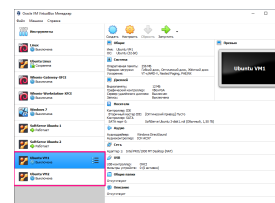




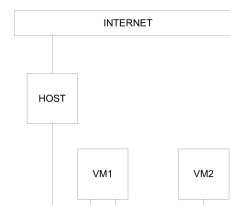
# Пробрасываем шлюз (gateway) от одной машины на другую в VitrualBox

## Предварительная настройка "Ubuntu VM1" и "Ubuntu VM2":

1. Создаем две виртуальные машины в VirtualBox. Я выбрал две Ubuntu 14.

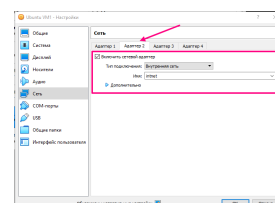
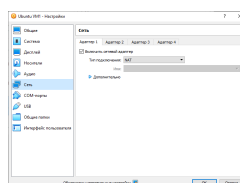


2. В моем случае, "Ubuntu VM1" будет шлюзом для "Ubuntu VM2" и структура будет иметь следующий вид:



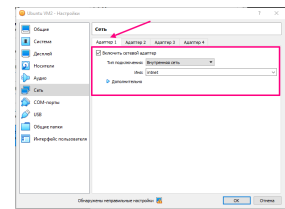
3. Меняем настройки сети для "Ubuntu VM1".

Адаптер 1 оставляем без изменений - NAT. Это наше подключение к самому интернету. Адаптер 2 включаем, меняем тип подключения на внутреннюю сеть и выбираем имя сети, в моем случае "intnet".

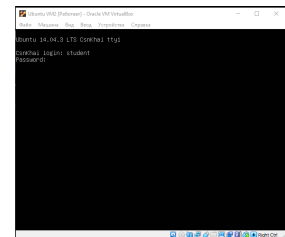
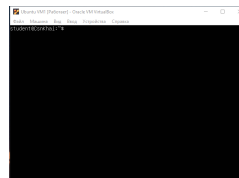


4. Меняем настройки сети для "Ubuntu VM2".

Тип подключения для адаптера 1 ставим "Внутренняя сеть" и выбираем то же имя, что и для "Ubuntu VM1". В моем случае "intnet".

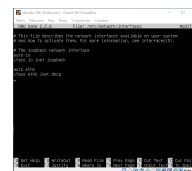
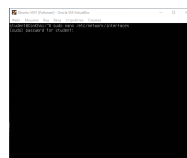


5. Настройка сети в настройках VirtualBox закончена. Теперь запускаем обе машины. Авторизируемся на обеих машинах.

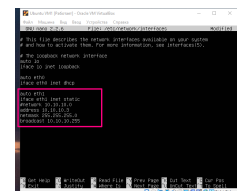


## Настройка "Ubuntu VM1":

6. Производим команду `sudo nano /etc/network/interfaces` и дописываем в файл указанные на скриншоте строки. Сохраняем файл.

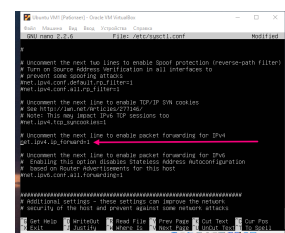
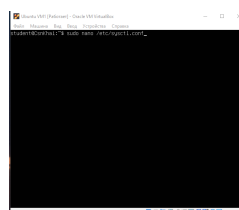


Изначальный вид файла

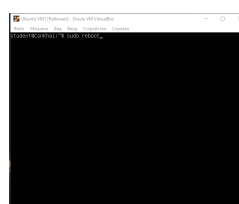


Отредактированный файл

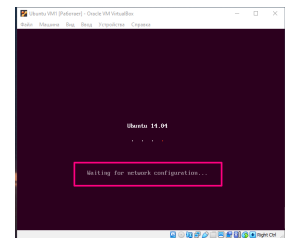
7. Заходим в файл `sudo nano /etc/sysctl.conf` и убираем решетку со следующей строки. Сохраняем файл.



8. Перезагружаем машину. После проделанного у Ubuntu 14 выводится информация о перенастройке сетевого подключения. Теперь переходим к

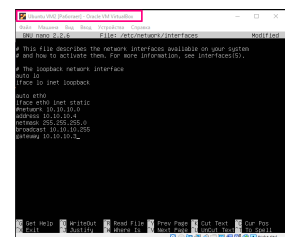
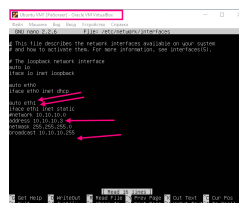


настройке "Ubuntu VM2". К дополнительной настройке "Ubuntu VM1" мы вернемся позже.

