

Министерство высшего образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)  
Кафедра безопасности информационных систем (БИС)

LINUX, DOCKER, BASH, ОКРУЖЕНИЕ ДЛЯ РАЗРАБОТКИ

Отчет по лабораторной работе №1

по дисциплине «Системное программирование»

Студентка гр.737-1

\_\_\_\_\_ Агеева В.С.

\_\_\_\_.\_\_\_\_.2021г

Принял

Руководитель

доцент кафедры БИС

\_\_\_\_\_ Романов А.С.

\_\_\_\_.\_\_\_\_.2021г

## 1 Введение

Целью лабораторной работы является ознакомление с операционной системой и основными командами ОС Linux, возможностями Docker для контейнеризации окружения программного обеспечения и его зависимостей, средств разработки и сборки ПО. Необходимо подготовить Dockerfile, в котором осуществляется виртуализация операционной системы и устанавливаются необходимые пакеты.

В ходе лабораторной работы требуется выполнить задание и разработать скрипт, соответствующее 1 варианту (рисунок 1.1).

### Варианты заданий (образ ОС для Docker)

#### 1. CentOS

##### Задание 1

Разработать скрипт, который:

- запрашивает у пользователя имя пакета
- если указанный пакет установлен, то выводит информацию о пакете
- если указанный пакет не установлен, то выводит сообщение об этом и производит поиск пакета в репозиториях
- если пакет найден в репозиториях, то предлагает пользователю его установить или выйти из скрипта

Для выполнения задания используйте команды rpm и yum.

Рисунок 1.1 – Вариант и задание на лабораторную работу

После выполнения задания, необходимо загрузить код готовой программы и Dockerfile на github.com.

## 2 Ход работы

Для начала лабораторной работы рекомендуется реализовывать установку Docker на операционной системе Ubuntu, запустить которую можно на виртуальной машине (рисунок 2.1).

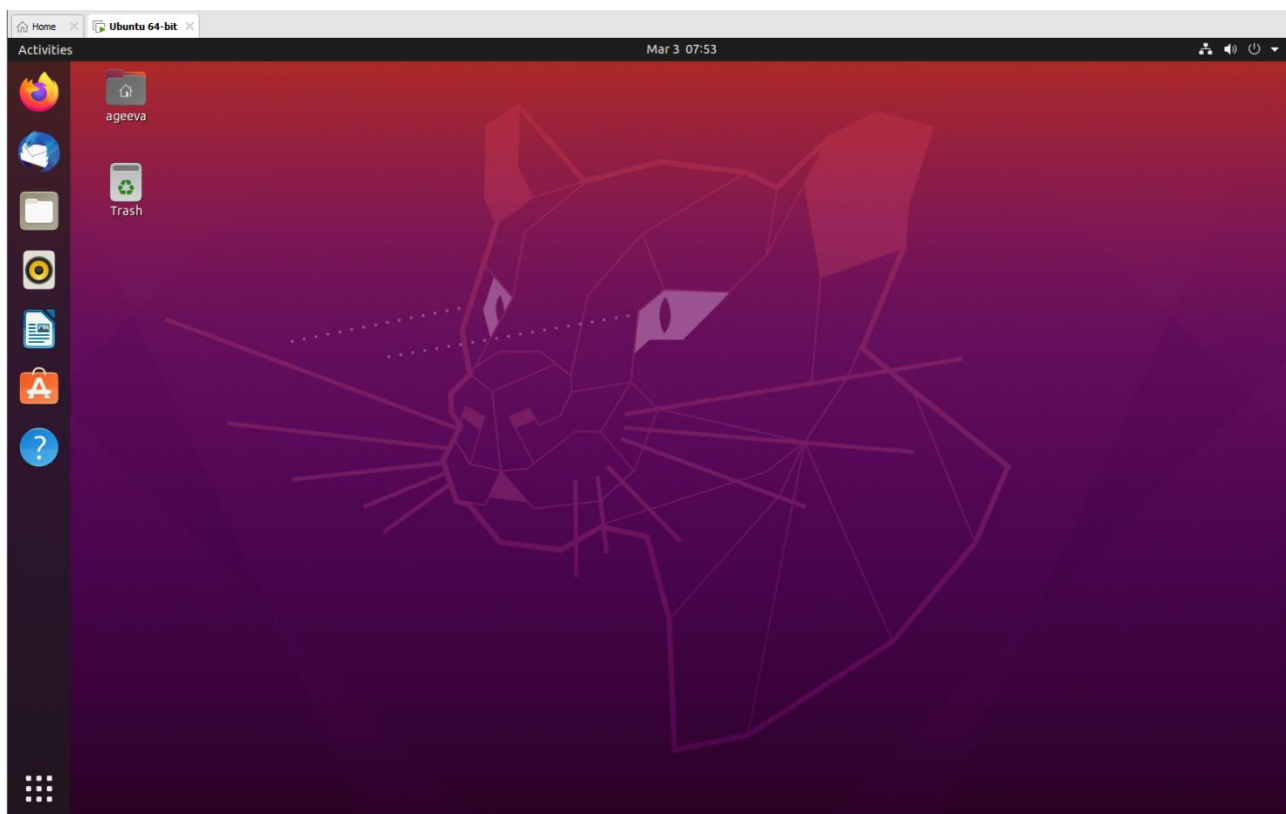


Рисунок 2.1 – Ubuntu 20.04.2.0, реализованная на VMware

После приступим к работе с терминалом. Для вызова терминала нужно нажать сочетание клавиш `Ctrl+Alt+T`. Запустив его, можно увидеть следующее окно (рисунок 2.2).

