



# Същност на Виртуализацията

Виртуални Машини

Жизнен цикъл на VM

Параметризация на VM. Vagrant

# Съдържание

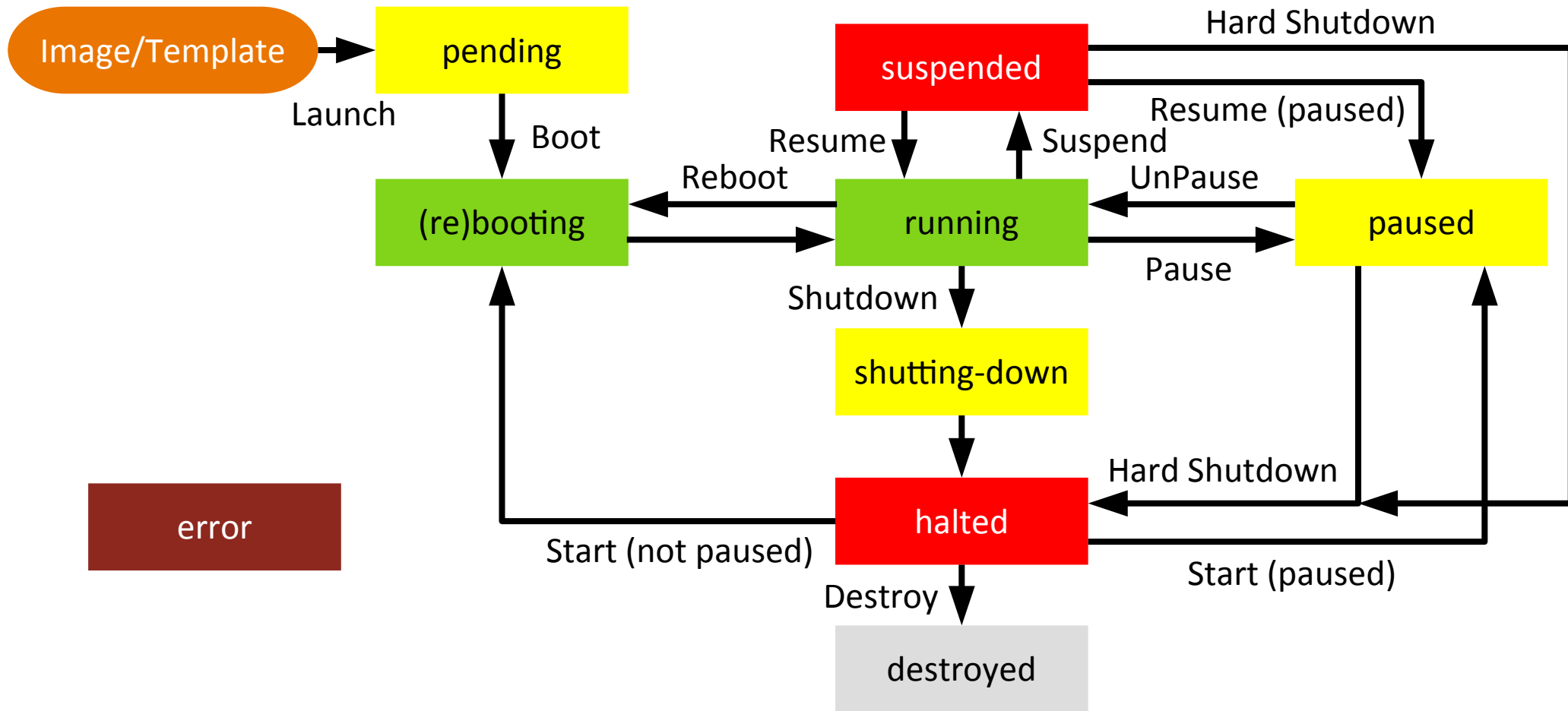
1. Жизнен цикъл на виртуалната машина
2. VirtualBox. Параметризация на виртуалната машина
3. Инсталация на Linux
4. Vagrant

# Жизнен Цикъл на VM

# Етапи в жизнения цикъл на ВМ

- ❖ **Планиране** (Planning) – Анализ на изискванията; Планиране на оптималното място за ВМ;
- ❖ **Обезпечаване** (Provisioning) – Създаване на ВМ (включително от темплейт/шаблон); Инсталация на ОС; Инсталиране на VMware tools или MS integration services и др.; Тестване;
- ❖ **Експлоатация** (Production) – Въвеждане в експлоатация (Deploy); Мониторинг (Статус, Производителност, Наличност); Създаване на Snapshot-и; Резервни копия (Backup); Скалиране на ресурсите на ВМ; Миграция; Възстановяване при бедствия; Клонирание;
- ❖ **Унищожаване** (Decommission/Destroy) – Освобождаване на ресурсите на ВМ; Премахване на ВМ;

# Състояния при работа на ВМ



# Състояния при работа на ВМ

- ❖ **Pending** – ВМ очаква ресурси за да бъде стартирана;
- ❖ **Booting/Rebooting** – ВМ е в процес на стартиране/рестартиране;
- ❖ **Running** – ВМ работи;
- ❖ **Paused** – ВМ е временно спряна. Не извършва комуникации, не натоварва процесора, но пълното и състояние не е съхранено на диска, а продължава да е в паметта, макар и неактивно. Възстановяването на работата и е почти мигновено;
- ❖ **Suspended** – ВМ е временно спряна. Пълното състояние на машината е съхранено на диска. ВМ не консумира ресурси (RAM/CPU/...). Възстановяването на работата изисква прочитане на състоянието, но не и рестартиране;
- ❖ **Shutting-down** – ВМ е в процес на спиране;
- ❖ **Halted** – ВМ е спряна напълно. (Може да бъдат правени промени); Възобновяването на работата на ВМ изисква повторното и стартиране;
- ❖ **Destroyed** – ВМ е премахната от host-а. Всички ресурси за ВМ са освободени;
- ❖ **Error** – ВМ не работи поради грешка в настройките или ресурсите;

# Параметризация на VM

*(VirtualBox)*

# Параметри на ВМ

- ❖ Основна мета информация: Име, Описание, Тип на ВМ, Guest ОС, Къде се намира ВМ (папка и др.) и др.;
- ❖ Описващи виртуалния системен хардуер: Процесор, Брой, Процент на натоварване, Разширени възможности на процесора, Дънна платка, Чипсет, Размер на основната памет, Boot параметри (UEFI, Boot устройство и др.), Таймер, Вид хипервайзор, Хардуерно ускорение и др.;
- ❖ Описание на виртуалната видеокарта: Вид, Видео памет, Брой монитори, Хардуерно ускорение, Поддръжка на RemoteDesktop и др.;
- ❖ Описание на устройствата за съхранение: Контролери за В/И, Дисковете свързани за тях, Вид и големина на виртуалните Имиджи, Споделени папки и др.;
- ❖ Параметри на мрежата: Адаптери, Вид, Скорост, Адреси, Начин на свързване, Комутатори и др.;
- ❖ Параметри за други виртуални В/И устройства: Аудио, Сериини, Паралелни и USB интерфейси (и устройствата свързани за тях) и др.

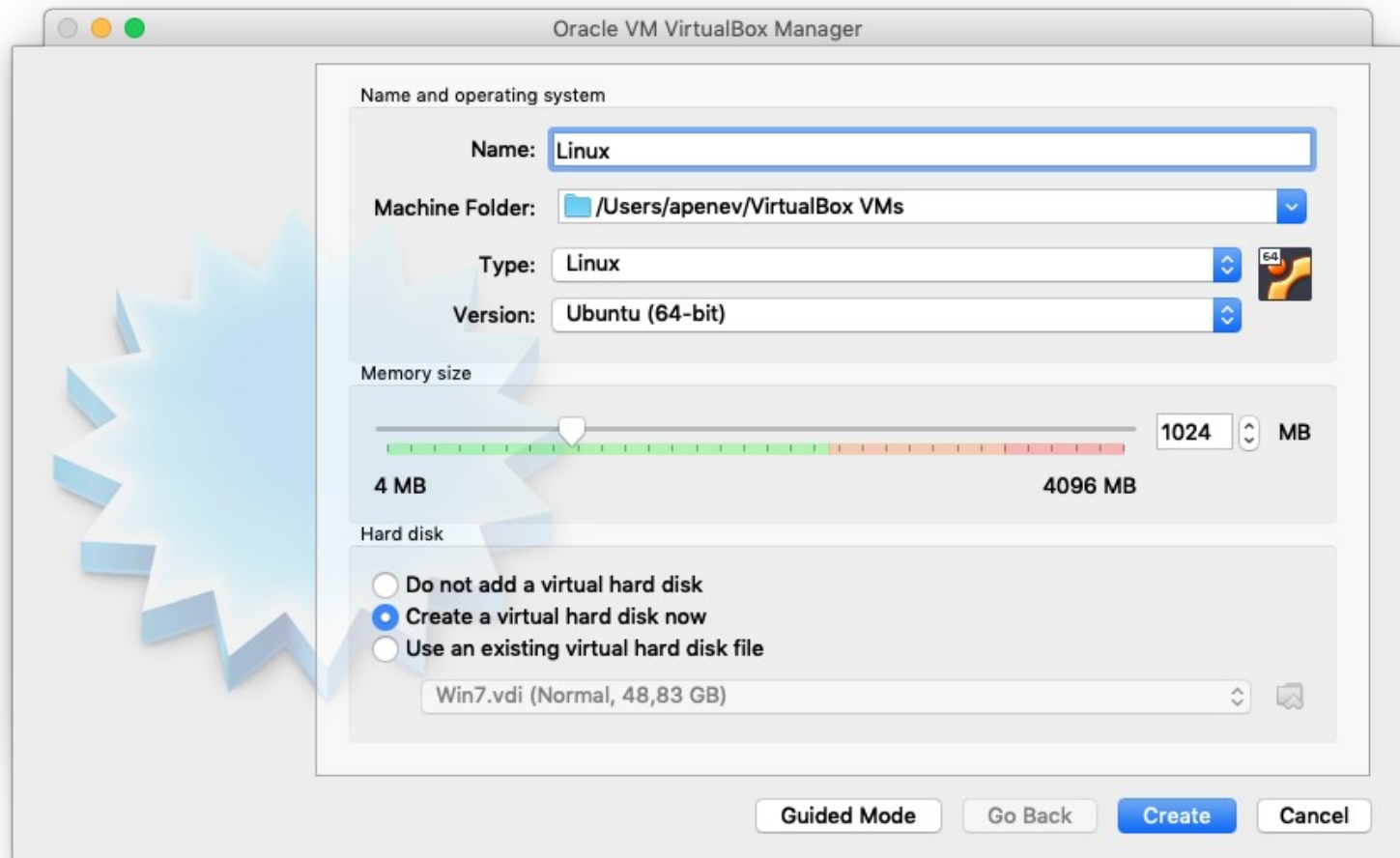
# Oracle VM VirtualBox



- ❖ VirtualBox е хипервайзор от Тип 2. Създава и управлява VM;
- ❖ Поддържаните операционни системи-домакин включват Linux, OS X, Windows XP и по-късно Solaris и OpenSolaris. Налични са също и портове за FreeBSD;
- ❖ Поддържа: 64-битови гост системи, Изображения на състоянието (Snapshots), Seamless режим, Споделен клипборд, Споделени папки, Вложено пейджиране за AMD-V и Intel VT (само за процесори, поддържащи SLAT), Частична поддръжка на 3D графично ускорение, Миграция в реално време, 2D ускорение и много други;
- ❖ Емулация на разнообразни устройства за съхранение на данни и имидж формати, мрежови карти, USB и др.;
- ❖ Разширителен пакет;
- ❖ и др.;

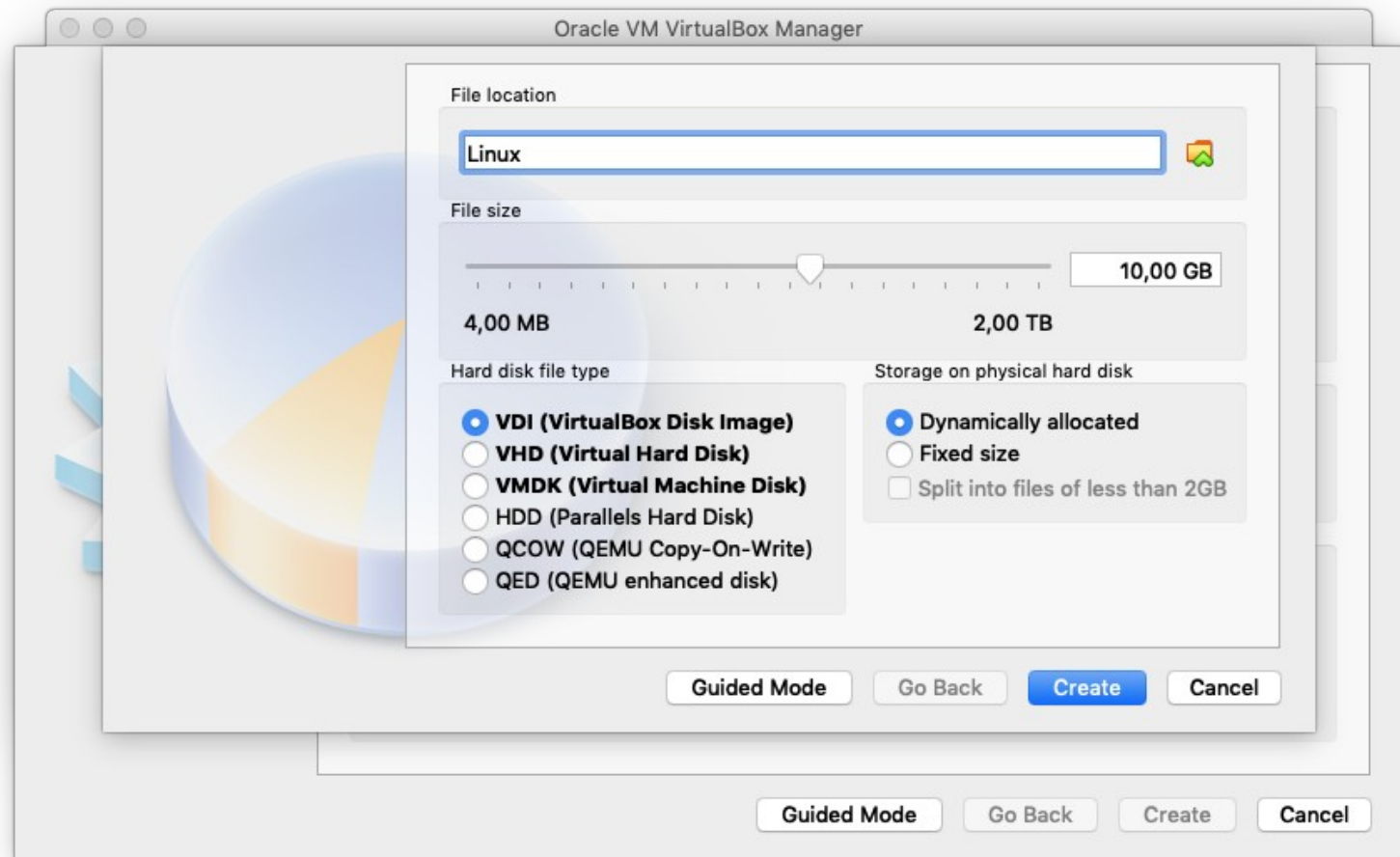
# VirtualBox – създаване на VM

След избор на режима за създаване на нова VM, въвеждаме името и вида на машината, големината на RAM паметта, както и какъв вид виртуален диск ще има VM.



# VirtualBox – параметри на нов виртуален диск

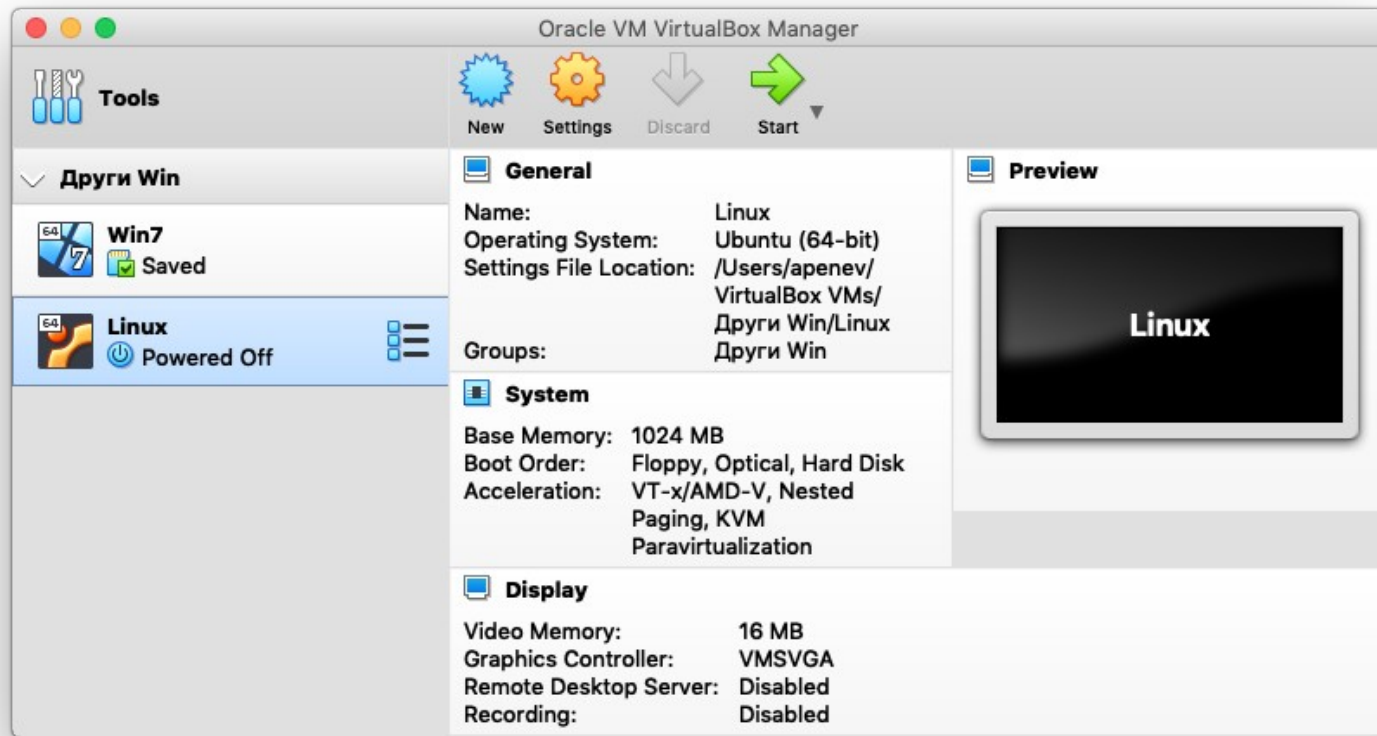
Избираме параметрите на новия виртуален диск, като име, размер, тип на Имиджа, дали да е динамичен или с фиксирана големина и др.



# VirtualBox – VM е създадена

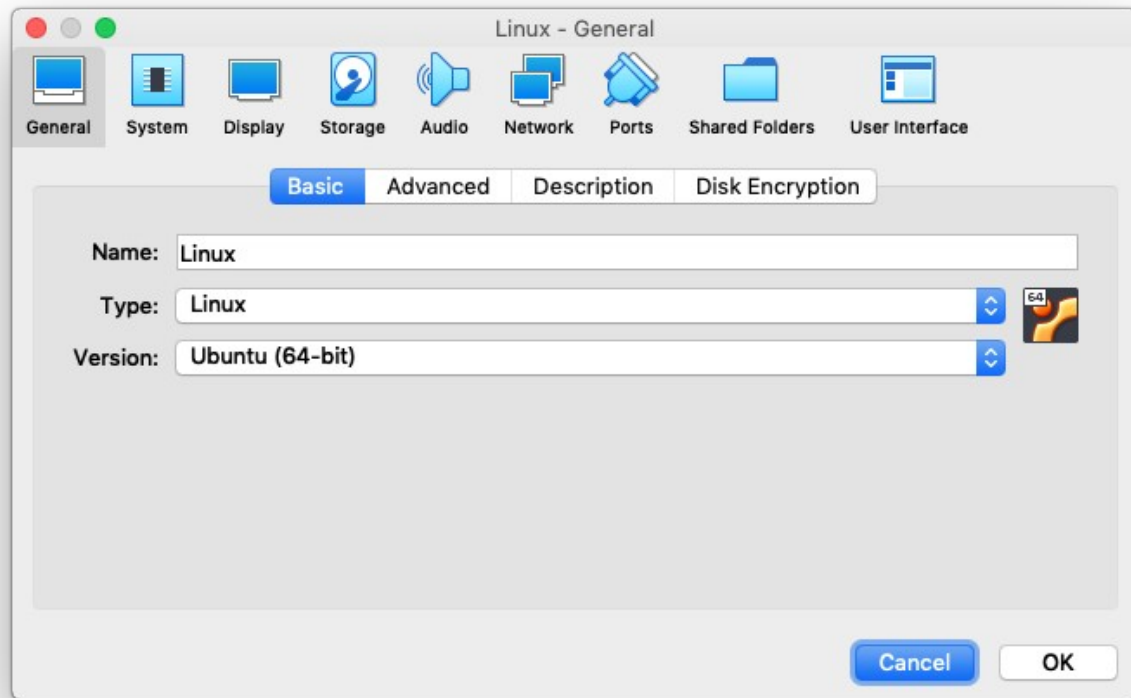
VirtualBox създава VM и нейните ресурси, като използва много от параметрите по подразбиране.

Вече може да стартираме VM, но преди това е добре да прегледаме и настроим другите параметри с режима Settings.



