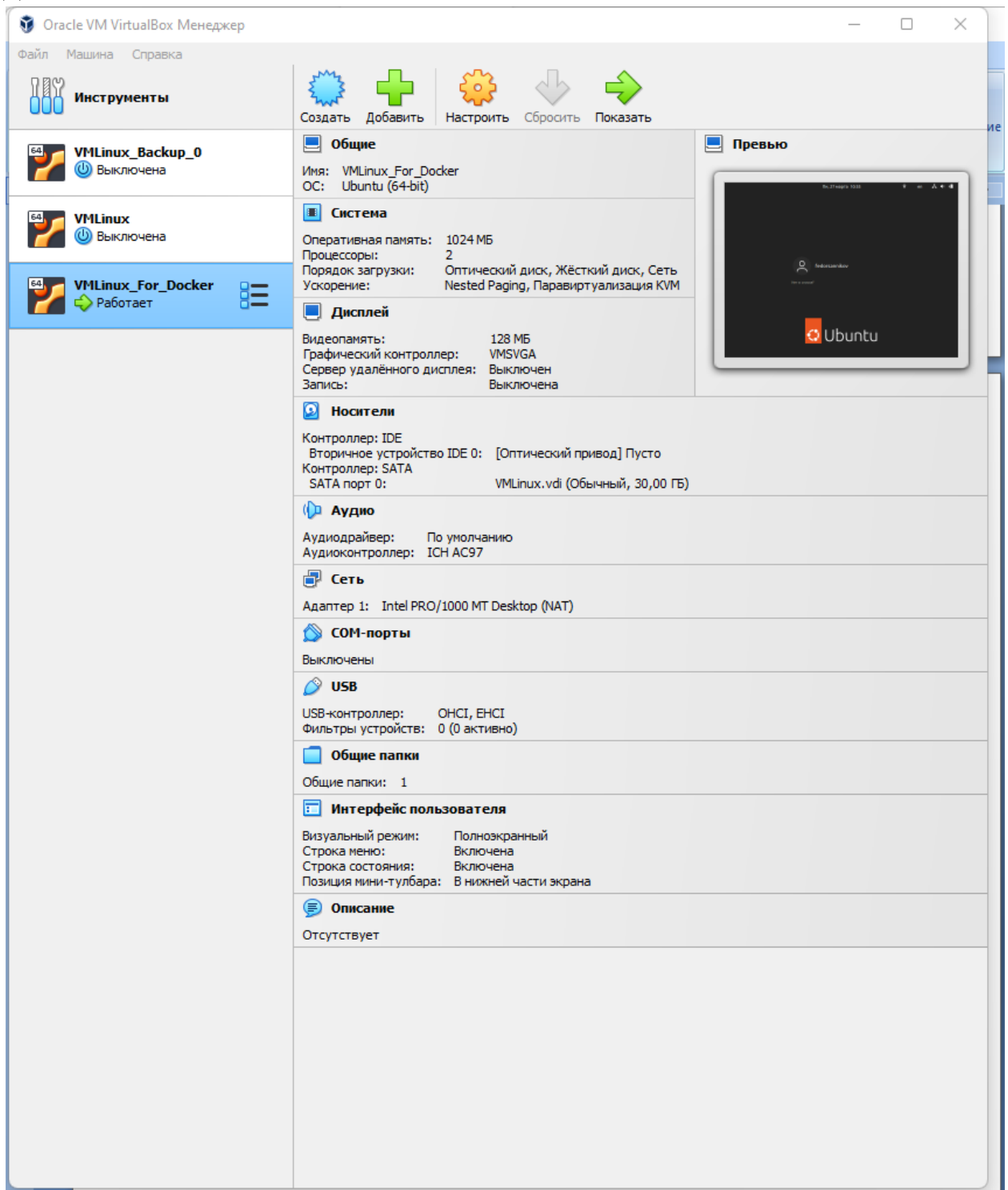


Домашняя работа №7:

1. Переустановить операционную систему (по желанию, для дополнительной практики):

Устанавливал ранее 3-й раз подряд, так же как и делал резервную копию системы. Сейчас не вижу смысла в этом.

