

Homework_Lesson2_OS_2

1. Создайте 2 виртуальных машины (далее - VM1, VM2. Вы можете дать любое удобное вам название). Используйте образ [ubuntu24.10](#)
2. Пройдите полностью все этапы установки и вручную разбейте свободное пространство на диски.
3. Настройте SSH-соединение следующим образом: хостовая ОС -> VM1, VM1 -> хостовая ос, VM2 -> VM1, VM2 -> хостовая ОС. Запрет соединения можно осуществить любым удобной полиси через iptables.

* с помощью инструмента Hashicorp Packer создайте образы двух виртуальных машин с заранее подготовленными предустановками, описанными выше. Должно быть 2 конфига.

Выполнение ДЗ Homework_Lesson2_OS_2

1. Создание двух VM VM1_ubuntu_server и VM2_ubuntu_server.

1.1 Скачан образ ОС [ubuntu24.10 \(ubuntu-24.10-live-server-amd64.iso\)](#)

1.2 Пример установки VM Ubuntu_server будет приводится на VM1_ubuntu_server, т.к. две VM будут идентичными.

1.3 Настройка VM в VirtualBox перед установкой.

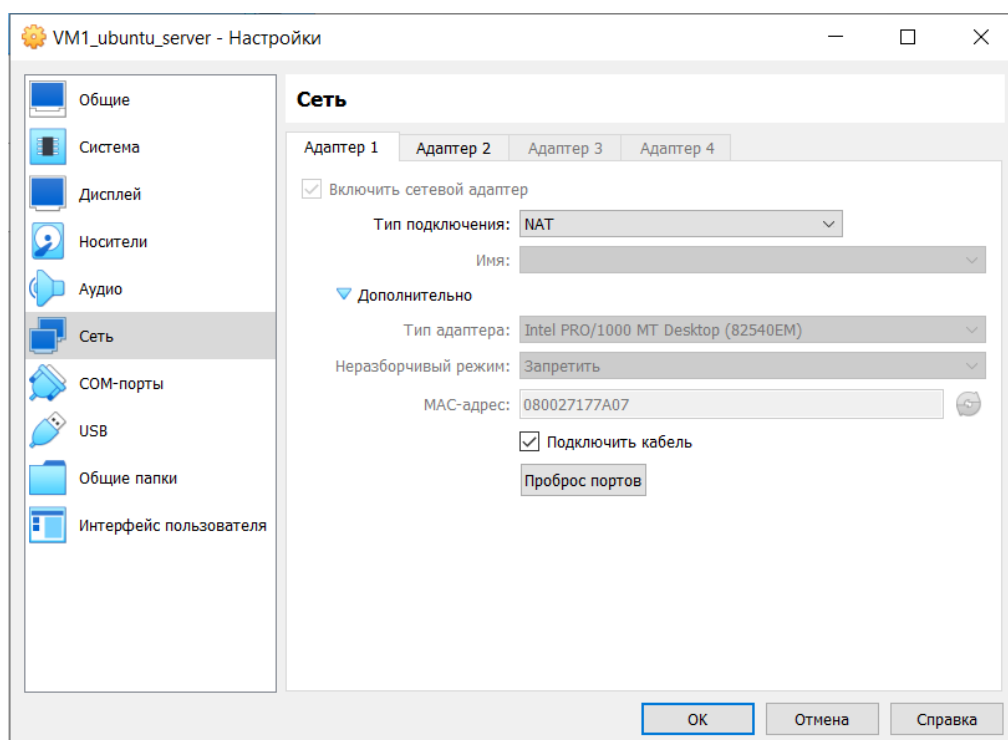
- подключаем iso образ [ubuntu24.10 \(ubuntu-24.10-live-server-amd64.iso\)](#)