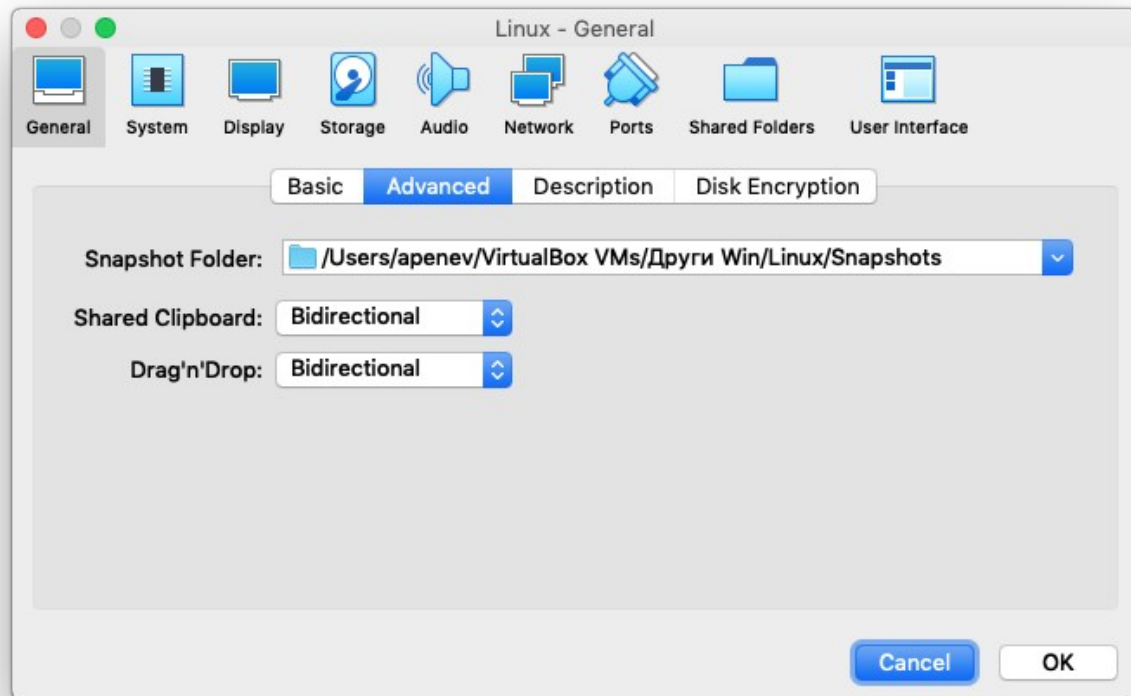
# VirtualBox – Настройки – Основни – Начални

Може да променяме името, типа на ОС, както и нейната версия.



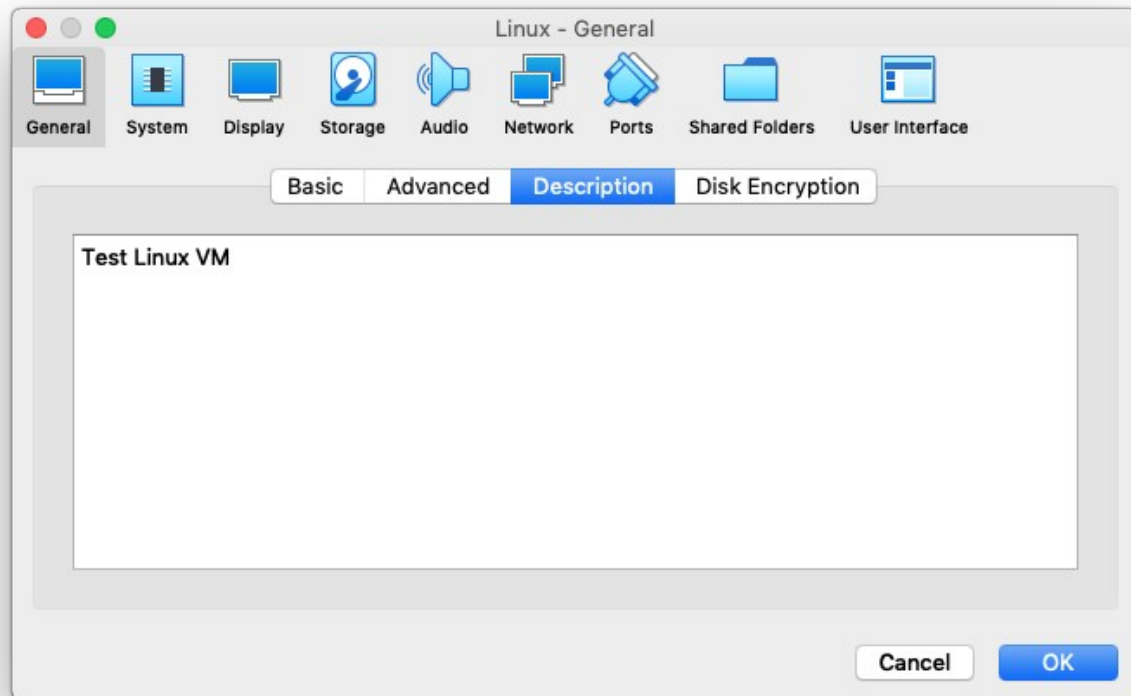
# VirtualBox – Настройки – Основни – Разширени

Може да променяме къде да се съхраняват Snapshot-ите, как да се третира споделения клипборд и др.



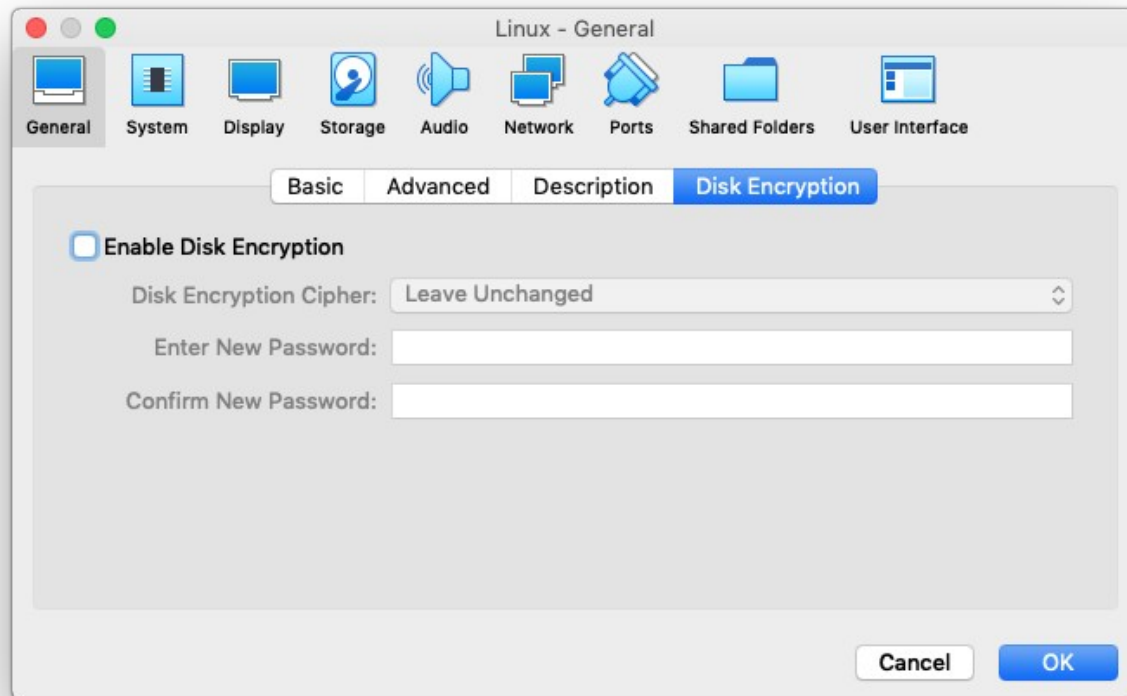
# VirtualBox – Настройки – Основни – Описание

Може да променяме краткото  
текстово описание на VM.



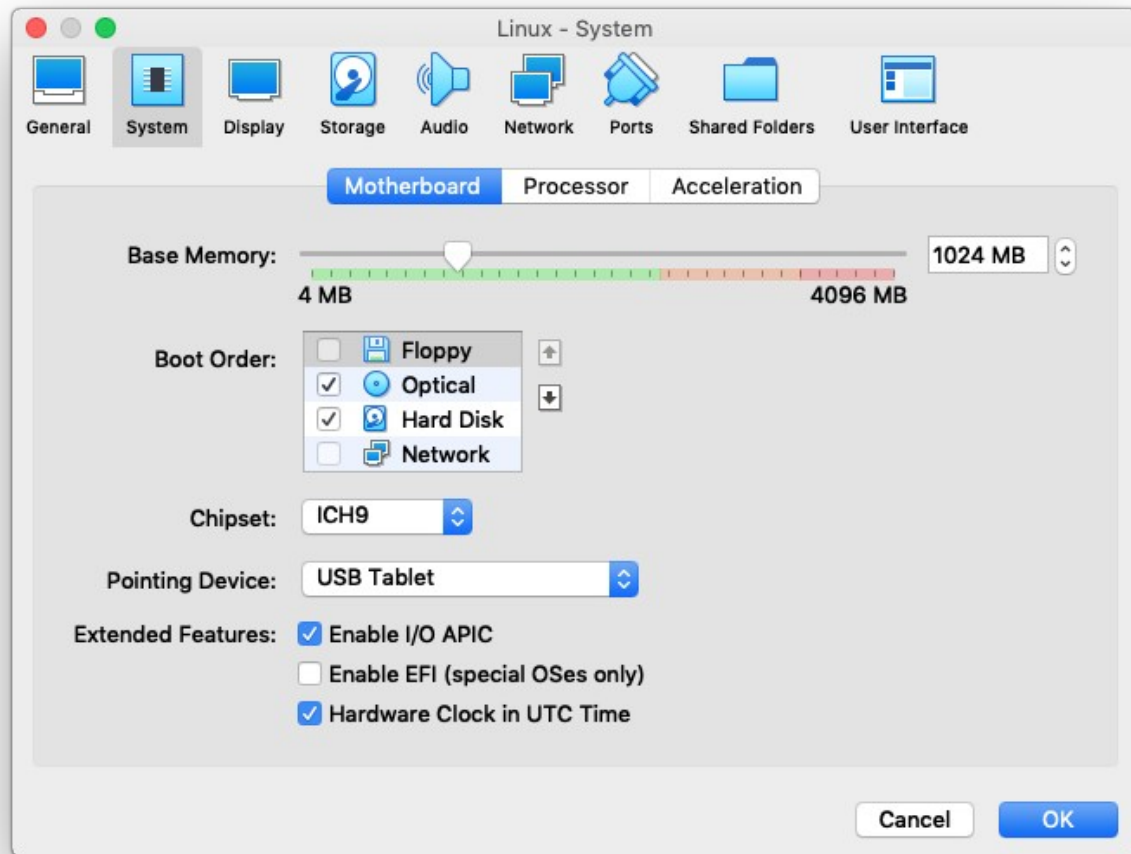
# VirtualBox – Настройки – Основни – Disk Encryption

Може да променяме дали и как да  
бъде криптиран диска на VM.



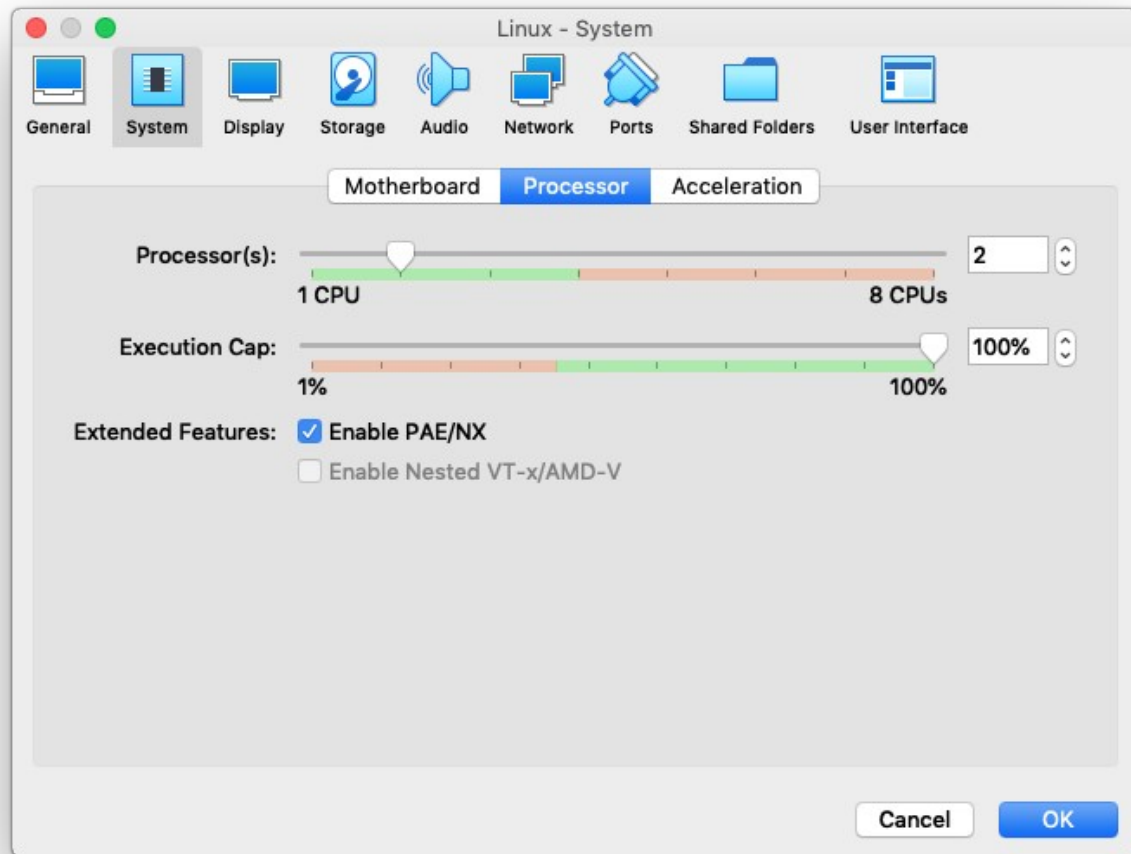
# VirtualBox – Настройки – Система – Дънна платка

Може да променяме размера на основната памет, кое устройство да бъде устройство по подразбиране за първоначално зареждане (Boot), вида на чипсета, както и други настройки на дънната платка на VM.



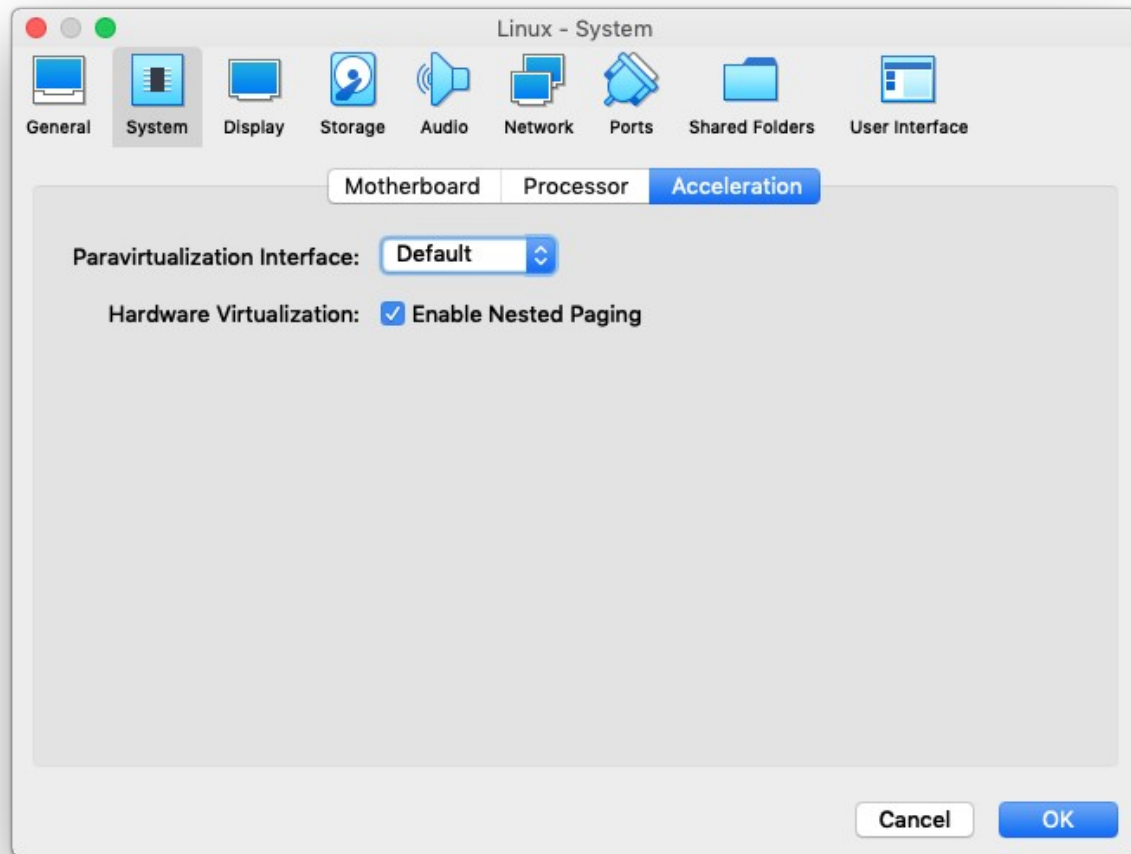
# VirtualBox – Настройки – Система – Процессор

Може да променяме броя на процесорите, тяхното максимално позволено натоварване, както и други настройки на процесора.



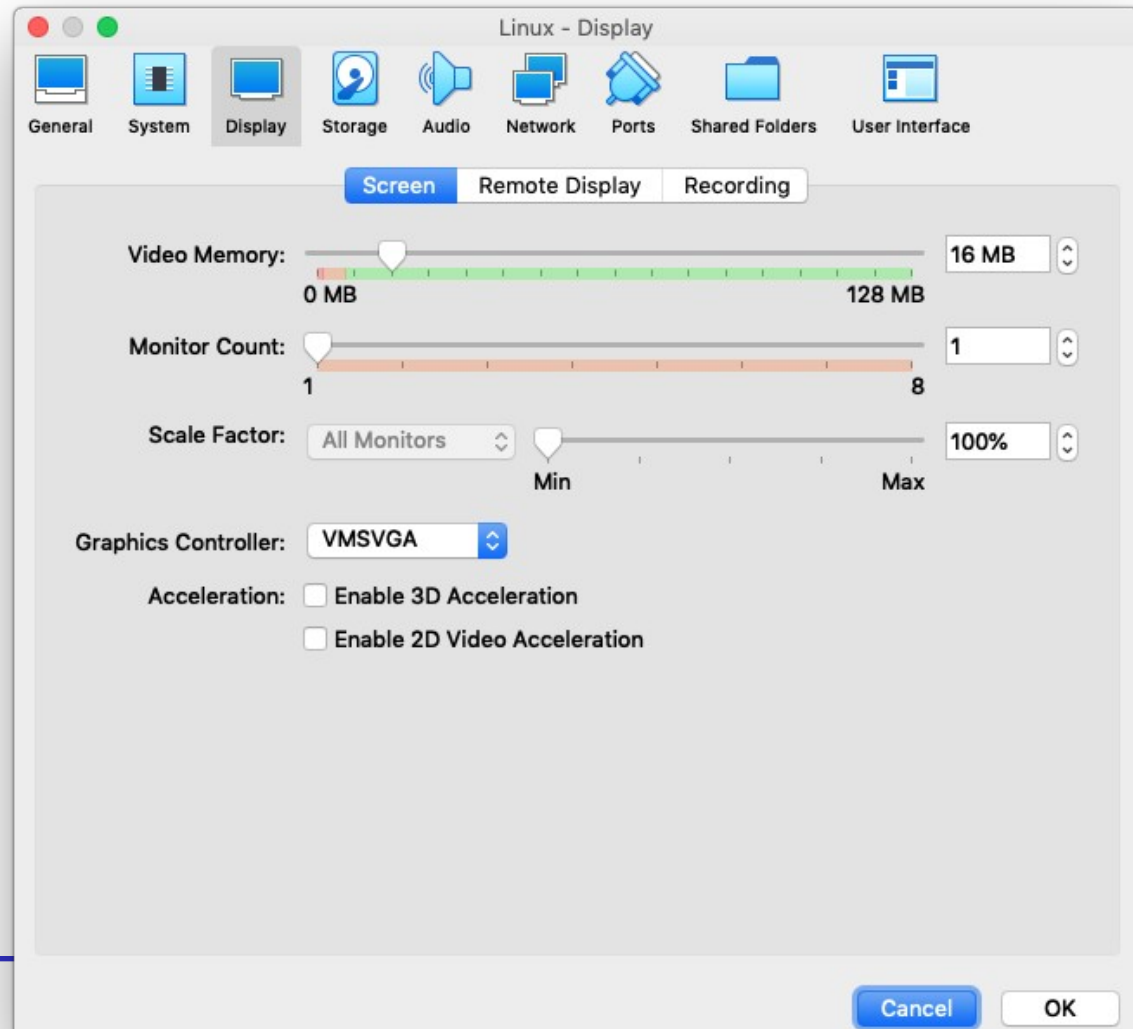
# VirtualBox – Настройки – Система – Ускорение

Може да променяме вида на хипервайзора и др.

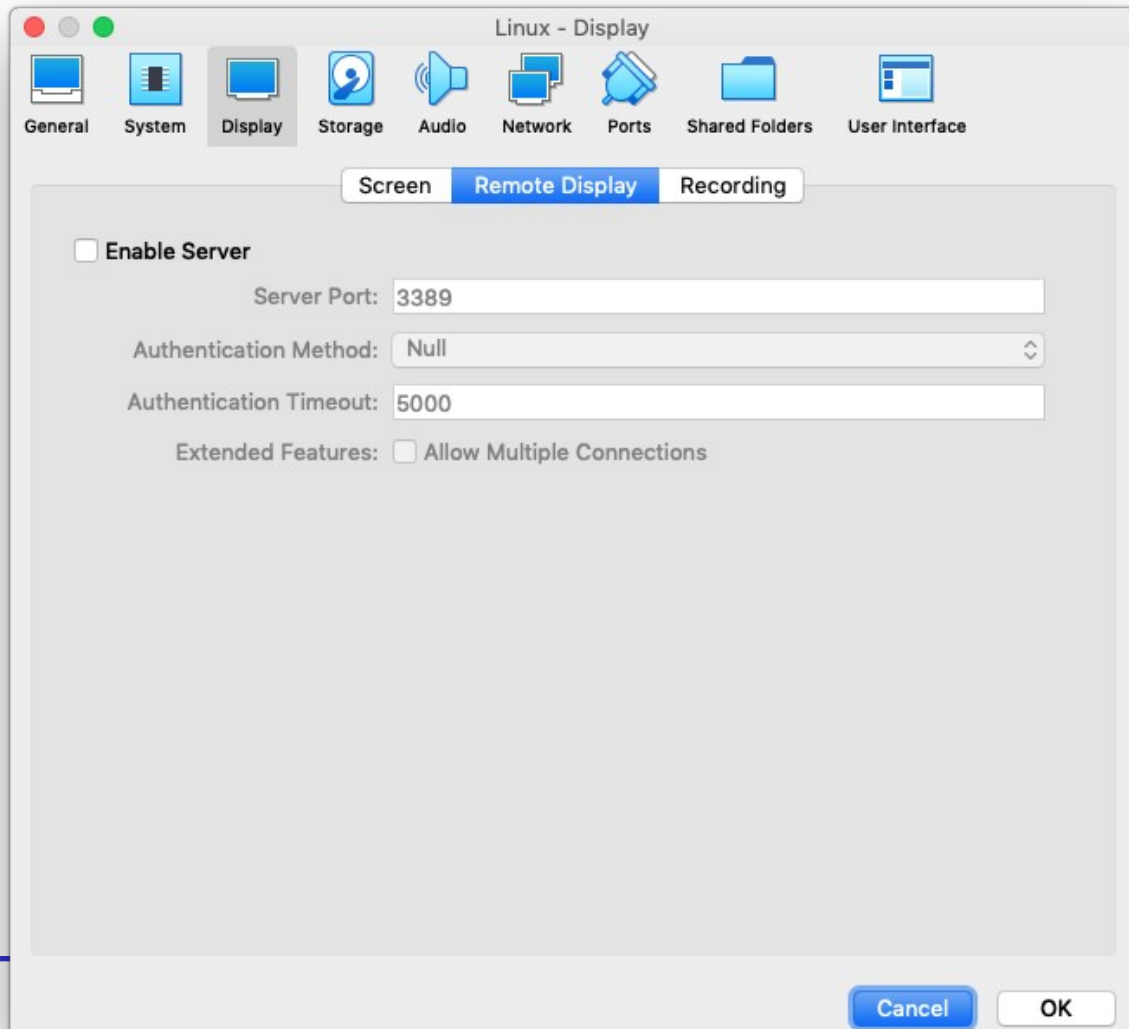


# VirtualBox – Настройки – Възпроизвеждане – Екран

Може да променяме размера на видео паметта, броя монитори, вида на емулираната видеокарта, 2D и 3D ускорението и др.



