

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной математики и информатики

## ОТЧЕТ

### ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

дисциплина:      *Операционные системы*

Студент: Мглинец Варвара Александровна

Группа: НПМбд-02-21

МОСКВА

2022 г.

**Цель работы:** Целью данной работы является приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.

### Ход работы:

1) Запускаем VirtualBox и создаем виртуальную машину, выбирая необходимые настройки (рисунки 1-4).

```
amglinec@dkn59 ~ $ /var/tmp
ash: /var/tmp: это каталог
amglinec@dkn59 ~ $ cd /var/tmp
amglinec@dkn59 /var/tmp $ ls
tisakhanyan
edora-Live-Desktop-1686-19-1.iso
avolchok
aabdulgalimov
root
systemd-private-f5f1ebc06f9f4914b8510a253df11d84-colord.service-rRH
7N
systemd-private-f5f1ebc06f9f4914b8510a253df11d84-systemd-logind.ser
vice-k0YAM3
systemd-private-f5f1ebc06f9f4914b8510a253df11d84-systemd-resolved,s
ervice-p5MGac
systemd-private-f5f1ebc06f9f4914b8510a253df11d84-systemd-timesyncd,
service-6FH89L
systemd-private-f5f1ebc06f9f4914b8510a253df11d84-upower.service-wfV
cx
amglinec
vduplenskikh
amglinec@dkn59 /var/tmp $ VirtualBox&
[1] 3679
```

