

**Министерство науки и высшего образования Российской Федерации**  
**федеральное государственное автономное образовательное учреждение высшего образования**  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**  
**(Университет ИТМО)**

**Факультет программной инженерии и компьютерной техники**

**Отчет по лабораторной работе №4**  
**по дисциплине «Системное администрирование»**

Выполнила:  
студент гр. № Р33212  
Ян Цзяфэн

Санкт-Петербург

2021

# Установка приложений

Перед началом проверить сетевое соединение.

## 1. Обновление установленных пакетов с помощью apt через терминал

**АРТ** (Advanced Package Tool) — инструмент командной строки для взаимодействия с системой управления пакетами. Вы можете использовать его для поиска и установки новых пакетов, обновления пакетов, удаления пакетов и т. д.

**Вся работа выполняется в Терминале**

а. Обновления базы данных пакетов

**\$sudo apt update**

Что означают типы строк: **Hit**(Сущ), **Get** (Пол), **Ign**(Игн)?

```
yang@yang-VirtualBox:~$ sudo apt update
[sudo] password for yang:
Игн http://ru.archive.ubuntu.com trusty InRelease
В кэше http://ru.archive.ubuntu.com trusty-updates InRelease
В кэше http://ru.archive.ubuntu.com trusty-backports InRelease
Игн http://extras.ubuntu.com trusty InRelease
В кэше http://ru.archive.ubuntu.com trusty Release.gpg
В кэше http://ru.archive.ubuntu.com trusty-updates/main Sources
Получено:1 http://extras.ubuntu.com trusty Release.gpg [72 B]
В кэше http://security.ubuntu.com trusty-security InRelease
В кэше http://ru.archive.ubuntu.com trusty-updates/restricted Sources
В кэше http://ru.archive.ubuntu.com trusty-updates/universe Sources
В кэше http://extras.ubuntu.com trusty Release
В кэше http://ru.archive.ubuntu.com trusty-updates/multiverse Sources
В кэше http://security.ubuntu.com trusty-security/main Sources
В кэше http://extras.ubuntu.com trusty/main Sources
В кэше http://ru.archive.ubuntu.com trusty-updates/main amd64 Packages
В кэше http://ru.archive.ubuntu.com trusty-updates/restricted amd64 Packages
В кэше http://extras.ubuntu.com trusty/main amd64 Packages
В кэше http://ru.archive.ubuntu.com trusty-updates/universe amd64 Packages
В кэше http://security.ubuntu.com trusty-security/restricted Sources
В кэше http://ru.archive.ubuntu.com trusty-updates/multiverse amd64 Packages
В кэше http://extras.ubuntu.com trusty/main i386 Packages
В кэше http://ru.archive.ubuntu.com trusty-updates/main i386 Packages
```

**Hit** означает, что apt проверяет метку времени в списке пакетов, и эти совпадения не изменились.

**Get** означает проверить отметку времени в списке пакетов, есть изменения и будут загружены.

**Ign** означает проверить отметку времени в списке пакетов, есть изменения и будут загружены.

б. Обновление установленных пакетов

**\$sudo apt upgrade**

```

yang@yang-VirtualBox:~$ sudo apt upgrade
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Расчёт обновлений...Готово
НОВЫЕ пакеты, которые будут установлены:
  libyaml-0-2 linux-base linux-headers-4.4.0-148
  linux-headers-4.4.0-148-generic linux-image-4.4.0-148-generic
  linux-modules-4.4.0-148-generic linux-modules-extra-4.4.0-148-generic
  python3-yaml
Пакеты, которые будут обновлены:
  apt apt-transport-https apt-utils busybox-initramfs busybox-static firefox
  ghostscript ghostscript-x gir1.2-gudev-1.0 grub-common grub-pc grub-pc-bin
  grub2-common intel-microcode libapt-inst1.5 libapt-pkg4.12 libc-bin
  libc-dev-bin libc6 libc6-dbg libc6-dev libgs9 libgs9-common libgudev-1.0-0
  libpolkit-agent-1-0 libpolkit-backend-1-0 libpolkit-gobject-1-0
  libpoppler-glib8 libpoppler44 libsmbclient libtiff5 libudev1 libwbclient0
  libxslt1.1 linux-generic-lts-xenial linux-headers-generic-lts-xenial
  linux-image-generic-lts-xenial linux-libc-dev multiarch-support policykit-1
  poppler-utils python-samba python3-distupgrade samba-common samba-common-bin
  samba-libs smbclient tzdata ubuntu-advantage-tools
  ubuntu-release-upgrader-core ubuntu-release-upgrader-gtk udev
  update-notifier update-notifier-common wget wpasupplicant
обновлено 56, установлено 8 новых пакетов, для удаления отмечено 0 пакетов, и 0
пакетов не обновлено.
Необходимо скачать 152 МБ архивов.
После данной операции, объём занятого дискового пространства возрастёт на 309 МБ
.