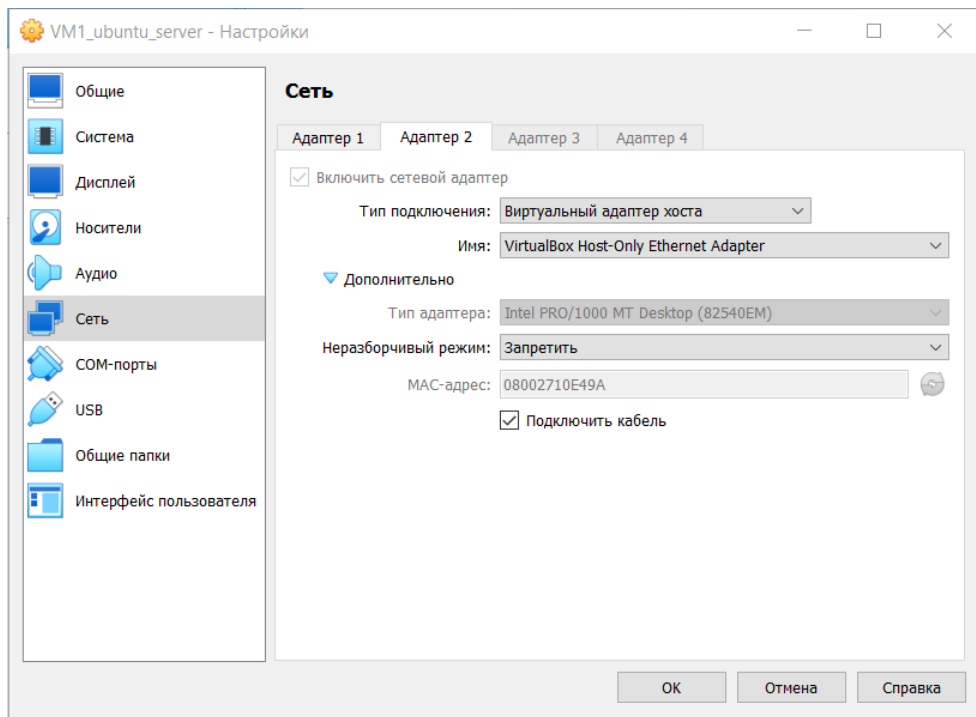
- выделяем 4 GB ОЗУ

- 25 GB vdi-диска (виртуального жесткого диска).

Также для работы сети настраиваем два адаптера (NAT) и (Виртуальный адаптер хоста).

Первый нам нужен для выхода в инет, для скачивания необходимых пакетов, а второй для настройки сети между VM машинами и хостом, с своими IP-адресами для коннекта по SSH.



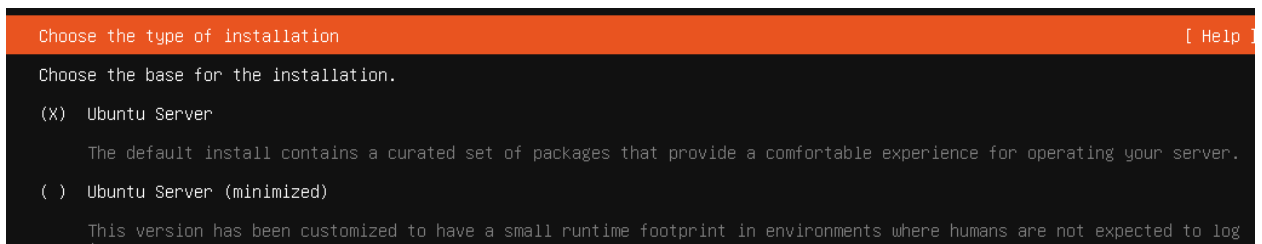


2. Установка ОС Ubuntu-server 24.10, на примере VM1_ubuntu_server.

2.1 Запускаем установку ОС. Выбираем язык.



2.2 Выбираем Ubuntu Server



2.3 Доходим до разбития диска:

```
Storage configuration

FILE SYSTEM SUMMARY

MOUNT POINT    SIZE    TYPE    DEVICE TYPE
[ /             15.000G new ext4 new partition of local disk ▶ ]
[ /home         5.997G  new ext4 new partition of local disk ▶ ]
[ SWAP          4.000G  new swap  new partition of local disk ▶ ]

AVAILABLE DEVICES

No available devices

[ Create software RAID (md) ▶ ]
[ Create volume group (LVM) ▶ ]

USED DEVICES

DEVICE                                TYPE    SIZE
[ VBOX_HARDDISK_VB580a9769-f20381b7  local disk  25.000G ▶ ]
partition 1  new, BIOS grub spacer    1.000M ▶ ]
partition 2  new, to be formatted as swap 4.000G ▶ ]
partition 3  new, to be formatted as ext4, mounted at / 15.000G ▶ ]
partition 4  new, to be formatted as ext4, mounted at /home 5.997G ▶ ]
```

2.4 Создаем своего пользователя “username” и даем название hostname (vm1server).

Аналогично для второй ВМ “username” и даем название hostname (vm2server).

