Associations (1 up, 0 connecting):
  ec2-to-vm2[8]: ESTABLISHED 68 seconds ago, 172.31.22.24[3.128.28.86]...213.110.102.151[
213.110.102.151]
  ec2-to-vm2{7}:  INSTALLED, TUNNEL, reqid 7, ESP in UDP SPIs: c9313782_i c8d74a97_o
  ec2-to-vm2{7}:   0.0.0.0/0 === 10.0.3.0/24
[ec2-user@ip-172-31-22-24 ~]$
```

```
danylenko@VM2:~/router$ sudo ip xfrm policy
src 10.0.3.0/24 dst 0.0.0.0/0
        dir out priority 387711
        tmpl src 10.0.3.15 dst 3.128.28.86
                proto esp spi 0xc9313782 reqid 1 mode tunnel
src 0.0.0.0/0 dst 10.0.3.0/24
        dir fwd priority 387711
        tmpl src 3.128.28.86 dst 10.0.3.15
                proto esp reqid 1 mode tunnel
src 0.0.0.0/0 dst 10.0.3.0/24
        dir in priority 387711
        tmpl src 3.128.28.86 dst 10.0.3.15
                proto esp reqid 1 mode tunnel
src 192.168.56.0/24 dst 192.168.56.0/24
        dir fwd priority 175423
src 192.168.56.0/24 dst 192.168.56.0/24
        dir in priority 175423
src 192.168.56.0/24 dst 192.168.56.0/24
        dir out priority 175423
```

продемонстировать трейс с VM 1 до google.com

VM1 > google.com

```
danylenko@debian1:~$ traceroute google.com
traceroute to google.com (216.58.192.238), 30 hops max, 60 byte packets
 1  _gateway (192.168.56.2)  1.726 ms  1.615 ms  1.401 ms
 2  172.31.22.24 (172.31.22.24)  156.739 ms  156.094 ms  155.983 ms
 3  ec2-52-15-0-99.us-east-2.compute.amazonaws.com (52.15.0.99)  176.170 ms  175
.229 ms ec2-52-15-0-97.us-east-2.compute.amazonaws.com (52.15.0.97)  157.394 ms
 4  100.65.25.48 (100.65.25.48)  157.970 ms 100.65.26.32 (100.65.26.32)  212.687
 ms 100.65.27.16 (100.65.27.16)  162.417 ms
 5  100.66.12.152 (100.66.12.152)  162.757 ms 100.66.12.226 (100.66.12.226)  157
.466 ms 100.66.12.24 (100.66.12.24)  162.143 ms
 6  100.66.14.132 (100.66.14.132)  169.560 ms 100.66.14.34 (100.66.14.34)  169.2
90 ms 100.66.15.78 (100.66.15.78)  176.142 ms
 7  100.66.7.163 (100.66.7.163)  171.483 ms 100.66.7.5 (100.66.7.5)  162.108 ms
100.66.6.101 (100.66.6.101)  171.312 ms
```

VM2 > google.com

```
danylenko@VM2:~/router$ traceroute google.com
traceroute to google.com (216.58.192.238), 30 hops max, 60 byte packets
 1  172.31.22.24 (172.31.22.24)  154.145 ms  156.151 ms  155.859 ms
 2  ec2-52-15-0-99.us-east-2.compute.amazonaws.com (52.15.0.99)  159.520 ms ec2-52-15-0-97.us-east-2.co
mpute.amazonaws.com (52.15.0.97)  162.037 ms  161.992 ms
 3  100.65.27.0 (100.65.27.0)  159.787 ms 100.65.24.0 (100.65.24.0)  160.619 ms 100.65.24.32 (100.65.24
.32)  160.596 ms
 4  100.66.12.70 (100.66.12.70)  157.442 ms^C
danylenko@VM2:~/router$
```

добавить на VM 1-3 правила фаервола, которые запретят все, но позволять работать ссш и трейсроуту.

Since VM1-3 firewall rules were mentioned:

I made rules to ACCEPT FORWARD traffic thorough gateways VM2,VM3,

Only to drop INTUP and OUTPUT traffic on them, and all traffic drops from VM1-debian

VM3 AWS