```

с. Вариант «быстрого» обновления БД и установленных пакетов

**\$sudo apt update && sudo apt upgrade -y**

```

yang@yang-VirtualBox:~$ sudo apt update && sudo apt upgrade -y
Игн http://ru.archive.ubuntu.com trusty InRelease
В кэше http://ru.archive.ubuntu.com trusty-updates InRelease
В кэше http://ru.archive.ubuntu.com trusty-backports InRelease
В кэше http://ru.archive.ubuntu.com trusty Release.gpg
В кэше http://ru.archive.ubuntu.com trusty Release
Игн http://extras.ubuntu.com trusty InRelease
В кэше http://ru.archive.ubuntu.com trusty-updates/main Sources
В кэше http://ru.archive.ubuntu.com trusty-updates/restricted Sources
В кэше http://extras.ubuntu.com trusty Release.gpg
В кэше http://ru.archive.ubuntu.com trusty-updates/universe Sources
В кэше http://ru.archive.ubuntu.com trusty-updates/multiverse Sources
В кэше http://ru.archive.ubuntu.com trusty-updates/main amd64 Packages
В кэше http://extras.ubuntu.com trusty Release
В кэше http://ru.archive.ubuntu.com trusty-updates/restricted amd64 Packages
В кэше http://security.ubuntu.com trusty-security InRelease
В кэше http://ru.archive.ubuntu.com trusty-updates/universe amd64 Packages
В кэше http://ru.archive.ubuntu.com trusty-updates/multiverse amd64 Packages
В кэше http://ru.archive.ubuntu.com trusty-updates/main i386 Packages
В кэше http://ru.archive.ubuntu.com trusty-updates/restricted i386 Packages
В кэше http://ru.archive.ubuntu.com trusty-updates/universe i386 Packages
В кэше http://ru.archive.ubuntu.com trusty-updates/multiverse i386 Packages
В кэше http://ru.archive.ubuntu.com trusty-updates/main Translation-en
В кэше http://ru.archive.ubuntu.com trusty-updates/multiverse Translation-en
В кэше http://ru.archive.ubuntu.com trusty-updates/restricted Translation-en
В кэше http://extras.ubuntu.com trusty/main Sources
В кэше http://ru.archive.ubuntu.com trusty-updates/universe Translation-en
В кэше http://ru.archive.ubuntu.com trusty-backports/main Sources
В кэше http://ru.archive.ubuntu.com trusty-backports/restricted Sources

```

d. Что выполняет команда apt full-upgrade?

full-upgrde включает в себя upgrade. apt full-upgrade выполняет функцию обновления, но удаляет установленные в данный момент пакеты, если это необходимо для обновления системы в целом.

```

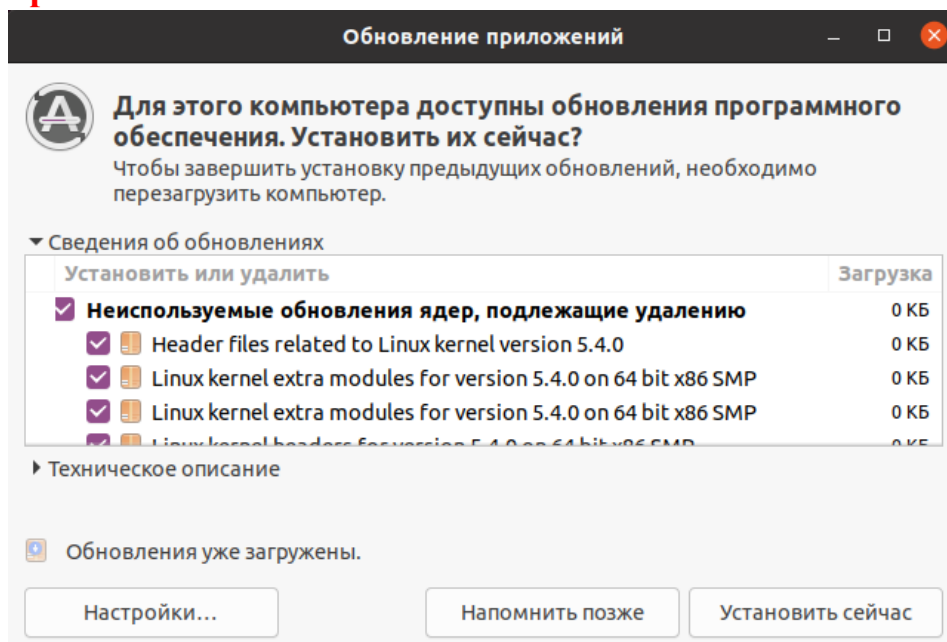
yang@yang-VirtualBox:~$ sudo apt full-upgrade
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Расчёт обновлений...Готово
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов
, и 0 пакетов не обновлено.

169 additional updates are available with UA Infrastructure ESM.
To see these additional updates run: apt list --upgradable
See https://ubuntu.com/advantage or run: sudo ua status

```

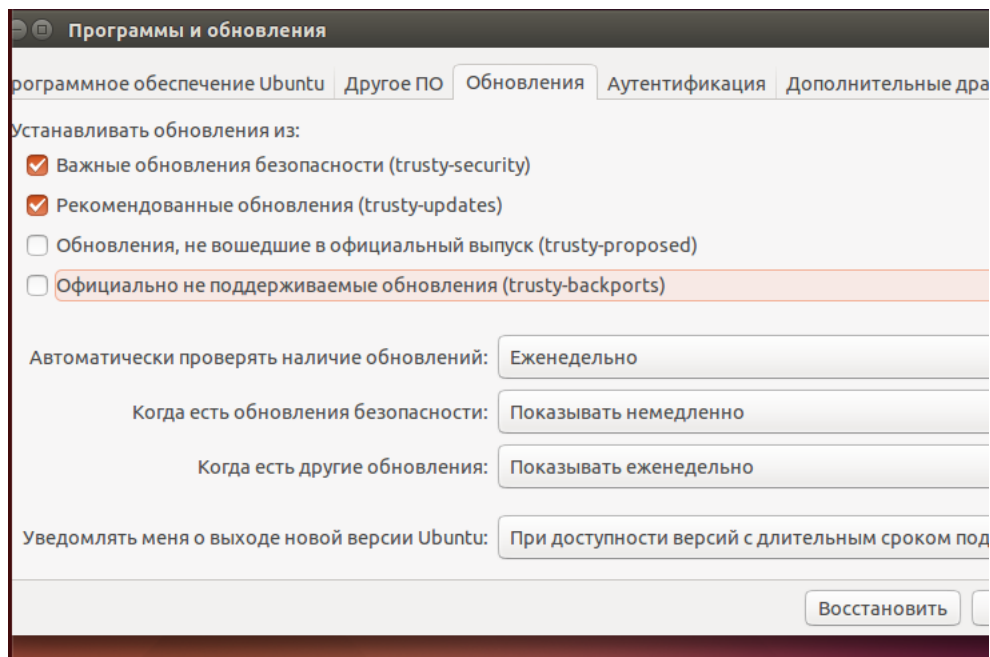
## 2. Обновление ПО через приложения

В графической оболочке запустить приложение — **Обновление приложений**



### 1. Выполнить настройки обновлений:

- Автоматически проверять обновления раз в неделю
- Уведомлять о выходе новой версии Ubuntu с длительным сроком поддержки
- Отменить обновление программ, ограниченных патентами или законами