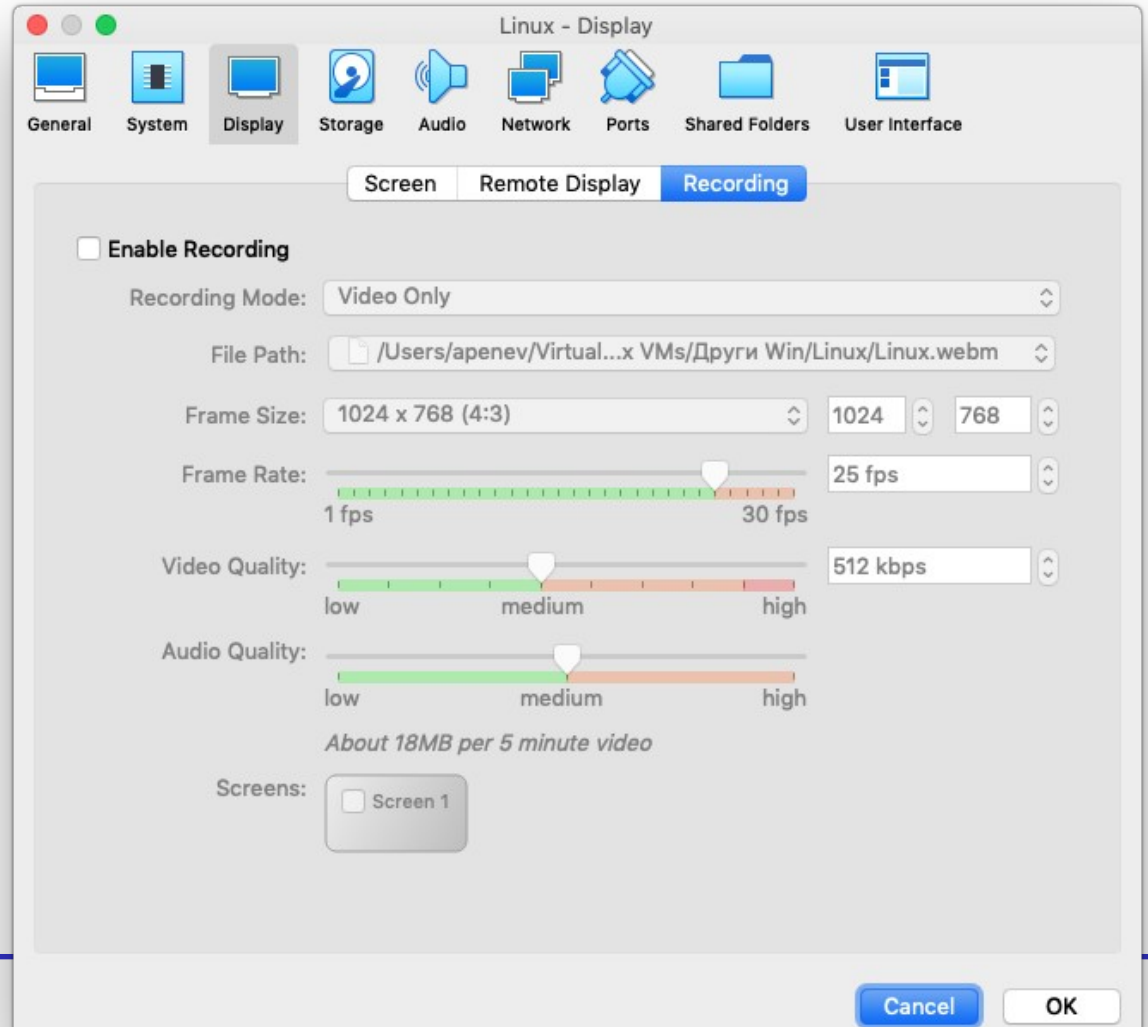
# *VirtualBox – Настройки – Възпроизвеждане – Отд.дисплей*



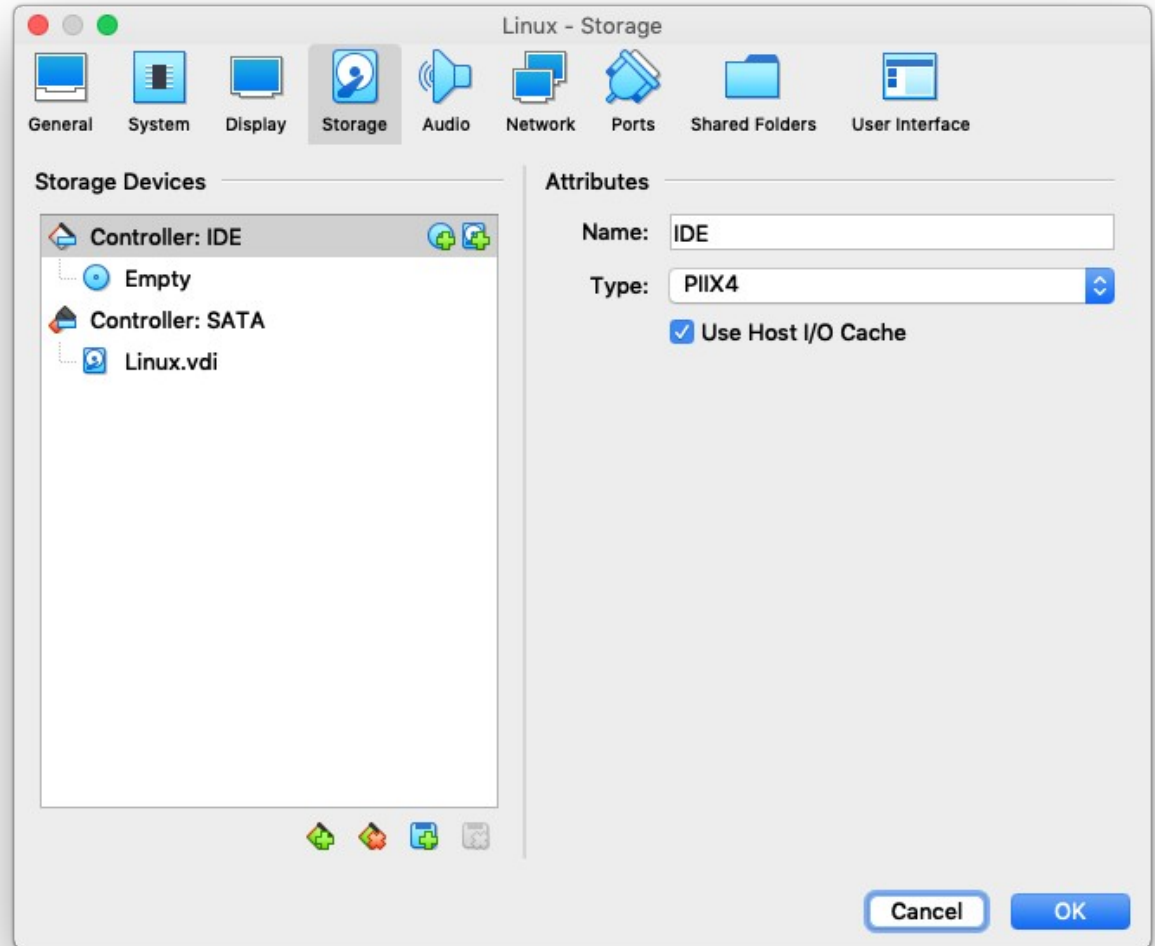
Може да променяме параметрите за  
отдалечен достъп до дисплея.

# VirtualBox – Настройки – Възпроизвеждане – Запис

Може да променяме вида и начина  
за запис на видео от дисплея на  
ВМ.



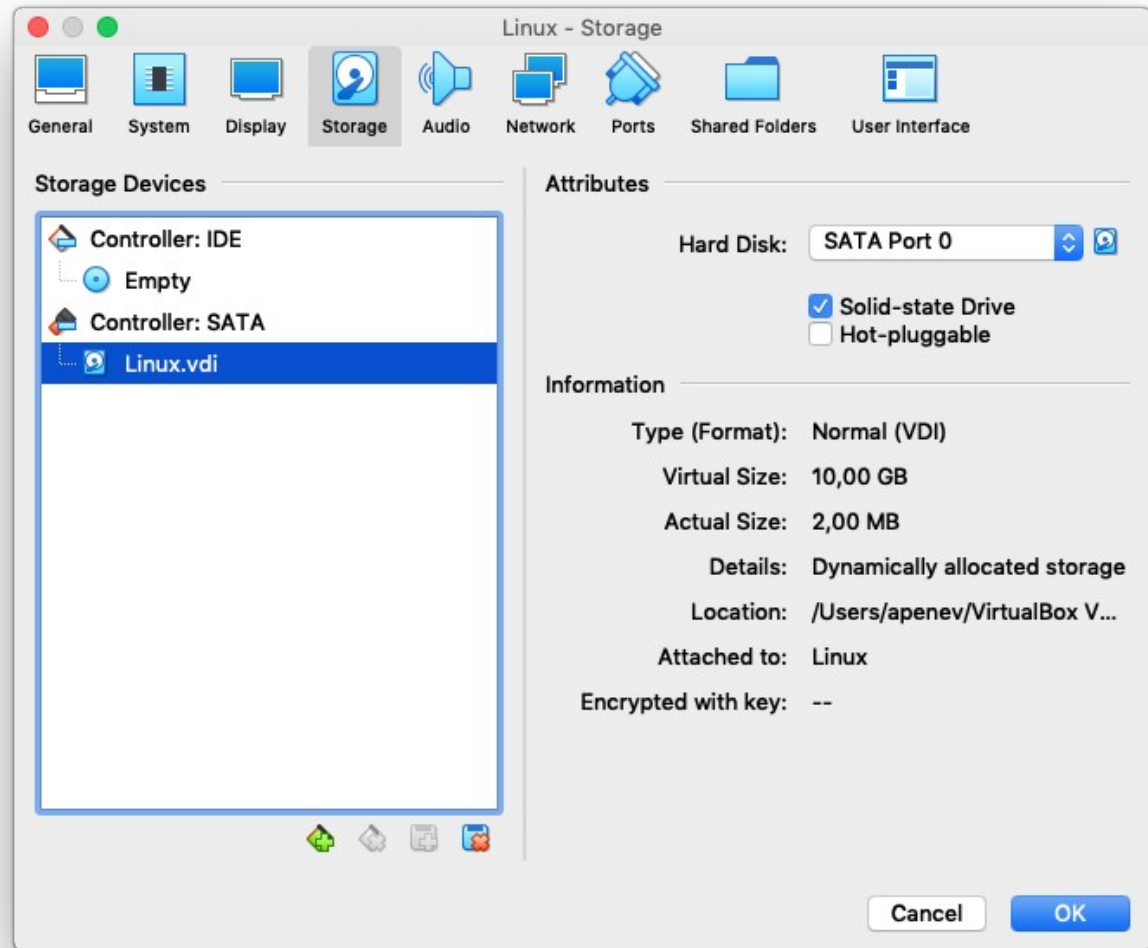
# VirtualBox – Настройки – Съхранение – Контролери



Може да променяме настройките на контролерите за дискове, ...

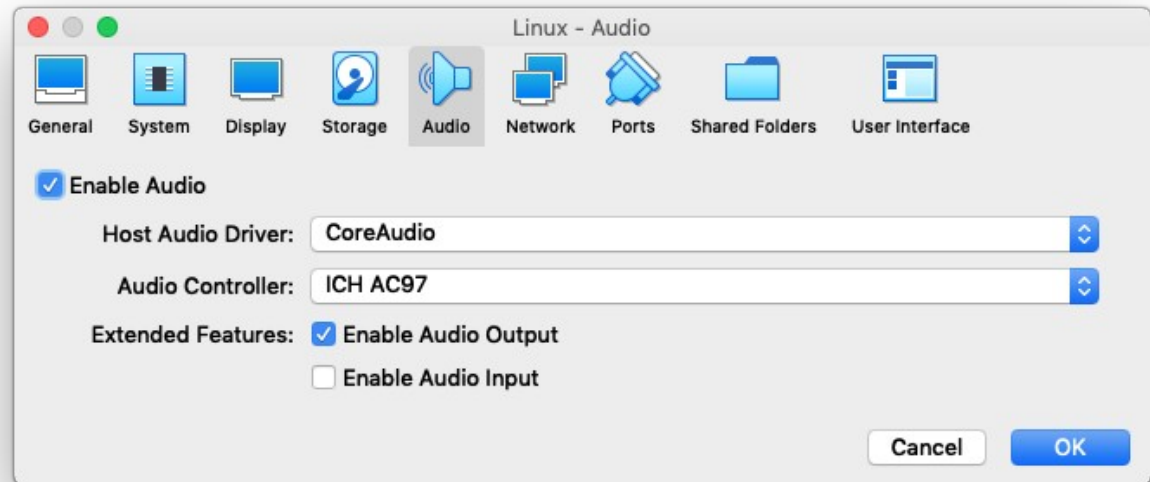
# VirtualBox – Настройки – Съхранение – Диск

... както и свързаните към тях виртуални дискове (Имиджи).



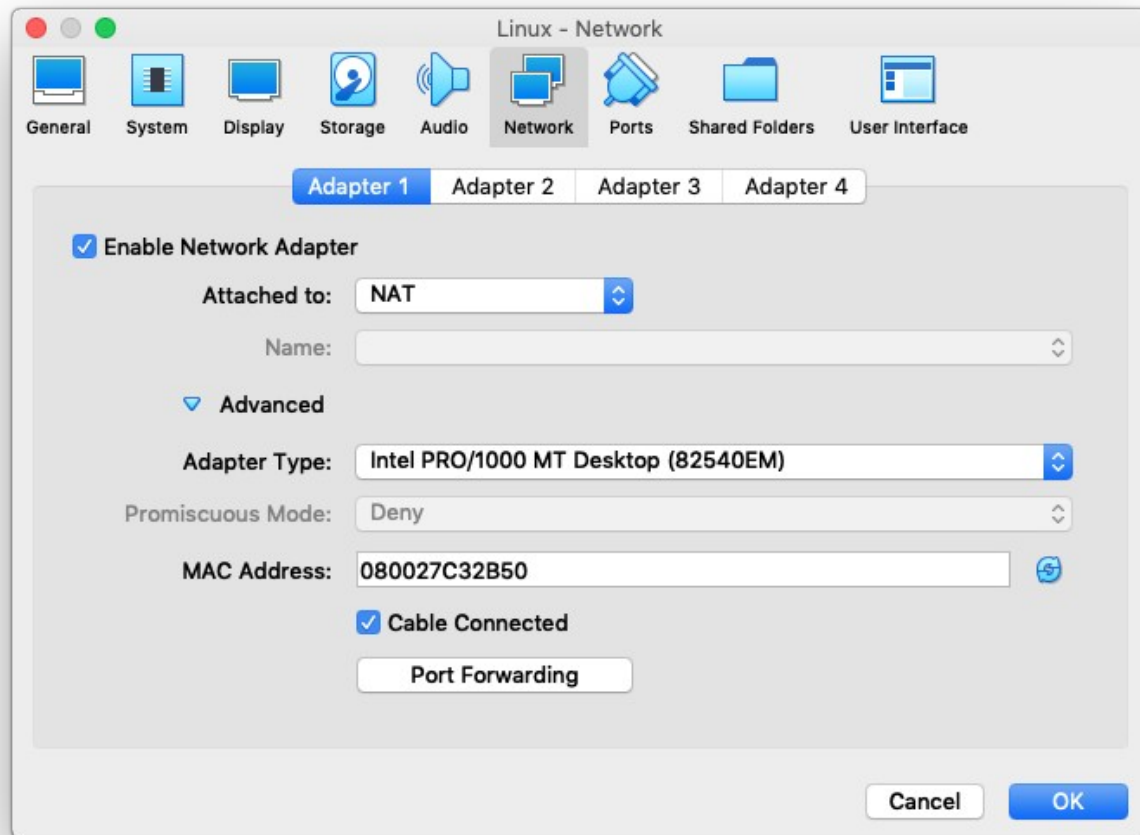
# VirtualBox – Настройки – Звук

Може да променяме параметрите на аудио картата на ВМ.



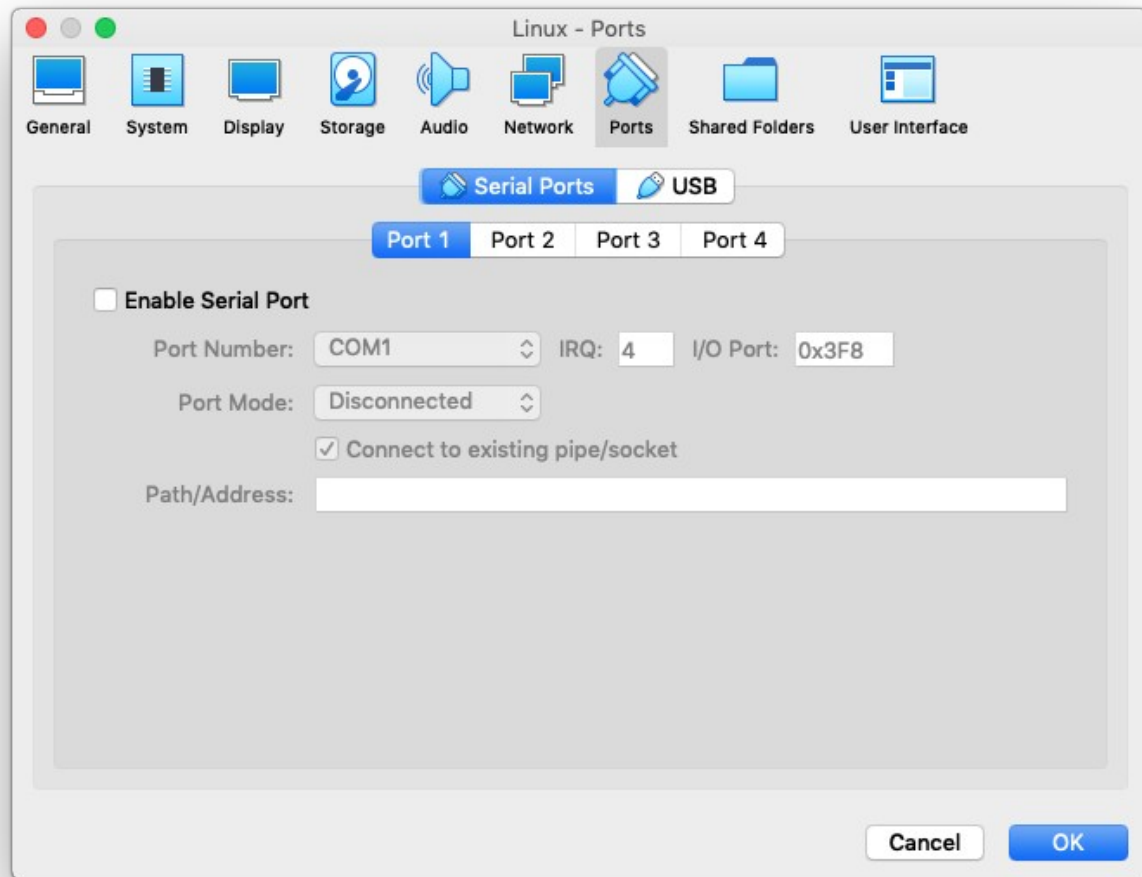
# VirtualBox – Настройки – Мрежа

Може да променяме параметрите на мрежовите карти, както и начина на тяхното свързване, адресите на мрежовите устройства, пренасочването на портове от Host към Guest и др.

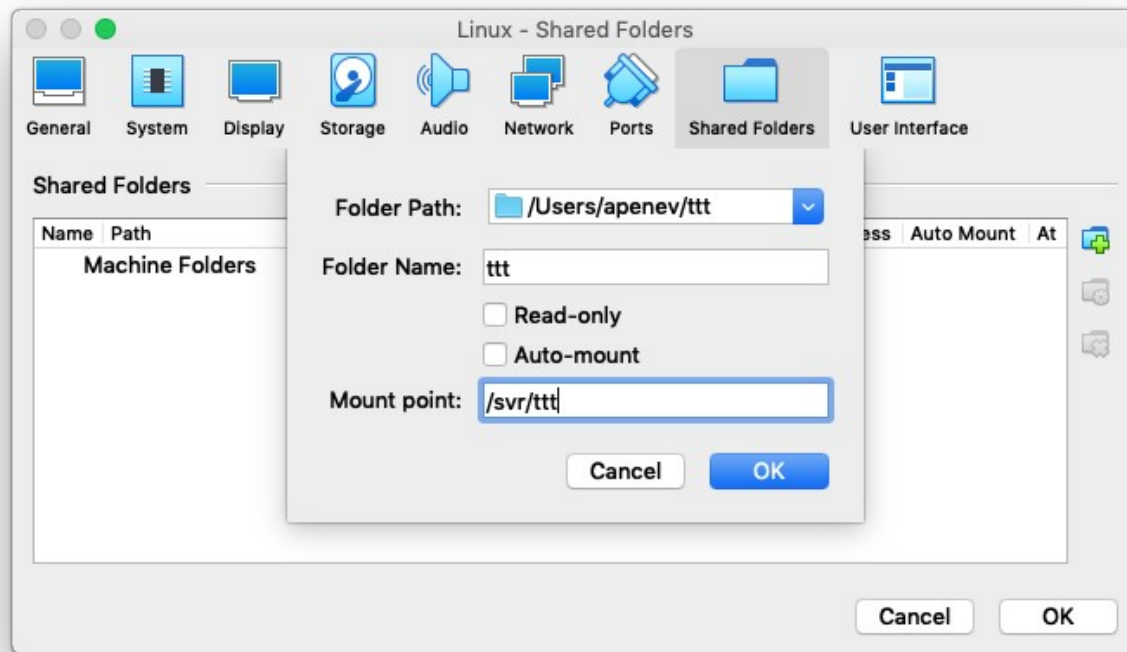


# VirtualBox – Настройки – Портове – Серийни портове

Може да променяме параметрите на серийните, USB и други портове, както и устройствата от Guest машината свързани с тях.



