

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Отчет
По лабораторной работе №3

Группа: Р3415

Выполнил: Карташев В.С.

Проверил:
к.т.н. преподаватель Белозубов А.В.

Санкт-Петербург, 2025 г.

Оглавление

Оглавление	2
Цели работы	3
Знакомство с Docker	4
Установка Docker	4
Настройка Apache2	5
Настройка MariaDB	7
Настройка NextCloud	8
Настройка phpVirtualBox	9
Настройка окружения с помощью Docker Compose	10
Запуск игры с DockerHub	12
Использованные команды	13
Основы работы с LXC/LXD	14
Установка LXD	14
Создание 3 контейнеров	14
Список, содержащий только IP-адреса контейнеров	15
Список, содержащий только остановленные контейнеры	15
Удаление контейнера MC	15
Список, содержащий только тип контейнера	15
Конфигурация и выполнение команд LXC/LXD	16
Проверка работы LXD	16
Создание и запуск 3 контейнеров	16
Конфигурация каждого контейнера	16
Остановка контейнера Big	16
Изменение ограничения CPU для Big на 2	16
Изменение ограничения RAM для Big на 300 MiB	16
Запуск контейнера Big	17
Отправка утилиты free в контейнер Big	17
Запуск top в Medium и изменение RAM	18
Запуск top в Small и изменение RAM	19
Переименование Small в BigBig	19
Практическое применение LXD	20
Проверка работы LXD	20
Установка ограничения RAM = 500 MiB	20
Установка Apache2 на контейнере Apache	20
Создание страницы с ФИО	21
Получение IP-адреса контейнера Apache	21
Проброс портов для NextCloud	22
Установка NextCloud	22
Установка Midnight Commander на MC	23
Заключение	24

Цели работы

Освоить основы работы с технологиями контейнеризации Docker и LXC/LXD. Изучить создание, настройку и управление контейнерами, сборку образов с помощью Dockerfile, orchestration с использованием Docker Compose, а также практическое применение LXD для развертывания изолированных сред.

Знакомство с Docker

Установка Docker

```
vladimir@lenovo:~/virtualization/lab3$ docker --version
Docker version 28.5.1, build e180ab8
vladimir@lenovo:~/virtualization/lab3$ docker run Hello-Kartashev
docker: invalid reference format: repository name (library/Hello-Kartashev) must be lowercase

Run 'docker run --help' for more information
vladimir@lenovo:~/virtualization/lab3$ docker run hello-kartashev
Unable to find image 'hello-kartashev:latest' locally
docker: Error response from daemon: pull access denied for hello-kartashev, repository does not exist or may require 'docker login': denied: requested access to the resource is denied

Run 'docker run --help' for more information
```

Рисунок 1. Проверка версии Docker

Мы получаем *ошибку*, так как образа *hello-kartashev* не существует как локально, так и в удаленном репозитории образов.

Исправим *ситуацию*:

```
vladimir@lenovo:~/virtualization/lab3/hello$ ls
Dockerfile
vladimir@lenovo:~/virtualization/lab3/hello$ cat Dockerfile
FROM alpine:latest
CMD echo "Hello-Kartashev"

vladimir@lenovo:~/virtualization/lab3/hello$ docker build -t hello-kartashev .
[+] Building 1.2s (5/5) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 86B
=> [internal] load metadata for docker.io/library/alpine:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> CACHED [1/1] FROM docker.io/library/alpine:latest@sha256:4b7ce07002c69e8f3d704a9c5d6fd3053be500b7f1c69fc0d80990c2ad8dd412
=> exporting to image
=> => exporting layers
=> => writing image sha256:65f5bdac4ffca122601d321d40014cd9fdebba17c3f14d1375180e92f8546c86
=> => naming to docker.io/library/hello-kartashev

1 warning found (use docker --debug to expand):
 - JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 4)
vladimir@lenovo:~/virtualization/lab3/hello$ docker run hello-kartashev
Hello-Kartashev
```

Рисунок 2. Создание образа *hello-kartashev*

Настройка Apache2

```
vladimir@lenovo:~/virtualization/lab3/apache$ ls
Dockerfile  html
vladimir@lenovo:~/virtualization/lab3/apache$ cat html/index.html
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Карташев P3415</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      max-width: 800px;
      margin: 50px auto;
      padding: 20px;
      background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
      color: white;
    }
    .card {
      background: rgba(255,255,255,0.1);
      padding: 30px;
      border-radius: 10px;
      backdrop-filter: blur(10px);
    }
    h1 { margin-top: 0; }
  </style>
</head>
<body>
  <div class="card">
    <h1>Информация о студенте</h1>
    <p><strong>ФИО:</strong> Карташев Владимир Сергеевич</p>
    <p><strong>Группа:</strong> P3415</p>
    <p><strong>Лабораторная работа:</strong> №3 - Знакомство с Docker </p>
    <p><strong>Дата:</strong> 2 ноября 2025</p>
  </div>
</body>
</html>
vladimir@lenovo:~/virtualization/lab3/apache$ cat Dockerfile
FROM httpd:2.4-alpine

# internal
WORKDIR /usr/local/apache2/htdocs/

# external into internal
COPY html/index.html /usr/local/apache2/htdocs/

EXPOSE 80
```

Рисунок 3. Настройка Dockerfile для Apache2

Запуск:

```
docker build -t apache-kartashev .
```

```
docker run -d -p 8080:80 --name apache-kartashev-container  
apache-kartashev
```

```
vladimir@lenovo:~/virtualization/lab3/apache$ docker build -t apache-kartashev .
[+] Building 1.8s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 193B
=> [internal] load metadata for docker.io/library/httpd:2.4-alpine
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/2] FROM docker.io/library/httpd:2.4-alpine@sha256:87b2fab7029a0b8aeb2e0fd02651c28fe22c21c5b5a59d6ff5b022791fcd89e
=> [internal] load build context
=> => transferring context: 61B
=> CACHED [2/3] WORKDIR /usr/local/apache2/htdocs/
=> CACHED [3/3] COPY html/index.html /usr/local/apache2/htdocs/
=> exporting to image
=> => exporting layers
=> => writing image sha256:1fe0483ba5e75c6d3a774530a3e79534c93fb225599bc94554994177229d7ac8
=> => naming to docker.io/library/apache-kartashev
vladimir@lenovo:~/virtualization/lab3/apache$ docker run -d -p 8080:80 --name apache-kartashev-container apache-kartashev
0b7d6cfd8d0b6ced946b65231c0897bf153d41a2b4ecd93767749d7e6f21fa
vladimir@lenovo:~/virtualization/lab3/apache$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
0b7d6cfd8d0b   apache-kartashev  "httpd -foreground"     4 seconds ago Up 3 seconds  0.0.0.0:8080->80/tcp, [::]:8080->80/tcp  apache-kartashev-container
```