### 3. Установка программного обеспечения

1. Установка ПО через терминал

2. Установить новый пакет:

**\$ sudo apt install имя-пакета**

3. Установить несколько пакетов:

**\$ sudo apt install имя\_пакета-1 имя\_пакета-2 имя\_пакета-3**

4. Установить несколько пакетов:

**\$ sudo apt install имя\_пакета-1 имя\_пакета-2 имя\_пакета-3**

5. Удалить пакет:

**\$ sudo apt remove имя-пакета**

6. Удалить пакет и файлы конфигурации:

**\$ sudo apt purge имя-пакета**

7. Удалить пакет по части имени пакета:

**\$ sudo apt remove имя-пак\***

8. Удаления пакетов, которые были установлены автоматически для удовлетворения зависимостей других пакетов, и теперь больше не нужны, так как зависимости изменены или пакет, нуждающийся в них, был удален.

**\$ sudo apt autoremove**

9. Удалить пакет **LibreOffice**

```
yang@yang-VirtualBox:~$ sudo apt remove LibreOffice
[sudo] password for yang:
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Package 'libreoffice' is not installed, so not removed
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
```

10. Установить ПО **Scilab**

```

yang@yang-VirtualBox:~$ sudo apt install Scilab
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  antlr bwidget ca-certificates-java default-jre default-jre-headless
  docbook-xml docbook-xsl fonts-dejavu-extra fop java-common java-wrappers
  javahelp2 junit libamd2.3.1 libantlr-java libapache-pom-java libarpack2
  libatk-wrapper-java libatk-wrapper-java-jni libavalon-framework-java
  libavcodec54 libavformat54 libavutil52 libbatik-java libblas3 libbsf-java
  libcamd2.3.1 libccolamd2.8.0 libcholmod2.1.2 libcommons-io-java
  libcommons-logging-java libcommons-parent-java libdc1394-22 libdom4j-java
  libecj-java libfftw3-double3 libflexdock-java libfop-java
  libfreehep-export-java libfreehep-graphics2d-java
  libfreehep-graphicsio-emf-java libfreehep-graphicsio-java
  libfreehep-graphicsio-tests-java libfreehep-io-java libfreehep-swing-java

```

11. Проверить конфигурацию системы (в случае необходимости установить необходимое ПО):

a. сетевого(вых) интерфейсов – `ifconfig -a` и `ifconfig -s`

```

yang@yang-VirtualBox:~$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:00:27:93:2f:18
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.0
          inet6 addr: fe80::a00:27ff:fe93:2f18/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:103877 errors:0 dropped:0 overruns:0 frame:0
          TX packets:19933 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:150001539 (150.0 MB)  TX bytes:1330489 (1.3 MB)

lo        Link encap:Локальная петля (Loopback)
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1271 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1271 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:124346 (124.3 KB)  TX bytes:124346 (124.3 KB)

```

```

yang@yang-VirtualBox:~$ ifconfig -s
Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0   1500  0    103877      0      0  0      19933      0      0      0  B
MRU
lo     65536  0      1271      0      0  0      1271      0      0      0  L
RU

```

b. таблицу маршрутизации – `route`

```

yang@yang-VirtualBox:~$ route
Таблица маршрутизации ядра протокола IP
Destination Gateway Genmask Flags Metric Ref Use Iface
default    10.0.2.2   0.0.0.0   UG    0      0      0 eth0
10.0.2.0   *          255.255.255.0   U    1      0      0 eth0

```

c. сетевую статистику – `netstat -a`. Сократите вывод команды, выведите информацию только по tcp и udp пакетам, как в примере:

`netstat -a`

```

unix  2      [ ]          DGRAM        0
unix  3      [ ]          STREAM       CONNECTED    9865
unix  3      [ ]          STREAM       CONNECTED    15031
unix  3      [ ]          STREAM       CONNECTED    14304    @/tmp/dbus-draMvzdfF9
unix  3      [ ]          STREAM       CONNECTED    24303    @/tmp/dbus-draMvzdfF9
unix  3      [ ]          STREAM       CONNECTED    913689   @/tmp/dbus-draMvzdfF9
unix  3      [ ]          STREAM       CONNECTED    17949
unix  3      [ ]          STREAM       CONNECTED    15733    @/tmp/dbus-ueZHML8gbL
unix  3      [ ]          STREAM       CONNECTED    14467
unix  3      [ ]          STREAM       CONNECTED    11389    /var/run/dbus/system_
bus_socket
unix  2      [ ]          STREAM       CONNECTED    17156    @/tmp/dbus-hrWqfaVd
unix  3      [ ]          STREAM       CONNECTED    15033
unix  3      [ ]          STREAM       CONNECTED    24299
unix  3      [ ]          STREAM       CONNECTED    14311    @/tmp/dbus-draMvzdfF9
unix  3      [ ]          STREAM       CONNECTED    14239    @/tmp/.X11-unix/X0
unix  3      [ ]          STREAM       CONNECTED    14121
unix  3      [ ]          STREAM       CONNECTED    9780
unix  3      [ ]          STREAM       CONNECTED    17953
unix  3      [ ]          STREAM       CONNECTED    15460    @/tmp/dbus-draMvzdfF9
unix  3      [ ]          STREAM       CONNECTED    9103
unix  2      [ ]          STREAM       CONNECTED    918753

