Создать VM 1 на локальных ресурсах Debian OS.

I already have RaspberryPI for experiments with Raspbian, based on Debian OS.

I will use it like VM1:

192.168.1.99 - PuTTY

```
login as: root
root@192.168.1.99's password:
Linux SmartHome 4.19.97+ #1294 Thu Jan 30 13:10:54 GMT 2020 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed May 20 11:56:49 2020 from 192.168.1.131
root@SmartHome:~# lsb_release -a
No LSB modules are available.
Distributor ID: Raspbian
Description:    Raspbian GNU/Linux 10 (buster)
Release:        10
Codename:       buster
root@SmartHome:~# cat /proc/version
Linux version 4.19.97+ (dom@buildbot) (gcc version 4.9.3 (crosstool-NG crosstool
-ng-1.22.0-88-g8460611)) #1294 Thu Jan 30 13:10:54 GMT 2020
root@SmartHome:~#
```

Создать VM 2 на локальных ресурсах Ubuntu Os

I planned to change my router (Old Laptop with several netcards and FreeBSD 10) to new industrial PC (https://www.aliexpress.com/item/32817795313.html?spm=a2g0s.9042311.0.0.27424c4dONspAb), as Linux Router + SmartHome core (www.home-assistant.io - needs docker), so I use this task as a chance to do it ;)

I installed Ubuntu 20.04 LTS Server:

root@mini-pc: ~

```
root@mini-pc:~# cat /proc/version
Linux version 5.4.0-31-generic (buildd@lgw01-amd64-059) (gcc version 9.3.0 (Ubun
tu 9.3.0-10ubuntu2)) #35-Ubuntu SMP Thu May 7 20:20:34 UTC 2020
root@mini-pc:~# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04 LTS
Release:        20.04
Codename:       focal
root@mini-pc:~# ip  addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t 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: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 00:e0:67:19:45:4c brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.2/24 brd 192.168.1.255 scope global enp1s0
       valid_lft forever preferred_lft forever
    inet6 fe80::2e0:67ff:fe19:454c/64 scope link
       valid_lft forever preferred_lft forever
3: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 00:e0:67:19:45:4d brd ff:ff:ff:ff:ff:ff
    inet 5.248.37.215/21 brd 5.248.39.255 scope global dynamic enp2s0
       valid_lft 17643sec preferred_lft 17643sec
    inet6 fe80::2e0:67ff:fe19:454d/64 scope link
       valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOW
N group default
    link/ether 02:42:c5:dd:35:3b brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
root@mini-pc:~#
```

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

Rename and set netinterfaces on Ubuntu gateway via netplan conf:

```
root@mini-pc: ~
Last login: Sun May 24 13:17:20 2020 from 192.168.1.131
root@mini-pc:~# cat /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    lan:
      match:
       macaddress: 00:e0:67:19:45:4c
      addresses:
      - 192.168.1.2/24
      optional: true
      set-name: lan0
      nameservers:
        addresses:
        - 4.4.4.4
        - 8.8.8.8
    wan:
      match:
       macaddress: 00:e0:67:19:45:4d
      dhcp4: true
      optional: true
      set-name: wan0
  version: 2
root@mini-pc:~#
```

After checking by (netplan try) and reboot (it is needed for interface name changing):

```
root@mini-pc: ~
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.2  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::2e0:67ff:fe19:454c  prefixlen 64  scopeid 0x20<link>
        ether 00:e0:67:19:45:4c  txqueuelen 1000  (Ethernet)
        RX packets 170  bytes 22677 (22.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 56  bytes 9362 (9.3 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 84  bytes 6324 (6.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 84  bytes 6324 (6.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 5.248.37.215  netmask 255.255.248.0  broadcast 5.248.39.255
        inet6 fe80::2e0:67ff:fe19:454d  prefixlen 64  scopeid 0x20<link>
        ether 00:e0:67:19:45:4d  txqueuelen 1000  (Ethernet)
        RX packets 60  bytes 34554 (34.5 KB)
        RX errors 0  dropped 19  overruns 0  frame 0
        TX packets 49  bytes 8188 (8.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@mini-pc:~#
```

Настроить роутинг: VM 2 - дефаулт гейтвей для VM 1, для VM 2 - дефаулт роутер - хост с гипервизором.

Enable nat and forwarding on VM2:

sysctl net.ipv4.ip_forward=1

iptables -t nat -A POSTROUTING -o wan0 -j MASQUERADE

add ufw configuration

set default router VM2 on VM1 and check traceroute:

192.168.1.99 - PuTTY

```
root@SmartHome:~# ip route
default via 192.168.1.2 dev eth0
169.254.0.0/16 dev veth590ac82 scope link src 169.254.169.242 metric 206
169.254.0.0/16 dev veth8aef5b4 scope link src 169.254.84.233 metric 208
169.254.0.0/16 dev veth2bea1f6 scope link src 169.254.39.143 metric 210
169.254.0.0/16 dev veth38eb8c6 scope link src 169.254.177.218 metric 212
169.254.0.0/16 dev veth3064cc7 scope link src 169.254.156.211 metric 214
169.254.0.0/16 dev veth6134a25 scope link src 169.254.99.206 metric 216
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1
172.30.32.0/23 dev hassio proto kernel scope link src 172.30.32.1
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.99 metric 100
192.168.1.0/24 dev eth0 proto dhcp scope link src 192.168.1.99 metric 202
root@SmartHome:~# traceroute google.com
traceroute to google.com (172.217.16.142), 30 hops max, 60 byte packets
 1  192.168.1.2 (192.168.1.2)  0.861 ms  1.433 ms  0.655 ms
 2  * * *
 3  81-23-23-75.ip.kyivstar.net (81.23.23.75)  8.416 ms  8.292 ms  10.393 ms
 4  74.125.32.160 (74.125.32.160)  9.942 ms  10.183 ms  10.025 ms
 5  108.170.248.147 (108.170.248.147)  8.380 ms 108.170.248.131 (108.170.248.131
)  8.603 ms  8.109 ms
 6  209.85.248.105 (209.85.248.105)  23.940 ms  23.994 ms  23.376 ms
 7  209.85.241.99 (209.85.241.99)  39.039 ms  38.377 ms  37.565 ms
 8  209.85.242.78 (209.85.242.78)  36.102 ms  35.501 ms  35.808 ms
 9  108.170.251.129 (108.170.251.129)  36.586 ms 108.170.252.1 (108.170.252.1)
36.627 ms 108.170.251.129 (108.170.251.129)  35.585 ms
10  66.249.94.245 (66.249.94.245)  38.302 ms 66.249.95.169 (66.249.95.169)  38.9
70 ms 66.249.94.245 (66.249.94.245)  38.506 ms
11  fra15s46-in-f14.1e100.net (172.217.16.142)  37.908 ms  38.666 ms  38.515 ms
root@SmartHome:~#
```

Настроить IPSEC VPN с VM 2 до VM3

transport mode + racoon + PSK. (http://www.ipsec-howto.org/ipsec-howto.pdf)

I am planning to use these tutorials for VM2 and AMI2 (VM3)

https://www.digitalocean.com/community/tutorials/how-to-set-up-an-ikev2-vpn-server-with-strongswan-on-ubuntu-18-04-2

https://www.peternijssen.nl/connect-multiple-aws-regions-strongswan/

I organized IPSEC tunnel between my Lan 192.168.1.0/24 and AWS LAN 172.31.32.0/20 using strongswan

```
root@mini-pc: /etc
  auto=start
  type=tunnel
  left=37.115.53.98
  leftid=37.115.53.98
  leftsubnet=192.168.1.0/24
  leftauth=psk
  right=204.236.252.156
  rightsubnet=172.31.32.0/20
  rightauth=psk
  ike=aes128-sha1-modp1024
  esp=aes128-sha1-modp1024
root@mini-pc:/etc# cat  ipsec.conf
config setup
  strictcrlpolicy=no
  charondebug=all
conn %default
  ikelifetime=60m
  keylife=20m
  rekeymargin=3m
  keyingtries=1
  keyexchange=ikev2
conn AWS
  authby=secret
  auto=start
  type=tunnel
  left=37.115.53.98
  leftid=37.115.53.98
  leftsubnet=192.168.1.0/24
  leftauth=psk
  right=204.236.252.156
  rightsubnet=172.31.32.0/20
  rightauth=psk
  ike=aes128-sha1-modp1024
  esp=aes128-sha1-modp1024
root@mini-pc:/etc#
```

It is nedded to use

iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o wan0 -m policy --pol ipsec --dir out -j ACCEPT

before the rule:

iptables -t nat -A POSTROUTING -o wan0 -j MASQUERADE

to avoid implement NAT for IPSEC traffic.

```
root@ip-172-31-41-151:~
[root@ip-172-31-41-151 ~]# cat /etc/strongswan/ipsec.conf
config setup
  strictcrlpolicy=no
  charondebug=all
conn %default
  ikelifetime=60m
  keylife=20m
  rekeymargin=3m
  keyingtries=1
  keyexchange=ikev2
conn Kiyvstar
  authby=secret
  auto=start
  type=tunnel
  left=172.31.41.151
  leftid=204.236.252.156
  leftsubnet=172.31.32.0/20
  leftauth=psk
  right=37.115.53.98
  rightsubnet=192.168.1.0/24
  rightauth=psk
  ike=aes128-sha1-modp1024
  esp=aes128-sha1-modp1024
[root@ip-172-31-41-151 ~]#
```