Рисунок 4. Сборка образа *apache-kartashev*

Перейдем по адресу <http://localhost:8080> на хостовой системе (localhost, так как мы (хостовой компьютер) не находимся в сети докера)

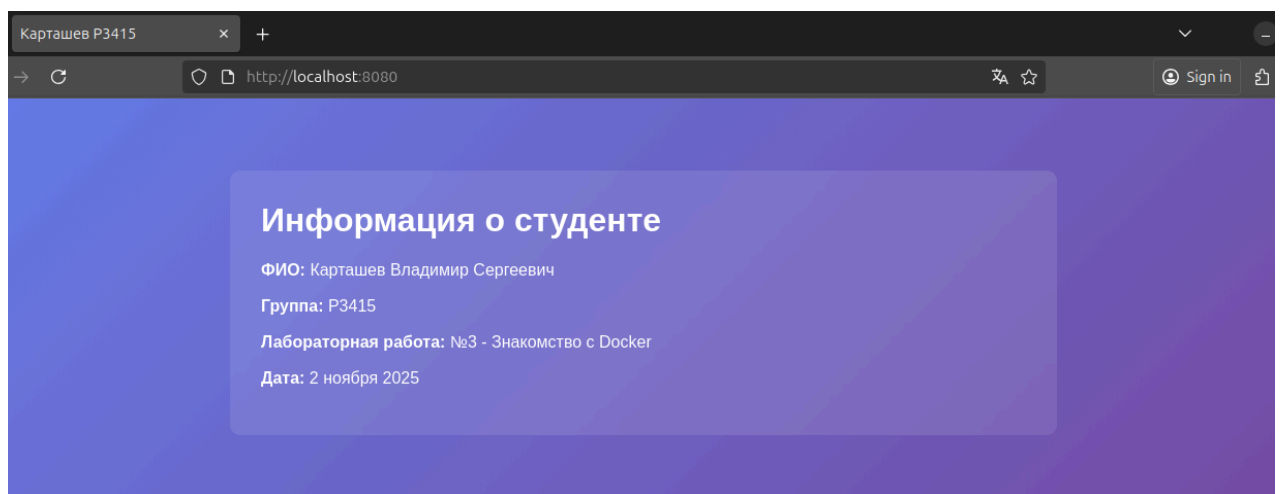


Рисунок 4. Отображение статического сайта, запущенного из Docker контейнера

Настройка MariaDB

Запуск:

```
docker run -d \
  --name mariadb-container \
  -e MYSQL_ROOT_PASSWORD=admin \
  -e MYSQL_DATABASE=mariadb \
  -v
/home/vladimir/virtualization/lab3/mariadb/db-data:/var/lib/m
ysql \
  -p 3306:3306 \
  mariadb:12.1.1-rc
```

```
docker exec -it mariadb-container mariadb -u root -padmin
```

```
CREATE DATABASE IF NOT EXISTS Karatshev;
SHOW DATABASES;
```

```
vladimir@lenovo:~/virtualization/lab3/mariadb$ ls
db-data
vladimir@lenovo:~/virtualization/lab3/mariadb$ docker run -d \
  --name mariadb-container \
  -e MYSQL_ROOT_PASSWORD=admin \
  -e MYSQL_DATABASE=mariadb \
  -v /home/vladimir/virtualization/lab3/mariadb/db-data:/var/lib/mysql \
  -p 3306:3306 \
  mariadb:12.1.1-rc
0e935adfb49c648b6fdbbc7575ef1d2b0431f7ad4051fd1b04cfd0c70bd1c8cf4
vladimir@lenovo:~/virtualization/lab3/mariadb$ docker exec -it mariadb-container mariadb -u root -padmin
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 12.1.1-MariaDB-ubu2404 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database Karatshev;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| Karatshev |
| information_schema |
| mariadb |
| mysql |
| performance_schema |
| sys |
+-----+
6 rows in set (0.001 sec)

MariaDB [(none)]> exit
Bye
vladimir@lenovo:~/virtualization/lab3/mariadb$ ls db-data/
aria_log.000000001  ddl_recovery.log  ibdata1  ibtmp1  mariadb  multi-master.info  performance_schema  tc.log  undo002
aria_log_control  ib_buffer_pool  ib_logfile0  Karatshev  mariadb_upgrade.info  mysql  sys  undo001  undo003
```

Рисунок 5. Настройка MariaDB

В конечном итоге БД отобразилась директорией в директории, проецируемой внутрь контейнера (то есть db-data).