```



```

tcp      0      0 localhost:domain  0.0.0.0:*          LISTEN
tcp      0      0 localhost:ipp     0.0.0.0:*          LISTEN
tcp      0      0 alexander-VBox:56090  ec2-52-10-100-61.:https ESTABLISHED
udp      0      0 localhost:domain  0.0.0.0:*          LISTEN
udp      0      0 alexander-VBox:bootpc _gateway:bootpc    ESTABLISHED
udp      0      0 0.0.0.0:631       0.0.0.0:*          LISTEN
udp      0      0 0.0.0.0:58206     0.0.0.0:*          LISTEN
udp      0      0 0.0.0.0:mdns      0.0.0.0:*          LISTEN

```

```

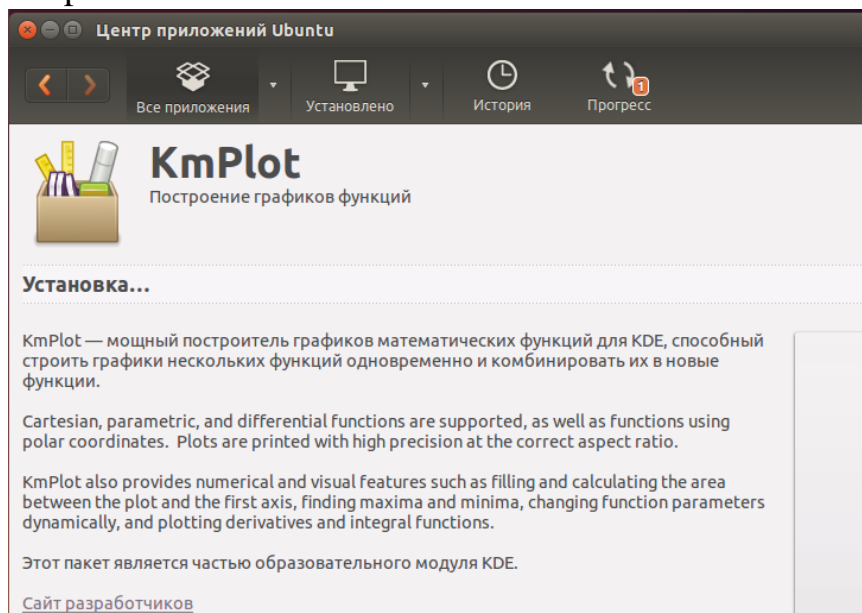
yang@yang-VirtualBox:~$ netstat -a -t -u
Активные соединения с интернетом (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp      0      0 yang-VirtualBox:domain *:.*              LISTEN
tcp      0      0 localhost:ipp     *:.*              LISTEN
tcp6     0      0 ip6-localhost:ipp [::]:.*          LISTEN
tcp6     1      0 ip6-localhost:41402 ip6-localhost:ipp CLOSE_WAIT
udp      0      0 *:ipp             *:.*              LISTEN
udp      0      0 *:52339           *:.*              LISTEN
udp      0      0 *:mdns            *:.*              LISTEN
udp      0      0 *:19850           *:.*              LISTEN
udp      0      0 yang-VirtualBox:domain *:.*              LISTEN
udp      0      0 *:bootpc          *:.*              LISTEN
udp6     0      0 [::]:43613        [::]:.*          LISTEN
udp6     0      0 [::]:mdns         [::]:.*          LISTEN
udp6     0      0 [::]:47121        [::]:.*          LISTEN

```

## 4. Установка ПО с помощью Ubuntu Software

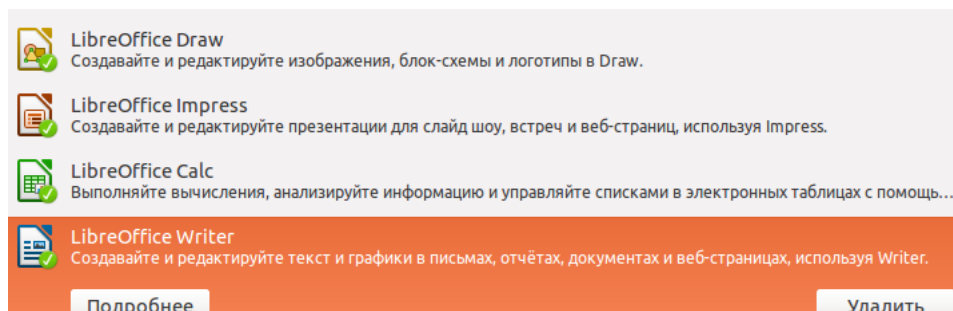
### 1. Установите приложения и проверьте работу ПО:

#### a. Kmplot



#### b. Drawio

#### c. Libreoffice.Write



## 5. Запуск приложений в Docker

### 1. Установите Docker:

Дистрибутив Docker, доступный в официальном репозитории Ubuntu, не всегда является последней версией программы. Лучше установить последнюю версию, загрузив ее из официального репозитория Docker.