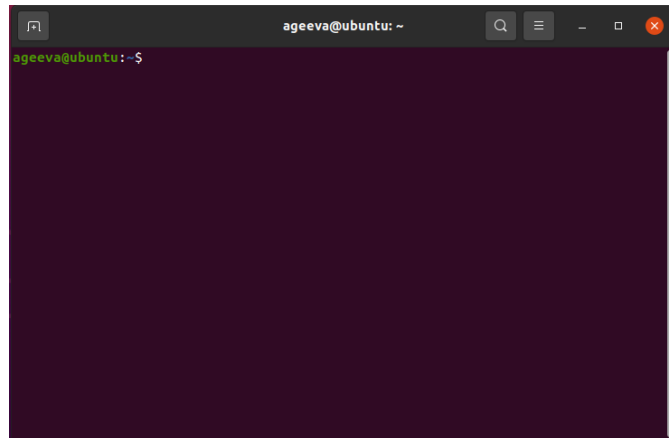


Рисунок 2.2 – Терминал

Для того чтобы подготовить систему к установке Docker, для начала нужно обновить индекс пакетов (загружаем списки пакетов из репозитория и "обновляем" их, чтобы получить информацию о новейших версиях пакетов и их зависимостях) (рисунок 2.3).

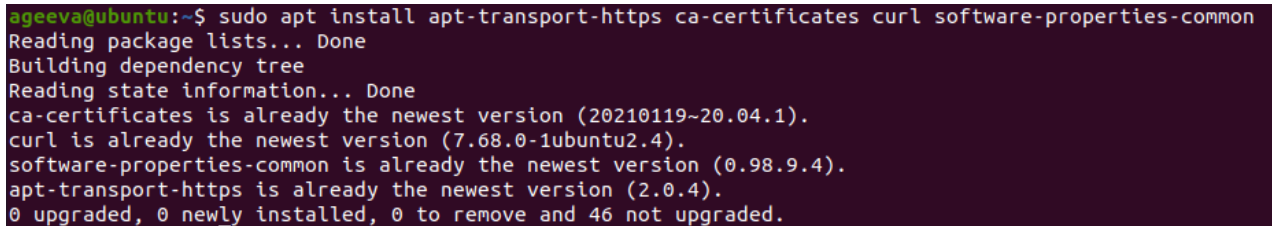
Команда: `sudo apt-get update`

```
ageeva@ubuntu:~$ sudo apt-get update
[sudo] password for ageeva:
Get:1 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:3 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages [17.1 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [521 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [434 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [24.3 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [7,072 B]
Get:11 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [841 kB]
Get:12 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [542 kB]
Get:13 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [58.1 kB]
Get:14 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [10.4 kB]
Get:15 http://us.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [201 kB]
Get:16 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [264 kB]
Get:17 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [12.8 kB]
Get:18 http://us.archive.ubuntu.com/ubuntu focal-updates/universe i386 Packages [555 kB]
Get:19 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [746 kB]
Get:20 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [302 kB]
Get:21 http://us.archive.ubuntu.com/ubuntu focal-updates/universe DEP-11 64x64 Icons [349 kB]
Get:22 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [16.2 kB]
Get:23 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [2,468 B]
Get:24 http://us.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [1,768 B]
Fetched 5,294 kB in 12s (445 kB/s)
Reading package lists... Done
```

Рисунок 2.3 - Обновление индекс пакетов

Затем установим необходимые пакеты, которые позволяют apt использовать пакеты по HTTPS (рисунок 2.4).

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

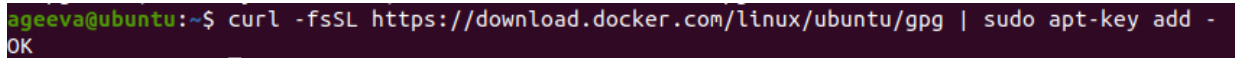


```
ageeva@ubuntu:~$ sudo apt install apt-transport-https ca-certificates curl software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20210119-20.04.1).
curl is already the newest version (7.68.0-1ubuntu2.4).
software-properties-common is already the newest version (0.98.9.4).
apt-transport-https is already the newest version (2.0.4).
0 upgraded, 0 newly installed, 0 to remove and 46 not upgraded.
```

Рисунок 2.4 – Установка необходимых пакетов

Затем добавим в свою систему ключ GPG официального репозитория Docker (рисунок 2.5).

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



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

Рисунок 2.5 – Добавление в систему ключа GPG

Добавим репозиторий Docker в список источников пакетов APT (рисунок 2.6).

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

```
ageeva@ubuntu:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:5 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
```

Рисунок 2.6 – Добавление репозитория Docker в список источников пакетов АРТ

Затем обновим базу данных пакетов информацией о пакетах Docker из вновь добавленного репозитория (рисунок 2.7).

Команда: `sudo apt-get update`

```
ageeva@ubuntu:~$ sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu bionic InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:5 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
```

Рисунок 2.7 – Обновление базы данных пакетов

Следует убедиться, что устанавливаем Docker из репозитория Docker, а не из репозитория по умолчанию Ubuntu.

Команда: `apt-cache policy docker-ce`

Вывод получится приблизительно следующий (рисунок 2.8).

```

ageeva@ubuntu:~$ apt-cache policy docker-ce
docker-ce:
  Installed: 5:20.10.4~3-0~ubuntu-bionic
  Candidate: 5:20.10.5~3-0~ubuntu-bionic
  Version table:
     5:20.10.5~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
*** 5:20.10.4~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     100 /var/lib/dpkg/status
     5:20.10.3~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:20.10.2~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:20.10.1~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:20.10.0~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.15~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.14~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.13~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.12~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.11~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.10~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.9~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.8~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.7~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.6~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.5~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.4~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.3~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
     5:19.03.2~3-0~ubuntu-bionic 500
        500 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages

```

Рисунок 2.8 – Проверка установки

Далее установим Docker (рисунок 2.9).