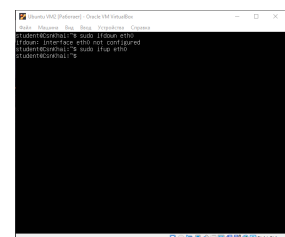


## Настройка "Ubuntu VM2":

9. Как и в "Ubuntu VM1" меняем настройки "*sudo nano /etc/network/interfaces*", только вместо eth1 ставим eth0, адресс ставим также другой, в конце указываем шлюз (gateway) с ip через которую будет идти интернет-соединение. В нашем случае адресс "Ubuntu VM1". Сохраняем файл.



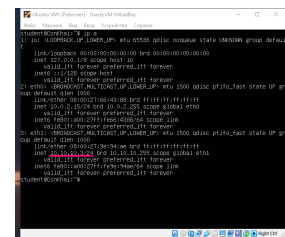
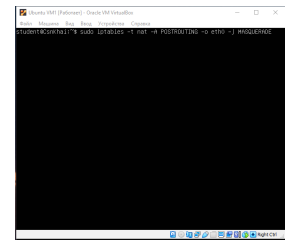
10. Выключаем и включаем сетевой интерфейс eth0, который мы добавили в "*/etc/network/interfaces*".



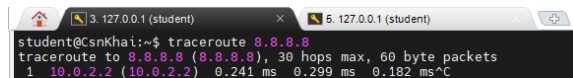
11. Выполняем команду "*ping 8.8.8.8*" и если сигнал проходит, то значит что интернет-соединение присутствует. Также можно выполнить команду "*traceroute 8.8.8.8*" и посмотреть, проходит ли сигнал через нашу "Ubuntu VM1". Это можно определить по ip адресу, который мы указали. Для того, чтобы проверить, можем ли мы послать сигнал на "Ubuntu VM1", мы выполняем команду "*ping*" с ip адресом нашей "Ubuntu VM1". В моем случае "*ping 10.10.10.3*".

## Дополнительная настройка "Ubuntu VM1"

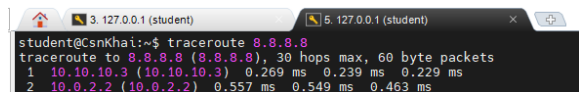
12. Выполняем команду `"sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE"`. На некоторых версиях Linux нужно вместо `"eth0"` использовать `"enp0s3"`. В таком случае команда будет выглядеть так `"sudo iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE"`.
13. Выполняем команду `"ip a"` или `"ifconfig"` и если результат похож на ситуацию на слайде, то настройка "Ubuntu VM1" закончена.