- Сначала обновляем существующий перечень пакетов:

```
$ sudo apt update
```

- Затем устанавливаем пакеты, которые позволяют apt использовать протокол HTTPS:

```
$ sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

```
yang@yang-VirtualBox:~$ sudo apt install apt-transport-https ca-certificates curl software-properties-common
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Уже установлена самая новая версия apt-transport-https.
Уже установлена самая новая версия ca-certificates.
ca-certificates помечен как установленный вручную.
Уже установлена самая новая версия curl.
curl помечен как установленный вручную.
Уже установлена самая новая версия software-properties-common.
software-properties-common помечен как установленный вручную.
Следующие пакеты устанавливались автоматически и больше не требуются:
  kde-l10n-engb kde-l10n-ru
Для их удаления используйте «apt-get autoremove».
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
```

- Затем добавляем в свою систему ключ GPG официального репозитория Docker:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

```
yang@yang-VirtualBox:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
```

- Добавляем репозиторий Docker в список источников пакетов apt (все в одну строчку):

```
$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
```

```
yang@yang-VirtualBox:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
yang@yang-VirtualBox:~$
```

- Затем обновим базу данных пакетов информацией о пакетах репозитория Docker:



## **\$ sudo apt update**

```
Игн https://download.docker.com bionic/stable Translation-ru_RU
Игн https://download.docker.com bionic/stable Translation-ru
Игн https://download.docker.com bionic/stable Translation-en
Получено 83,3 kB за 7с (11,5 kB/s)
Чтение списков пакетов... Готово
```

- Далее устанавливаем Docker:

## **\$ sudo apt install docker-ce**

```
yang@yang-VirtualBox:~$ sudo apt-get install docker-ce=17.06.0~ce-0~ubuntu
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
  kde-l10n-engb kde-l10n-ru
Для их удаления используйте «apt-get autoremove».
Будут установлены следующие дополнительные пакеты:
  aufs-tools git git-man liberror-perl
Предлагаемые пакеты:
  git-daemon-run git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-arch git-bzr git-cvs git-mediawiki git-svn
Следующие НОВЫЕ пакеты будут установлены:
  aufs-tools docker-ce git git-man liberror-perl
Обновлено 0 пакетов, установлено 5 новых пакетов, для удаления отмечено 0 пакетов,
и 0 пакетов не обновлено.
Необходимо скачать 24,0 МВ архивов.
После данной операции объем занятого дискового пространства возрастет на 118 МВ.
Хотите продолжить? [Д/н] у
Получено:1 http://ru.archive.ubuntu.com/ubuntu/ trusty/universe aufs-tools amd64
1:3.2+20130722-1.1 [92,3 kB]
Получено:2 http://ru.archive.ubuntu.com/ubuntu/ trusty/main liberror-perl all 0.
```

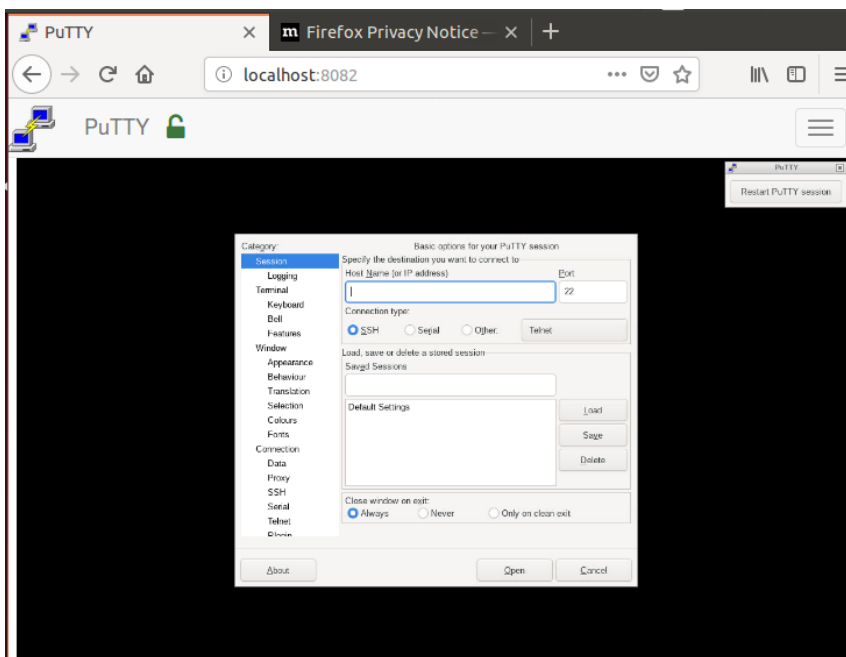
- Запуск приложения Putty
- Выполните в терминале команду:

## **\$ sudo docker run -d --name=putty -p 8082:5800 -v /docker/appdata/putty:/config:rw jlesage/putty**