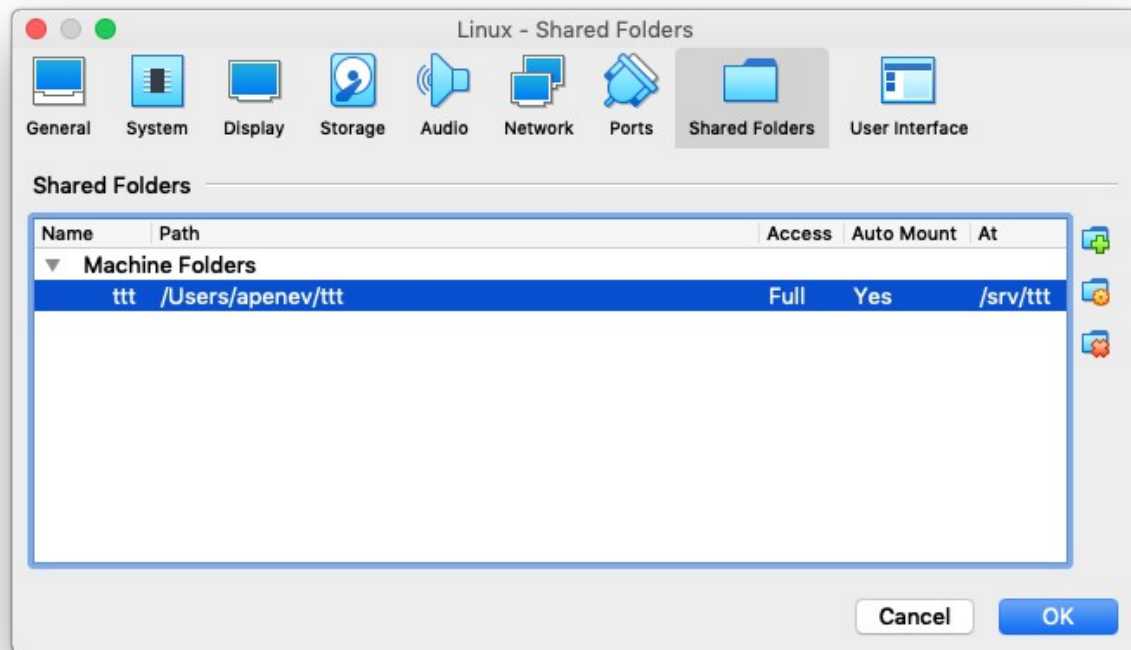
# VirtualBox – Настройки – Споделени папки



Може да променяме споделените папки с VM.

# VirtualBox – Настройки – Споделени папки

Може да променяме и премахваме споделените папки с VM.



# VirtualBox – ВМ – Готова за старт

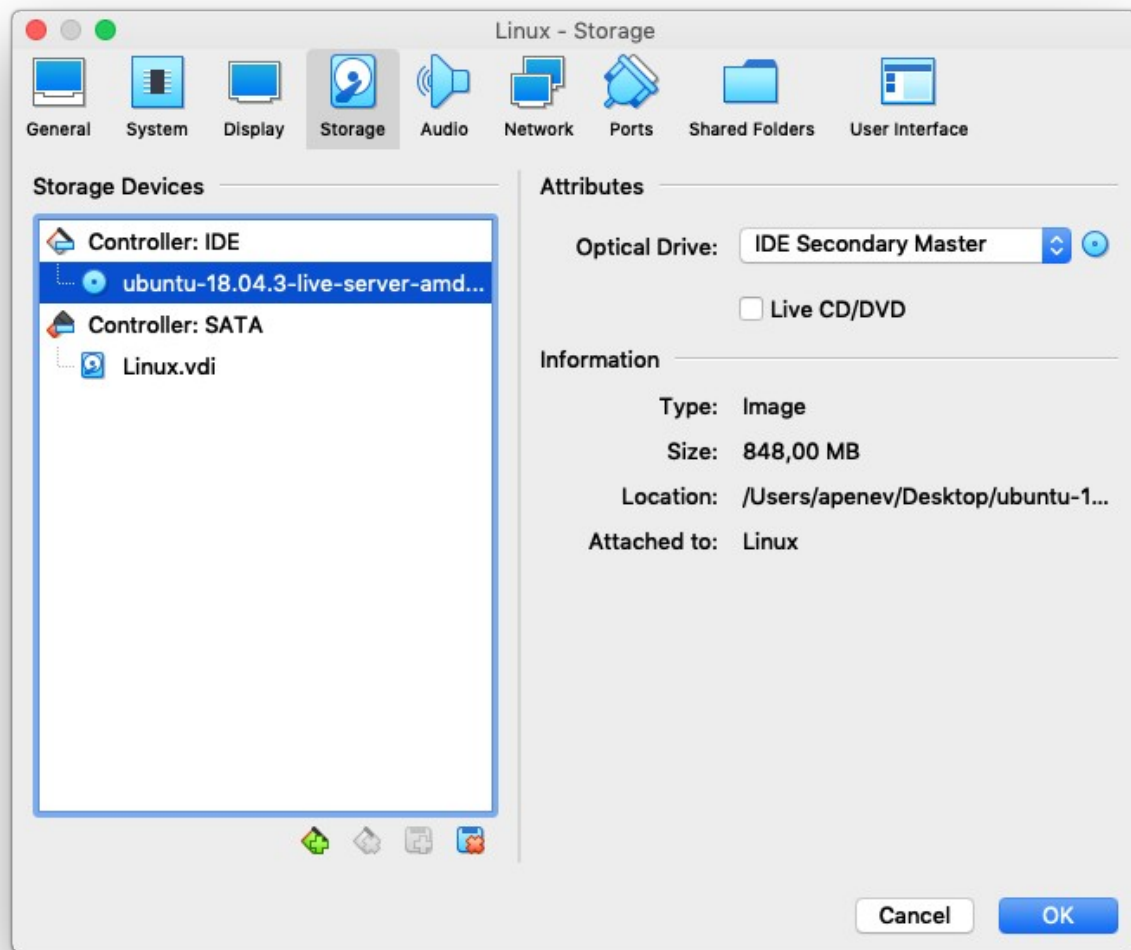
Виртуалната машина е  
настроена и готова за  
Start.



# Инсталация на Linux

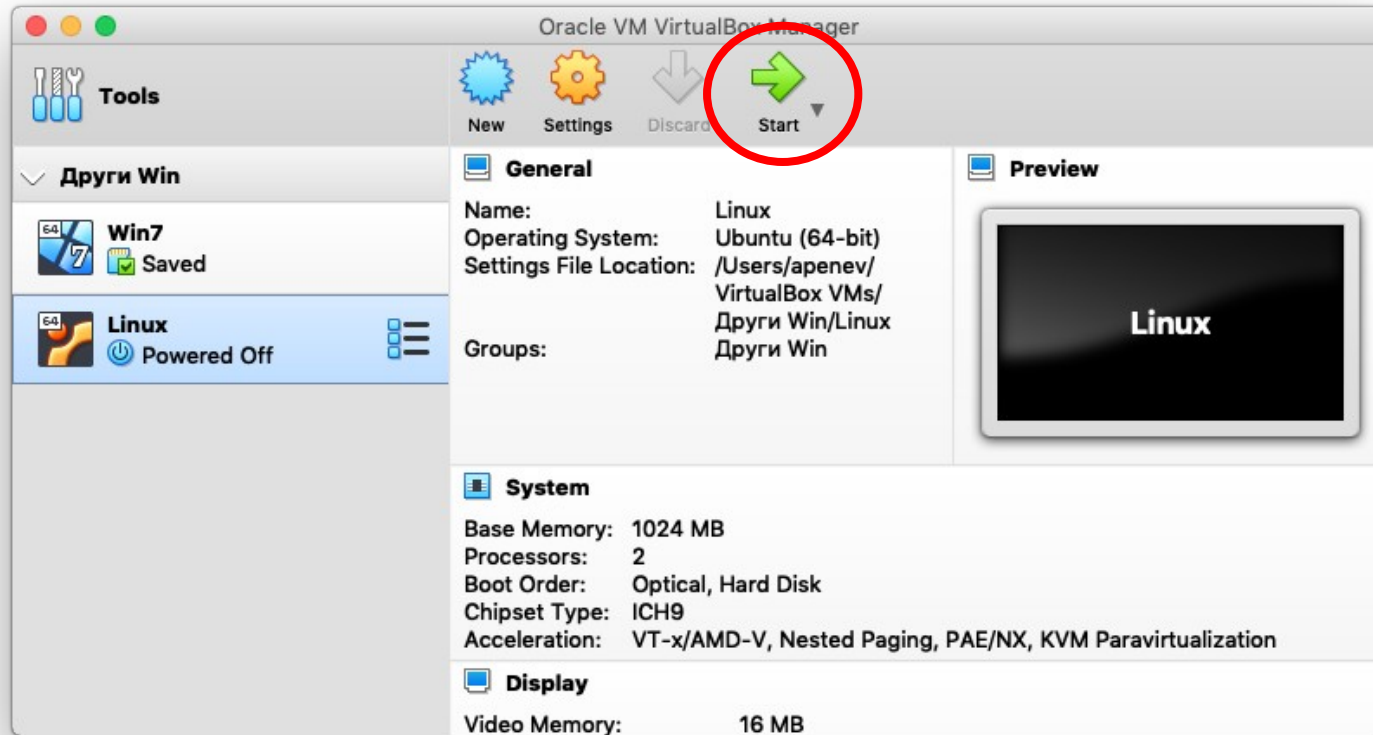
*(Ubuntu 18.04 LTS)*

# Инсталация на Linux във VM

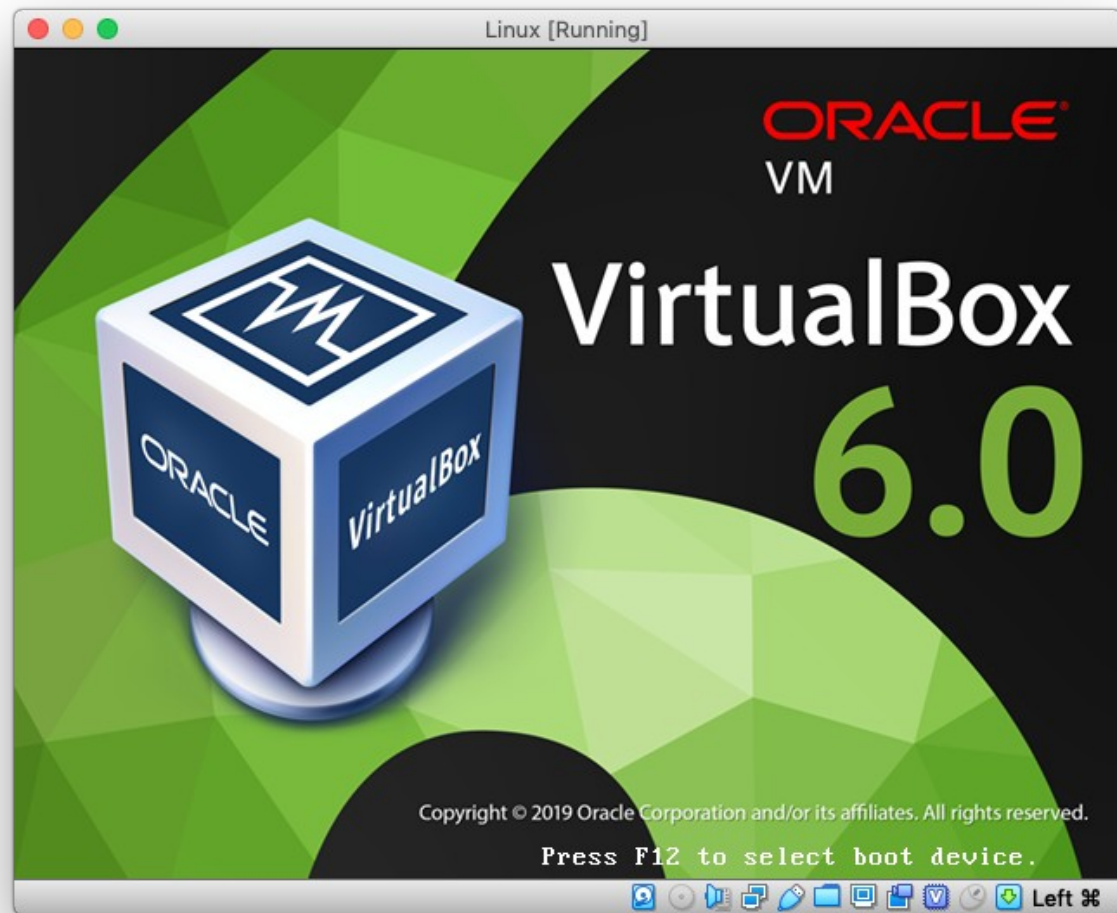


Преди да стартираме VM, ще свалим и закачим за CD устройството на VM един инсталационен ISO имидж с Ubuntu 18.04 LTS.

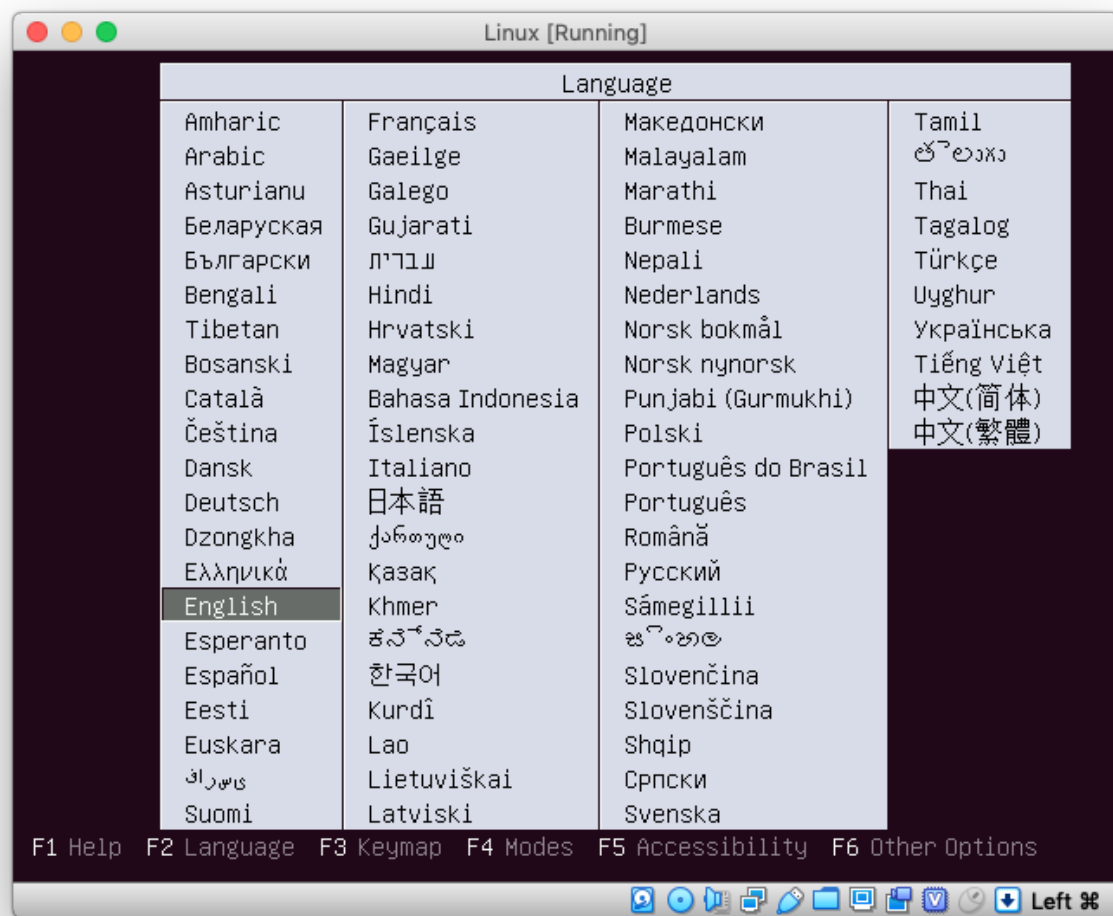
# VirtualBox – ВМ – Готова за старт



# Инсталация на Linux във VM

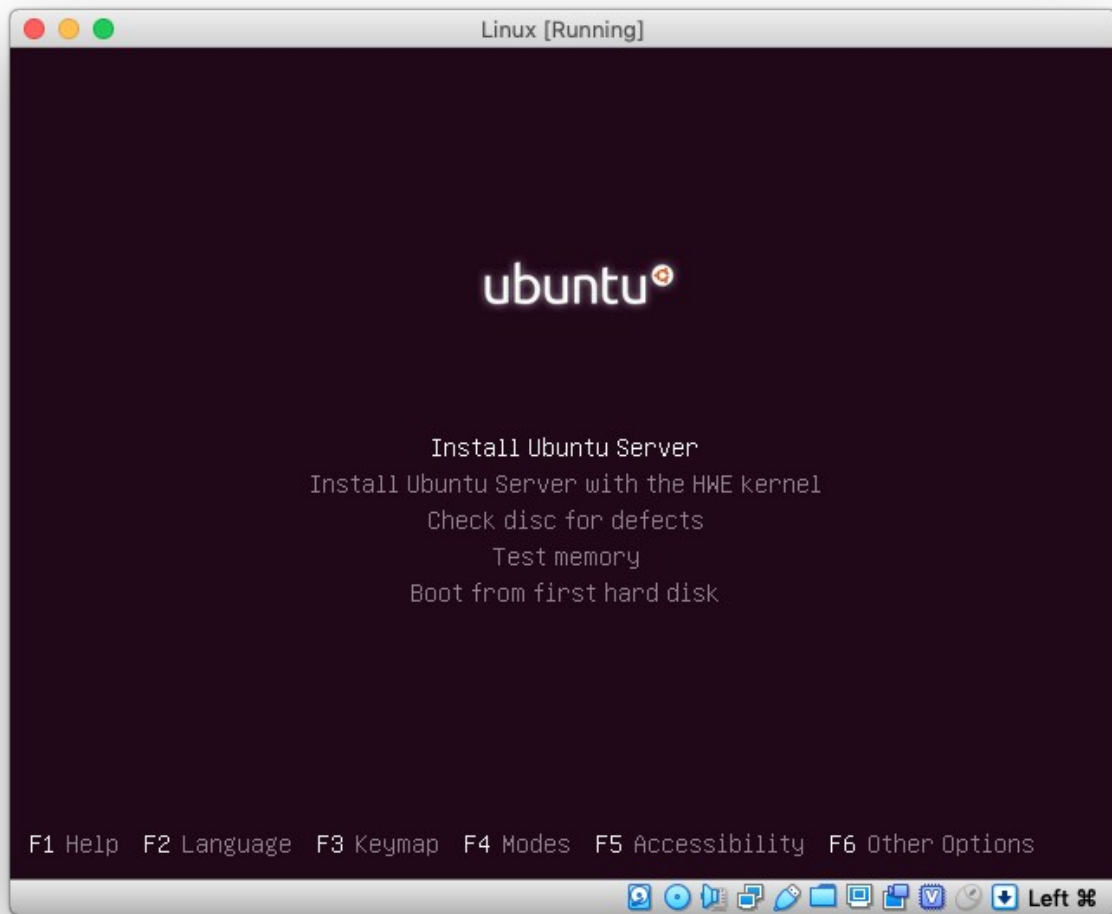


# Инсталация на Linux във ВМ



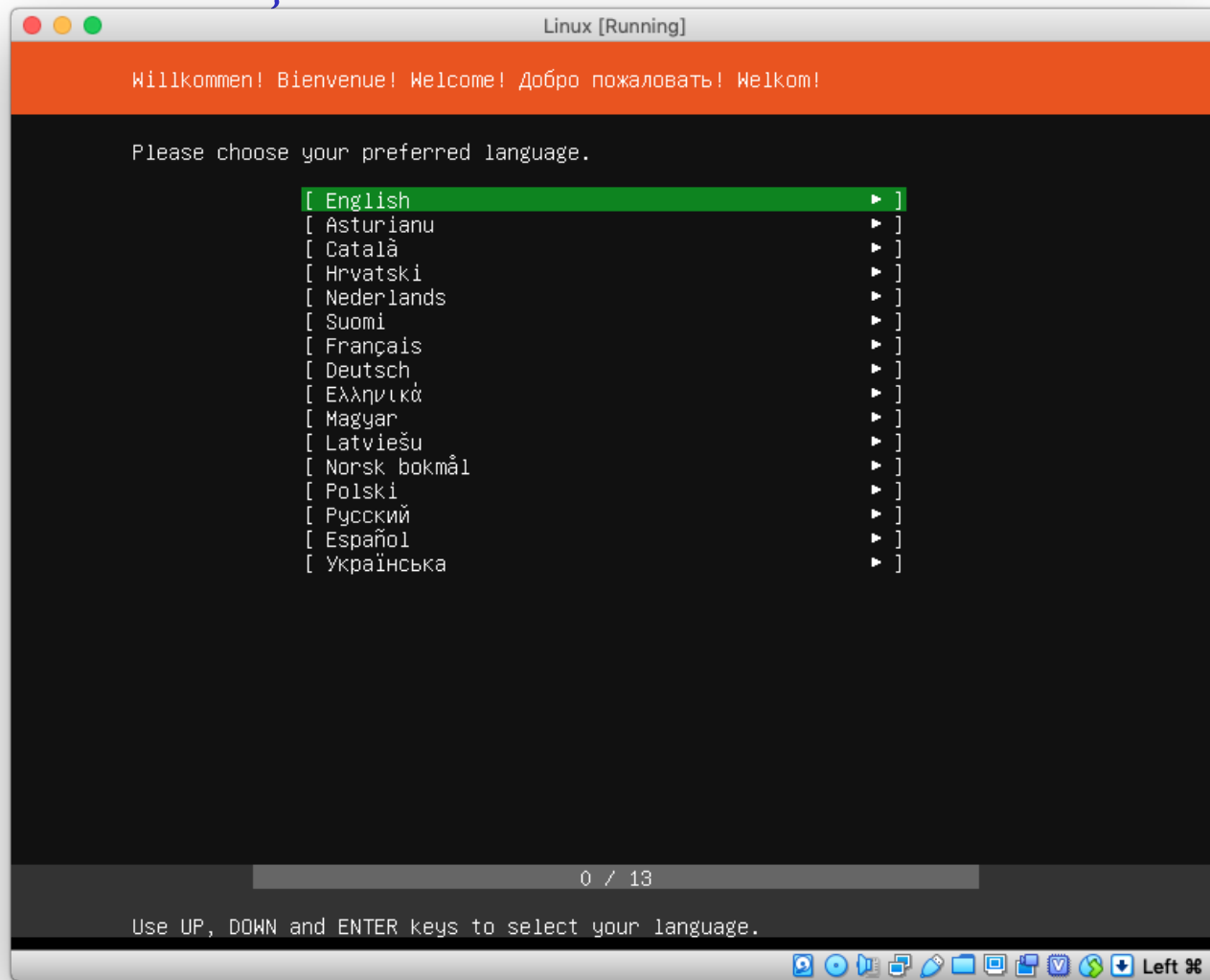
Избираме език за инсталацията.