Команда: `sudo apt install docker-ce`

```
ageeva@ubuntu:~$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following packages will be upgraded:
  docker-ce
1 upgraded, 0 newly installed, 0 to remove and 20 not upgraded.
Need to get 24.8 MB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 https://download.docker.com/linux/ubuntu bionic/stable amd64 docker-ce amd64 5:20.10.5~3-0-ubuntu-bionic [24.8 MB]
Fetched 24.8 MB in 2s (10.6 MB/s)
(Reading database ... 183983 files and directories currently installed.)
Preparing to unpack .../docker-ce_5%3a20.10.5~3-0-ubuntu-bionic_amd64.deb ...
Unpacking docker-ce (5:20.10.5~3-0-ubuntu-bionic) over (5:20.10.4~3-0-ubuntu-bionic) ...
Setting up docker-ce (5:20.10.5~3-0-ubuntu-bionic) ...
Processing triggers for systemd (245.4-4ubuntu3.4) ...
```

Рисунок 2.9 – Установка docker

Теперь Docker установлен, демон запущен, и процесс будет запускаться при загрузке системы. Убедимся, что процесс запущен (рисунок 2.10).

Команда: `sudo systemctl status docker`

```
ageeva@ubuntu:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2021-03-03 08:08:18 PST; 1min 5s ago
 TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 11777 (dockerd)
      Tasks: 10
     Memory: 40.4M
    CGroup: /system.slice/docker.service
            └─11777 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

Рисунок 2.10 – Проверка запущен ли процесс

Вывод должен быть похож на представленный выше, сервис должен быть запущен и активен (рисунок 2.10).

Чтобы не вводить `sudo` каждый раз при запуске команды `docker`, добавим имя своего пользователя в группу `docker` (рисунок 2.11).

Команда: `sudo usermod -aG docker ${USER}`



```
ageevasp@ubuntu:~$ sudo usermod -aG docker ageevasp
ageevasp@ubuntu:~$
```

Рисунок 2.11 – Добавление пользователя в группу docker

Для применения этих изменений в составе группы необходимо разлогиниться и снова залогиниться на сервере или задать следующую команду (рисунок 2.12).

Команда: `su - ${USER}`

```
ageeva@ubuntu:~$ su - ageeva
Password:
ageeva@ubuntu:~$
```

Рисунок 2.12 – Применение изменений через команду

Для продолжения работы необходимо ввести пароль пользователя.

Убедимся, что пользователь добавлен в группу docker можно следующим образом (рисунок 2.13).

Команда: `id -nG`

```
ageeva@ubuntu:~$ id -nG
ageeva adm cdrom sudo dip plugdev lpadmin lxd sambashare docker
ageeva@ubuntu:~$
```

Рисунок 2.13 – Проверка добавления пользователя

Команда docker позволяет использовать различные опции, команды с аргументами. Синтаксис выглядит следующим образом:

`docker [option] [command] [arguments]`

Для просмотра всех доступных подкоманд введем следующую команду

(рисунок 2.14).

Команда: docker

```
ageeva@ubuntu:~$ docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default "/home/ageeva/.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 "/home/ageeva/.docker/ca.pem")
  --tlscert string     Path to TLS certificate file (default "/home/ageeva/.docker/cert.pem")
  --tlskey string      Path to TLS key file (default "/home/ageeva/.docker/key.pem")
  --tlsverify          Use TLS and verify the remote
  -v, --version        Print version information and quit

Management Commands:
  app*                Docker App (Docker Inc., v0.9.1-beta3)
  builder             Manage builds
  buildx*             Build with BuildKit (Docker Inc., v0.5.1-docker)
  config              Manage Docker configs
  container            Manage containers
  context              Manage contexts
  image               Manage images
  manifest             Manage Docker image manifests and manifest lists
  network              Manage networks
  node                 Manage Swarm nodes
  plugin              Manage plugins
  secret              Manage Docker secrets
  service              Manage services
  stack               Manage Docker stacks
  swarm               Manage Swarm
  system              Manage Docker
  trust                Manage trust on Docker images
  volume              Manage volumes

Commands:
  attach              Attach local standard input, output, and error streams to a running container
  build               Build an image from a Dockerfile
  commit              Create a new image from a container's changes
  cp                  Copy files/folders between a container and the local filesystem
  create              Create a new container
  diff                Inspect changes to files or directories on a container's filesystem
  events              Get real time events from the server
  exec                Run a command in a running container
  export              Export a container's filesystem as a tar archive
  history              Show the history of an image
  images              List images
  import              Import the contents from a tarball to create a filesystem image
  info                Display system-wide information
  inspect             Return low-level information on Docker objects
  kill                Kill one or more running containers
  load                Load an image from a tar archive or STDIN
  login               Log in to a Docker registry
  logout              Log out from a Docker registry
  logs                Fetch the logs of a container
  pause               Pause all processes within one or more containers
  port                List port mappings or a specific mapping for the container
  ps                  List containers
  pull                Pull an image or a repository from a registry
  push                Push an image or a repository to a registry
  rename              Rename a container
  restart             Restart one or more containers
  rm                  Remove one or more containers
  rmi                 Remove one or more images
  run                 Run a command in a new container
  save                Save one or more images to a tar archive (streamed to STDOUT by default)
  search              Search the Docker Hub for images
  start               Start one or more stopped containers
  stats               Display a live stream of container(s) resource usage statistics
  stop                Stop one or more running containers
  tag                 Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
  top                 Display the running processes of a container
  unpause             Unpause all processes within one or more containers
  update              Update configuration of one or more containers
  version             Show the Docker version information
  wait                Block until one or more containers stop, then print their exit codes

Run 'docker COMMAND --help' for more information on a command.