Рисунок 1

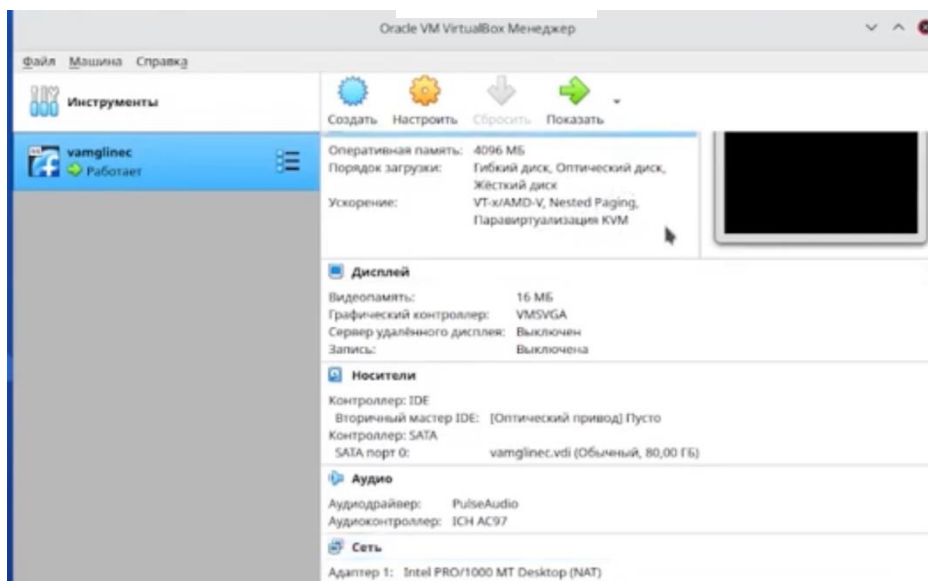


Рисунок 2

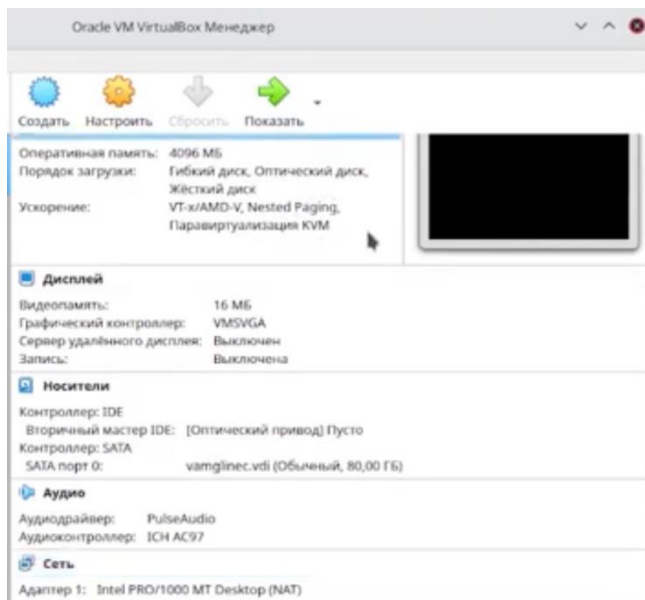


Рисунок 3

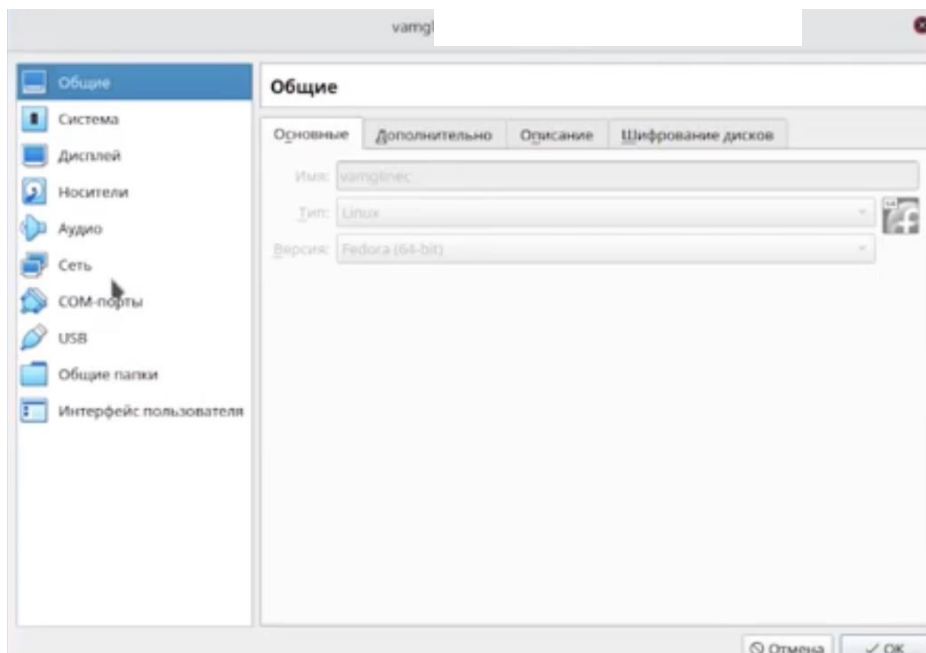


Рисунок 4

2) После создания виртуальной машины и создания пользователя, запускаем виртуальную машину заново и заходим, используя введенные ранее данные (рисунок 5).