```
yang@yang-VirtualBox:~$ sudo docker run -d --name=putty -p 8082:5800 -v /docker/
appdata/putty:/config:rw jlesage/putty
Unable to find image 'jlesage/putty:latest' locally
latest: Pulling from jlesage/putty
532819f3e44c: Pull complete
ae204b32d40f: Pull complete
1a5680e9f6cf: Pull complete
8bc8cd2482dd: Pull complete
07f9edd9d766: Pull complete
fcb0e57ce707: Pull complete
76e40d713c42: Pull complete
1938062d7bee: Pull complete
aa15a1f2194c: Pull complete
d9bcf3a8827f: Pull complete
fb5ea3142131: Pull complete
fb4cfbcb1d41: Pull complete
d25272dca9ff: Pull complete
6e76f3de4e93: Pull complete
d3cab8a785cf: Pull complete
c8fea395911a: Pull complete
de2fb3add290: Pull complete
51bed3d61dfe: Pull complete
bfde7ae05b07: Pull complete
Digest: sha256:fd5be3fe99ddd6e7abd2db861063e3e62fb14d90833a97e5d81f1f98a9223e1a
Status: Downloaded newer image for jlesage/putty:latest
9f5b1cebac36430d5d220bebfc5c4c09f395ecd0f2afae7cd4d961606c4dea71
```

- Проверьте работу приложения, запустите браузер и введите адрес:

<http://localhost:8082>



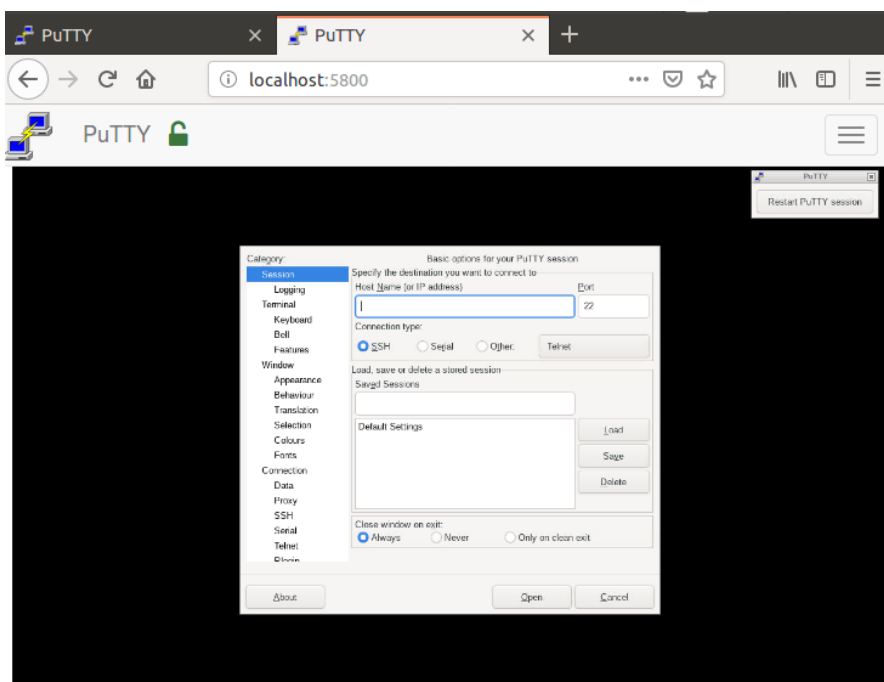
- Измените значение порта и запустите копию приложения:

```
$ sudo docker run -d --name=putty2 -p 5800:5800 -v /docker/appdata/putty:/config:rw jlesage/putty
```

```
yang@yang-VirtualBox:~$ sudo docker run -d --name=putty2 -p 5800:5800 -v /docker/appdata/putty:/config:rw jlesage/putty  
41e4313e7ffdae18103d9d3e2d8a7d0b012025951b93547aaab298c9a277531c
```

- Проверьте работу приложения, запустите браузер и введите адрес:

<http://localhost:5800>



- Посмотрите список запущенных контейнеров:

**\$docker ps**

```
yang@yang-VirtualBox:~$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED
STATUS        PORTS          NAMES
41e4313e7ffd   jlesage/putty  "/init"                 About a minute ago
Up About a minute   0.0.0.0:5800->5800/tcp, 5900/tcp  putty2
9f5b1cebac36   jlesage/putty  "/init"                 5 minutes ago
Up 5 minutes      5900/tcp, 0.0.0.0:8082->5800/tcp  putty
```

- Останавливаем все запущенные контейнеры и проверяем выполнение команды:

**\$ sudo docker kill \$(docker ps -q)**

```
yang@yang-VirtualBox:~$ sudo docker kill $(sudo docker ps -q)
41e4313e7ffd
9f5b1cebac36
```

## 2. *Создание образа в Docker:*

- Создайте пустой каталог (MyLabs-Docker).

```
yang@yang-VirtualBox:~$ su
Пароль:
root@yang-VirtualBox:/home/yang# cd ..
root@yang-VirtualBox:/home# mkdir MyLabs-Docker
```

- Войдите в каталог и создайте файл с именем Dockerfile

```
root@yang-VirtualBox:/home# cd MyLabs-Docker
root@yang-VirtualBox:/home/MyLabs-Docker# cat > Dockerfile
```

- Скопируйте и вставьте в этот файл следующее содержимое и сохраните его.

```
FROM python:2.7-slim

# Укажем рабочий каталог /app
WORKDIR /app

# Скопируем текущее содержимое каталога в контейнер по адресу /app
ADD . /app

# Установить список пакетов из файла requirements.txt
RUN pip install --trusted-host pypi.python.org -r requirements.txt

# Открыть 80 порт
EXPOSE 80

# Определение переменной окружения
ENV NAME World

# Команда запуска приложения
CMD ["python", "app.py"]
```

```

root@yang-VirtualBox:/home/MyLabs-Docker# cat > Dockerfile
FROM python:2.7-slim

# Укажем рабочий каталог /app
WORKDIR /app

# Скопируем текущее содержимое каталога в контейнер по адресу /app
ADD . /app

# Установить список пакетов из файла requirements.txt
RUN pip install --trusted-host pypi.python.org -r requirements.txt

# Открыть 80 порт
EXPOSE 80

# Определение переменной окружения
ENV NAME World

# Команда запуска приложения
CMD ["python", "app.py"]
^Z
[1]+  Остановлено  cat > Dockerfile

```

- Создать файлы с содержимым:

#### **requirements.txt:**

```

Flask
Redis

```

#### **app.py:**

```

from flask import Flask
from redis import Redis, RedisError
import os
import socket

# Connect to Redis
redis = Redis(host="redis", db=0, socket_connect_timeout=2,
socket_timeout=2)

app = Flask(__name__)

@app.route("/")
def hello():
    try:
        visits = redis.incr("counter")
    except RedisError:
        visits = "<i>cannot connect to Redis, counter disabled</i>"

    html = "<h3>Hello {name}!</h3>" \
"<b>Hostname:</b> {hostname}<br/>" \
"<b>Visits:</b> {visits}"

```