2.5 Ставим галочку на установку OpenSSH Server.

```
SSH configuration

You can choose to install the OpenSSH server package to enable secure remote access to your server.

[X] Install OpenSSH server

[X] Allow password authentication over SSH

[ Import SSH key ▶ ]

AUTHORIZED KEYS

No authorized key
```

2.6 Ожидаем конца инсталляции ОС Ubuntu.

```
Installing system [ Help ]

subiquity/load_cloud_config/extract_autoinstall:
subiquity/Early/apply_autoinstall_config:
subiquity/Reporting/apply_autoinstall_config:
subiquity/Error/apply_autoinstall_config:
subiquity/Userdata/apply_autoinstall_config:
subiquity/Package/apply_autoinstall_config:
subiquity/Debconf/apply_autoinstall_config:
subiquity/Kernel/apply_autoinstall_config:
subiquity/KernelCrashDumps/apply_autoinstall_config:
subiquity/Zdev/apply_autoinstall_config:
subiquity/Ad/apply_autoinstall_config:
subiquity/Late/apply_autoinstall_config:
configuring apt
curtin command in-target
installing system
executing curtin install initial step
executing curtin install partitioning step
curtin command install
configuring storage
running 'curtin block-meta simple'
curtin command block-meta
removing previous storage devices
configuring disk: disk-0da
configuring partition: partition-0
configuring partition: partition-1
configuring format: format-0
configuring mount: mount-0
configuring partition: partition-2
configuring format: format-1
configuring mount: mount-1
configuring partition: partition-3
configuring format: format-2
configuring mount: mount-2
executing curtin install extract step
curtin command install
writing install sources to disk
running 'curtin extract'
curtin command extract
acquiring and extracting image from cp:///tmp/tmp6tv455Sp/mount |
```

3. Настройка SSH между ВМ vm1server, vm2server и хостовой Windows.

3.1 После установки ОС ubuntu на ВМ проверяем конфигурации сетевых адаптеров:

```
root@vm1server:~# ip a
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 noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:17:7a:07 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 82838sec preferred_lft 82838sec
    inet6 fe80::a00:27ff:fe17:7a07/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:10:e4:9a brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 metric 100 brd 192.168.56.255 scope global dynamic enp0s8
        valid_lft 342sec preferred_lft 342sec
    inet6 fe80::a00:27ff:fe10:e49a/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
root@vm1server:~#
```

vm1server (192.168.56.102)

```
root@vm2server:~# ip a
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 noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:32:74:74 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86379sec preferred_lft 86379sec
    inet6 fe80::a00:27ff:fe32:7474/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:4d:f7:2a brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 metric 100 brd 192.168.56.255 scope global dynamic enp0s8
        valid_lft 579sec preferred_lft 579sec
    inet6 fe80::a00:27ff:fe4d:f72a/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
root@vm2server:~#
```

vm2server (192.168.56.101)

```
PS C:\Windows\system32> ipconfig

Настройка протокола IP для Windows

Адаптер Ethernet Ethernet:

    Состояние среды. . . . . : Среда передачи недоступна.
    DNS-суффикс подключения . . . . . :

Неизвестный адаптер OpenVPN Wintun:

    Состояние среды. . . . . : Среда передачи недоступна.
    DNS-суффикс подключения . . . . . :

Адаптер Ethernet Ethernet 2:

    DNS-суффикс подключения . . . . . :
    Локальный IPv6-адрес канала . . . : fe80::b1fc:32ec:ec1:bf0c%22
    IPv4-адрес. . . . . : 192.168.56.1
    Маска подсети . . . . . : 255.255.255.0
    Основной шлюз. . . . . :
```

Хостовая Windows (192.168.56.1)

Также все: хостовая Windows и ВМ друг друга видят (пингуются).

```
PS C:\Windows\system32> ping 192.168.56.102
```

```
Обмен пакетами с 192.168.56.102 по с 32 байтами данных:  
Ответ от 192.168.56.102: число байт=32 время<1мс TTL=64  
Ответ от 192.168.56.102: число байт=32 время<1мс TTL=64
```

```
PS C:\Windows\system32> ping 192.168.56.101
```

```
Обмен пакетами с 192.168.56.101 по с 32 байтами данных:  
Ответ от 192.168.56.101: число байт=32 время<1мс TTL=64  
Ответ от 192.168.56.101: число байт=32 время<1мс TTL=64
```

```
makarov1@vm1server:~$ ping 192.168.56.1  
PING 192.168.56.1 (192.168.56.1) 56(84) bytes of data.  
64 bytes from 192.168.56.1: icmp_seq=1 ttl=128 time=0.528 ms  
64 bytes from 192.168.56.1: icmp_seq=2 ttl=128 time=0.428 ms
```

```
makarov1@vm1server:~$ ping 192.168.56.101  
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.  
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=11.1 ms  
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=1.20 ms
```