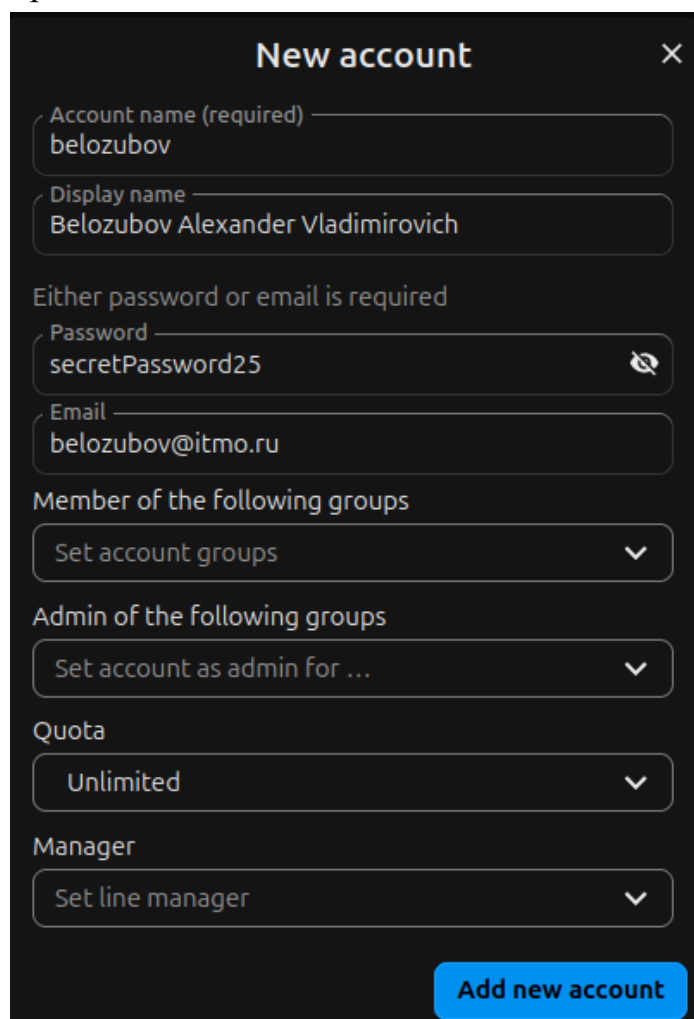
Настройка NextCloud

Запуск:

```
docker run -d \
  --name nextcloud-container \
  -p 8088:80 \
  -v
/home/vladimir/virtualization/lab3/nextcloud:/var/www/html
\
  nextcloud:latest
```

Логин и пароль от NextCloud – admin:admin

Создадим аккаунт преподавателя:



New account ×

Account name (required)
belozubov

Display name
Belozubov Alexander Vladimirovich

Either password or email is required

Password
secretPassword25

Email
belozubov@itmo.ru

Member of the following groups
Set account groups

Admin of the following groups
Set account as admin for ...

Quota
Unlimited

Manager
Set line manager

Add new account

Рисунок 6. Создание аккаунта преподавателя

Настройка phpVirtualBox

Запуск:

```
vboxwebsrv -H 0.0.0.0 -A null -b

docker run -d \
  --name phpvirtualbox \
  --restart=always \
  -p 8089:80 \
  -e SRV1_HOSTPORT=host.docker.internal:18083 \
  -e SRV1_NAME=LocalVBox \
  --add-host=host.docker.internal:host-gateway \
  jazzdd/phpvirtualbox
```

Логин и пароль от phpVirtualBox – admin:admin

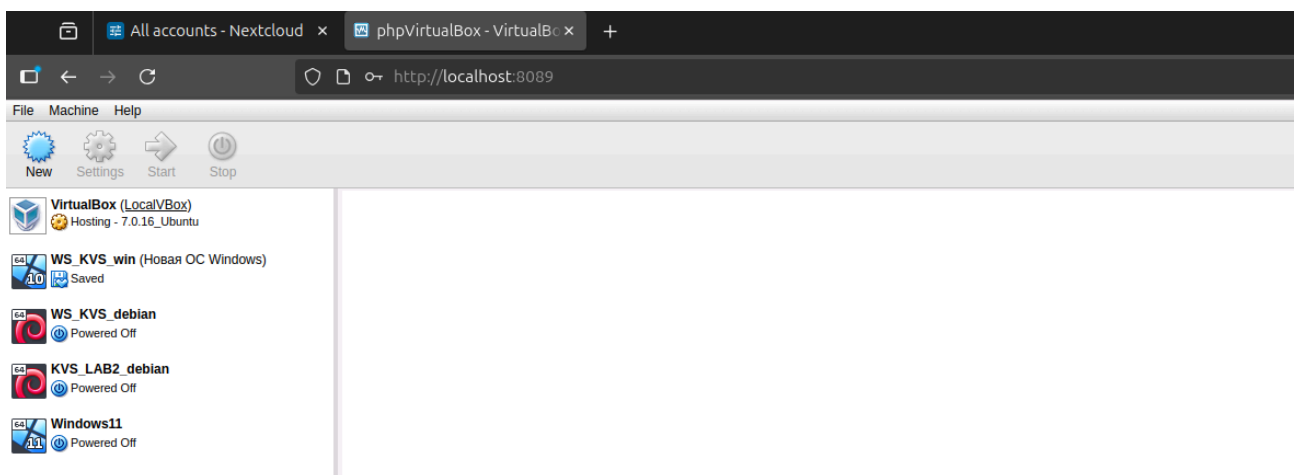


Рисунок 7. Веб-страница управления VirtualBox через phpVirtualBox

Можно увидеть виртуальные машины из Лабораторной 1 и Лабораторной 2

Настройка окружения с помощью Docker Compose

Запуск:

```
docker compose up -d
```

docker-compose.yml

```
services:
  db:
    image: mariadb:12.1.1-rc
    container_name: nextcloud-mariadb
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: admin
      MYSQL_DATABASE: mariadb
    volumes:
      - /home/vladimir/virtualization/lab3/mariadb/nextcloud-data:/var/lib/mysql
    networks:
      - nextcloud-network

  webserver:
    build: ../apache
    container_name: apache-webserver
    restart: always
    ports:
      - "8080:80"
    volumes:
      - /home/vladimir/virtualization/lab3/apache/html:/usr/local/apache2/htdocs
    networks:
      - nextcloud-network

  nextcloud:
    image: nextcloud:latest
    container_name: nextcloud-app
    restart: always
    ports:
      - "2022:80"
    environment:
      MYSQL_HOST: db
      MYSQL_DATABASE: mariadb
      MYSQL_USER: root
      MYSQL_PASSWORD: admin
    volumes:
      - /home/vladimir/virtualization/lab3/nextcloud:/var/www/html
    depends_on:
      - db
      - webserver
    networks:
      - nextcloud-network

networks:
  nextcloud-network:
    driver: bridge
```