To get more help with docker, check out our guides at https://docs.docker.com/go/guides/
```

Рисунок 2.14 – Подкоманды Docker

Для просмотра всей информации о Docker используется следующая команда (рисунок 2.15).

Команда: `docker info`

```
ageeva@ubuntu:~$ docker info
Client:
 Context:      default
 Debug Mode: false
 Plugins:
  app: Docker App (Docker Inc., v0.9.1-beta3)
  buildx: Build with BuildKit (Docker Inc., v0.5.1-docker)

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 20.10.5
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: true
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 io.containerd.runtime.v1.linux runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 269548fa27e0089a8b8278fc4fc781d7f65a939b
 runc version: ff819c7e9184c13b7c2607fe6c30ae19403a7aff
 init version: de40ad0
 Security Options:
  apparmor
  seccomp
   Profile: default
 Kernel Version: 5.8.0-44-generic
 Operating System: Ubuntu 20.04.2 LTS
 OSType: linux
 Architecture: x86_64
 CPUs: 2
 Total Memory: 3.815GiB
 Name: ubuntu
 ID: TYKX:ZFNU:DX2T:SXRD:GZGV:W76K:677Q:HDUF:7YC5:KGAM:3T6V:33TI
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: false
 Insecure Registries:
  127.0.0.0/8
 Live Restore Enabled: false
```

Рисунок 2.15 – Информация о docker

Для поиска необходимых образов на Docker Hub (рисунок 2.16) используется команда `docker` и подкоманда `search`. Чтобы найти образ Ubuntu, нужно ввести:

Команда: `docker search ubuntu`



NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
ubuntu	Ubuntu is a Debian-based Linux operating sys...	11894	[OK]	
dorowu/ubuntu-desktop-lxde-vnc	Docker image to provide HTML5 VNC interface ...	500		[OK]
websphere-liberty	WebSphere Liberty multi-architecture images ...	268	[OK]	
rastasheep/ubuntu-sshd	Dockerized SSH service, built on top of offi...	250		[OK]
consol/ubuntu-xfce-vnc	Ubuntu container with "headless" VNC session...	233		[OK]
ubuntu-upstart	Upstart is an event-based replacement for th...	110	[OK]	
neurodebian	NeuroDebian provides neuroscience research s...	80	[OK]	
1andiinternet/ubuntu-16-nginx-php-phpmyadmin-mysql-5	ubuntu-16-nginx-php-phpmyadmin-mysql-5	50		[OK]
ubuntu-debootstrap	debootstrap --variant=minbase --components=m...	44	[OK]	
open-liberty	Open Liberty multi-architecture images based...	42	[OK]	
i386/ubuntu	Ubuntu is a Debian-based Linux operating sys...	24		
nuagebec/ubuntu	Simple always updated Ubuntu docker images w...	24		[OK]
1andiinternet/ubuntu-16-apache-php-5.6	ubuntu-16-apache-php-5.6	14		[OK]
1andiinternet/ubuntu-16-apache-php-7.0	ubuntu-16-apache-php-7.0	13		[OK]
1andiinternet/ubuntu-16-nginx-php-phpmyadmin-mariadb-10	ubuntu-16-nginx-php-phpmyadmin-mariadb-10	11		[OK]
1andiinternet/ubuntu-16-nginx-php-5.6-wordpress-4	ubuntu-16-nginx-php-5.6-wordpress-4	8		[OK]
1andiinternet/ubuntu-16-apache-php-7.1	ubuntu-16-apache-php-7.1	6		[OK]
darksheer/ubuntu	Base Ubuntu Image -- Updated hourly	5		[OK]
pivotaldata/ubuntu	A quick freshening-up of the base Ubuntu doc...	4		
1andiinternet/ubuntu-16-nginx-php-7.0	ubuntu-16-nginx-php-7.0	4		[OK]
pivotaldata/ubuntu16.04-build	Ubuntu 16.04 image for GPDB compilation	2		
smartentry/ubuntu	ubuntu with smartentry	1		[OK]
pivotaldata/ubuntu-gpdb-dev	Ubuntu images for GPDB development	1		
1andiinternet/ubuntu-16-php-7.1	ubuntu-16-php-7.1	1		[OK]
pivotaldata/ubuntu16.04-test	Ubuntu 16.04 image for GPDB testing	0		

Рисунок 2.16 – Поиск в Docker Hub

Скрипт просматривает Docker Hub и возвращает список всех образов, имена которых подходят под заданный поиск.

Проще говоря, с помощью команды `search` был организован поиск, по ключевому слову Ubuntu, образов в Docker Hub.

В столбце OFFICIAL строка ОК показывает, что образ построен и поддерживается компанией, которая занимается разработкой этого проекта.

После чего с помощью команды `pull` нужно выбрать необходимую версию и приступить к скачиванию (рисунок 2.17).

Команда: `docker pull ubuntu`

```
ageeva@ubuntu:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
83ee3a23efb7: Pull complete
db98fc6f11f0: Pull complete
f611acd52c6c: Pull complete
Digest: sha256:703218c0465075f4425e58fac086e09e1de5c340b12976ab9eb8ad26615c3715
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

Рисунок 2.17 – Скачивание образа

Для просмотра загруженных на компьютер образов нужно ввести следующую команду (рисунок 2.18).

Команда: `docker images`

```
ageeva@ubuntu:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest    f63181f19b2f   5 weeks ago    72.9MB
ageeva@ubuntu:~$
```

Рисунок 2.18 – список загруженных на компьютер образов

После загрузки образа можно запустить контейнер с загруженным образом с помощью подкоманды `run` (рисунок 2.19).