# Инсталация на Linux във VM



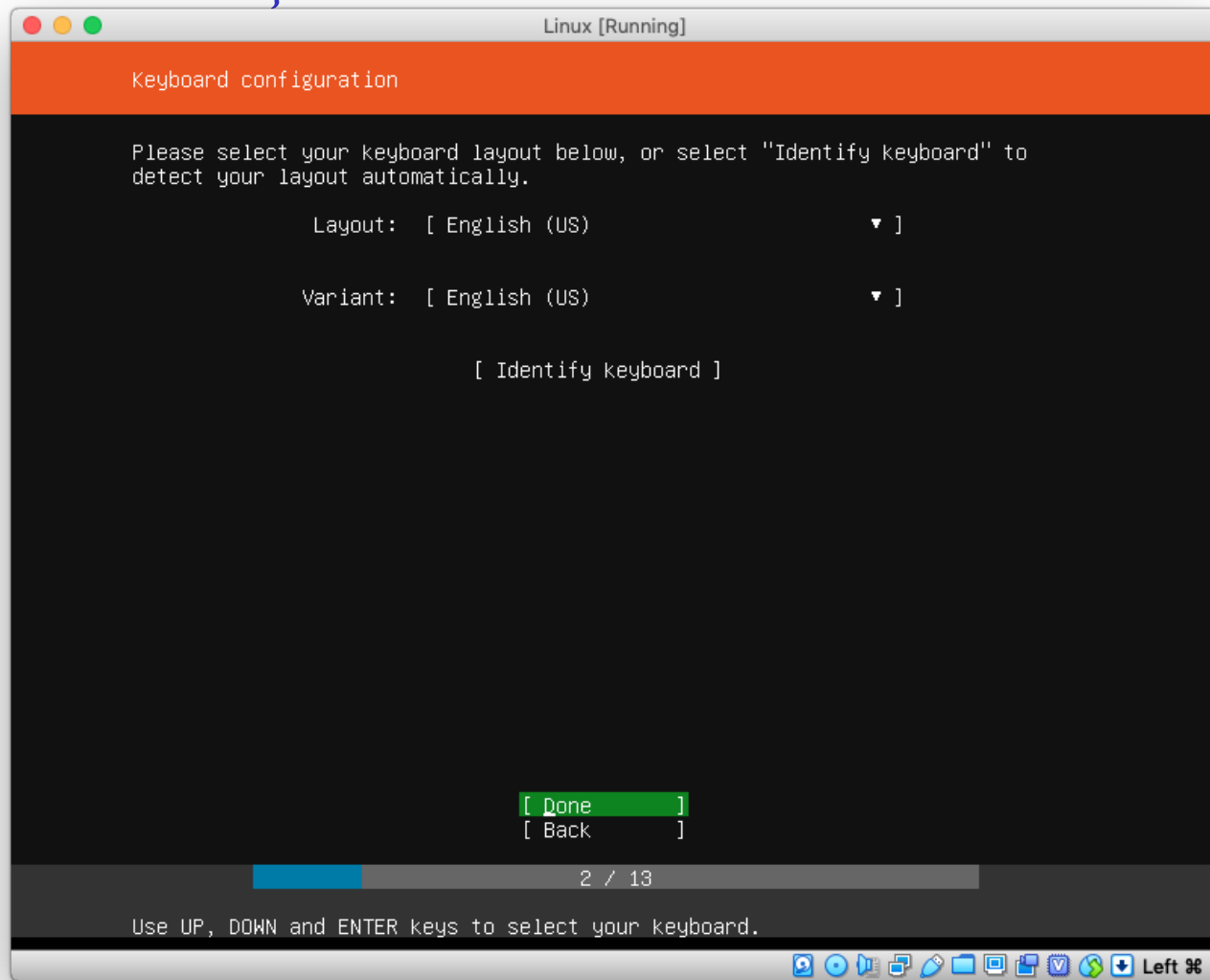
Избираме инсталация на  
Ubuntu Server.

# Инсталация на Linux във ВМ



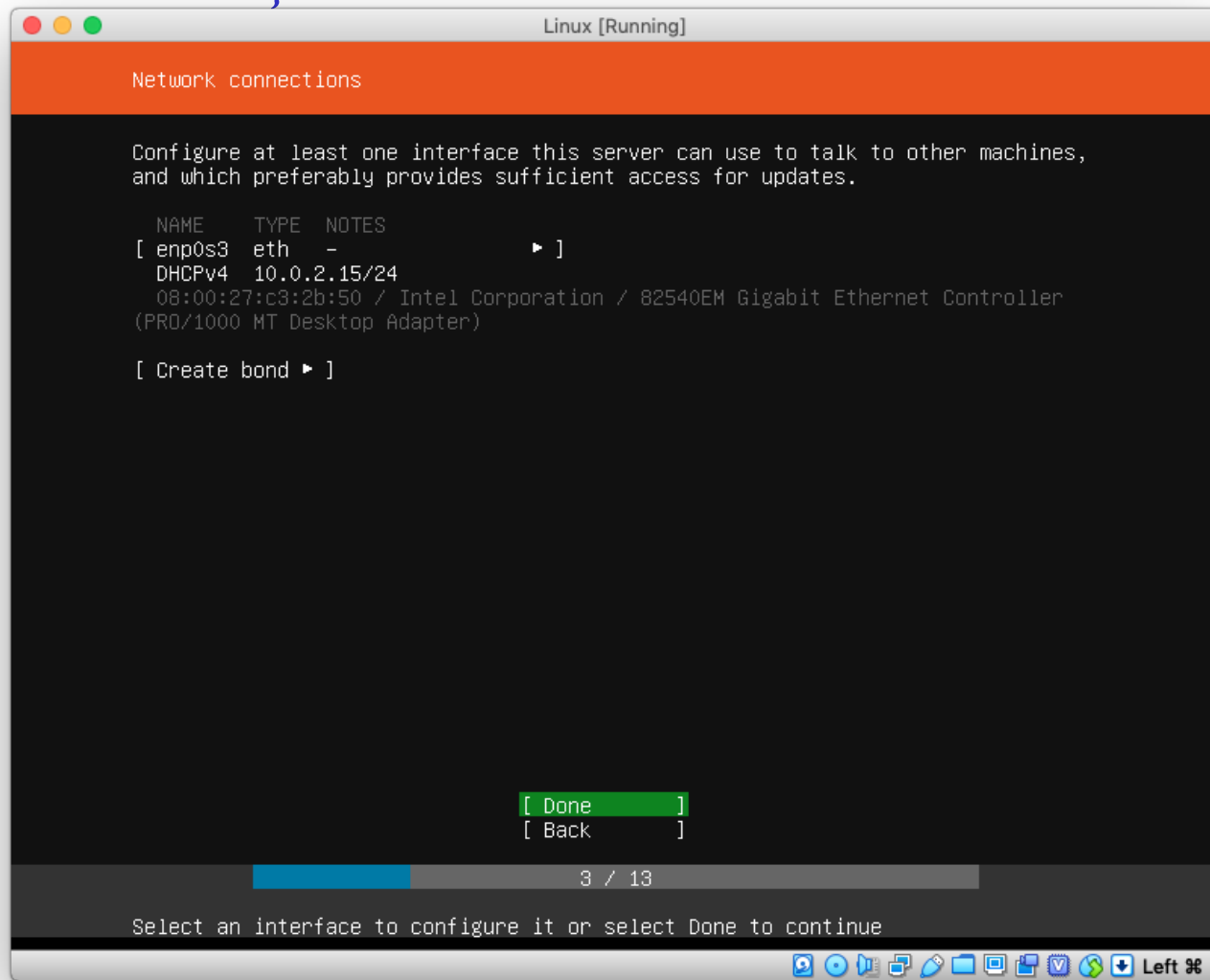
Език.

# Инсталация на Linux във ВМ



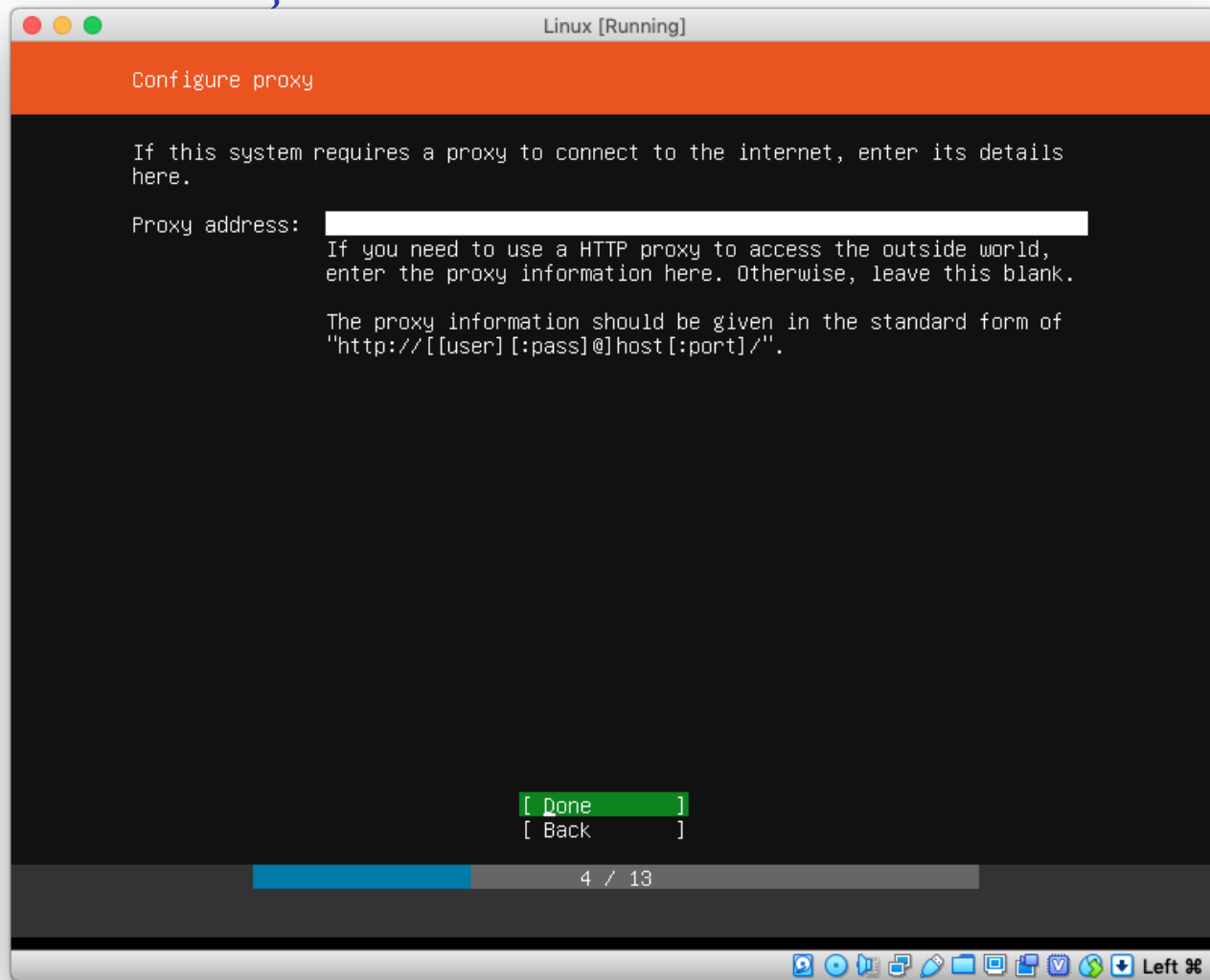
Подредба на клавиатурата.

# Инсталация на Linux във VM



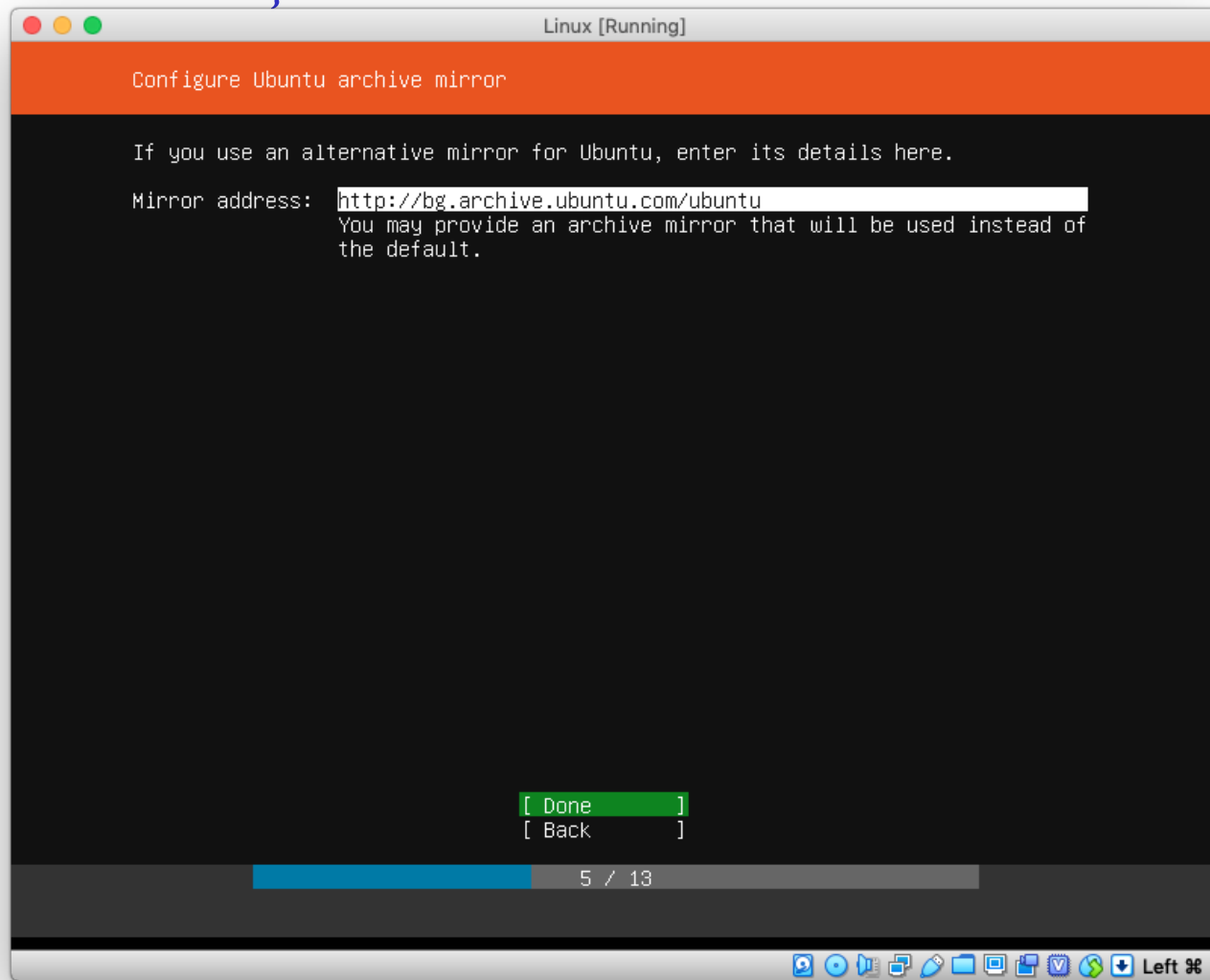
Параметризираме  
мрежовите карти.

# Инсталация на Linux във VM



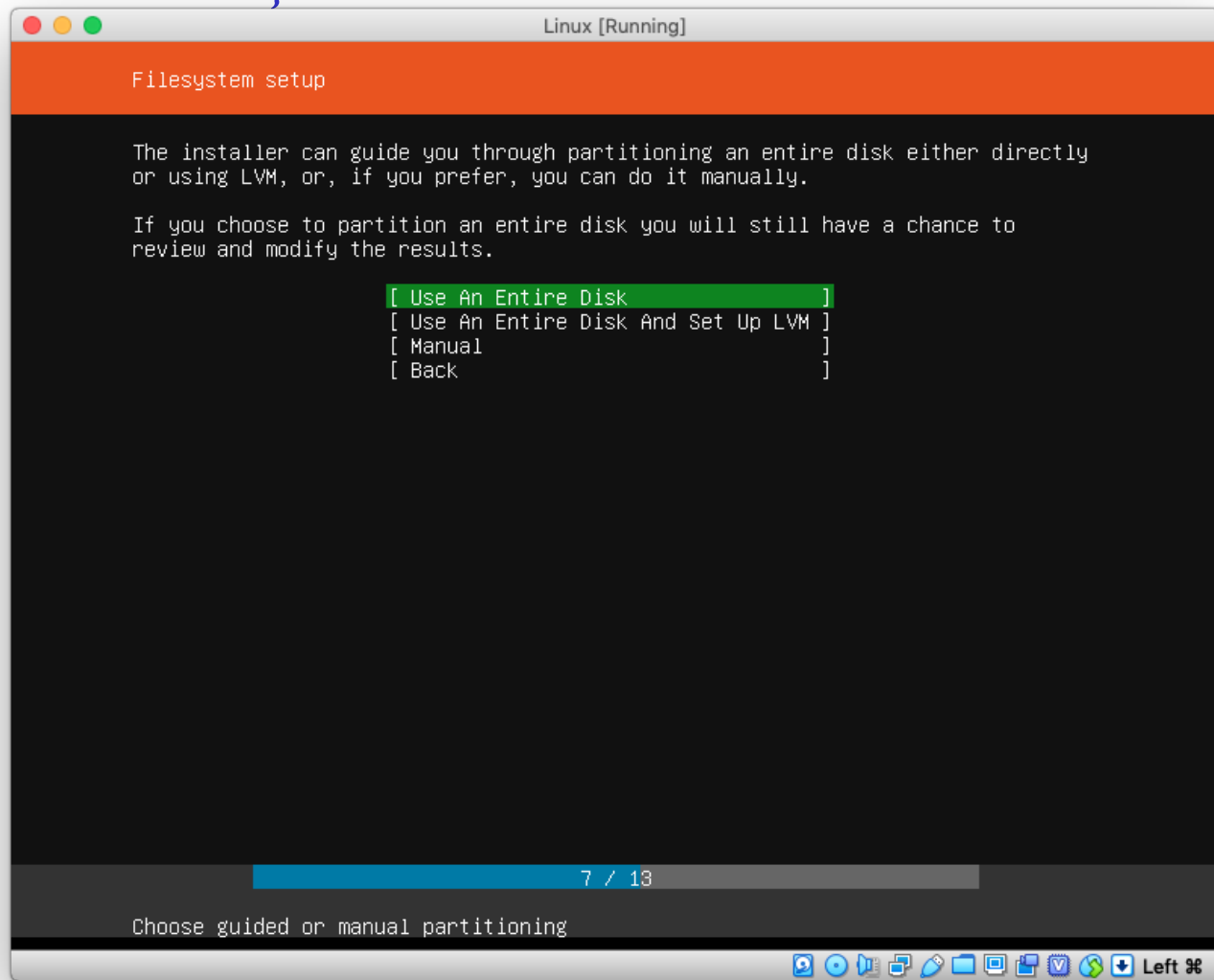
Прокси сървър за достъп до мрежата и Интернет.

# Инсталация на Linux във ВМ



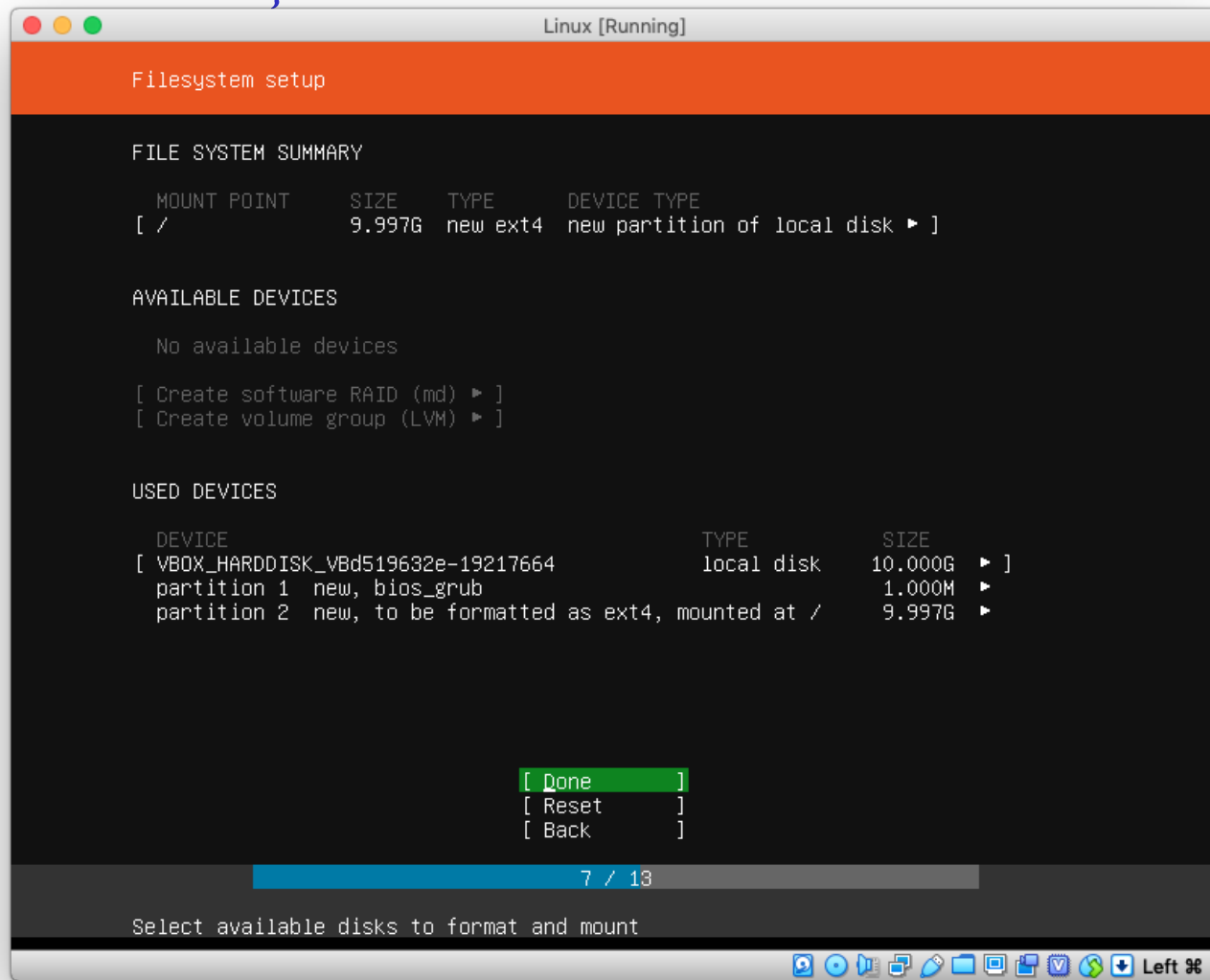
Мирър сайт за сваляне на пакети и обновления за Ubuntu.