Логин и пароль от NextCloud – admin:admin

Судя по содержимому нового контейнера mariadb - инициализация прошла успешно:

```
MariaDB [mariadb]> SHOW TABLES;
+-----+
| Tables_in_mariadb |
+-----+
| oc_accounts        |
| oc_accounts_data   |
| oc_activity        |
| oc_activity_mq     |
| oc_addressbookchanges |
| oc_addressbooks    |
| oc_appconfig       |
| oc_appconfig_ex    |
| oc_authorized_groups |
| oc_authtoken       |
| oc_bruteforce_attempts |
| oc_calendar_invitations |
| oc_calendar_reminders |
| oc_calendar_resources |
| oc_calendar_resources_md |
+-----+
```

Рисунок 8. Таблицы, созданные NextCloud в MariaDB

Загрузка в DockerHub

```
docker login -u colddirol
docker tag nextcloud-compose-webserver
colddirol/apache-nextcloud:v1.0
docker push colddirol/apache-nextcloud:v1.0
```

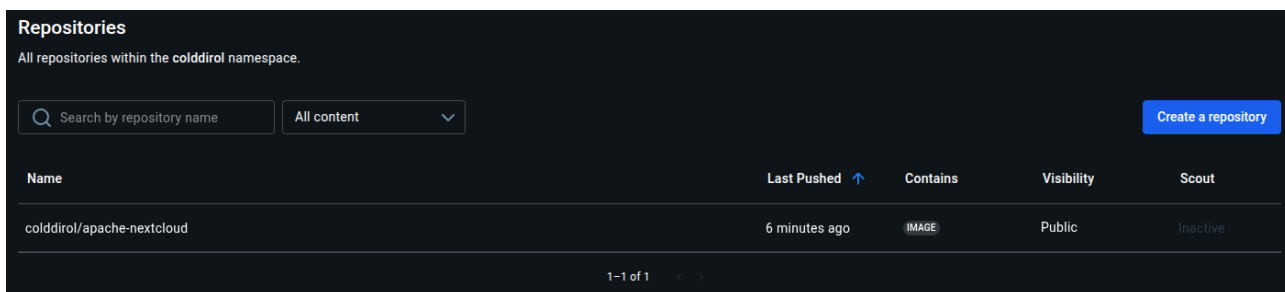


Рисунок 9. Репозиторий colddirol/apache-nextcloud в DockerHub

Запуск игры с DockerHub

Запуск:

```
docker run -d -p 8082:80 devtimnbr/minesweeper
```

Minesweeper



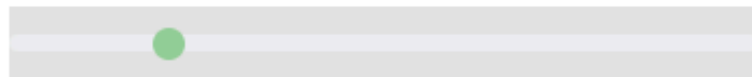
Rows

9

Cols

9

Difficulty:



Lobby List

Reset Game

You win!

Рисунок 10. Игра Minesweeper, запущенная из Docker

Использованные команды

```
# Управление образами
docker images                # Список образов
docker pull image:tag        # Скачать образ
docker build -t name .       # Собрать образ из Dockerfile
docker rmi image_id           # Удалить образ
docker tag source target     # Создать тег образа

# Управление контейнерами
docker ps                    # Список запущенных контейнеров
docker ps -a                 # Список всех контейнеров
docker run [options] image   # Создать и запустить контейнер
docker start container_id     # Запустить контейнер
docker stop container_id     # Остановить контейнер
docker restart container_id  # Перезапустить контейнер
docker rm container_id       # Удалить контейнер
docker exec -it container bash # Выполнить команду в контейнере

# Docker Compose
docker compose up -d         # Запустить проект в фоне
docker compose down          # Остановить и удалить контейнеры
docker compose ps            # Статус сервисов
docker compose logs          # Логи сервисов
docker compose restart       # Перезапустить сервисы

# Информация и логи
docker logs container_id     # Логи контейнера
docker inspect container_id  # Детальная информация
docker stats                 # Статистика ресурсов

# Работа с Docker Hub
docker login                 # Вход в Docker Hub
docker push image:tag        # Загрузить образ
docker pull image:tag        # Скачать образ
docker search keyword         # Поиск образов
```

Основы работы с LXC/LXD

Установка LXD

```
vladimir@lenovo:~/virtualization/lab3/nextcloud-compose$ lxc --version
6.5
```

Рисунок 11. Проверка версии LXC

```
vladimir@lenovo:~/virtualization/lab3/lxd$ sudo lxd init
Would you like to use LXD clustering? (yes/no) [default=no]: no
Do you want to configure a new storage pool? (yes/no) [default=yes]: yes
Name of the new storage pool [default=default]: My_KVS_fs
Name of the storage backend to use (btrfs, powerflex, zfs, ceph, dir, lvm, pure) [default=zfs]: dir
Would you like to connect to a MAAS server? (yes/no) [default=no]: no
Would you like to create a new local network bridge? (yes/no) [default=yes]: yes
What should the new bridge be called? [default=lxdbr0]: lxdbr0
What IPv4 address should be used? (CIDR subnet notation, "auto" or "none") [default=auto]: 10.100.100.1/24
Would you like LXD to NAT IPv4 traffic on your bridge? [default=yes]: yes
What IPv6 address should be used? (CIDR subnet notation, "auto" or "none") [default=auto]: none
Would you like the LXD server to be available over the network? (yes/no) [default=no]: no
Would you like stale cached images to be updated automatically? (yes/no) [default=yes]: yes
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]: no
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc profile list
+-----+-----+-----+
| NAME | DESCRIPTION | USED BY |
+-----+-----+-----+
| default | Default LXD profile | 0 |
+-----+-----+-----+
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc network list
+-----+-----+-----+-----+-----+-----+-----+-----+
| NAME | TYPE | MANAGED | IPV4 | IPV6 | DESCRIPTION | USED BY | STATE |
+-----+-----+-----+-----+-----+-----+-----+-----+
| docker0 | bridge | NO | | | | 0 | |
+-----+-----+-----+-----+-----+-----+-----+-----+
| enp2s0 | physical | NO | | | | 0 | |
+-----+-----+-----+-----+-----+-----+-----+-----+
| lxdbr0 | bridge | YES | 10.100.100.1/24 | none | | 1 | CREATED |
+-----+-----+-----+-----+-----+-----+-----+-----+
| wlp4s0 | physical | NO | | | | 0 | |
+-----+-----+-----+-----+-----+-----+-----+-----+
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc storage list
+-----+-----+-----+-----+-----+-----+-----+-----+
| NAME | DRIVER | SOURCE | DESCRIPTION | USED BY | STATE |
+-----+-----+-----+-----+-----+-----+-----+-----+
| My_KVS_fs | dir | /var/snap/lxd/common/lxd/storage-pools/My_KVS_fs | | 1 | CREATED |
+-----+-----+-----+-----+-----+-----+-----+-----+
| default | dir | /var/snap/lxd/common/lxd/storage-pools/default | | 0 | CREATED |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Рисунок 12. Отображение профилей, сетей и хранилищ, созданных в LXC

Создание 3 контейнеров