В качестве примера запустим контейнер с помощью последней версии образа Ubuntu. Комбинация параметров `-i` и `-t` обеспечивает интерактивный доступ к командному процессору контейнера.

Команда: `docker run -it ubuntu`

```
ageeva@ubuntu:~$ sudo docker run -it ubuntu
root@ad72bb58decc:/#
```

Рисунок 2.19 – Запуск контейнера через команду `run`

Далее обновим индекс пакетов (рисунок 2.20).

Команда: `sudo apt-get update`

```
ageeva@ubuntu:~$ docker run -it ubuntu
root@8f22c8ef4090:/# apt -get update
E: Command line option 'g' [from -get] is not understood in combination with the other options.
root@8f22c8ef4090:/# apt-get update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [21.6 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [671 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [177 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [651 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [209 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1062 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [29.6 kB]
Get:16 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [934 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [4301 B]
Fetched 17.2 MB in 26s (656 kB/s)
Reading package lists... Done
root@8f22c8ef4090:/#
```

Рисунок 2.20 – Обновление индекса пакетов для гостевой ОС

Для выполнения работы необходимо установить текстовый редактор Nano (рисунок 2.21).

Nano - текстовый редактор для консоли. Используется в Unix-подобных ОС, распространяется свободно.

Команда: apt install nano

```
root@8f22c8ef4090:/# apt install nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  hunspell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 6 not upgraded.
Need to get 269 kB of archives.
After this operation, 868 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 nano amd64 4.8-1ubuntu1 [269 kB]
Fetched 269 kB in 1s (214 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package nano.
(Reading database ... 4121 files and directories currently installed.)
Preparing to unpack .../nano_4.8-1ubuntu1_amd64.deb ...
Unpacking nano (4.8-1ubuntu1) ...
Setting up nano (4.8-1ubuntu1) ...
update-alternatives: using /bin/nano to provide /usr/bin/editor (editor) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/editor.1.gz because associated file /usr/share/man/man1/nano.1.gz (of link group editor) doesn't exist
update-alternatives: using /bin/nano to provide /usr/bin/pico (pico) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/pico.1.gz because associated file /usr/share/man/man1/nano.1.gz (of link group pico) doesn't exist
root@8f22c8ef4090:/#
```

Рисунок 2.21 - Установка текстового редактора

Для создания текстового файла используем следующую команду (рисунок 2.22).

Команда: `nano «название файла».sh`

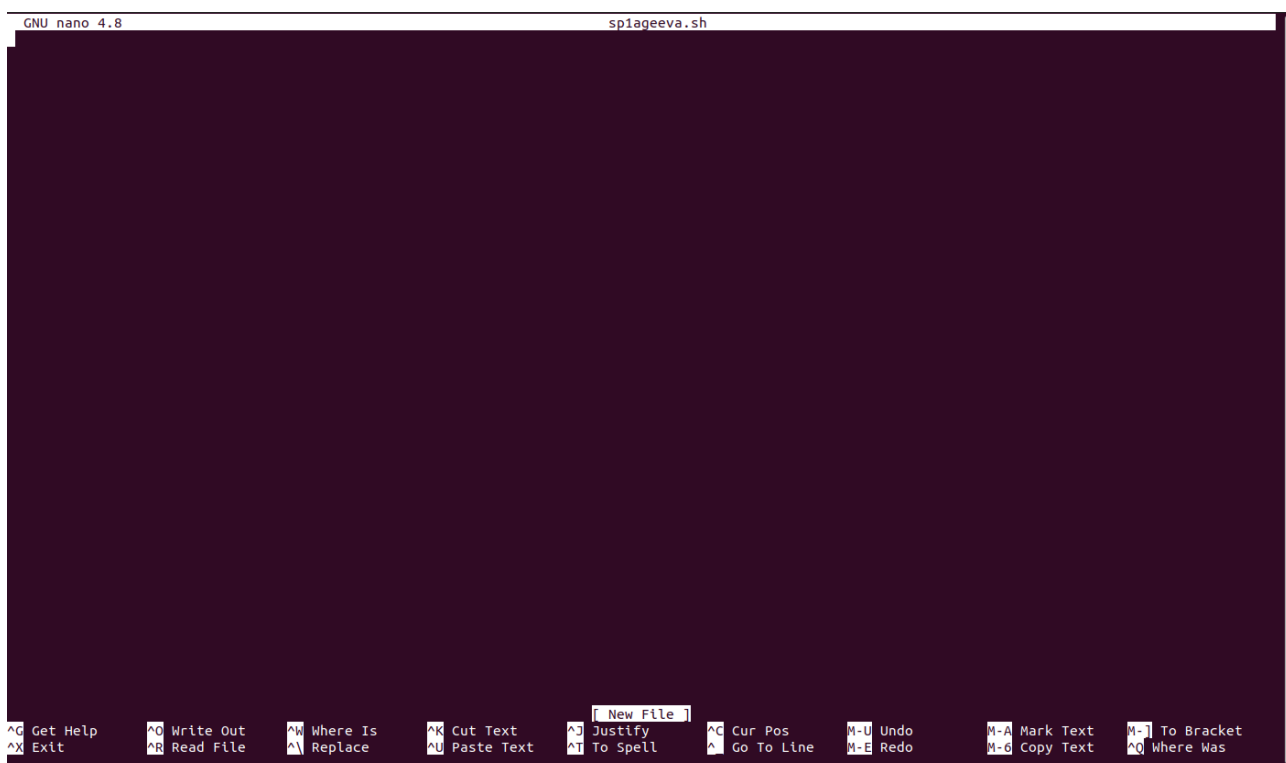


Рисунок 2.22 – Создание файла

Проверим работу текстовых файлов с помощью команды `echo` (рисунок 2.23).