# Инсталация на Linux във VM



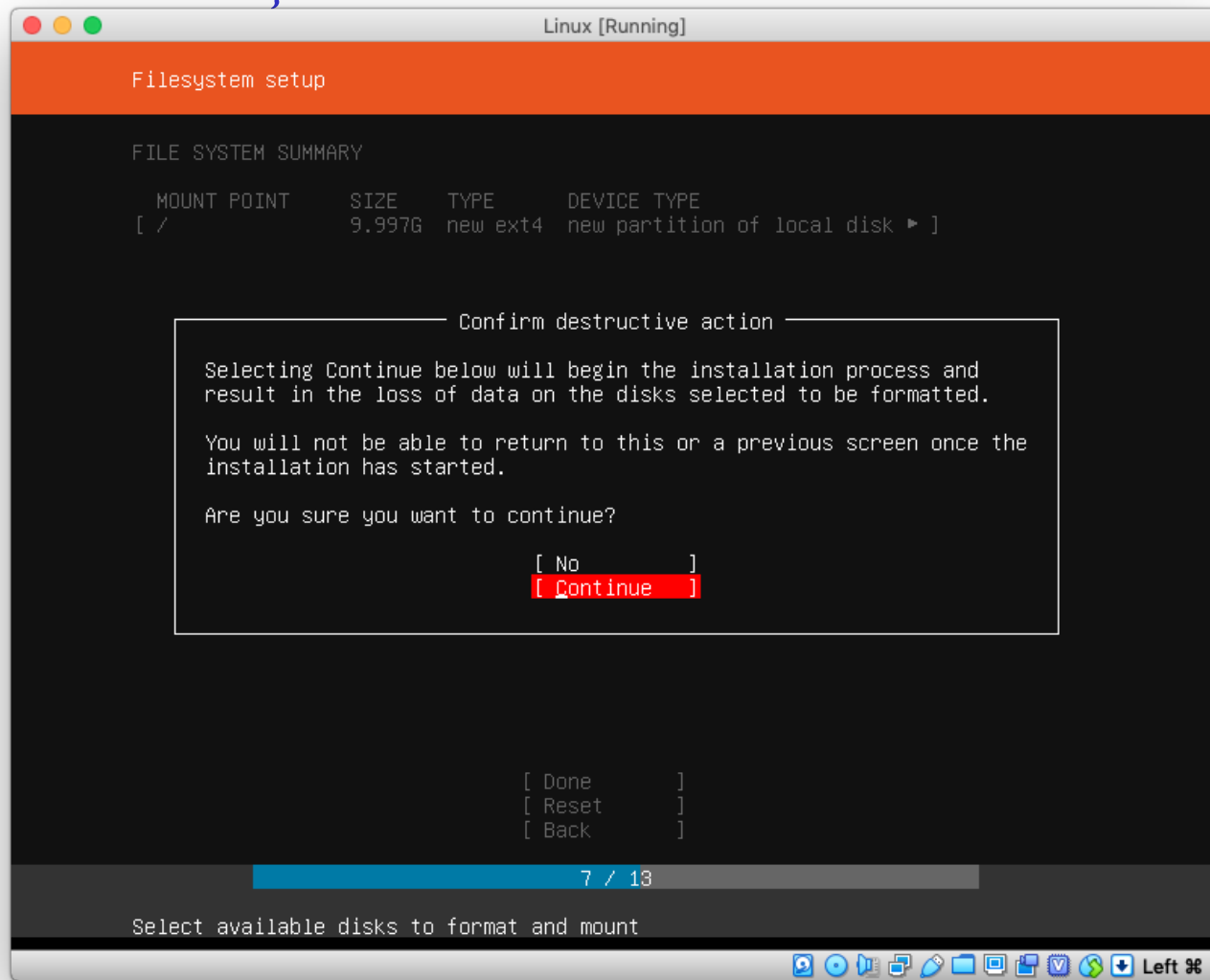
Как да разделим диска.  
Оставяме по подразбиране.

# Инсталация на Linux във VM



Как да разделим диска.  
Оставяме по подразбиране.

# Инсталация на Linux във VM



Да, искаме да презапишем виртуалния диск на VM и да продължим с инсталацията.

# Инсталация на Linux във VM

Linux [Running]

Profile setup

Enter the username and password you will use to log in to the system. You can configure SSH access on the next screen but a password is still needed for sudo.

Your name:

Your server's name:   
The name it uses when it talks to other computers.

Pick a username:

Choose a password:

Confirm your password:

[ Done ]

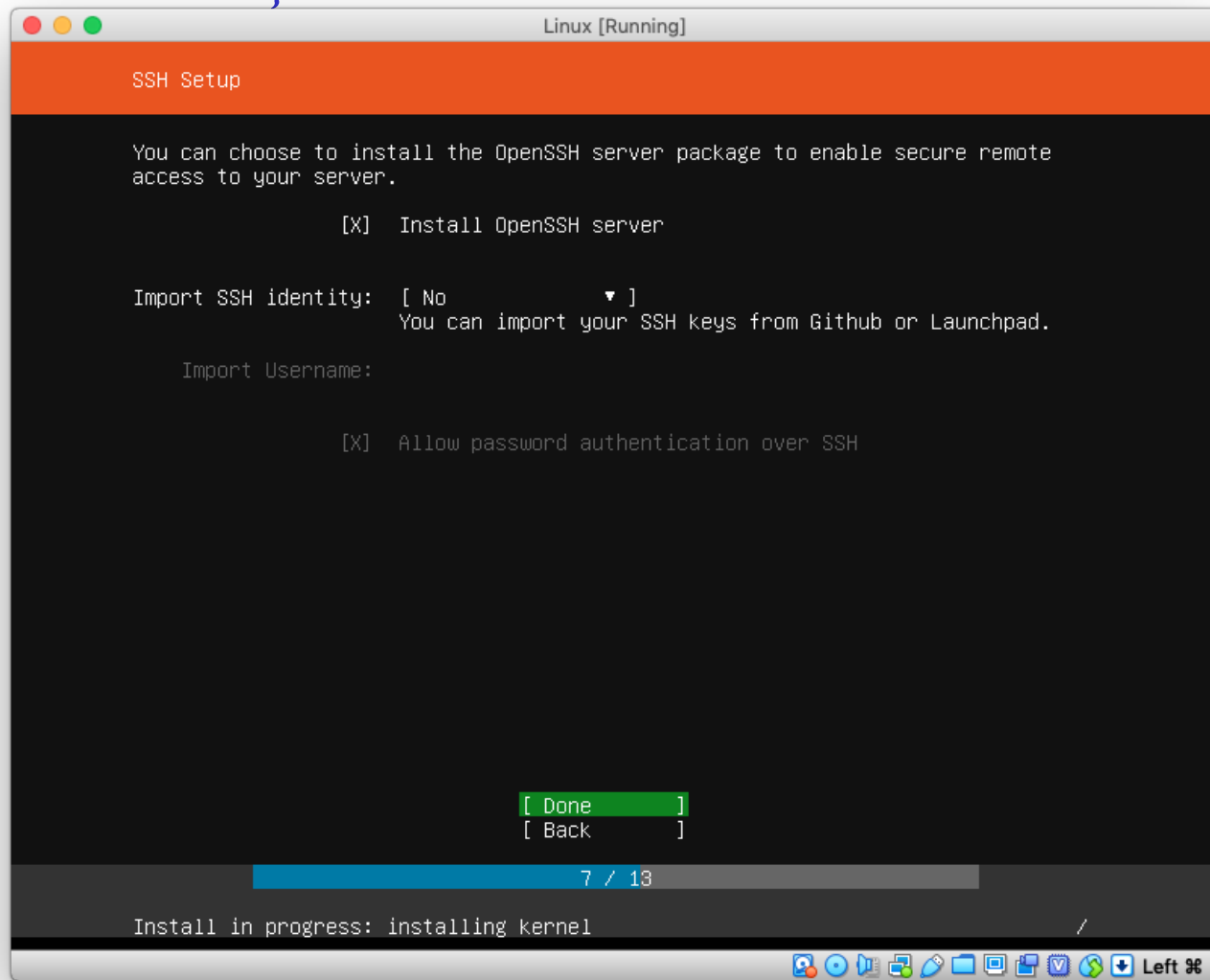
7 / 13

Install in progress: curtin command in-target

Left %

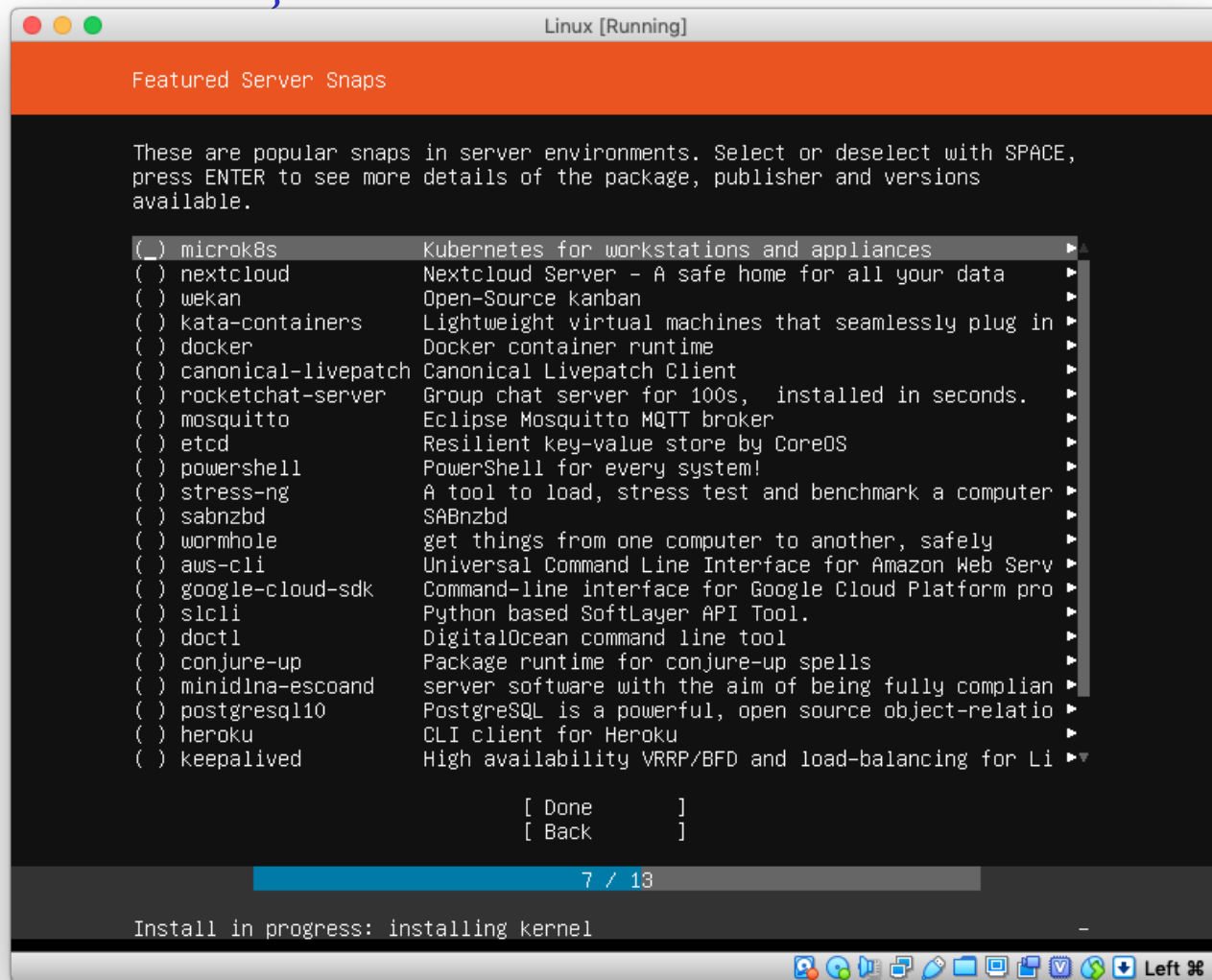
Въвеждаме нашето име,  
име на сървъра,  
потребителско име и  
парола.

# Инсталация на Linux във VM



Добре да да инсталираме SSH сървър на VM, за да може да правим отдалечен достъп до нея. Това не винаги е нужно, защото винаги може да направим достъп до нея локално чрез VirtualBox.

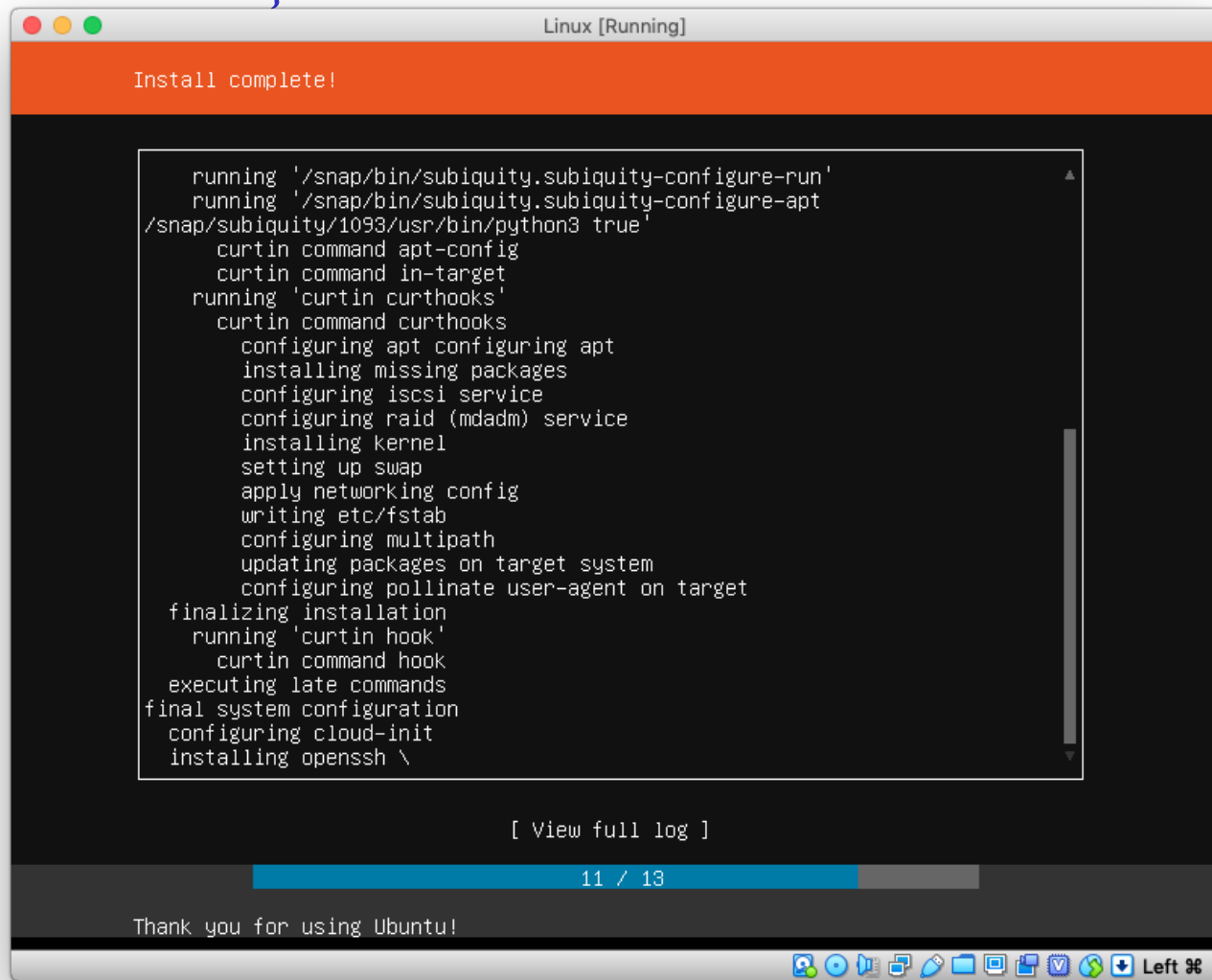
# Инсталация на Linux във VM



Кои пакети искаме да се инсталират по време на инсталацията на Ubuntu.

За сега оставяме без промени. Искаме минимална инсталация, а и в последствие винаги може да ги инсталираме с apt-get

# Инсталация на Linux във ВМ



```
Linux [Running]

Install complete!

running '/snap/bin/subiquity.subiquity-configure-run'
running '/snap/bin/subiquity.subiquity-configure-apt'
/snap/subiquity/1093/usr/bin/python3 true'
curtin command apt-config
curtin command in-target
running 'curtin curthooks'
curtin command curthooks
configuring apt configuring apt
installing missing packages
configuring iscsi service
configuring raid (mdadm) service
installing kernel
setting up swap
apply networking config
writing etc/fstab
configuring multipath
updating packages on target system
configuring pollinate user-agent on target
finalizing installation
running 'curtin hook'
curtin command hook
executing late commands
final system configuration
configuring cloud-init
installing openssh \

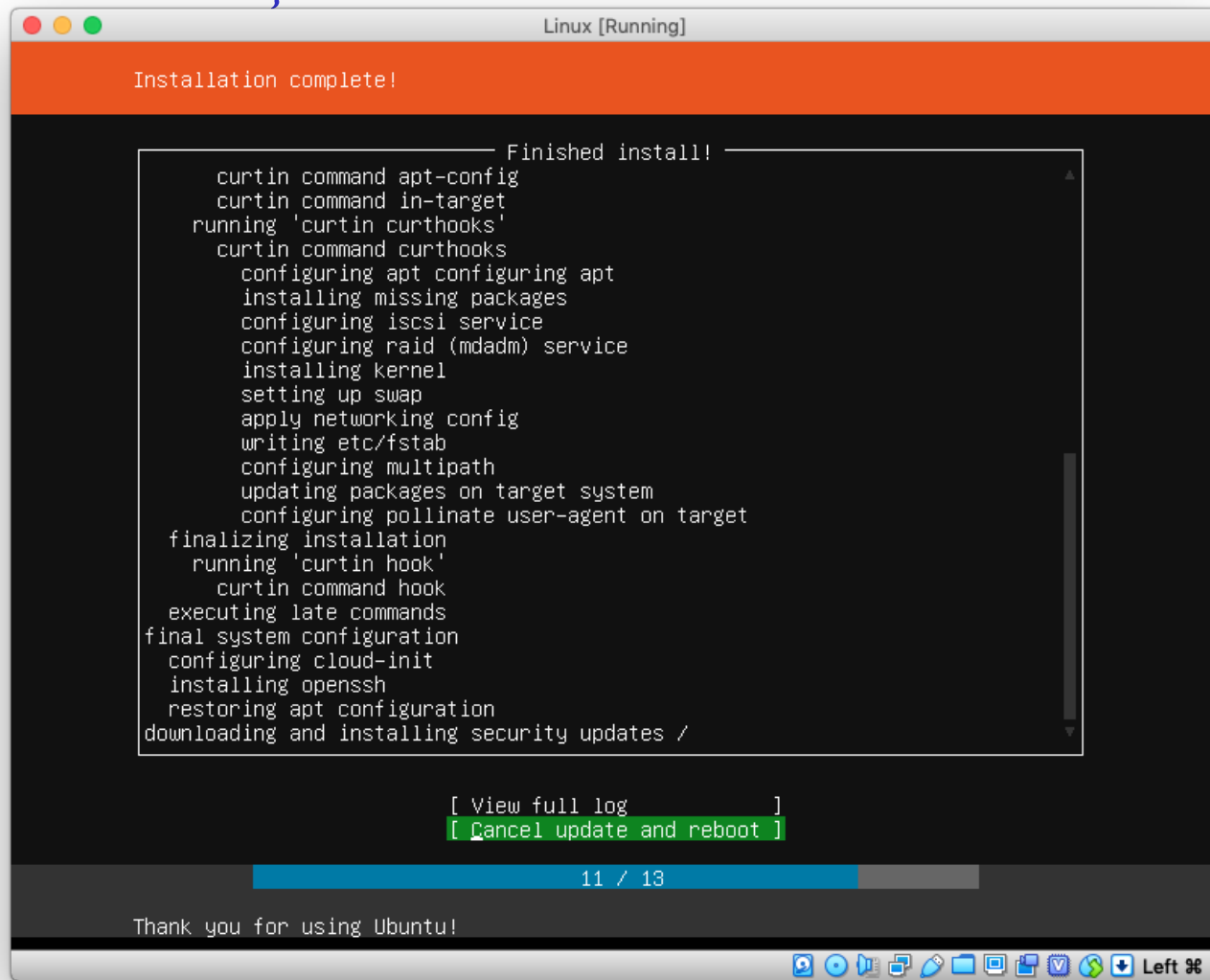
[ View full log ]

11 / 13

Thank you for using Ubuntu!
```

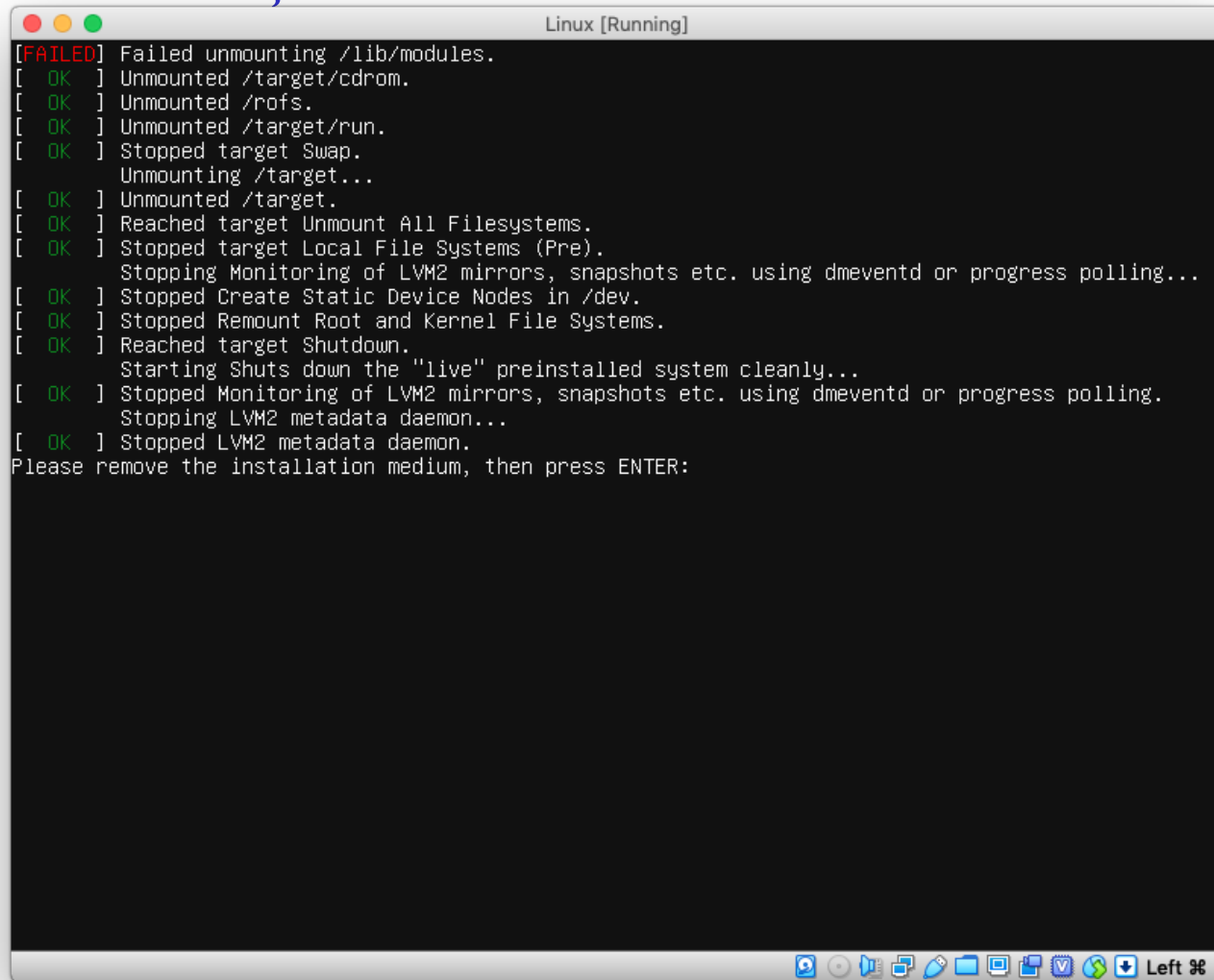
Инсталацията в ход...