```
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc launch ubuntu:22.04 Apache
Launching Apache
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc launch ubuntu:22.04 Chrome
Launching Chrome
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc launch ubuntu:22.04 MC
Launching MC
```

Рисунок 13. Создание контейнеров Apache, Chrome, MC

```
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc list
+-----+-----+-----+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS |
+-----+-----+-----+-----+-----+-----+-----+
| Apache | RUNNING | 10.100.100.25 (eth0) | | CONTAINER | 0 |
+-----+-----+-----+-----+-----+-----+-----+
| Chrome | RUNNING | 10.100.100.239 (eth0) | | CONTAINER | 0 |
+-----+-----+-----+-----+-----+-----+-----+
| MC | RUNNING | 10.100.100.249 (eth0) | | CONTAINER | 0 |
+-----+-----+-----+-----+-----+-----+-----+
```

Рисунок 14. Отображение контейнеров в LXC

Список, содержащий только IP-адреса контейнеров

Команда:

```
lxc list -c n4 --format compact
```

```
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc list -c n4 --format compact
NAME      IPV4
Apache    10.100.100.25 (eth0)
Chrome    10.100.100.239 (eth0)
MC         10.100.100.249 (eth0)
```

Рисунок 15. Отображение IP-адресов контейнеров

Список, содержащий только остановленные контейнеры

Команда:

```
lxc list | grep STOPPED
```

Удаление контейнера MC

Команда:

```
lxc stop MC
lxc delete MC
```

Список, содержащий только тип контейнера

Команда:

```
lxc list -c nt --format compact
```

```
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc list -c nt --format compact
NAME      TYPE
Apache    CONTAINER
Chrome    CONTAINER
```

Рисунок 16. Отображение типов контейнеров

Конфигурация и выполнение команд LXC/LXD

Проверка работы LXD

```
vladimir@lenovo:~/virtualization/lab3/nextcloud-compose$ lxc --version  
6.5
```

Рисунок 17. Проверка версии LXC

Создание и запуск 3 контейнеров

```
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc launch ubuntu:22.04 Big  
Launching Big  
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc launch ubuntu:22.04 Medium  
Launching Medium  
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc launch ubuntu:22.04 Small  
Launching Small
```

Рисунок 18. Запуск разных контейнеров

Конфигурация каждого контейнера

Команда:

```
lxc info Big  
lxc info Medium  
lxc info Small
```

Остановка контейнера Big

Команда:

```
lxc stop Big
```

Изменение ограничения CPU для Big на 2

Команда:

```
lxc config set Big limits.cpu 2
```

Изменение ограничения RAM для Big на 300 MiB

Команда:

```
lxc config set Big limits.memory 300MiB
```


Запуск контейнера Big

Команда:

```
lxc start Big
```

Отправка утилиты free в контейнер Big

Команда:

```
lxc exec Big -- free -h
```

```
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc exec Big -- free -h
total        used        free        shared  buff/cache   available
Mem:        300Mi       100Mi       187Mi         0.0Ki        11Mi       199Mi
Swap:        300Mi         0B        300Mi
```

Рисунок 19. Отправка утилиты free в контейнер Big

Запуск top в Medium и изменение RAM

Терминал 1:

```
lxc exec Medium -- top
```

Терминал 2:

```
lxc config set Medium limits.memory 200MiB
```

```
top - 23:30:19 up 11 min, 0 users, load average: 4.89, 6.12, 4.78
Tasks: 22 total, 2 running, 20 sleeping, 0 stopped, 0 zombie
%Cpu(s): 30.2 us, 7.0 sy, 0.0 ni, 59.4 id, 3.3 wa, 0.0 hi, 0.1 si, 0.0 st
MiB Mem : 27864.2 total, 27726.8 free, 110.8 used, 26.6 buff/cache
MiB Swap: 8192.0 total, 8192.0 free, 0.0 used, 27753.1 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
33722	root	20	0	2286888	56472	22488	S	40.0	0.2	0:03.63	snappd
346	root	20	0	41300	27656	14132	S	0.0	0.1	0:00.20	python3
341	root	20	0	1759420	25628	16016	S	0.0	0.1	0:00.76	snap
370	root	20	0	110140	21716	13780	S	0.0	0.1	0:00.06	unattended-upgr
339	root	20	0	33072	19444	10612	S	0.0	0.1	0:00.06	networkd-dispat
302	systemd+	20	0	26332	14336	9352	S	0.0	0.1	0:00.07	systemd-resolve
58	root	20	0	31344	14116	13348	S	0.0	0.0	0:00.55	systemd-journal
1	root	20	0	167280	12380	8412	R	8.7	0.0	0:27.17	systemd
362	root	20	0	15440	9356	7820	S	0.0	0.0	0:00.00	sshd
300	systemd+	20	0	16128	8312	7416	S	0.0	0.0	0:00.02	systemd-network
396	root	20	0	235468	7908	6972	S	0.0	0.0	0:00.01	polkitd
344	root	20	0	15020	7000	6232	S	0.3	0.0	0:01.10	systemd-logind
98	root	20	0	11100	5856	4576	S	0.0	0.0	0:00.09	systemd-udev
33916	root	20	0	452180	5712	1448	S	3.3	0.0	0:00.10	snappd
340	syslog	20	0	152768	5676	4652	S	0.0	0.0	0:00.16	rsyslogd
335	message+	20	0	8592	4680	4168	S	0.7	0.0	0:02.77	dbus-daemon
34407	root	20	0	11124	3976	3720	S	0.0	0.0	0:00.00	systemctl
33717	root	20	0	10784	3784	3272	R	0.3	0.0	0:00.02	top
333	root	20	0	7288	2872	2616	S	0.0	0.0	0:00.00	cron
365	root	20	0	6220	2316	2188	S	0.0	0.0	0:00.00	agetty
34040	root	20	0	227888	1832	1448	S	0.0	0.0	0:00.00	snappd
1233	root	20	0	302652	1428	1300	S	0.0	0.0	0:00.98	snappd