```

return      html.format(name=os.getenv("NAME",      "world"),
hostname=socket.gethostname(), visits=visits)

if __name__ == "__main__":
app.run(host='0.0.0.0', port=80)

```

```

root@yang-VirtualBox:/home/MyLabs-Docker# cat > requirements.txt
Flask
Redis
^Z
[2]+  Остановлено  cat > requirements.txt
root@yang-VirtualBox:/home/MyLabs-Docker# cat > app.py
from flask import Flask
from redis import Redis, RedisError
import os
import socket

redis=Redis(host="redis", db=0,socket_connect_timeout=2, socket_timeout=2)

app=Flask(__name__)

@app.route("/")
def hello():
    try:
        visits=redis.incr("counter")
    except RedisError:
        visits="<i>cannot connect to Redis, counter disabled</i>"
    html="<h3>Hello {name}</h3>" \
        "<b>Hostname:</b> {hostname}<br/>" \
        "<b>Visits:</b>{visits}"
    return html.format(name=os.getenv("NAME", "world"), hostname=socket.gethostname(), visits=visits)

if __name__=="__main__":
    app.run(host='0.0.0.0',port=80)
^Z
[3]+  Остановлено  cat > app.py

```

- Описать что произойдет при первом запуске скрипта. Что происходит при повторных запусках?

при первом запуске скрипта:(download requirements)

```

root@yang-VirtualBox:/home/MyLabs-Docker# sudo docker build -t friendlyhello
"docker build" requires exactly 1 argument(s).
See 'docker build --help'.

Usage:  docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile
root@yang-VirtualBox:/home/MyLabs-Docker# sudo docker build -t friendlyhello
Sending build context to Docker daemon  5.12kB
Step 1/7 : FROM python:2.7-slim
2.7-slim: Pulling from library/python
123275d6e508: Pull complete
dd1cd6637523: Pull complete
0c4e6d630f2c: Pull complete
13e9cd8f0ea1: Pull complete
Digest: sha256:6c1ffdf499e29ea663e6e67c9b6b9a3b401d554d2c9f061f9a45344e39923
Status: Downloaded newer image for python:2.7-slim
--> eeb27ee6b893
Step 2/7 : WORKDIR /app
--> 04328744a3f3
Removing intermediate container d498f416e720

```

при повторных запусках:

```

root@yang-VirtualBox:/home/MyLabs-Docker# sudo docker build -t friendlyhello .
Sending build context to Docker daemon 5.12kB
Step 1/7 : FROM python:2.7-slim
--> eeb27ee6b893
Step 2/7 : WORKDIR /app
--> Using cache
--> 04328744a3f3
Step 3/7 : ADD . /app
--> Using cache
--> 93d64414f9bd
Step 4/7 : RUN pip install --trusted-host pypi.python.org -r requirements.txt
--> Using cache
--> 326fb516cedc
Step 5/7 : EXPOSE 80
--> Using cache
--> f0c131f6434d
Step 6/7 : ENV NAME World
--> Using cache
--> 9fff6d60a4cf
Step 7/7 : CMD python app.py

```

- Запустите скрипт и следующие команды

**\$ sudo docker build -t friendlyhello**

**\$ sudo docker images**

```

root@yang-VirtualBox:/home/MyLabs-Docker# sudo docker build -t friendlyhello
"docker build" requires exactly 1 argument(s).
See 'docker build --help'.

Usage:  docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile
root@yang-VirtualBox:/home/MyLabs-Docker# sudo docker build -t friendlyhello
Sending build context to Docker daemon 5.12kB
Step 1/7 : FROM python:2.7-slim
2.7-slim: Pulling from library/python
123275d6e508: Pull complete
dd1cd6637523: Pull complete
0c4e6d630f2c: Pull complete
13e9cd8f0ea1: Pull complete
Digest: sha256:6c1ffdf499e29ea663e6e67c9b6b9a3b401d554d2c9f061f9a45344e39923
Status: Downloaded newer image for python:2.7-slim
--> eeb27ee6b893
Step 2/7 : WORKDIR /app
--> 04328744a3f3
Removing intermediate container d498f416e720

Successfully built f8ffe5b6725a
Successfully tagged friendlyhello:latest

```

```

root@yang-VirtualBox:/home/MyLabs-Docker# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
friendlyhello       latest             f8ffe5b6725a       5 seconds ago
159MB
jlesage/putty       latest            da76ab168a08       6 weeks ago
105MB
python              2.7-slim          eeb27ee6b893       14 months ago
148MB

```

- Запустите приложение и откройте следующие порты (последние 4 цифры вашего ID в ИСУ) **4000** и **80**

**\$ sudo docker run -p 80:80 friendlyhello**



```

root@yang-VirtualBox:/home/MyLabs-Docker# sudo docker run -p 80:80 friendlyhello
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
172.17.0.1 - - [28/Jun/2021 22:15:41] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [28/Jun/2021 22:16:16] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [28/Jun/2021 22:16:16] "GET /favicon.ico HTTP/1.1" 404 -

```



**Hello World!**

**Hostname:** 6d401e70541c

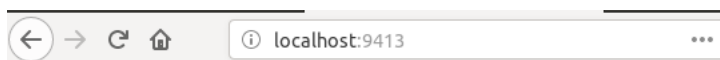
**Visits:** cannot connect to Redis, counter disabled

**\$ sudo docker run -p 4000:80 friendlyhello**

```

root@yang-VirtualBox:/home/MyLabs-Docker# sudo docker run -p 9413:80 friendlyhello
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
172.17.0.1 - - [28/Jun/2021 22:22:02] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [28/Jun/2021 22:22:02] "GET /favicon.ico HTTP/1.1" 404 -

```



**Hello World!**

**Hostname:** 8471d4b04f92

**Visits:** cannot connect to Redis, counter disabled

Проверьте работу в браузере <http://localhost:4000> и <http://localhost:80>

- Выполните команду

**\$ sudo docker container ls**