# Инсталация на Linux във VM



Инсталацията в ход...

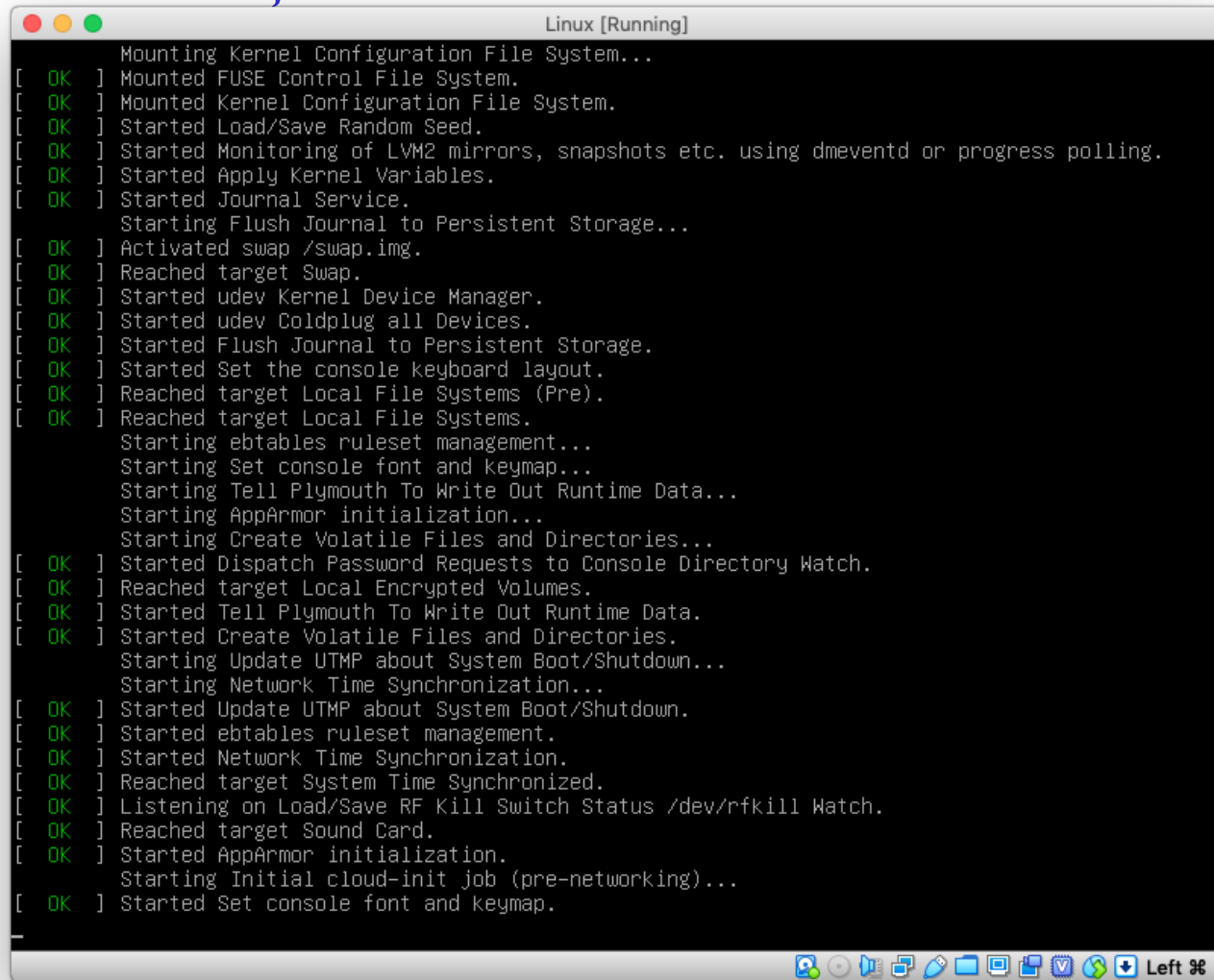
# Инсталация на Linux във VM



```
Linux [Running]
[FAILED] Failed unmounting /lib/modules.
[ OK ] Unmounted /target/cdrom.
[ OK ] Unmounted /rofs.
[ OK ] Unmounted /target/run.
[ OK ] Stopped target Swap.
      Unmounting /target...
[ OK ] Unmounted /target.
[ OK ] Reached target Unmount All Filesystems.
[ OK ] Stopped target Local File Systems (Pre).
      Stopping Monitoring of LVM2 mirrors, snapshots etc. using dmeventd or progress polling...
[ OK ] Stopped Create Static Device Nodes in /dev.
[ OK ] Stopped Remount Root and Kernel File Systems.
[ OK ] Reached target Shutdown.
      Starting Shuts down the "live" preinstalled system cleanly...
[ OK ] Stopped Monitoring of LVM2 mirrors, snapshots etc. using dmeventd or progress polling.
      Stopping LVM2 metadata daemon...
[ OK ] Stopped LVM2 metadata daemon.
Please remove the installation medium, then press ENTER:
```

Системата рестартира и Ubuntu започва зареждането си. Ще се стартира сървърната версия на системата без GUI.

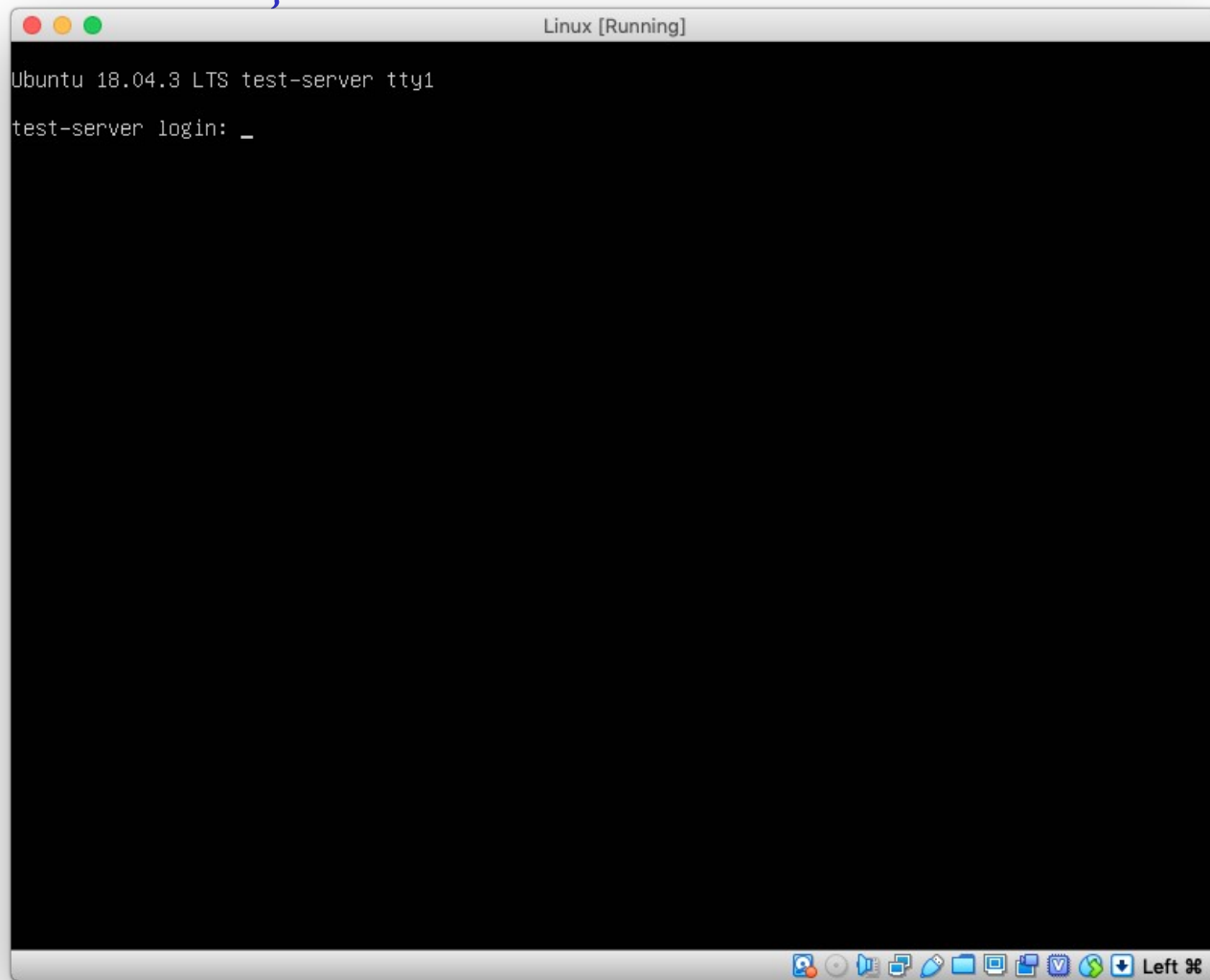
# Инсталация на Linux във ВМ



```
Linux [Running]
Mounting Kernel Configuration File System...
[ OK ] Mounted FUSE Control File System.
[ OK ] Mounted Kernel Configuration File System.
[ OK ] Started Load/Save Random Seed.
[ OK ] Started Monitoring of LVM2 mirrors, snapshots etc. using dmeventd or progress polling.
[ OK ] Started Apply Kernel Variables.
[ OK ] Started Journal Service.
Starting Flush Journal to Persistent Storage...
[ OK ] Activated swap /swap.img.
[ OK ] Reached target Swap.
[ OK ] Started udev Kernel Device Manager.
[ OK ] Started udev Coldplug all Devices.
[ OK ] Started Flush Journal to Persistent Storage.
[ OK ] Started Set the console keyboard layout.
[ OK ] Reached target Local File Systems (Pre).
[ OK ] Reached target Local File Systems.
Starting ebttables ruleset management...
Starting Set console font and keymap...
Starting Tell Plymouth To Write Out Runtime Data...
Starting AppArmor initialization...
Starting Create Volatile Files and Directories...
[ OK ] Started Dispatch Password Requests to Console Directory Watch.
[ OK ] Reached target Local Encrypted Volumes.
[ OK ] Started Tell Plymouth To Write Out Runtime Data.
[ OK ] Started Create Volatile Files and Directories.
Starting Update UTMP about System Boot/Shutdown...
Starting Network Time Synchronization...
[ OK ] Started Update UTMP about System Boot/Shutdown.
[ OK ] Started ebttables ruleset management.
[ OK ] Started Network Time Synchronization.
[ OK ] Reached target System Time Synchronized.
[ OK ] Listening on Load/Save RF Kill Switch Status /dev/rfkill Watch.
[ OK ] Reached target Sound Card.
[ OK ] Started AppArmor initialization.
Starting Initial cloud-init job (pre-networking)...
[ OK ] Started Set console font and keymap.
```

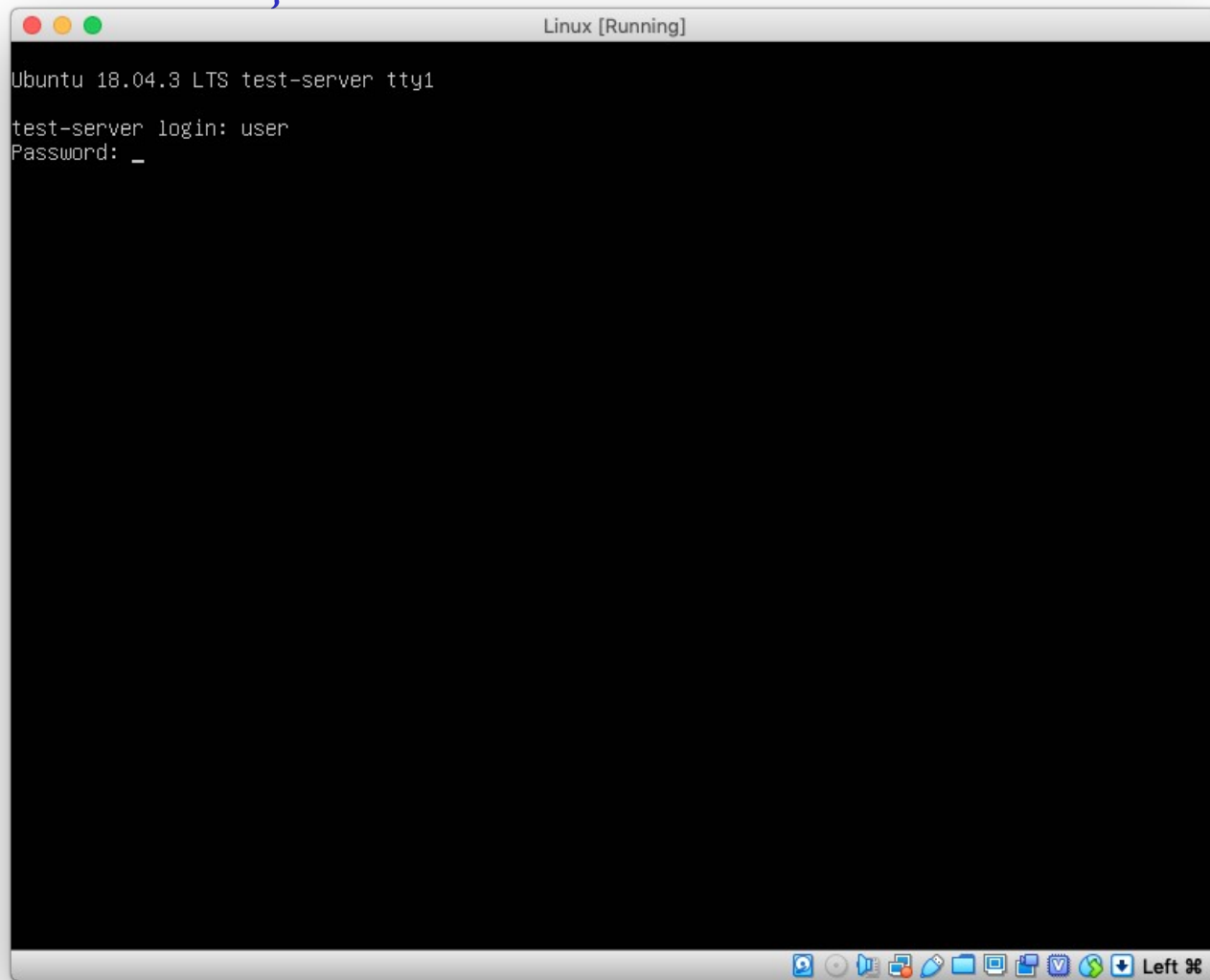
Зареждане на Ubuntu.

# Инсталация на Linux във VM



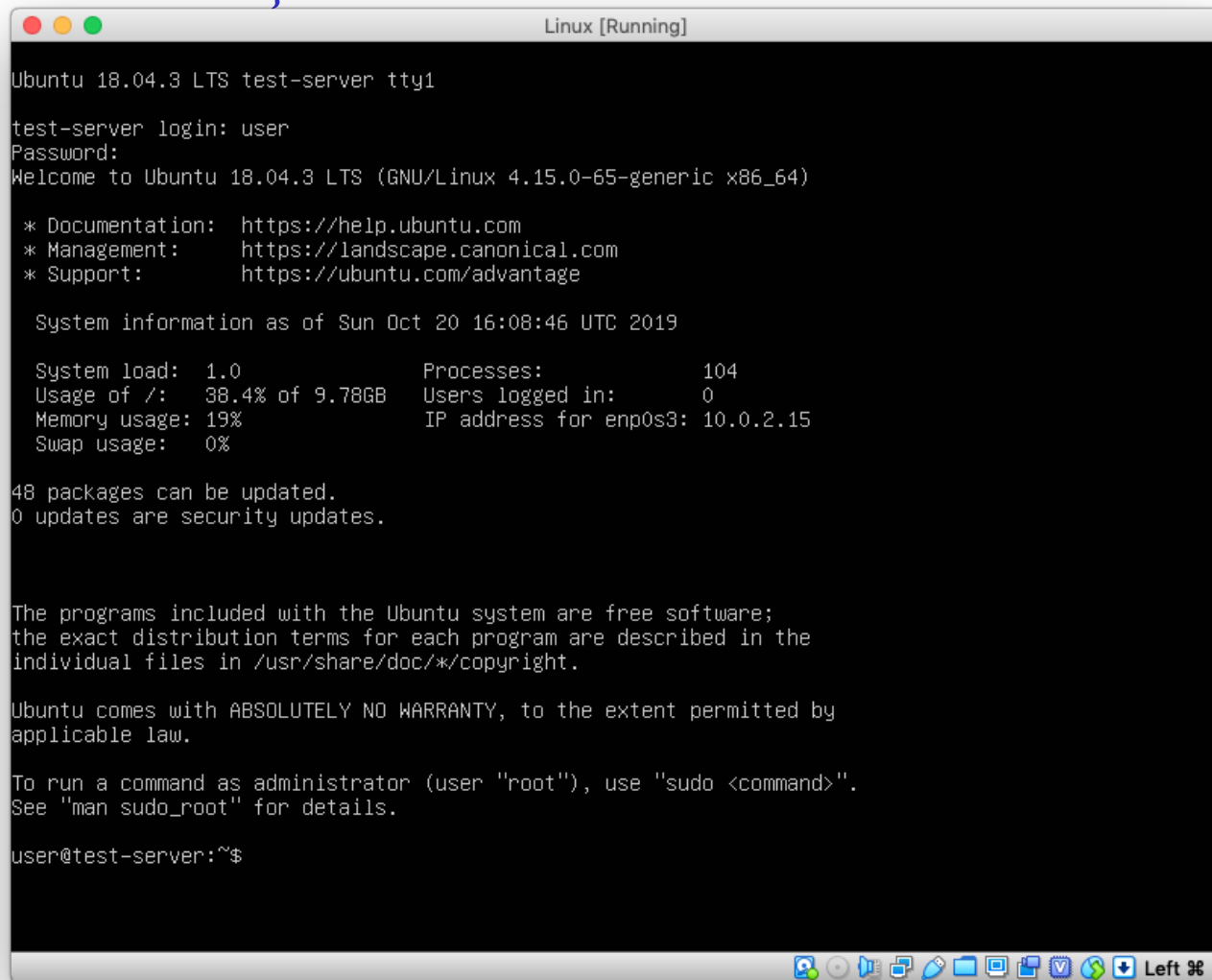
Готово. Остава само да напишем потребителското си име и парола.

# Инсталация на Linux във VM



Готово. Остава само да напишем потребителското си име и парола.

# Инсталация на Linux във ВМ



```
Linux [Running]

Ubuntu 18.04.3 LTS test-server tty1

test-server login: user
Password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-65-generic x86_64)

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

System information as of Sun Oct 20 16:08:46 UTC 2019

System load:  1.0           Processes:            104
Usage of /:   38.4% of 9.78GB Users logged in:          0
Memory usage: 19%          IP address for enp0s3: 10.0.2.15
Swap usage:   0%

48 packages can be updated.
0 updates are security updates.

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

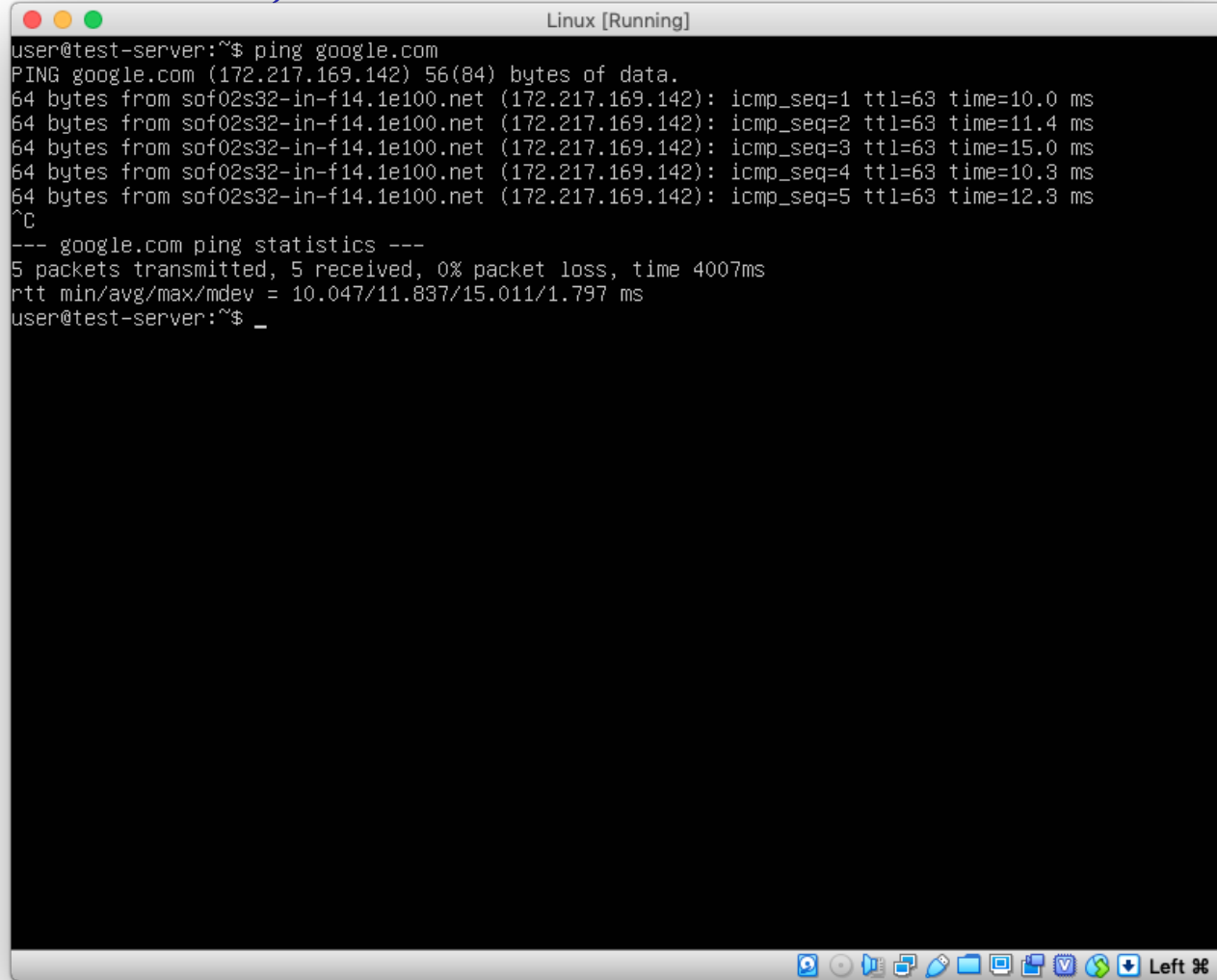
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

user@test-server:~$
```

Готови сме за работа.