Рисунок 20. Состояние процессов в контейнере Medium до установки лимита

```
top - 23:30:57 up 11 min, 0 users, load average: 6.81, 6.53, 4.97
Tasks: 20 total, 2 running, 18 sleeping, 0 stopped, 0 zombie
%Cpu(s): 16.8 us, 6.3 sy, 0.0 ni, 74.6 id, 2.3 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 200.0 total, 83.7 free, 95.4 used, 20.9 buff/cache
MiB Swap: 200.0 total, 200.0 free, 0.0 used, 104.5 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
36858	root	20	0	2064924	45688	21624	S	27.7	22.3	0:00.23	snappd
346	root	20	0	41300	27656	14132	S	0.0	13.5	0:00.20	python3
341	root	20	0	1759420	26012	16016	S	1.2	12.7	0:00.81	snap
370	root	20	0	110140	21716	13780	S	0.0	10.6	0:00.06	unattended-upgr
339	root	20	0	33072	19444	10612	S	0.0	9.5	0:00.06	networkd-dispat
58	root	20	0	39536	14864	14096	S	0.0	7.3	0:00.59	systemd-journal
302	systemd+	20	0	26332	14336	9352	S	0.0	7.0	0:00.07	systemd-resolve
1	root	20	0	167280	12380	8412	S	2.4	6.0	0:29.20	systemd
362	root	20	0	15440	9356	7820	S	0.0	4.6	0:00.00	sshd
300	systemd+	20	0	16128	8312	7416	S	0.0	4.1	0:00.02	systemd-network
396	root	20	0	235468	7908	6972	S	0.0	3.9	0:00.01	polkitd
344	root	20	0	15020	7000	6232	S	0.0	3.4	0:01.17	systemd-logind
98	root	20	0	11100	5856	4576	S	0.0	2.9	0:00.09	systemd-udev
340	syslog	20	0	152768	5676	4652	S	0.0	2.8	0:00.18	rsyslogd
36927	root	20	0	11604	5568	4032	R	9.6	2.7	0:00.08	udevadm
335	message+	20	0	8592	4680	4168	S	0.0	2.3	0:02.97	dbus-daemon
33717	root	20	0	10784	3784	3272	R	0.0	1.8	0:00.03	top
333	root	20	0	7288	2872	2616	S	0.0	1.4	0:00.00	cron
365	root	20	0	6220	2316	2188	S	0.0	1.1	0:00.00	agetty
1233	root	20	0	302652	1428	1300	S	0.0	0.7	0:01.05	snappd

Рисунок 21. Состояние процессов в контейнере Medium после установки лимита

С 27.8 Гб суммарное использование памяти сократилось до 200 Мб

Запуск top в Small и изменение RAM

Терминал 1:

```
lxc exec Small -- top
```

Терминал 2:

```
lxc config set Small limits.memory 150MiB
# wait
lxc config set Small limits.memory 800MiB
```

```
top - 23:34:47 up 15 min, 0 users, load average: 4.79, 5.99, 5.12
Tasks: 18 total, 1 running, 17 sleeping, 0 stopped, 0 zombie
%Cpu(s): 7.1 us, 2.5 sy, 0.0 ni, 89.8 id, 0.6 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 27864.2 total, 27773.7 free, 68.0 used, 22.5 buff/cache
MiB Swap: 8192.0 total, 8192.0 free, 0.0 used, 27796.1 avail Mem
```

Рисунок 22. Состояние процессов в контейнере Small до установки лимита

```
top - 23:35:05 up 15 min, 0 users, load average: 5.13, 6.01, 5.14
Tasks: 19 total, 1 running, 18 sleeping, 0 stopped, 0 zombie
%Cpu(s): 32.8 us, 6.6 sy, 0.0 ni, 58.0 id, 2.6 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 150.0 total, 20.2 free, 107.0 used, 22.8 buff/cache
MiB Swap: 150.0 total, 150.0 free, 0.0 used, 42.8 avail Mem
```

Рисунок 23. Состояние процессов в контейнере Small после установки лимита в 150 Мб

```
top - 23:35:21 up 16 min, 0 users, load average: 5.31, 6.01, 5.15
Tasks: 19 total, 1 running, 18 sleeping, 0 stopped, 0 zombie
%Cpu(s): 17.3 us, 3.7 sy, 0.0 ni, 76.5 id, 2.5 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 800.0 total, 733.1 free, 53.0 used, 13.9 buff/cache
MiB Swap: 800.0 total, 760.1 free, 39.9 used, 746.9 avail Mem
```

Рисунок 24. Состояние процессов в контейнере Small после установки лимита в 800 Мб

С 27.8 Гб суммарное использование памяти сократилось до 150 Мб, затем возросло до 800 Мб

Переименование Small в BigBig

Команда:

```
lxc stop Small
lxc rename Small BigBig
lxc list
```

```
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc stop Small
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc rename Small BigBig
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc list
+-----+-----+-----+-----+-----+-----+
| NAME   | STATE | IPV4   | IPV6   | TYPE   | SNAPSHOTS |
+-----+-----+-----+-----+-----+-----+
| Apache | RUNNING | 10.100.100.25 (eth0) | | CONTAINER | 0 |
+-----+-----+-----+-----+-----+-----+
| Big    | RUNNING | 10.100.100.148 (eth0) | | CONTAINER | 0 |
+-----+-----+-----+-----+-----+-----+
| BigBig | STOPPED | | | CONTAINER | 0 |
+-----+-----+-----+-----+-----+-----+
| Chrome | RUNNING | 10.100.100.239 (eth0) | | CONTAINER | 0 |
+-----+-----+-----+-----+-----+-----+
| Medium | RUNNING | 10.100.100.133 (eth0) | | CONTAINER | 0 |
+-----+-----+-----+-----+-----+-----+
vladimir@lenovo:~/virtualization/lab3/lxd$
```

Рисунок 25. Отображение списка контейнеров после переименования контейнера Small в BigBig

Практическое применение LXD

Проверка работы LXD