Для доказательства:



2 Установить Docker.

Устанавливаем:

apt install docker.io

```
fedorsannikov@VMLinux:~$ sudo apt install docker.io
```

Проверяем работоспособность:

```
fedorsannikov@VMLinux:~$ sudo docker
Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default "/root/.docker")
  -c, --context string  Name of the context to use to connect to the daemon (overrides DOCKER_HOST env var and default context set with "docker context use")
  -D, --debug           Enable debug mode
  -H, --host list       Daemon socket(s) to connect to
  -l, --log-level string Set the logging level ("debug"|"info"|"warn"|"error"|"fatal") (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string    Trust certs signed only by this CA (default "/root/.docker/ca.pem")
  --tlscert string       Path to TLS certificate file (default "/root/.docker/cert.pem")
  --tlskey string        Path to TLS key file (default "/root/.docker/key.pem")
  --tlsverify           Use TLS and verify the remote
  -v, --version         Print version information and quit
```

Создаем и запускаем контейнер hello-world так же для проверки работоспособности docker:

docker run hello-world

```
fedorsannikov@VMLinux:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:fffb13da98453e0f04d33a6eee5bb8e46ee50d08ebe17735fc0779d0349e889e9
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Список запущенных контейнеров:

```
sudo docker ps
```

```
fedorsannikov@VMLinux:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
```

Список всех контейнеров:

```
sudo docker ps -a
```

```
fedorsannikov@VMLinux:~$ sudo docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
64177134d3a9   hello-world  "/hello"   3 minutes ago   Exited (0) 3 minutes ago   elastic_elbakyan
d90678b965b2   hello-world  "/hello"   3 minutes ago   Exited (0) 3 minutes ago   flamboyant_agnesi
5d2dd3e09638   hello-world  "/hello"   6 minutes ago   Exited (0) 6 minutes ago   crazy_greider
```

P.S.: Запускал три раза один контейнер .

Удаляем все три контейнера:

```
docker rm 64177134d3a9
```

```
root@VMLinux:/home/fedorsannikov# docker rm 64177134d3a9
64177134d3a9
```

```
docker rm d90678b965b2
```

```
root@VMLinux:/home/fedorsannikov# docker rm d90678b965b2
d90678b965b2
```

```
docker rm 5d2dd3e09638
```

```
root@VMLinux:/home/fedorsannikov# docker rm 5d2dd3e09638
5d2dd3e09638
```

Список всех образов контейнеров на машине:

```
sudo docker images
```

```
fedorsannikov@VMLinux:~$ sudo su
root@VMLinux:/home/fedorsannikov# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world    latest    feb5d9fea6a5  18 months ago  13.3kB
```

Удаляем образ контейнера hello-world на машине:

```
docker rmi hello-world
```

```
root@VMLinux:/home/fedorsannikov# docker rmi hello-world
Untagged: hello-world:latest
Untagged: hello-world@sha256:ffb13da98453e0f04d33a6eee5bb8e46ee50d08ebe17735fc0779d0349e889e9
Deleted: sha256:feb5d9fea6a5e9606aa995e879d862b825965ba48de054caab5ef356dc6b3412
Deleted: sha256:e07ee1baac5fae6a26f30cabfe54a36d3402f96afda318fe0a96cec4ca393359
```

docker images

```
root@VMLinux:/home/fedorsannikov# docker images
REPOSITORY    TAG        IMAGE ID    CREATED    SIZE
root@VMLinux:/home/fedorsannikov#
```

2 Запустить контейнер с Ubuntu

ищем контейнер с ubuntu:

docker search ubuntu