3.2 Т.к. для первого задания использовал WSL, для выполнения данного домашнего задания нужно настроить SSH. На хостовой машине Windows нужно установить sshd-server. В PowerShell из под Администратора выполнить строку:

```
Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH.Server*' | Add-  
WindowsCapability -Online
```

После запустить службу sshd.

```
PS C:\Windows\system32> Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH.Server*' | Add-WindowsCapability -Online  
  
Path :  
Online : True  
RestartNeeded : False  
  
PS C:\Windows\system32> Start-Service sshd
```

3.3 Создаем SSH-ключи на ВМ vm1server и vm2server:

3.3.1 Для ВМ vm1server и vm2server:

```
ssh-keygen -t rsa
```

```
ssh-copy-id makarov2@192.168.56.101
```

```
ssh-copy-id Nec@192.168.56.1
```

3.3.2 На хостовой машине Windows выполняем следующие действия:

```
scp makarov2@192.168.56.101:~/.ssh/id_rsa.pub C:\Users\Nec\.ssh\id_rsa.pub – копирование  
публичного ключа
```

```
PS C:\Windows\system32> scp makarov2@192.168.56.101:~/.ssh/id_rsa.pub C:\Users\Nec\
.ssh\id_rsa.pub
makarov2@192.168.56.101's password:
id_rsa.pub 100% 572 186.2KB/s 00:00
```

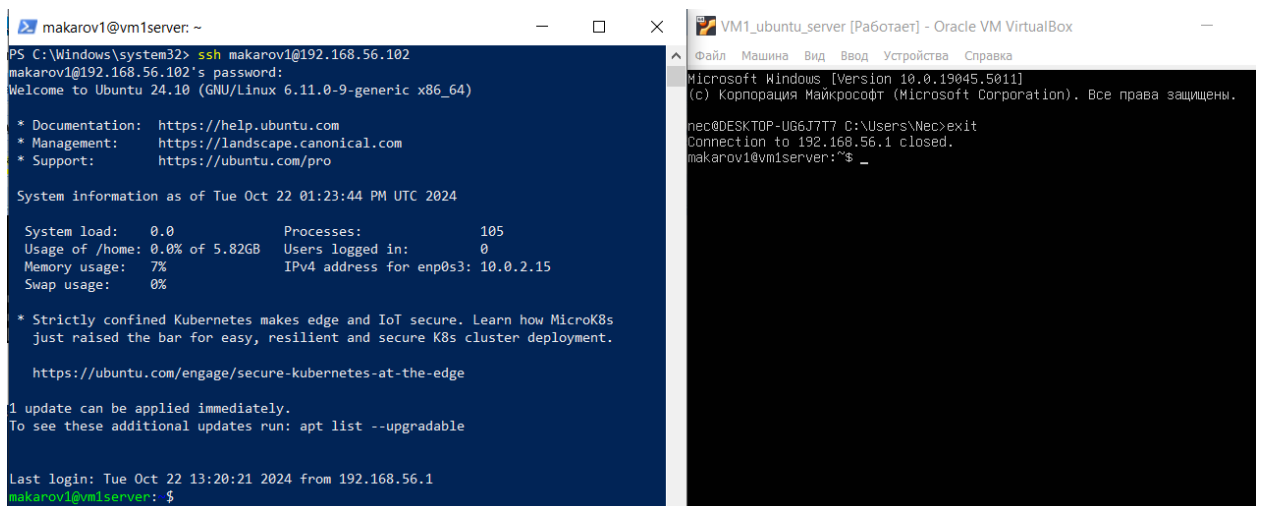
move C:\Users\Nec\.ssh\id_rsa.pub C:\Users\Nec\.ssh\authorized_keys – переименование публичного ключа

Права доступа для Windows пользователя Nec:

icacls C:\Users\Nec\.ssh\authorized_keys /inheritance:r

icacls C:\Users\Nec\.ssh\authorized_keys /grant:r Nec:F

3.4 Пробуем подключиться по SSH: хостовая ОС -> VM1, VM1 -> хостовая ос, VM2 -> VM1, VM2 -> хостовая ОС.