And the result is (from AWS machine):

```
root@ip-172-31-41-151:~
  leftauth=psk
  right=37.115.53.98
  rightsubnet=192.168.1.0/24
  rightauth=psk
  ike=aes128-sha1-modp1024
  esp=aes128-sha1-modp1024
[root@ip-172-31-41-151 ~]# traceroute 192.168.1.99
traceroute to 192.168.1.99 (192.168.1.99), 30 hops max, 60 byte packets
 1  ip-192-168-1-2.ec2.internal (192.168.1.2)  137.617 ms  137.699 ms  137.857 m
s
 2  ip-192-168-1-99.ec2.internal (192.168.1.99)  139.274 ms  139.624 ms  140.716
 ms
[root@ip-172-31-41-151 ~]# ssh  192.168.1.99
root@192.168.1.99's password:
Linux SmartHome 4.19.97+ #1294 Thu Jan 30 13:10:54 GMT 2020 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun May 31 11:05:28 2020 from 172.31.41.151
root@SmartHome:~#
```

SO it is like Secure connection between my office network and Amazon Virtual Private Cloud (extra task) - but for FREE using strongswan :))

Next step

Modify ipsec configuration and forward to VM3 all traffic from 192.168.1.0/24

It is mandatory to add type=passthrough for 192.168.1.0/24 net before setting rightsubnet=0.0.0.0/0

https://wiki.strongswan.org/issues/1283

Final config from VM2 side:

```
root@mini-pc: ~
root@mini-pc:~# vi /etc/ipsec.conf
root@mini-pc:~# cat /etc/ipsec.conf
config setup
  strictcrlpolicy=no
  charondebug=all
conn %default
  ikelifetime=60m
  keylife=20m
  rekeymargin=3m
  keyingtries=1
  keyexchange=ikev2
conn pass
    right=127.0.0.1 # so this connection does not get used for other purposes
    leftsubnet=192.168.1.0/24
    rightsubnet=192.168.1.0/24
    type=passthrough
    auto=route
conn AWS
  authby=secret
  auto=start
  type=tunnel
  left=37.115.53.98
  leftid=37.115.53.98
  leftsubnet=192.168.1.0/24
  leftauth=psk
  right=204.236.252.156
  rightsubnet=0.0.0.0/0
  rightauth=psk
  ike=aes128-sha1-modp1024
  esp=aes128-sha1-modp1024
root@mini-pc:~#
```

Продемонстировать трейс с VM 1 до google.com

192.168.1.99 - PuTTY

```
2 packets transmitted, 2 received, 0% packet loss, time 3ms
rtt min/avg/max/mdev = 262.257/262.276/262.295/0.019 ms
root@SmartHome:~# traceroute google.com
traceroute to google.com (172.217.7.142), 30 hops max, 60 byte packets
 1  192.168.1.2 (192.168.1.2)  1.351 ms  0.745 ms  1.397 ms
 2  172.31.41.151 (172.31.41.151)  136.702 ms  137.400 ms  136.553 ms
 3  216.182.226.44 (216.182.226.44)  155.771 ms 216.182.238.103 (216.182.238.103
)  187.227 ms 216.182.229.153 (216.182.229.153)  137.990 ms
 4  100.66.9.126 (100.66.9.126)  148.872 ms 100.66.12.86 (100.66.12.86)  153.259
 ms 100.65.83.112 (100.65.83.112)  138.663 ms
 5  100.66.11.8 (100.66.11.8)  158.707 ms 100.66.48.106 (100.66.48.106)  137.610
 ms 100.66.44.254 (100.66.44.254)  138.479 ms
 6  100.66.7.223 (100.66.7.223)  149.059 ms 100.66.42.172 (100.66.42.172)  156.4
75 ms 100.66.6.29 (100.66.6.29)  158.965 ms
 7  100.66.6.63 (100.66.6.63)  150.760 ms 100.66.5.237 (100.66.5.237)  155.047 m
s 100.66.5.93 (100.66.5.93)  160.390 ms
 8  100.65.13.113 (100.65.13.113)  138.031 ms 100.66.5.43 (100.66.5.43)  505.907
 ms 100.66.5.129 (100.66.5.129)  155.650 ms
 9  100.65.13.225 (100.65.13.225)  136.841 ms 100.65.15.17 (100.65.15.17)  138.6
60 ms 52.93.28.161 (52.93.28.161)  139.293 ms
10  100.100.2.38 (100.100.2.38)  138.336 ms 52.93.28.169 (52.93.28.169)  138.055
 ms 52.93.28.143 (52.93.28.143)  137.657 ms
11  99.83.65.3 (99.83.65.3)  138.599 ms 99.82.181.23 (99.82.181.23)  138.101 ms
99.83.65.3 (99.83.65.3)  138.242 ms
12  108.170.246.1 (108.170.246.1)  138.747 ms * 99.82.181.23 (99.82.181.23)  137
.621 ms
13  72.14.234.134 (72.14.234.134)  140.252 ms 108.170.228.150 (108.170.228.150)
 138.255 ms *
14  iad30s08-in-f142.1e100.net (172.217.7.142)  137.860 ms  138.242 ms  137.723
ms
root@SmartHome:~#
```

Now we a getting access to internet via AWS VM3.

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

It is needed to allow:

500,4500/UDP - for IPSEC

22/TCP - for SSH

ICMP type 8 (echo request), type11 (Time exceeded) - for ICMP traceroute

33434-33534/UDP - for Unix traceroute type.

And after deny all