```
vladimir@lenovo:~/virtualization/lab3/nextcloud-compose$ lxc --version  
6.5
```

Рисунок 26. Проверка версии LXC

```
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc launch ubuntu:22.04 Apache  
Launching Apache  
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc launch ubuntu:22.04 NextCloud  
Launching NextCloud  
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc launch ubuntu:22.04 MC  
Launching MC  
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc list
```

NAME	STATE	IPV4	IPV6	TYPE	SNAPSHOTS
Apache	RUNNING	10.100.100.237 (eth0)		CONTAINER	0
MC	RUNNING	10.100.100.14 (eth0)		CONTAINER	0
NextCloud	RUNNING	10.100.100.40 (eth0)		CONTAINER	0

Рисунок 27. Запуск разных контейнеров

Установка ограничения RAM = 500 MiB

Команда:

```
lxc config set Apache limits.memory 500MiB  
lxc config set NextCloud limits.memory 500MiB  
lxc config set MC limits.memory 500MiB
```

Установка Apache2 на контейнере Apache

Команда:

```
lxc exec Apache -- bash -c "apt update && apt install -y apache2"
```

Создание страницы с ФИО

Команда:

```
lxc exec Apache -- bash <<'EOF'
cat > /var/www/html/index.html <<'HTML'
<html>
<head>
    <meta charset="UTF-8">
</head>
<body>
    <h1>Карташев Владимир Сергеевич</h1>
</body>
</html>
HTML
EOF
```

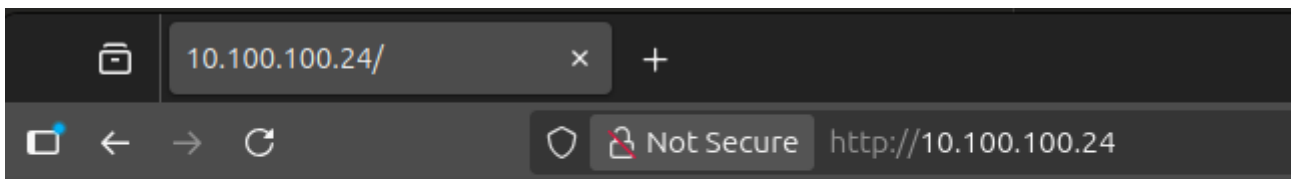
Получение IP-адреса контейнера Apache

Команда:

```
lxc list Apache -c 4
```

```
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc list Apache -c 4
+-----+
|          IPV4          |
+-----+
| 10.100.100.24 (eth0) |
+-----+
```

Рисунок 28. Отображение IP-адреса контейнера Apache



Карташев Владимир Сергеевич

Рисунок 29. Отображение статической веб-страницы из контейнера Apache

Конфигурация портов для NextCloud

Команда:

```
lxc config device add NextCloud myport80 proxy  
listen=tcp:0.0.0.0:8080 connect=tcp:127.0.0.1:80
```

```
vladimir@lenovo:~/virtualization/lab3/lxd$ lxc config device add NextCloud myport80 proxy  
listen=tcp:0.0.0.0:8080 connect=tcp:127.0.0.1:80  
Device myport80 added to NextCloud
```

Рисунок 30. Конфигурация портов для NextCloud

Установка NextCloud

Команда:

```
lxc exec NextCloud -- bash -c "apt update && apt install -y snapd"  
lxc exec NextCloud -- bash -c "snap install nextcloud"
```

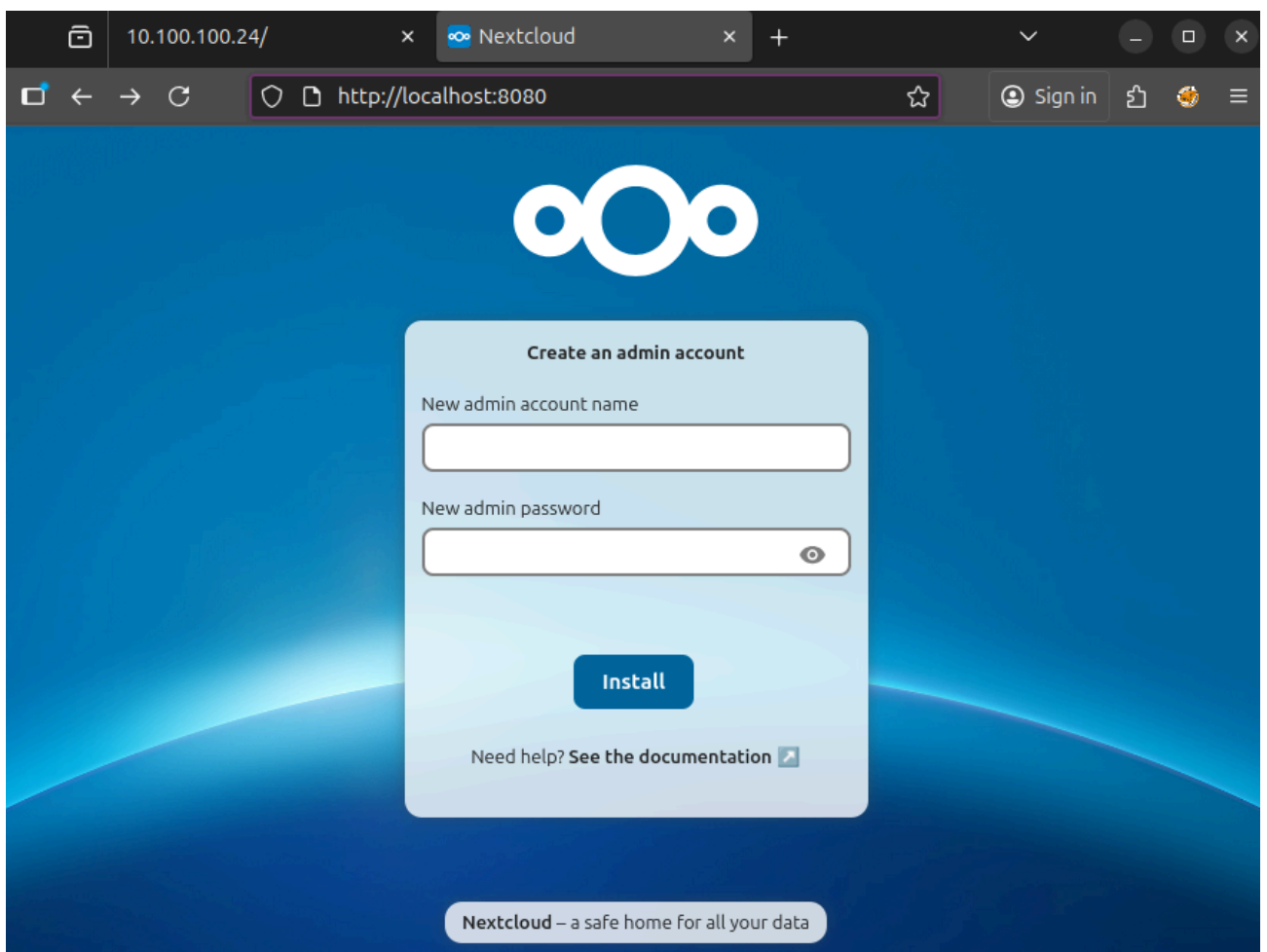


Рисунок 30. Веб-страница с окном входа из контейнера NextCloud

Установка Midnight Commander на MC

Команда:

```
lxc exec MC -- bash -c "apt update && apt install -y mc"
lxc exec MC -- mc
```

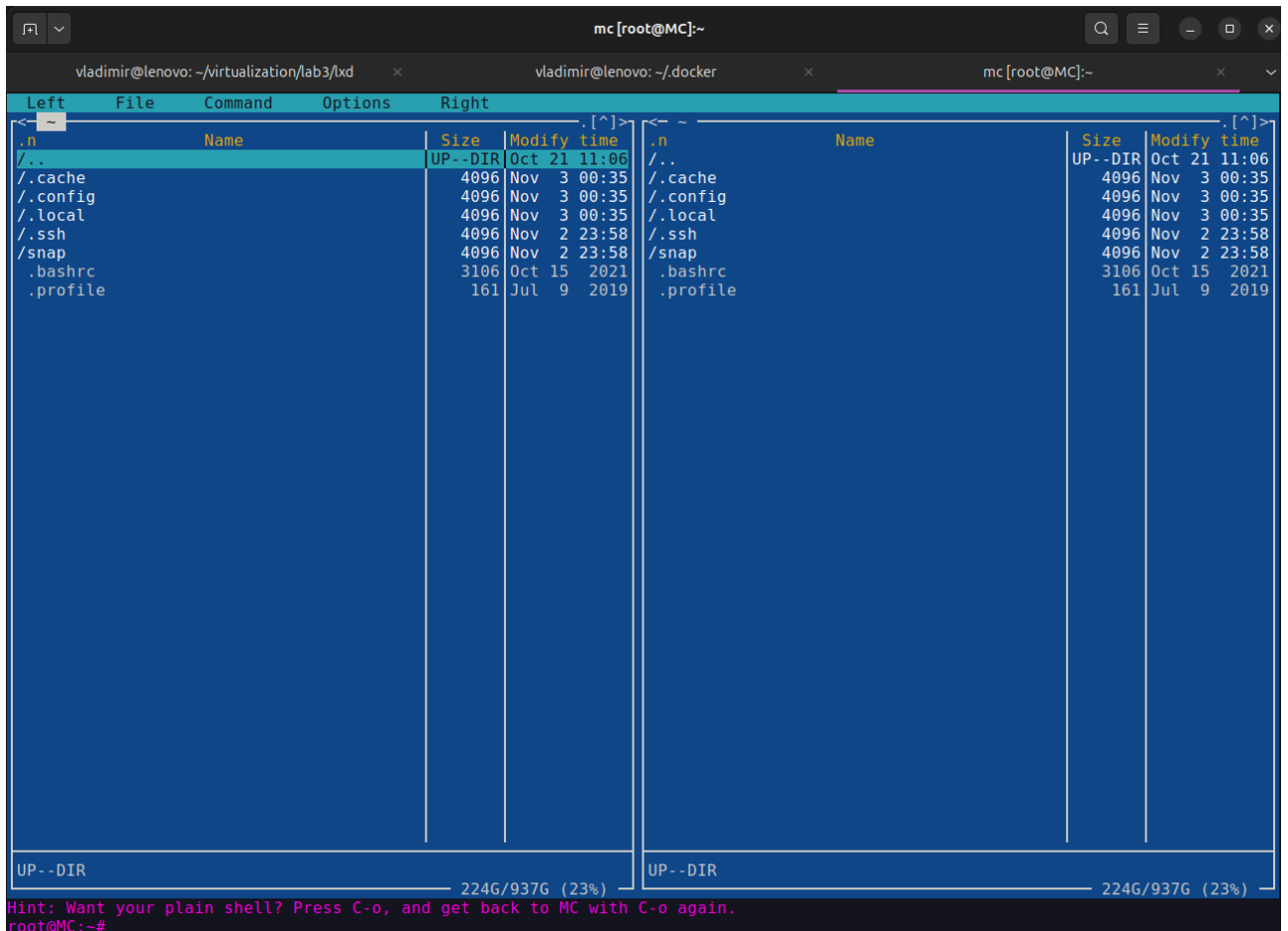


Рисунок 31. Веб-страница с окном входа из контейнера NextCloud

Заключение

В ходе лабораторной работы были успешно установлены и настроены Docker и LXC/LXD. Созданы и запущены контейнеры с различными приложениями (Apache, MariaDB, NextCloud), настроены их сетевые параметры и тома. Освоены основные команды управления контейнерами и образами, сборка проектов через Dockerfile и Docker Compose. В части LXD изучено создание и конфигурация контейнеров, управление ресурсами и сетевыми настройками. Работа продемонстрировала эффективность контейнеризации для развертывания и управления приложениями в изолированных средах.