# Инсталация на Linux във ВМ



```
Linux [Running]
user@test-server:~$ ping google.com
PING google.com (172.217.169.142) 56(84) bytes of data:
64 bytes from sof02s32-in-f14.1e100.net (172.217.169.142): icmp_seq=1 ttl=63 time=10.0 ms
64 bytes from sof02s32-in-f14.1e100.net (172.217.169.142): icmp_seq=2 ttl=63 time=11.4 ms
64 bytes from sof02s32-in-f14.1e100.net (172.217.169.142): icmp_seq=3 ttl=63 time=15.0 ms
64 bytes from sof02s32-in-f14.1e100.net (172.217.169.142): icmp_seq=4 ttl=63 time=10.3 ms
64 bytes from sof02s32-in-f14.1e100.net (172.217.169.142): icmp_seq=5 ttl=63 time=12.3 ms
^C
--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 10.047/11.837/15.011/1.797 ms
user@test-server:~$ _
```

Малко тестове на мрежата.  
Имаме Интернет.

# Инсталация на Linux във ВМ

```
Linux [Running]
user@test-server:~$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:1b:ac:b8:7c txqueuelen 0 (Ethernet)
    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

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fec3:2b50 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:c3:2b:50 txqueuelen 1000 (Ethernet)
    RX packets 393438 bytes 381178820 (381.1 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 55877 bytes 3916424 (3.9 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

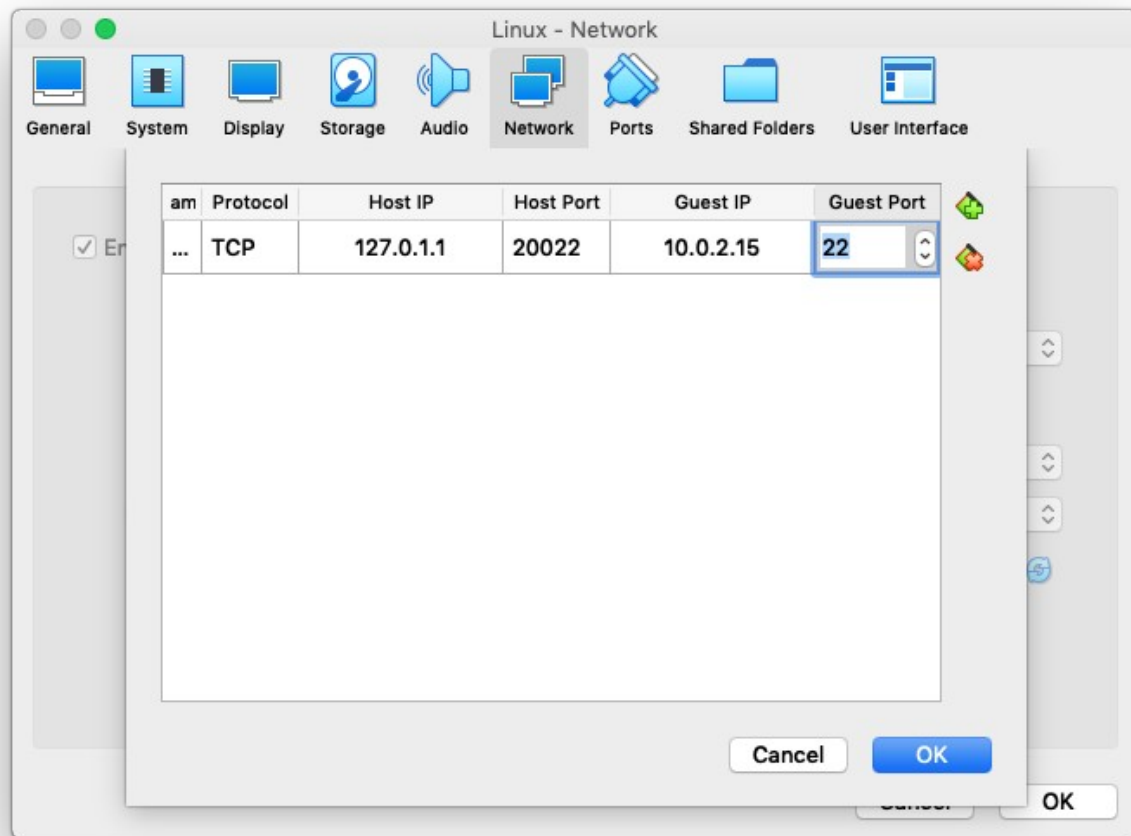
flannel.1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1450
    inet 10.1.32.0 netmask 255.255.255.255 broadcast 0.0.0.0
    inet6 fe80::9ccf:e6ff:fe7e:ab63 prefixlen 64 scopeid 0x20<link>
    ether 9e:cf:e6:7e:ab:63 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 10 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 23562 bytes 7065729 (7.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 23562 bytes 7065729 (7.0 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

user@test-server:~$ _
```

Виждаме мрежовите карти  
и техните адреси

# Инсталация на Linux във VM



Може да пренасочим порт 20022 на Host-а към порт 22 на VM за достъп през SSH.

# Инсталация на Linux във ВМ



A terminal window with a title bar containing three colored circles (red, yellow, green) and the text "apenev — -bash — 72x27". The terminal content shows the command "apenev~\$ ssh user@127.0.0.1 -p 20022" with a cursor at the end of the line.

```
apenev~$ ssh user@127.0.0.1 -p 20022
```

SSH?

# Инсталация на Linux във ВМ

```
apenev — user@test-server: ~ — ssh user@127.0.0.1 -p 20022 — 72x27
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[127.0.0.1]:20022' (ECDSA) to the list of kn
own hosts.
[user@127.0.0.1's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-65-generic x86_64)

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

System information as of Sun Oct 20 16:21:16 UTC 2019

System load:  0.29           Processes:            114
Usage of /:   42.8% of 9.78GB Users logged in:      1
Memory usage: 60%           IP address for enp0s3: 10.0.2.15
Swap usage:   1%            IP address for docker0: 172.17.0.1

48 packages can be updated.
0 updates are security updates.

Last login: Sun Oct 20 16:08:46 2019
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

user@test-server:~$
```

Да. Готови сме за разработка, инсталации, тестове и др.

Може да си инсталираме Apache, както и да пренасочим порт 80 на външен адрес към порт 80 на ВМ и вече сме в „нета“.

# Автоматизация на Създаването на VM (*Vagrant*)

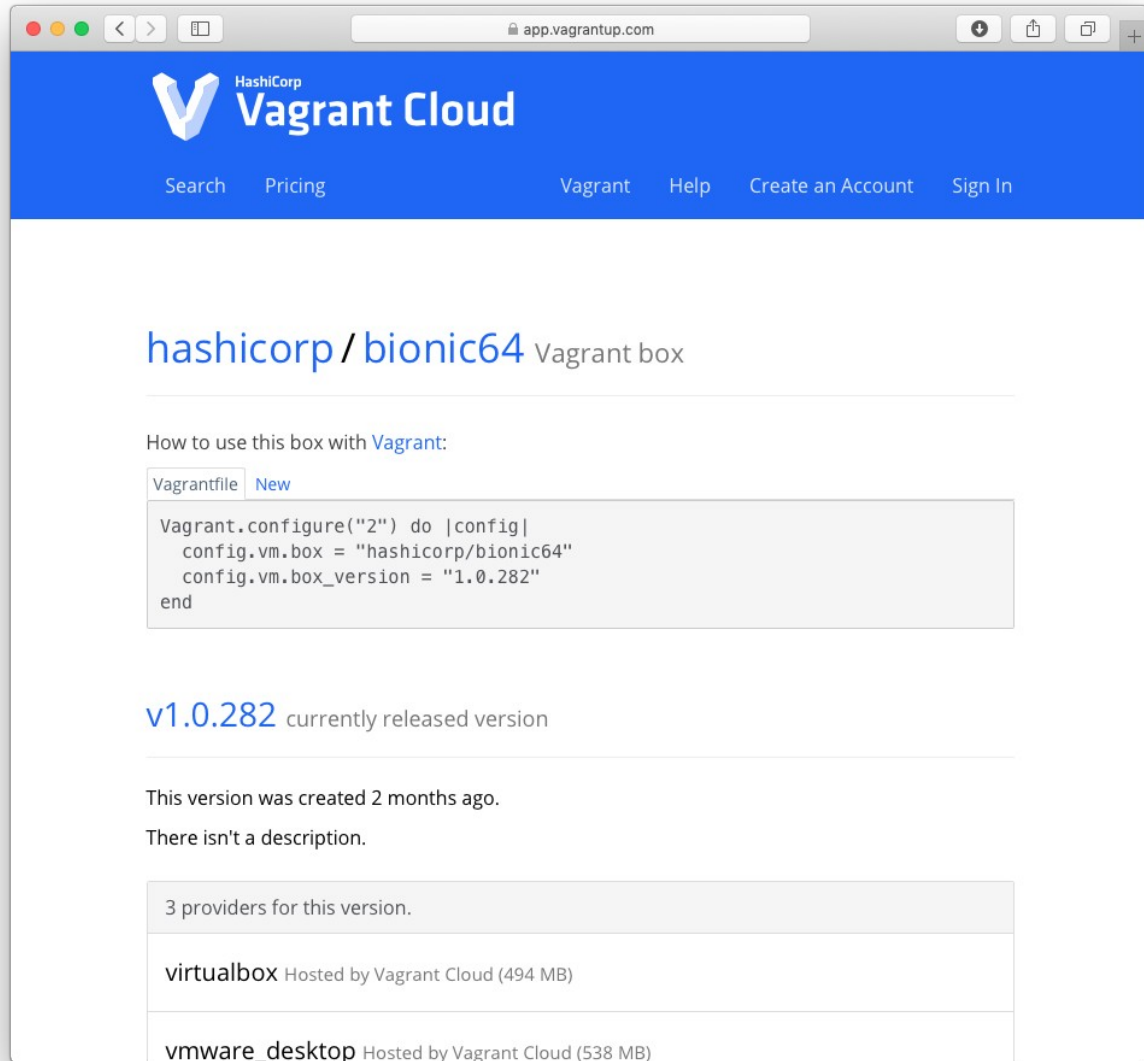


- ❖ Инструмент за изграждане и управление на среди на виртуални машини в един работен процес. С лесен за използване работен процес и с акцент върху автоматизацията. Понижава времето за настройка на средата за разработка;
- ❖ Осигурява лесна за конфигуриране, възпроизводима и преносима работна среда, изградена на базата на стандартната за индустрията технология и контролирана от единен последователен работен процес, за да помогне за увеличаване на производителността и гъвкавостта;
- ❖ Машините се предлагат за VirtualBox, VMware, AWS или друг доставчик;
- ❖ Стандартните за индустрията инструменти за осигуряване, като скриптове за shell, Chef или Puppet, могат автоматично да инсталират и конфигурират софтуер на виртуалната машина;



# Vagrant Cloud

- ❖ Намираме желаната „рецепта“ (Vagrant Box) за желаната VM;
- ❖ Пишем няколко команди в shell-а и сме готови с VM;



The screenshot shows the Vagrant Cloud website in a browser window. The URL is `app.vagrantup.com`. The page has a blue header with the Vagrant Cloud logo and navigation links: Search, Pricing, Vagrant, Help, Create an Account, and Sign In. The main content area displays the box `hashicorp / bionic64` Vagrant box. Below the box name, it says "How to use this box with Vagrant:". There are two tabs: "Vagrantfile" (selected) and "New". The "Vagrantfile" tab shows a code snippet for configuring the box. Below the code, it says "v1.0.282 currently released version". Further down, it says "This version was created 2 months ago." and "There isn't a description." At the bottom, there is a section titled "3 providers for this version." with a table listing the providers.

| 3 providers for this version. |                                  |
|-------------------------------|----------------------------------|
| virtualbox                    | Hosted by Vagrant Cloud (494 MB) |
| vmware_desktop                | Hosted by Vagrant Cloud (538 MB) |

## Инсталиране Linux на VM – „Vagrant Edition“

```
$ mkdir linux
```

```
$ cd linux
```

```
$ vagrant init hashicorp/bionic64
```

```
$ vagrant up
```

... НЯКОЛО МИНУТИ ПО-КЪСНО ...

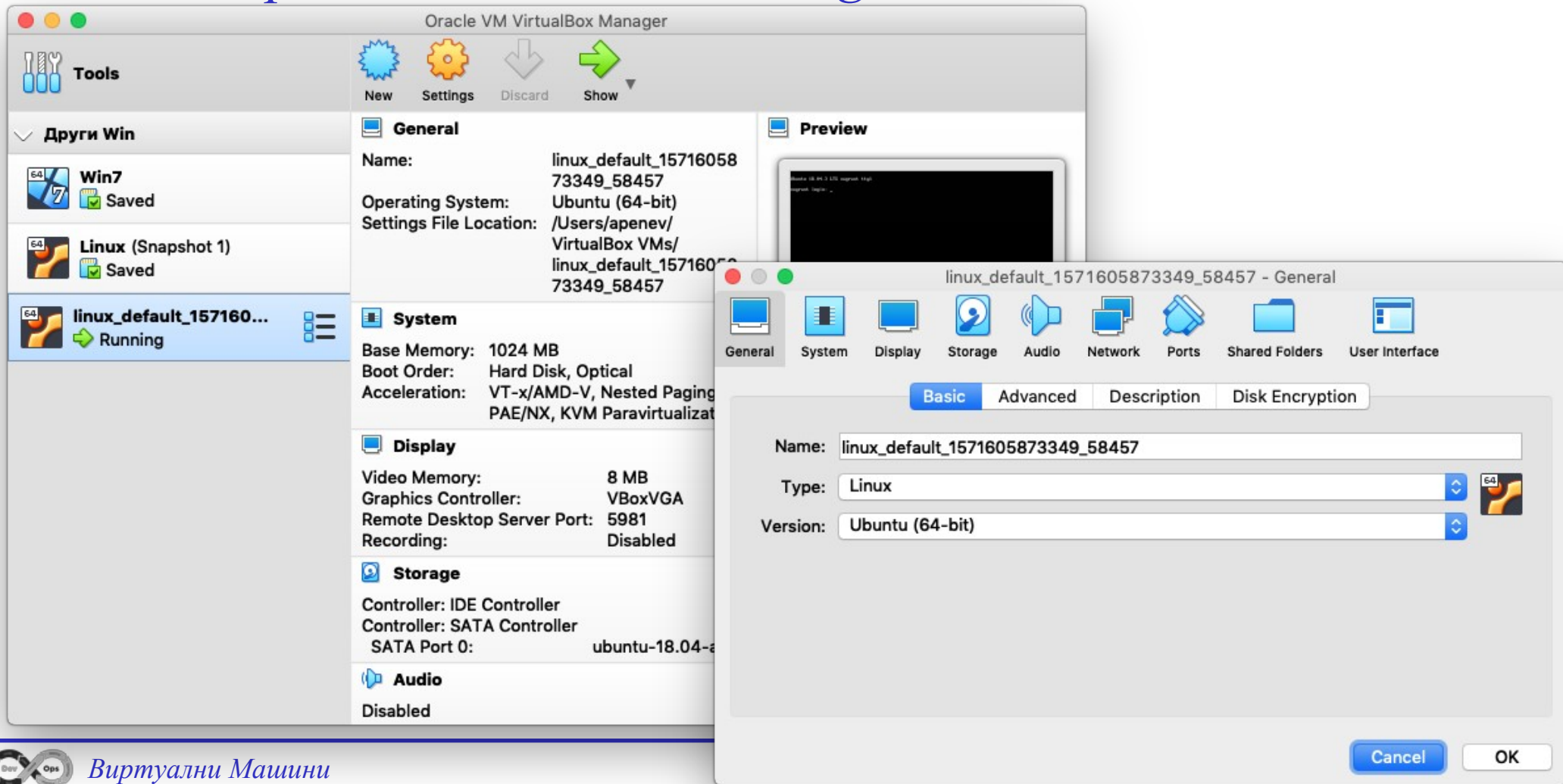
```
$ vagrant ssh
```

```
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0)
```

```
vagrant@vagrant:~$ ls
```

...

# Инсталиране Linux на VM – „Vagrant Edition“



# *Linux+Apache+Network*

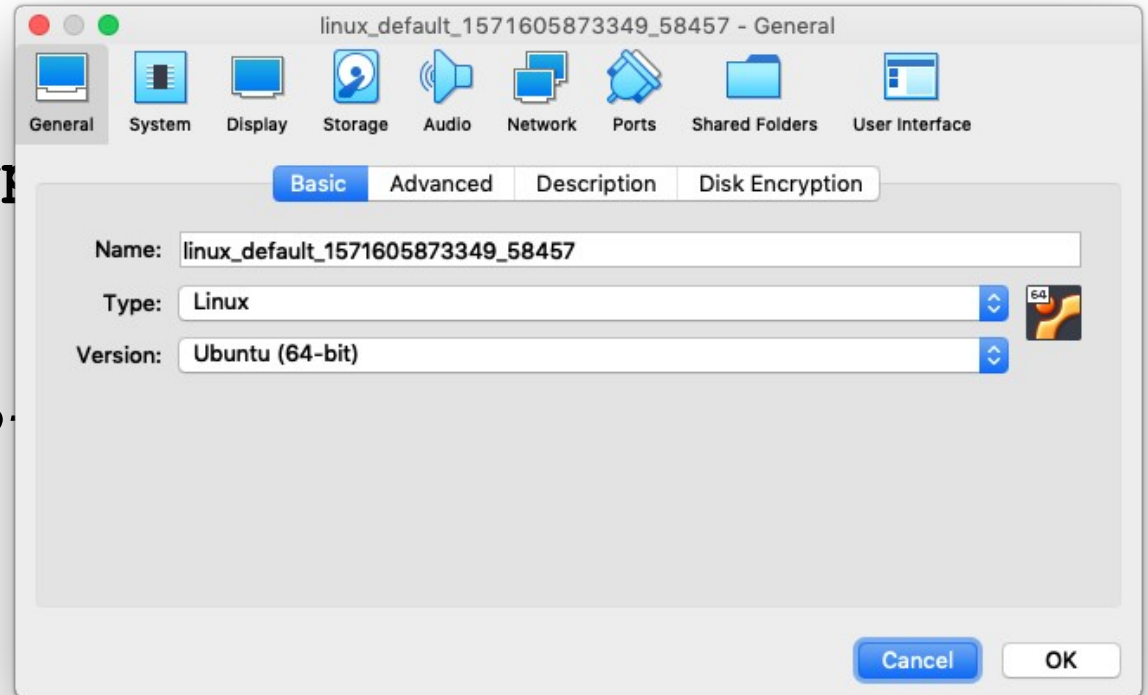
```
$ mkdir linux
$ cd linux
$ vagrant init hashicorp/bionic64
$ nano Vagrantfile
$ vagrant up
... НЯКОЛО МИНУТИ ПО-КЪСНО ...
$ vagrant ssh
$ wget 127.0.0.1:8080
```

# *Linux+Apache+Network*

```
$ mkdir linux
$ cd linux
$ vagrant init hashicorp/bionic64
$ nano Vagrantfile
$ v  config.vm.network "forwarded_port", guest: 80, host: 8080
... config.vm.provision "shell", inline: <<-SHELL
$ v      apt-get update
$ w      apt-get install -y apache2
      SHELL
```

# Linux+Apache+Network

```
$ mkdir linux
$ cd linux
$ vagrant init hashicorp
$ nano Vagrantfile
$ vagrant up
..... НЯКОЛО МИНУТИ ПО
$ vagrant ssh
$ wget 127.0.0.1:8080
```



# Linux+Apache+Network

```
$ mkdir linux
$ cd linux
$ vagrant init hashicorp
$ nano Vagrantfile
$ vagrant up
..... НЯКОЛО МИНУТИ
$ vagrant ssh
$ wget 127.0.0.1:8080
```

127.0.0.1:8080

## Apache2 Ubuntu Default Page

### ubuntu

#### It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

#### Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the

# HashiCorp Vagrant - команди



Usage: vagrant [options] <command> [<args>]

- ❖ **box** – manages boxes: installation, removal, etc.
- ❖ **cloud** – manages everything related to Vagrant Cloud
- ❖ **destroy** – stops and deletes all traces of the vagrant machine
- ❖ **global-status** – outputs status Vagrant environments for this user
- ❖ **halt** – stops the vagrant machine
- ❖ **help** – shows the help for a subcommand
- ❖ **init** – initializes a new Vagrant environment by creating a Vagrantfile
- ❖ **login**
- ❖ **package** – packages a running vagrant environment into a box
- ❖ **plugin** manages plugins: install, uninstall, update, etc.
- ❖ **port** – displays information about guest port mappings
- ❖ **powershell** – connects to machine via powershell remoting
- ❖ **provision** – provisions the vagrant machine
- ❖ **push** – deploys code in this environment to a configured destination
- ❖ **rdp** – connects to machine via RDP
- ❖ **reload** – restarts vagrant machine, loads new Vagrantfile configuration
- ❖ **resume** – resume a suspended vagrant machine
- ❖ **snapshot** – manages snapshots: saving, restoring, etc.
- ❖ **ssh** – connects to machine via SSH
- ❖ **ssh-config** – outputs OpenSSH valid configuration to connect to the machine
- ❖ **status** – outputs status of the vagrant machine
- ❖ **suspend** – suspends the machine
- ❖ **up** – starts and provisions the vagrant environment
- ❖ **upload** – upload to machine via communicator
- ❖ **validate** – validates the Vagrantfile
- ❖ **version** – prints current and latest Vagrant version
- ❖ **winrm** – executes commands on a machine via WinRM
- ❖ **winrm-config** – outputs WinRM configuration to connect to the machine

***Въпроси?***

*apenev@uni-plovdiv.bg*