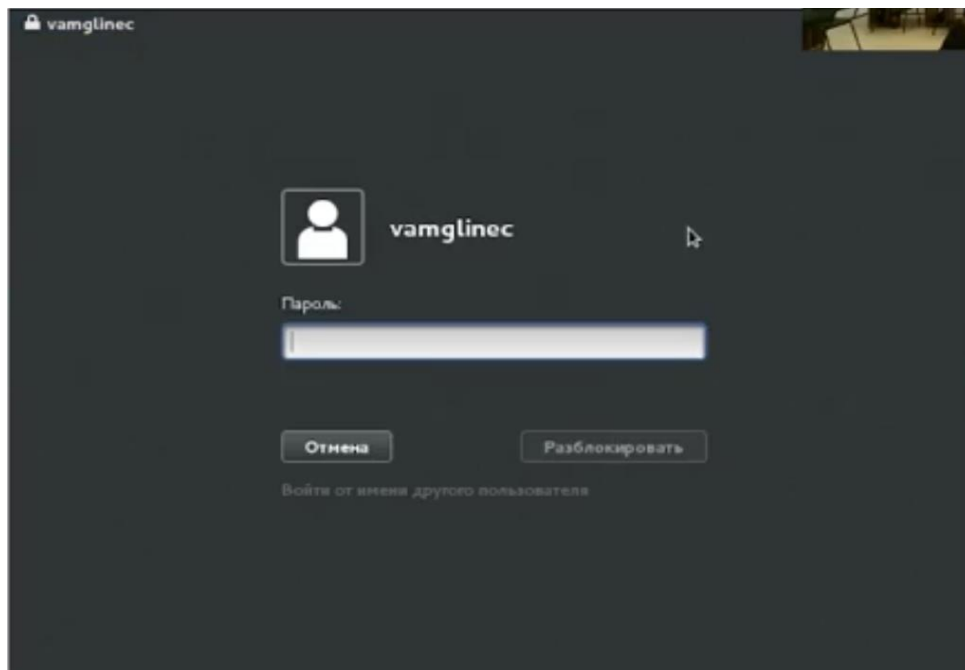


Рисунок 5

**3) Подключаем образ диска дополнений гостевой ОС, система выдаёт ошибку.**

#### **4) Домашнее задание**

В виртуальной машине запускаем терминал и получаем необходимую информацию:

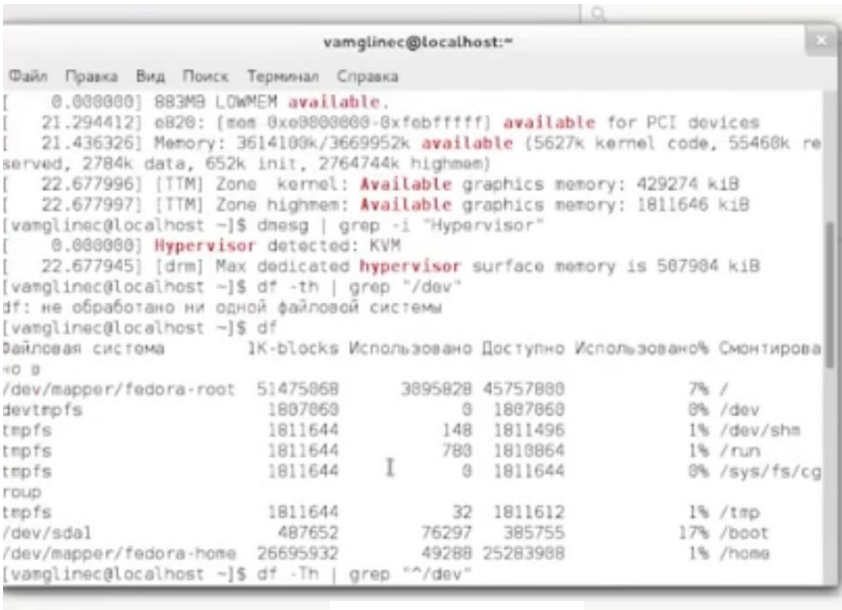
1. Версия ядра Linux (Linux version).
2. Частота процессора (Detected Mhz processor).
3. Модель процессора (CPU0).
4. Объем доступной оперативной памяти (Memory available).
5. Тип обнаруженного гипервизора (Hypervisor detected).
6. Тип файловой системы корневого раздела.
7. Последовательность монтирования файловых систем (рисунки 6-8).

```
vamglinec@localhost:~  
Файл Правка Вид Поиск Терминал Справка  
[vamglinec@localhost ~]$ dmesg | grep -i "Linux version"  
grep: i: Нет такого файла или каталога  
grep: Linux version: Нет такого файла или каталога  
[vamglinec@localhost ~]$ dmesg | grep -i "Linux "  
grep: i: Нет такого файла или каталога  
grep: Linux : Нет такого файла или каталога  
[vamglinec@localhost ~]$ dmesg | grep -i "Linux version"  
[ 0.000000] Linux version 3.9.5-301.fc19.i686 (mockbuild@bkernel02) (gcc vers  
ion 4.8.1 20130603 (Red Hat 4.8.1-1) (GCC) ) #1 SMP Tue Jun 11 20:01:50 UTC 2013  
[vamglinec@localhost ~]$ dmesg | grep -i "Mhz"  
[ 21.447551] tsc: Detected 1703.994 MHz processor  
[ 22.980856] tsc: Refined TSC clocksource calibration: 1704.246 MHz  
[ 71.551236] e1000 0000:00:03:0 eth0: (PCI:33MHz:32-bit) 08:00:27:40:a8:10  
[vamglinec@localhost ~]$ dmesg | grep -i "CPU0"  
[ 21.534609] smpboot: CPU0: Intel(R) Core(TM) i5-8400T CPU @ 1.70GHz (fam: 06,  
model: 9e, stepping: 0a)  
[ 21.629336] NMI watchdog: disabled [cpu0]: hardware events not enabled  
[ 70.949165] microcode: CPU0 sig=0x906ea, pf=0x2, revision=0x0  
[vamglinec@localhost ~]$ dmesg | grep -i "available"  
[ 0.000000] 2699MB HIGHMEM available.  
[ 0.000000] 883MB LOWMEM available.  
[ 21.294412] e820: [mem 0xe0000000-0xfebffff] available for PCI devices  
[ 21.436326] Memory: 3614100k/3669952k available (5627k kernel code, 55460k re  
served, 2784k data, 652k init, 2764744k highmem)
```