Хостовая ОС -> VM1, VM1 -> Хостовая ос

```

makarov2@vm2server:~$ ssh makarov1@192.168.56.102
Welcome to Ubuntu 24.10 (GNU/Linux 6.11.0-9-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/pro

System information as of Tue Oct 22 01:55:21 PM UTC 2024

System load:  0.0               Processes:    104
Usage of /home: 0.0% of 5.82GB   Users logged in: 0
Memory usage:  7%              IPv4 address for enp0s3: 10.0.2.15
Swap usage:    0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
  just raised the bar for easy, resilient and secure K8s cluster deployment.

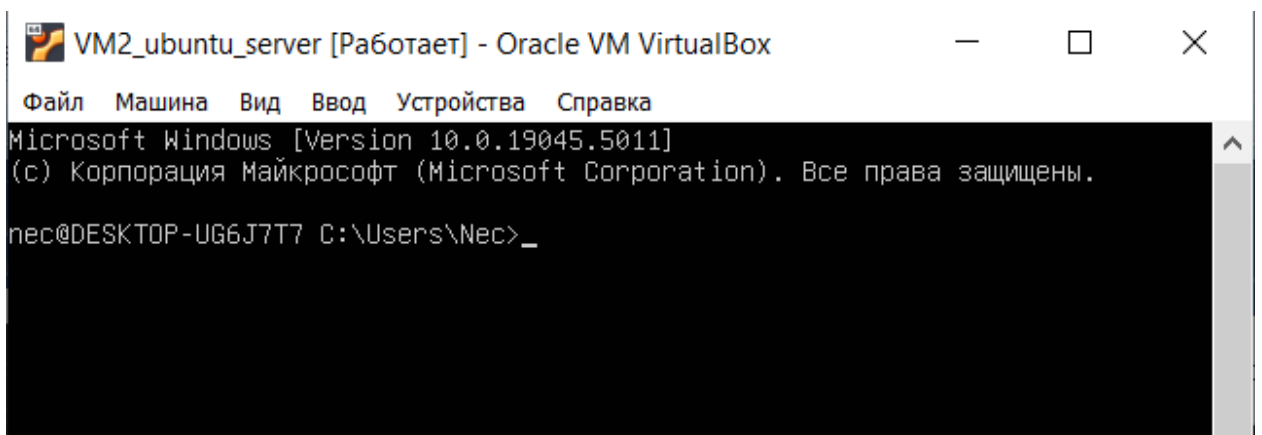
  https://ubuntu.com/engage/secure-kubernetes-at-the-edge

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Tue Oct 22 13:52:06 2024 from 192.168.56.1
makarov1@vm1server:~$

```

VM2 -> VM1



VM2 -> Хостовая ОС.

4. Запрет соединения через iptables.

Пример, на ВМ vm2server (192.168.56.101) закрываем доступ к ВМ vm1server (192.168.56.102) и хостовой Windows 192.168.56.1.

На скринке ниже показано, что по ssh мы подключаемся к хостовой Windows машине 192.168.56.1.

После по ssh делаем пробное подключение к ВМ vm1server (192.168.56.102).

Вводим запрет для ВМ vm1server (192.168.56.102):

```
sudo iptables -A OUTPUT -d 192.168.56.1 -j DROP
```

После пробуем подключиться по SSH к этой ВМ и без результата. Запрос зависает.

Аналогично вводим запрет для хостовой Windows машине:

```
sudo iptables -A OUTPUT -d 192.168.56.1 -j DROP
```

Также запрос не обрабатывает.

```
Microsoft Windows [Version 10.0.19045.5011]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

nec@DESKTOP-UG6J7T7 C:\Users\Nec>exit
Connection to 192.168.56.1 closed.
makarov2@vm2server:~$ ssh makarov1@192.168.56.102
Welcome to Ubuntu 24.10 (GNU/Linux 6.11.0-9-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Tue Oct 22 02:24:51 PM UTC 2024

System load:  0.0               Processes:    104
Usage of /home: 0.0% of 5.82GB   Users logged in: 0
Memory usage:  7%              IPv4 address for enp0s3: 10.0.2.15
Swap usage:    0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Tue Oct 22 13:55:21 2024 from 192.168.56.101
makarov1@vm1server:~$ exit
logout
Connection to 192.168.56.102 closed.
makarov2@vm2server:~$ sudo iptables -A OUTPUT -d 192.168.56.102 -j DROP
makarov2@vm2server:~$ ssh makarov1@192.168.56.102
^C
makarov2@vm2server:~$ sudo iptables -A OUTPUT -d 192.168.56.1 -j DROP
makarov2@vm2server:~$ ssh Nec@192.168.56.1
^C
makarov2@vm2server:~$ _
```

Проверяем правила iptables:

```
makarov2@vm2server:~$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT     0    --  0.0.0.0/0              0.0.0.0/0
DROP       0    --  192.168.56.101         0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
DROP       0    --  0.0.0.0/0              192.168.56.102
DROP       0    --  0.0.0.0/0              192.168.56.1
makarov2@vm2server:~$
```

Видим, что закрыт доступ к vm1server (192.168.56.102) и Windows (192.168.56.1).