```
[ec2-user@ip-172-31-22-24 ~]$ sudo iptables-save
# Generated by iptables-save v1.8.2 on Sun May 31 17:53:37 2020
*nat
:PREROUTING ACCEPT [2653:170492]
:INPUT ACCEPT [336:23739]
:OUTPUT ACCEPT [2002:156048]
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -o eth0 -m policy --dir out --pol ipsec -j ACCEPT
-A POSTROUTING -o eth0 -j MASQUERADE
COMMIT
# Completed on Sun May 31 17:53:37 2020
# Generated by iptables-save v1.8.2 on Sun May 31 17:53:37 2020
*filter
:INPUT DROP [0:0]
:FORWARD ACCEPT [166:14120]
:OUTPUT DROP [3:228]
-A INPUT -p icmp -m icmp --icmp-type 11 -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A INPUT -p udp -m udp --dport 33434:33524 -j ACCEPT
-A INPUT -p udp -m udp --dport 4500 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
-A INPUT -p udp -m udp --sport 4500 -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A INPUT -p udp -m udp --dport 500 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
-A INPUT -p udp -m udp --sport 500 -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A OUTPUT -p icmp -m icmp --icmp-type 11 -j ACCEPT
-A OUTPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A OUTPUT -p udp -m udp --dport 33434:33524 -j ACCEPT
-A OUTPUT -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A OUTPUT -p udp -m udp --sport 4500 -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A OUTPUT -p udp -m udp --dport 4500 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -p udp -m udp --sport 500 -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A OUTPUT -p udp -m udp --dport 500 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -p tcp -m tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT
COMMIT
# Completed on Sun May 31 17:53:37 2020
```

VM2 Ubuntu gateway

```
danylenko@VM2:~/router$ sudo iptables-save
# Generated by iptables-save v1.6.1 on Sun May 31 20:56:50 2020
*filter
:INPUT DROP [16:3678]
:FORWARD ACCEPT [637:52127]
:OUTPUT DROP [59:5069]
-A INPUT -p icmp -m icmp --icmp-type 11 -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A INPUT -p udp -m udp --dport 33434:33524 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A INPUT -p udp -m udp --sport 500 -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A INPUT -p udp -m udp --dport 500 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
-A INPUT -p udp -m udp --sport 4500 -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A INPUT -p udp -m udp --dport 4500 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -p icmp -m icmp --icmp-type 11 -j ACCEPT
-A OUTPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A OUTPUT -p udp -m udp --dport 33434:33524 -j ACCEPT
-A OUTPUT -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A OUTPUT -p tcp -m tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -p udp -m udp --dport 500 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -p udp -m udp --sport 500 -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A OUTPUT -p udp -m udp --dport 4500 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -p udp -m udp --sport 4500 -m conntrack --ctstate ESTABLISHED -j ACCEPT
COMMIT
# Completed on Sun May 31 20:56:50 2020
# Generated by iptables-save v1.6.1 on Sun May 31 20:56:50 2020
*nat
:PREROUTING ACCEPT [5589:430826]
:INPUT ACCEPT [658:69299]
:OUTPUT ACCEPT [3372:276953]
:POSTROUTING ACCEPT [1149:91238]
-A POSTROUTING -o enp0s8 -m policy --dir out --pol ipsec -j ACCEPT
-A POSTROUTING -o enp0s8 -j MASQUERADE
COMMIT
# Completed on Sun May 31 20:56:50 2020
```

VM1 client

```
danylenko@debian1:~$ sudo iptables -S
-P INPUT DROP
-P FORWARD DROP
-P OUTPUT DROP
-A INPUT -p icmp -m icmp --icmp-type 11 -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A INPUT -p udp -m udp --dport 33434:33524 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A OUTPUT -p icmp -m icmp --icmp-type 11 -j ACCEPT
-A OUTPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A OUTPUT -p udp -m udp --dport 33434:33524 -j ACCEPT
-A OUTPUT -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A OUTPUT -p tcp -m tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
danylenko@debian1:~$ scp danylenko@192.168.56.2:~/router/block-rules.save ~/block-rules
.save
danylenko@192.168.56.2's password:
block-rules.save                                      100% 1929     2.3MB/s   00:00
danylenko@debian1:~$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  _gateway (192.168.56.2)  0.423 ms  0.457 ms  0.444 ms
 2  172.31.22.24 (172.31.22.24)  154.323 ms  154.328 ms  154.283 ms
 3  52.15.0.95 (52.15.0.95)  157.324 ms 52.15.0.103 (52.15.0.103)  156.317 ms 52.15.0.9
9 (52.15.0.99)  167.125 ms
```

All Iptable and nat rules saved through iptable-save \ and applied on system restart on each VM with iptable-restore

```
[ec2-user@ip-172-31-22-24 ~]$ cat /etc/rc.local
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev rules
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure
# that this script will be executed during boot.

touch /var/lock/subsys/local
strongswan start
/home/ec2-user/iptables.sh
exit 0
[ec2-user@ip-172-31-22-24 ~]$ cat ~/iptables.sh
#!/bin/bash
iptables-restore < /etc/iptables.conf
```