Рисунок 6

```
vamglinec@localhost:~  
Файл Правка Вид Поиск Терминал Справка  
[ 67.059419] EXT4-fs (dm-1): re-mounted. Opts: (null)  
[ 89.665282] EXT4-fs (sda1): mounted filesystem with ordered data mode. Opts:  
(null)  
[ 93.488497] EXT4-fs (dm-2): mounted filesystem with ordered data mode. Opts:  
(null)  
[vamglinec@localhost ~]$ sudo dmesg | grep -i "Mount"  
  
We trust you have received the usual lecture from the local System  
Administrator. It usually boils down to these three things:  
  
#1) Respect the privacy of others.  
#2) Think before you type.  
#3) With great power comes great responsibility.  
  
[sudo] password for vamglinec:  
[ 21.447654] Mount-cache hash table entries: 512  
[ 39.496851] EXT4-fs (dm-1): mounted filesystem with ordered data mode. Opts:  
(null)  
[ 67.059419] EXT4-fs (dm-1): re-mounted. Opts: (null)  
[ 89.665282] EXT4-fs (sda1): mounted filesystem with ordered data mode. Opts:  
(null)  
[ 93.488497] EXT4-fs (dm-2): mounted filesystem with ordered data mode. Opts:  
(null)  
[vamglinec@localhost ~]$
```

Рисунок 7

A screenshot of a terminal window titled 'vamglinec@localhost:~'. The terminal shows various system status messages, including memory availability, PCI device status, and graphics memory. It also displays the output of the 'df' command, showing disk usage for various filesystems. The 'df' output is as follows:

Файловая система	1K-blocks	Использовано	Доступно	Использовано%	Смонтировано в
/dev/mapper/fedora-root	51475068	3895828	45757880	7%	/
devtmpfs	1887068	0	1887068	0%	/dev
tmpfs	1811644	148	1811496	1%	/dev/shm
tmpfs	1811644	780	1810864	1%	/run
tmpfs	1811644	0	1811644	0%	/sys/fs/cg
group					
tmpfs	1811644	32	1811612	1%	/tmp
/dev/sda1	487652	76297	385755	17%	/boot
/dev/mapper/fedora-home	26695932	49288	25283988	1%	/home

Рисунок 8

## 5) Контрольные вопросы

1. Учетная запись пользователя – это необходимая для системы информация о пользователе, хранящаяся в специальных файлах. Информация используется Linux для аутентификации пользователя и назначения ему прав доступа. Аутентификация – системная процедура, позволяющая Linux определить, какой именно пользователь осуществляет вход. Вся информация о пользователе обычно хранится в файлах /etc/passwd и /etc/group.

Учётная запись пользователя содержит:

2. Команды терминала:

- Для получения справки по команде:
- Для перемещения по файловой системе:

(Рисунок 34)

- Имя пользователя (user name)
- Идентификационный номер пользователя (UID)
- Идентификационный номер группы (GID).
- Пароль (password)
- Полное имя (full name)
- Домашний каталог (home directory)
- Начальную оболочку (login shell)

man [команда]. Например, команда «man ls» выведет справку о команде «ls».

cd [путь]. Например, команда «cd newdir» осуществляет переход в каталог newdir

- Для просмотра содержимого каталога:

ls [опции] [путь]. Например, команда «ls -a ~/newdir» отобразит имена скрытых файлов в каталоге newdir

- Для определения объёма каталога:

- Для создания / удаления каталогов / файлов:
- Для задания определённых прав на файл / каталог:
- Для просмотра истории команд:  
`du [опция] [путь]`. Например, команда «`du -k ~/newdir`» выведет размер каталога `newdir` в килобайтах  
`mkdir [опции] [путь] / rmdir [опции] [путь] / rm [опции] [путь]`. Например, команда «`mkdir -p ~/newdir1/newdir2`» создаст иерархическую цепочку подкаталогов, создав каталоги `newdir1` и `newdir2`; команда «`rmdir -v ~/newdir`» удалит каталог `newdir`; команда «`rm -r ~/newdir`» так же удалит каталог `newdir`  
`chmod [опции] [путь]`. Например, команда «`chmod g+r ~/text.txt`» даст группе право на чтение файла `text.txt`  
`history [опции]`. Например, команда «`history 5`» покажет список последних 5 команд

3. Файловая система имеет два значения: с одной стороны – это архитектура хранения битов на жестком диске, с другой – это организация каталогов в соответствии с идеологией Unix.

Файловая система (англ. «file system») – это архитектура хранения данных в системе, хранение данных в оперативной памяти и доступа к конфигурации ядра. Файловая система устанавливает физическую и логическую структуру файлов, правила их создания и управления ими. В физическом смысле файловая система Linux представляет собой пространство раздела диска, разбитое на блоки фиксированного размера. Их размер кратен размеру сектора: 1024, 2048, 4096 или 8120 байт. Существует несколько типов файловых систем:

- XFS – начало разработки 1993 год, фирма Silicon Graphics, в мае 2000 года предстала в GNU GPL, для пользователей большинства Linux систем стала доступна в 2001-2002 гг. Отличительная черта системы – прекрасная поддержка больших файлов и файловых томов, 8 эксбибайт (8\*260 байт) для 64-х битных систем.
- ReiserFS (Reiser3) – одна из первых журналируемых файловых систем под Linux, разработана Namesys, доступна с 2001 г. Максимальный объём тома для этой системы равен 16 тебибайт (16\*240 байт).
- JFS (Journaled File System) – файловая система, детище IBM, явившееся миру в далёком 1990 году для ОС AIX (Advanced Interactive eXecutive). В виде первого стабильного релиза, для пользователей Linux, система стала доступна в 2001 году. Из плюсов системы – хорошая масштабируемость. Из минусов – не особо активная поддержка на протяжении всего жизненного цикла. Максимальный размер тома 32 пэбибайта (32\*250 байт).
- ext (extended filesystem) – появилась в апреле 1992 года, это была первая файловая система, изготовленная специально под нужды Linux ОС. Разработана Remy Card с целью преодолеть ограничения файловой системы Minix.

- ext2 (second extended file system) – была разработана Remy Card в 1993 году. Не журналируемая файловая система, это был основной её недостаток, который исправит ext3.

- ext3 (third extended filesystem) – по сути расширение исконной для Linux ext2, способное к журналированию. Разработана Стивеном

Твиди (Stephen Tweedie) в 1999 году, включена в основное ядро Linux в ноябре 2001 года. На фоне других своих сослуживцев обладает более скромным размером пространства, до 4 тебибайт (4\*240 байт) для 32-х разрядных систем. На данный момент является наиболее стабильной и поддерживаемой файловой системой в среде Linux.

- Reiser4 – первая попытка создать файловую систему нового поколения для Linux. Впервые представленная в 2004 году, система включает в себя такие передовые технологии как транзакции,

задержка выделения пространства, а так же встроенная возможность кодирования и сжатия данных. Ханс Рейзер (Hans Reiser) – главный разработчик системы.

- ext4 – попытка создать 64-х битную ext3 способную поддерживать больший размер файловой системы (1 эксбибайт). Позже добавились возможности – непрерывные области дискового пространства, задержка выделения пространства, онлайн дефрагментация и прочие. Обеспечивается прямая совместимость с системой ext3 и ограниченная обратная совместимость при недоступной способности к непрерывным областям дискового пространства.

- Btrfs (B-tree FS или Butter FS) – проект изначально начатый компанией Oracle, впоследствии поддержанный большинством Linux систем.

Ключевыми особенностями данной файловой системы являются технологии: copy-on-write, позволяющая сделать снимки областей диска (снапшоты), которые могут пригодиться для последующего восстановления; контроль за целостностью данных и метаданных (с повышенной гарантией целостности); сжатие данных; оптимизированный режим для накопителей SSD (задаётся при монтировании) и прочие. Немаловажным фактором является возможность перехода с ext3 на Btrfs. С августа 2008 года данная система выпускается под GNU GPL.