```
root@VMLinux:/home/fedorsannikov# docker search ubuntu
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
ubuntu	Ubuntu is a Debian-based Linux operating sys...	15767	[OK]	
websphere-liberty	WebSphere Liberty multi-architecture images ...	293	[OK]	
ubuntu-upstart	DEPRECATED, as is Upstart (find other proces...	112	[OK]	
neurodebian	NeuroDebian provides neuroscience research s...	99	[OK]	
ubuntu/nginx	Nginx, a high-performance reverse proxy & we...	83		
open-liberty	Open Liberty multi-architecture images based...	59	[OK]	
ubuntu/apache2	Apache, a secure & extensible open-source HT...	57		
ubuntu/squid	Squid is a caching proxy for the Web. Long-t...	52		
ubuntu-debootstrap	DEPRECATED; use "ubuntu" instead	50	[OK]	
ubuntu/bind9	BIND 9 is a very flexible, full-featured DNS...	48		
ubuntu/mysql	MySQL open source fast, stable, multi-thread...	43		
ubuntu/prometheus	Prometheus is a systems and service monitori...	40		
ubuntu/postgres	PostgreSQL is an open source object-relatio...	27		
ubuntu/kafka	Apache Kafka, a distributed event streaming ...	26		
ubuntu/redis	Redis, an open source key-value store. Long-...	17		
ubuntu/prometheus-alertmanager	Alertmanager handles client alerts from Prom...	8		
ubuntu/dotnet-deps	Chiselled Ubuntu for self-contained .NET & A...	7		
ubuntu/grafana	Grafana, a feature rich metrics dashboard & ...	7		
ubuntu/memcached	Memcached, in-memory keyvalue store for smal...	5		
ubuntu/zookeeper	ZooKeeper maintains configuration informatio...	5		
ubuntu/dotnet-runtime	Chiselled Ubuntu runtime image for .NET apps...	5		
ubuntu/telegraf	Telegraf collects, processes, aggregates & W...	4		
ubuntu/dotnet-aspnet	Chiselled Ubuntu runtime image for ASP.NET a...	4		
ubuntu/cortex	Cortex provides storage for Prometheus. Long...	3		
ubuntu/cassandra	Cassandra, an open source NoSQL distributed ...	2		

ИЛИ

The screenshot shows the Docker Hub interface for the 'ubuntu' official image. At the top, there's a search bar and navigation links like 'Explore', 'Pricing', 'Sign In', and 'Register'. Below the header, the 'ubuntu' image is featured with its logo, a 'DOCKER OFFICIAL IMAGE' badge, and statistics (15+ downloads, 10K+ stars). A 'docker pull ubuntu' button is visible. The main content area is divided into sections: 'Quick reference' (listing maintainers Canonical and Tianon, and where to get help), 'Supported tags and respective Dockerfile links' (listing various Ubuntu versions like 18.04, 20.04, 22.04, 22.10, and 23.04 with their respective Dockerfiles), 'Recent Tags' (showing a list of recent tags like rolling, lunar-20230314, lunar, latest, kinetic-20230308, kinetic, jammy-20230308, jammy, focal-20230308, and focal), 'About Official Images' (explaining that these are curated, open-source images), and 'Why Official Images?' (stating they have clear documentation and promote best practices).

Хочу установить контейнер Ubuntu версии 22.10

Загружаем образ ubuntu версии 20.04 на свой компьютер:

docker pull ubuntu:20.04

```
root@VMLinux:/home/fedorsannikov# docker pull ubuntu:20.04
20.04: Pulling from library/ubuntu
06d39c85623a: Pull complete
Digest: sha256:24a0df437301598d1a4b62ddf59fa0ed2969150d70d748c84225e6501e9c36b9
Status: Downloaded newer image for ubuntu:20.04
docker.io/library/ubuntu:20.04
```

Список всех образов контейнеров на машине:

sudo docker images

```
docker.io/library/ubuntu:20.04
root@VMLinux:/home/fedorsannikov# sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        20.04     1c5c8d0b973a   2 weeks ago    72.8MB
```

Запускаем образ контейнера на машине (присвоив контейнеру имя Ubuntu2004):

docker run -it --name Ubuntu2004 ubuntu:20.04 /bin/bash

```
root@VMLinux:/home/fedorsannikov# docker run -it --name Ubuntu2004 ubuntu:20.04 /bin/bash
root@b88f3cbc680c:/# exit
exit
```

Примечание:

Создаем контейнер с командой, которая будет выполняться (для этого передаем команду после имени образа /bin/bash). Чтобы контейнер был доступен интерактивно, создаем с опций -i и -t.