```

root@yang-VirtualBox:/home/MyLabs-Docker# sudo docker container ls
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES

```

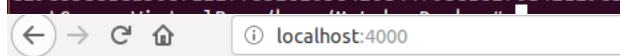
- Запустите программу app в фоновом режиме

**\$ sudo docker run -d -p 4000:80 friendlyhello**

```

root@yang-VirtualBox:/home/MyLabs-Docker# sudo docker run -d -p 4000:80 friendlyhello
c29c83ae1c1b6e71227f8b2e26bd426a44f0ac1e276a42129e36d109b020d749

```



**Hello World!**

**Hostname:** c29c83ae1c1b

**Visits:** cannot connect to Redis, counter disabled

- Выполните команду

**\$ sudo docker container ls**

```
root@yang-VirtualBox:/home/MyLabs-Docker# sudo docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
c29c83ae1c1b       friendlyhello      "python app.py"     About a minute ago
Up About a minute   0.0.0.0:4000->80/tcp  hardcore_brahmagupta
```

- Завершите работу контейнера friendlyhello по ID и имени

```
root@yang-VirtualBox:/home/MyLabs-Docker# sudo docker stop c29c83ae1c1b
c29c83ae1c1b
```

### 3. Установка Базы данных в Docker

- Проведем подготовку к работе приложения:
- Создайте необходимый каталог для хранения данных MariaDB

**\$sudo mkdir -p /var/lib/mysql**

```
root@yang-VirtualBox:/home# sudo mkdir -p /var/lib/mysql
```

- Установите клиентский пакет MySQL на локальной машине

**\$sudo apt install mysql-client**

```
root@yang-VirtualBox:/home# sudo apt install mysql-client
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
  kde-l10n-engb kde-l10n-gu
Для их удаления используйте «apt-get autoremove».
Будут установлены следующие дополнительные пакеты:
  libdbd-mysql-perl libdbi-perl libmysqlclient18 libterm-readkey-perl
  mysql-client-5.5 mysql-client-core-5.5 mysql-common
Предлагаемые пакеты:
  libmldbm-perl libnet-daemon-perl libplrpc-perl libsql-statement-perl
Следующие НОВЫЕ пакеты будут установлены:
```

- Скачаем образ базы данных MariaDB

**\$sudo docker pull mariadb**

```
root@yang-VirtualBox:/home# sudo docker pull mariadb
Using default tag: latest
latest: Pulling from library/mariadb
c549ccf8d472: Pull complete
26ea6552a462: Pull complete
329b1f41043f: Pull complete
9f8d09317d80: Pull complete
2bc055a5511d: Pull complete
e989e430508e: Pull complete
cdba2af19f87: Pull complete
04fe4f90eab8: Pull complete
389c6b423e31: Pull complete
bef640655d86: Pull complete
Digest: sha256:0c72b63198ac53df4e84db821876c73794b00509b2d8a77100d186a13e49ac31
Status: Downloaded newer image for mariadb:latest
```

- Просмотр образов Docker, установленных в системе

- Перечислите изображения Docker, установленные в вашей системе.

### **docker images**

```
root@yang-VirtualBox:/home# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
friendlyhello	latest	f8ffe5b6725a	18 minutes ago	59MB
mariadb	latest	6d5c5ed114ad	4 days ago	108MB
jlesage/putty	latest	da76ab168a08	6 weeks ago	105MB
python	2.7-slim	eeb27ee6b893	14 months ago	48MB

- Запустите контейнер MariaDB с постоянным хранением данных в папке /var/lib/mysql

**\$sudo docker run -d --name mariadb-server -p 3306:3306 -v /var/lib/mysql:/var/lib/mysql -e "MYSQL\_ROOT\_PASSWORD=user" mariadb**

```
root@yang-VirtualBox:/home# sudo docker run -d --name mariadb-server -p 3306:3306 -v /var/lib/mysql:/var/lib/mysql -e "MYSQL_ROOT_PASSWORD=user" mariadb
e4daf4dbed82ea1d7a2c760d8a162b13093686d70ff7b65970afb90f9a410053
```

- Проверьте доступ к БД MariaDB, запущенной в контейнере Docker

**\$sudo mysql -h 127.0.0.1 -u root -p**

```
root@yang-VirtualBox:/home# sudo mysql -h 127.0.0.1 -u root -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'172.17.0.1' (using password: YES)
```

## Приложение

1. `docker ps` — показывает список запущенных контейнеров.
  - `-a / --all` — список всех контейнеров;
  - `-q / --quiet` — перечислить только `id` контейнеров.
2. `docker pull` — Docker Hub содержит множество готовых образов, которые можно использовать вместо того, чтобы создавать и настраивать свой собственный. Чтобы скачать определённый образ или набор образов (репозиторий), используется команда `docker pull`.
3. `docker build` — эта команда собирает образ Docker из Dockerfile и «контекста».
4. `docker run` — запускает контейнер, на основе указанного образа.
5. `docker logs` — эта команда используется для просмотра логов указанного контейнера. Можно использовать флаг `--follow`, чтобы следить за логами работающей программы: `docker logs --follow my_container`.
6. `docker rm` — удаляет один и более контейнеров, например, `docker rm my_container`.
7. `docker rmi` — удаляет один и более образов, например, `docker rmi my_image`.
8. `docker stop` — останавливает один и более контейнеров.
  - `docker stop my_container` - остановит один контейнер
  - `docker stop $(docker ps -a -q)` — все запущенные.
9. Можно комбинировать эти команды. Например, для очистки всех контейнеров и образов:
  - `docker kill $(docker ps -q)` - Останавливаем все запущенные контейнеры.
  - `docker rm $(docker ps -a -q)` - Удаляем все остановленные контейнеры.