- Tux2 – известная, но так и не анонсированная публично файловая система. Создатель Дэниэл Филипс (Daniel Phillips). Система базируется на алгоритме «Фазового Древа», который как и журналирование защищает файловую систему от сбоев.

Организована как надстройка на ext2.

- Tux3 – система создана на основе FUSE (Filesystem in Userspace), специального модуля для создания файловых систем на Unix платформах.

Данный проект ставит перед собой цель избавиться от привычного журналирования, взамен предлагая версионное восстановление (состояние в определённый промежуток времени). Преимуществом используемой в данном случае версионной системы, является способ описания изменений, где для каждого



файла создаётся изменённая копия, а не переписывается текущая версия.

- Xiafs – задумка и разработка данной файловой системы принадлежат Frank Xia, основана на файловой системе MINIX. В настоящее время считается устаревшей и практически не используется. Наряду с ext2 разрабатывалась, как замена системе ext. В декабре 1993 года система была добавлена в стандартное ядро Linux. И хотя система обладала большей стабильностью и занимала меньше дискового пространства под контрольные структуры – она оказалась слабее ext2, ведущую роль сыграли ограничения максимальных размеров файла и раздела, а так же способность к дальнейшему расширению.

- ZFS (Zettabyte File System) – изначально созданная в Sun Microsystems файловая система, для небезызвестной операционной системы Solaris в 2005 году. Отличительные особенности – отсутствие фрагментации данных как таковой, возможности по управлению снапшотами (snapshots), пулами хранения (storage pools), варьируемый размер блоков, 64-х разрядный механизм контрольных сумм, а так же способность адресовать 128 бит информации. В Linux системах может использоваться посредством FUSE.

4. Команда «findmnt» или «findmnt --all» будет отображать все подмонтированные файловые системы или искать файловую систему.

5. Основные сигналы (каждый сигнал имеет свой номер), которые используются для завершения процесса:

- SIGINT – самый безобидный сигнал завершения, означает Interrupt. Он отправляется процессу, запущенному из терминала с помощью сочетания клавиш Ctrl+C. Процесс правильно завершает все свои действия и возвращает управление;

- SIGQUIT – это еще один сигнал, который отправляется с помощью сочетания клавиш, программе, запущенной в терминале. Он сообщает ей что нужно завершиться и программа может выполнить корректное завершение или проигнорировать сигнал. В отличие от

предыдущего, она генерирует дампы памяти. Сочетание клавиш Ctrl+Q;

- SIGHUP – сообщает процессу, что соединение с управляющим терминалом разорвано, отправляется, в основном, системой при разрыве соединения с интернетом;

- SIGTERM – немедленно завершает процесс, но обрабатывается программой, поэтому позволяет ей завершить дочерние процессы и освободить все ресурсы;

- SIGKILL – тоже немедленно завершает процесс, но, в отличие от предыдущего варианта, он не передается самому процессу, а обрабатывается ядром. Поэтому ресурсы и дочерние процессы остаются запущенными.

Также для передачи сигналов процессам в Linux используется утилита kill, её синтаксис: kill [-сигнал] [pid\_процесса] (PID – уникальный идентификатор процесса). Сигнал представляет собой один из выше перечисленных сигналов для завершения процесса.

Перед тем, как выполнить остановку процесса, нужно определить его PID. Для этого используют команды ps и grep. Команда ps предназначена для вывода списка активных процессов в системе и информации о них. Команда grep запускается одновременно с ps (в канале) и будет выполнять поиск по результатам команды ps.

Утилита pkill – это оболочка для kill, она ведет себя точно так же, и имеет тот же синтаксис, только в качестве идентификатора процесса ей нужно передать его имя.

killall работает аналогично двум предыдущим утилитам. Она тоже принимает имя процесса в качестве параметра и ищет его PID в директории /proc. Но эта утилита обнаружит все процессы с таким именем и завершит их.

**Вывод:** в ходе выполнения данной работы я приобрела практические навыки установки операционной системы на виртуальную машину и настройки минимально необходимых для дальнейшей работы сервисов.