Работать в контейнере крайне не желательно !

Запускаем контейнер:

docker start Ubuntu2004

```
root@VMLinux:/home/fedorsannikov# docker start Ubuntu2004
Ubuntu2004
```

Список запущенных контейнеров:

docker ps

```
root@VMLinux:/home/fedorsannikov# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS      PORTS          NAMES
b88f3cbc680c   ubuntu:20.04   "/bin/bash"             2 minutes ago   Exited (0)   2 minutes ago   Ubuntu2004
```

Останавливаем контейнер:

docker stop Ubuntu2004

```
root@VMLinux:/home/fedorsannikov# docker stop Ubuntu2004
Ubuntu2004
```

Список запущенных контейнеров:

docker ps

```
root@VMLinux:/home/fedorsannikov# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
root@VMLinux:/home/fedorsannikov#
```

Удаляем контейнер с присвоенным именем Ubuntu2004:

docker rm Ubuntu2004

```
root@VMLinux:/home/fedorsannikov# docker rm Ubuntu2004
Ubuntu2004
```

docker ps

```
root@VMLinux:/home/fedorsannikov# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
```

Удаляем образ контейнера Ubuntu2004 на машине:

docker rmi 1c5c8d0b973a

```
root@VMLinux:/home/fedorsannikov# docker rmi 1c5c8d0b973a
Untagged: ubuntu:20.04
Untagged: ubuntu@sha256:24a0df437301598d1a4b62ddf59fa0ed2969150d70d748c84225e6501e9c36b9
Deleted: sha256:1c5c8d0b973ab9d6d7742cf050beefa8ba2efb7779e09bfb1200d6192180766e
Deleted: sha256:62789ffcd78ce0eae545b2743541d5c063270fe9f45477acacf2031b93cea70f
```

docker images

```
root@VMLinux:/home/fedorsannikov# docker images
REPOSITORY    TAG       IMAGE ID   CREATED   SIZE
```

Полезные материалы:

<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-20-04-ru>

<https://losst.pro/zapusk-kontejnera-docker>

4 Используя Dockerfile, собрать связку nginx + PHP-FPM в одном контейнере.

Полезный материал:

<https://www.dmosk.ru/miniinstruktions.php?mini=docker-self-image>

и

<https://www.dmosk.ru/miniinstruktions.php?mini=docker-webserver>

и

<https://gist.github.com/bocharsky-bw/3ef82076f022ba11fad9>

Работаем под root:

```
sudo su
```

Создаем директорию (в которой будем создавать свой образ):

```
mkdir -p /opt/docker
```

Переходим в созданный каталог:

```
cd /opt/docker
```

Создаем докер-файл (в котором пишем “инструкцию” для создания образа):

```
nano Dockerfile
```

Содержание файла:

```
FROM ubuntu:22.04
```

```
MAINTAINER FedorSannikov
```

```
ENV TZ=Europe/Moscow
```

```
RUN apt update -y
```

```
RUN apt install -y nginx php php-fpm
```

```
RUN echo "\ndaemon off;" >> /etc/nginx/nginx.conf
```

```
RUN mkdir /run/php-fpm
```

```
CMD php-fpm start && nginx
```

```
EXPOSE 80
```


Собираем образ:

```
docker build -t fedorsannikov/nginx_php_fpm:v1 .
```

Примечание:

fedorsannikov - имя автора;

nginx_php_fpm - название для сборки;

v1 - указанием версии.

Точка на конце указывает, что поиск Dockerfile выполняем в текущей директории.

Просматриваем, создан образ или нет (хотя и так будет понятно создался он или нет):

```
docker images
```