Рисунок 2.23 – Вывод Hello World

Далее нужно проверить созданный файл с помощью команды `ls` (рисунок 2.24).

```
root@8f22c8ef4090:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin splageeva.sh srv sys tmp usr var
root@8f22c8ef4090:/#
```

Рисунок 2.24 – Проверка созданного файла

Необходимо сделать данный файл исполняемым, для этого используем команду `chmod` (рисунок 2.25).

Команда: `chmod ugo+x “название файла”.sh`

```
root@8f22c8ef4090:/# chmod ugo+x splageeva.sh
root@8f22c8ef4090:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin splageeva.sh srv sys tmp usr var
root@8f22c8ef4090:/#
```

Рисунок 2.25 – Установка прав на файл

Проверка как работает пример скрипта (рисунок 2.26).

Команда: `sh “название файла”.sh`

```
root@8f22c8ef4090:/# sh splageeva.sh
Hello World
root@8f22c8ef4090:/#
```

Рисунок 2.26 – Проверка работы скрипта

Чтобы выйти и остановить контейнер, введем команду `exit` (рисунок 2.27).

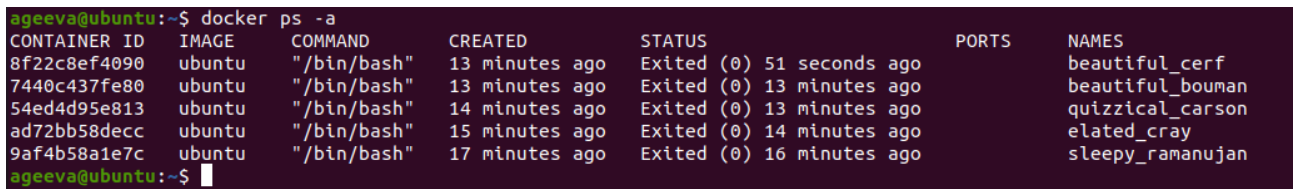
```
root@8f22c8ef4090:/# exit
exit
ageeva@ubuntu:~$
```

Рисунок 2.27 – Выход из контейнера



Чтобы посмотреть список всех контейнеров и их coin id (для подключения и запуска), нужно ввести следующую команду (рисунок 2.28).

Команда: `docker ps -a`



```
ageeva@ubuntu:~$ docker ps -a
```

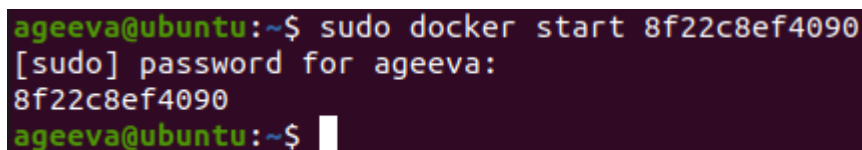
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8f22c8ef4090	ubuntu	"/bin/bash"	13 minutes ago	Exited (0) 51 seconds ago		beautiful_cerf
7440c437fe80	ubuntu	"/bin/bash"	13 minutes ago	Exited (0) 13 minutes ago		beautiful_bouman
54ed4d95e813	ubuntu	"/bin/bash"	14 minutes ago	Exited (0) 13 minutes ago		quizzical_carson
ad72bb58decc	ubuntu	"/bin/bash"	15 minutes ago	Exited (0) 14 minutes ago		elated_cray
9af4b58a1e7c	ubuntu	"/bin/bash"	17 minutes ago	Exited (0) 16 minutes ago		sleepy_ramanujan

```
ageeva@ubuntu:~$
```

Рисунок 2.28 – Список всех контейнеров

Для запуска остановленного контейнера, нужно ввести следующую команду (рисунок 2.29):

Команда: `sudo docker start "coin id"`

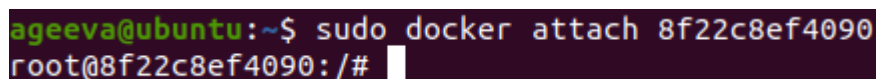


```
ageeva@ubuntu:~$ sudo docker start 8f22c8ef4090
[sudo] password for ageeva:
8f22c8ef4090
ageeva@ubuntu:~$
```

Рисунок 2.29 – Запуск остановленного контейнера

Подключение к существующему контейнеру осуществляется через следующую команду (рисунок 2.30)

Команда: `sudo docker attach "coin id"`



```
ageeva@ubuntu:~$ sudo docker attach 8f22c8ef4090
root@8f22c8ef4090:/#
```

Рисунок 2.30 – Подключение к существующему контейнеру

Далее создадим Dockerfile с помощью текстового редактора nano (рису-

нок 2.31).

Команда: `nano dockerfile`

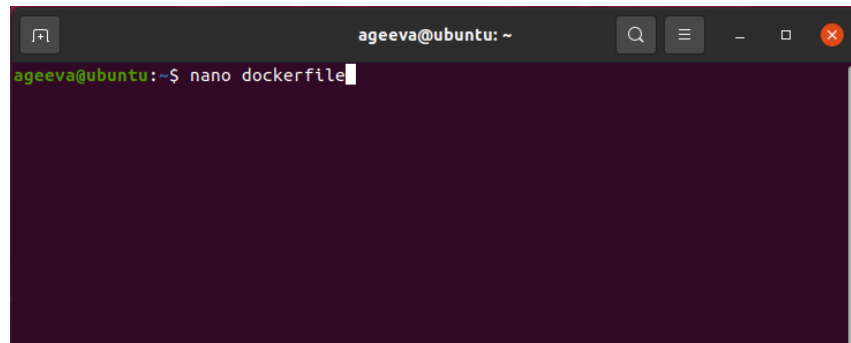


Рисунок 2.31 - Создание Dockerfile

После создания `dockerfile` откроется окно, где мы можем приступать к написанию кода (рисунок 2.32).