## Check the route from VM2 to Host

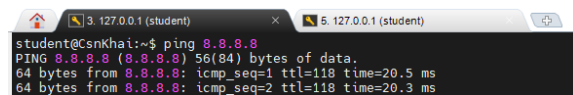


tracert for VM1

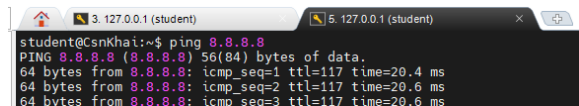


tracert for VM2

## Check the access to the Internet

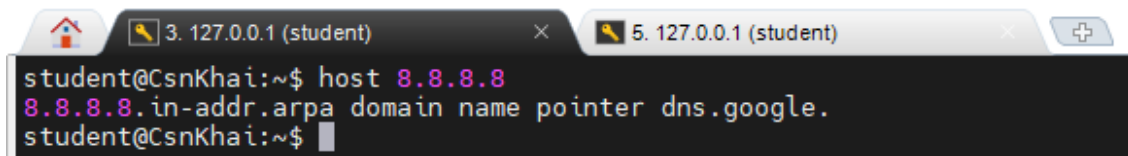


ping for VM1



ping for VM2

## Determine, which resource has an IP address 8.8.8.8

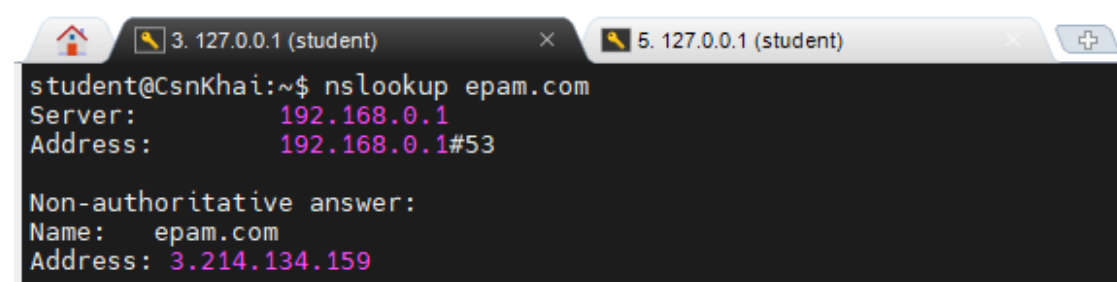


```
student@CsnKhai:~$ host 8.8.8.8
8.8.8.8.in-addr.arpa domain name pointer dns.google.
student@CsnKhai:~$
```

host

---

**Determine, which ip address belongs to resource epam.com**



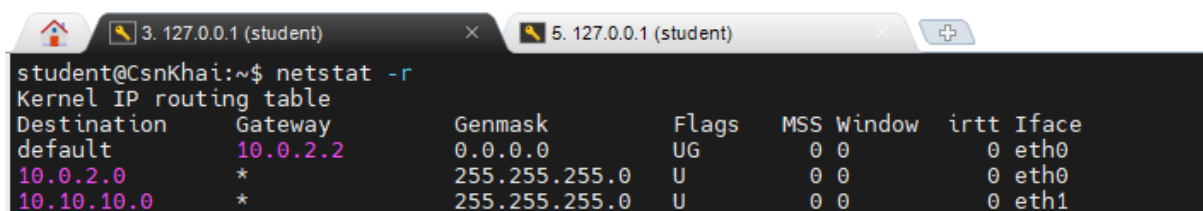
```
student@CsnKhai:~$ nslookup epam.com
Server:      192.168.0.1
Address:     192.168.0.1#53

Non-authoritative answer:
Name:   epam.com
Address: 3.214.134.159
```

nslookup

---

**Determine the default gateway for your HOST and display routing table**



```
student@CsnKhai:~$ netstat -r
Kernel IP routing table
Destination      Gateway         Genmask         Flags         MSS Window  irtt Iface
default          10.0.2.2        0.0.0.0         UG            0  0          0 eth0
10.0.2.0          *               255.255.255.0   U              0  0          0 eth0
10.10.10.0         *               255.255.255.0   U              0  0          0 eth1
```

netstat -r

---

**Trace the route to google.com**

By default traceroute uses UDP packages and I have the follow result:

```
student@CsnKhali:~$ traceroute google.com
traceroute to google.com (216.58.209.14), 30 hops max, 60 byte packets
 1 10.0.2.2 (10.0.2.2) 0.265 ms 0.239 ms 0.174 ms
 2 * * *
 3 * * *
 4 * * *
 5 * * *
 6 * * *
 7 * * *
 8 * * *
 9 * * *
10 * * *
11 * * *
12 * ^C
student@CsnKhali:~$
```

traceroute with UDP packages

Let's change UDP to ICMP packages:

```
student@CsnKhali:~$ sudo traceroute -I google.com
traceroute to google.com (142.250.203.206), 30 hops max, 60 byte packets
 1 10.0.2.2 (10.0.2.2) 0.116 ms 0.088 ms 0.079 ms
 2 192.168.0.1 (192.168.0.1) 0.953 ms 0.879 ms 0.872 ms
 3 254.57.57.37.triollan.net (37.57.57.254) 2.509 ms 2.773 ms 2.713 ms
 4 10.121.121.121 (10.121.121.121) 1.266 ms 1.212 ms 1.201 ms
 5 142.250.164.150 (142.250.164.150) 7.444 ms 7.567 ms 7.427 ms
 6 108.170.248.155 (108.170.248.155) 7.565 ms 7.705 ms 7.625 ms
 7 216.239.46.121 (216.239.46.121) 21.730 ms 21.063 ms 21.228 ms
 8 108.170.250.209 (108.170.250.209) 21.497 ms 21.557 ms 21.549 ms
 9 209.85.252.109 (209.85.252.109) 19.982 ms 20.317 ms 20.023 ms
10 waw02s22-in-f14.1e100.net (142.250.203.206) 20.715 ms 21.179 ms 20.926 ms
student@CsnKhali:~$
```

traceroute with ICMP packages