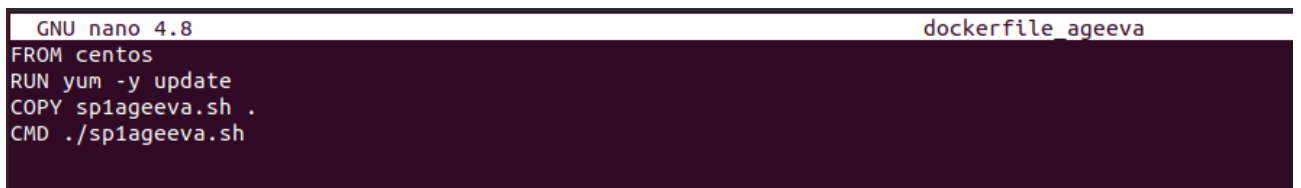


Рисунок 2.32 – Dockerfile

После чего стоит собрать образ (рисунок 2.33).

Команда: `docker build -t container_ageeva -f dockerfile_ageeva .`

Данная команда означает, что мы строим изображение контейнера (незапущенный контейнер) с названием `container_ageeva` по инструкции из файла `dockerfile_ageeva .`, рабочий каталог текущий (т.к. точка).

```

ageevasp@ubuntu:~$ docker build -t container_ageeva -f dockerfile_ageeva .
Sending build context to Docker daemon 8.558MB
Step 1/4 : FROM centos
--> 300e315adb2f
Step 2/4 : RUN yum -y update
--> Using cache
--> bb3a508d211b
Step 3/4 : COPY sp1ageeva.sh .
--> 611283a3edd8
Step 4/4 : CMD ./sp1ageeva.sh
--> Running in efe97451592d
Removing intermediate container efe97451592d
--> 6dc23af0e606
Successfully built 6dc23af0e606
Successfully tagged container_ageeva:latest

```

Рисунок 2.33 – Сборка образа

Команда: `docker run -it «имя файла»` (рисунок 2.34).

```

ageevasp@ubuntu:~$ docker run -it container_ageeva
Package search program.
A program for checking the existence of a package.
Developer: Ageeva Valeria
Enter the package name: focal
Error: No matching Packages to list
focal not installed yet
Last metadata expiration check: 3:30:36 ago on Mon Mar 8 07:47:00 2021.
Error: No matching Packages to list
Failed to search focal in repository
Would you like to continue? [y/N] y
Enter the package name: gcc
Error: No matching Packages to list
gcc not installed yet
Last metadata expiration check: 3:31:52 ago on Mon Mar 8 07:47:00 2021.
Available Packages
gcc.x86_64 8.3.1-5.1.el8 appstream
gcc exist in repository. Would you like to install it? [y/N] k
Invalid value, must be y or n
gcc exist in repository. Would you like to install it? [y/N] n
Would you like to continue? [y/N] n
ageevasp@ubuntu:~$ nano sp1ageeva.sh

```

Рисунок 2.34 – Запуск контейнера и тестирование его работы

```

ageevasp@ubuntu:~$ docker run -it container_ageeva
Package search program.
A program for checking the existence of a package.
Developer: Ageeva Valeria
-----
Enter the package name: pul
Error: No matching Packages to list
pul not installed yet
Last metadata expiration check: 3:34:36 ago on Mon Mar  8 07:47:00 2021.
Error: No matching Packages to list
Failed to search pul in repository
Would you like to continue? [y/N] y
Enter the package name: gcc
Error: No matching Packages to list
gcc not installed yet
Last metadata expiration check: 3:34:46 ago on Mon Mar  8 07:47:00 2021.
Available Packages
gcc.x86_64                                8.3.1-5.1.el8                                appstream
gcc exist in repository. Would you like to install it? [y/N] g
Invalid value, must be y or n
gcc exist in repository. Would you like to install it? [y/N] y
Last metadata expiration check: 3:34:51 ago on Mon Mar  8 07:47:00 2021.
Dependencies resolved.
=====
Package                                Architecture                                Version                                Size                                Repository                                Size
=====
Installing:
gcc                                    x86_64                                    8.3.1-5.1.el8                                23 M                                appstream
Installing dependencies:
cpp                                    x86_64                                    8.3.1-5.1.el8                                10 M                                appstream
glibc-devel                            x86_64                                    2.28-127.el8                                1.0 M                                baseos
glibc-headers                          x86_64                                    2.28-127.el8                                475 k                                baseos
isl                                     x86_64                                    0.16.1-6.el8                                841 k                                appstream
kernel-headers                        x86_64                                    4.18.0-240.15.1.el8_3                       5.6 M                                baseos
libgomp                                x86_64                                    8.3.1-5.1.el8                                204 k                                baseos
libmpc                                 x86_64                                    1.0.2-9.el8                                59 k                                appstream
libpkgconf                             x86_64                                    1.4.2-1.el8                                35 k                                baseos
libxpcrypt-devel                       x86_64                                    4.1.1-4.el8                                25 k                                baseos
pkgconf                                x86_64                                    1.4.2-1.el8                                38 k                                baseos
pkgconf-m4                             noarch                                     1.4.2-1.el8                                17 k                                baseos
pkgconf-pkg-config                     x86_64                                    1.4.2-1.el8                                15 k                                baseos
=====
Transaction Summary
-----
Install 13 Packages

Total download size: 42 M
Installed size: 99 M
Is this ok [y/N]: y
Downloading Packages:
(1/13): isl-0.16.1-6.el8.x86_64.rpm                                1.3 MB/s | 841 kB | 00:00
(2/13): cpp-8.3.1-5.1.el8.x86_64.rpm                                10 MB/s | 10 MB | 00:01
(3/13): libmpc-1.0.2-9.el8.x86_64.rpm                                43 kB/s | 59 kB | 00:01
(4/13): gcc-8.3.1-5.1.el8.x86_64.rpm                                9.3 MB/s | 23 MB | 00:02
(5/13): kernel-headers-4.18.0-240.15.1.el8_3.x86_64.rpm           10 MB/s | 5.6 MB | 00:00
(6/13): libgomp-8.3.1-5.1.el8.x86_64.rpm                           1.6 MB/s | 204 kB | 00:00
(7/13): libpkgconf-1.4.2-1.el8.x86_64.rpm                          546 kB/s | 35 kB | 00:00
(8/13): libxpcrypt-devel-4.1.1-4.el8.x86_64.rpm                   304 kB/s | 25 kB | 00:00
(9/13): pkgconf-1.4.2-1.el8.x86_64.rpm                             444 kB/s | 38 kB | 00:00
(10/13): pkgconf-m4-1.4.2-1.el8.noarch.rpm                         261 kB/s | 17 kB | 00:00
(11/13): pkgconf-pkg-config-1.4.2-1.el8.x86_64.rpm                 252 kB/s | 15 kB | 00:00
(12/13): glibc-devel-2.28-127.el8.x86_64.rpm                      354 kB/s | 1.0 MB | 00:02
(13/13): glibc-headers-2.28-127.el8.x86_64.rpm                    201 kB/s | 475 kB | 00:02
-----
Total                                                                    7.7 MB/s | 42 MB | 00:05
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
Installing : libmpc-1.0.2-9.el8.x86_64                                1/1
Running scriptlet: libmpc-1.0.2-9.el8.x86_64                        1/13
Installing : cpp-8.3.1-5.1.el8.x86_64                                2/13
Running scriptlet: cpp-8.3.1-5.1.el8.x86_64                        2/13
Installing : pkgconf-m4-1.4.2-1.el8.noarch                          3/13
Installing : libpkgconf-1.4.2-1.el8.x86_64                         4/13
Installing : pkgconf-1.4.2-1.el8.x86_64                            5/13
Installing : pkgconf-pkg-config-1.4.2-1.el8.x86_64                 6/13
Installing : libgomp-8.3.1-5.1.el8.x86_64                          7/13
Running scriptlet: libgomp-8.3.1-5.1.el8.x86_64                    7/13
Installing : kernel-headers-4.18.0-240.15.1.el8_3.x86_64           8/13
Running scriptlet: glibc-headers-2.28-127.el8.x86_64              9/13
Installing : glibc-headers-2.28-127.el8.x86_64                    9/13
Installing : libxpcrypt-devel-4.1.1-4.el8.x86_64                  10/13
Installing : glibc-devel-2.28-127.el8.x86_64                      11/13
Running scriptlet: glibc-devel-2.28-127.el8.x86_64                11/13
Installing : isl-0.16.1-6.el8.x86_64                               12/13
Running scriptlet: isl-0.16.1-6.el8.x86_64                         12/13
Installing : gcc-8.3.1-5.1.el8.x86_64                              13/13
Running scriptlet: gcc-8.3.1-5.1.el8.x86_64                        13/13
Verifying : gcc-8.3.1-5.1.el8.x86_64                                1/13
Verifying : gcc-8.3.1-5.1.el8.x86_64                                2/13
Verifying : isl-0.16.1-6.el8.x86_64                                3/13
Verifying : libmpc-1.0.2-9.el8.x86_64                              4/13
Verifying : glibc-devel-2.28-127.el8.x86_64                       5/13
Verifying : glibc-headers-2.28-127.el8.x86_64                     6/13
Verifying : kernel-headers-4.18.0-240.15.1.el8_3.x86_64           7/13
Verifying : libgomp-8.3.1-5.1.el8.x86_64                           8/13
Verifying : libpkgconf-1.4.2-1.el8.x86_64                          9/13
Verifying : libxpcrypt-devel-4.1.1-4.el8.x86_64                  10/13
Verifying : pkgconf-1.4.2-1.el8.x86_64                            11/13
Verifying : pkgconf-m4-1.4.2-1.el8.noarch                          12/13
Verifying : pkgconf-pkg-config-1.4.2-1.el8.x86_64                 13/13
-----
Installed:
cpp-8.3.1-5.1.el8.x86_64                                gcc-8.3.1-5.1.el8.x86_64                                glibc-devel-2.28-127.el8.x86_64                                glibc-headers-2.28-127.el8.x86_64
isl-0.16.1-6.el8.x86_64                                kernel-headers-4.18.0-240.15.1.el8_3.x86_64                libgomp-8.3.1-5.1.el8.x86_64                                libmpc-1.0.2-9.el8.x86_64
libpkgconf-1.4.2-1.el8.x86_64                            libxpcrypt-devel-4.1.1-4.el8.x86_64                        pkgconf-1.4.2-1.el8.x86_64                                pkgconf-m4-1.4.2-1.el8.noarch
pkgconf-pkg-config-1.4.2-1.el8.x86_64

Complete!
gcc exist in repository. Would you like to install it? [y/N] n
Would you like to continue? [y/N] n
ageevasp@ubuntu:~$

```

Рисунок 2.35 – Запуск контейнера и тестирование установки пакета

Рабочий скрипт и Dockerfile загружены на github (ссылка на репозиторий: [https://github.com/7371avs/SP\\_Lab1](https://github.com/7371avs/SP_Lab1)), а также приложены в архиве к отчету.

### 3 Заключение

В ходе лабораторной работы были получены навыки работы со средой контейнеризации ПО Docker и с командным интерпретатором bash. В ходе работы был написан скрипт и Dockerfile для воссоздания контейнера согласно варианту.