

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

Факультет программной инженерии и компьютерной техники

Лабораторная работа 5

Знакомство с LXC и LDC

Группа: P34102

Выполнил:

Лапин А.А.

Юнусов Р.Э.

Проверил:

к.т.н. преподаватель Белозубов А.В.

Санкт-Петербург

2024г.

Оглавление

Оглавление.....	2
Введение	3
Описание работы и инструментов.....	3
Цели и задачи	3
Задачи	4
Часть 1. Установка Proxmox	5
Цель	5
Текст задания.....	5
Выполнение	5
Часть 2. Конфигурация и выполнение команд LXC/LXD	15
Цель	15
Текст задания.....	15
Выполнение	15
Часть 3. Практическое применение LXD.	25
Цель	25
Текст задания.....	25
Выполнение	25
Заключение	33
Список литературы	34

Введение

Данная лабораторная работа посвящена изучению контейнерных систем LXD и LXC. LXC представляет собой систему контейнеризации, позволяющая запускать несколько изолированных экземпляров операционной системы Linux на одном узле. LXC не использует виртуальные машины, а создаёт виртуальное окружение с собственным пространством процессов и сетевым стеком; все экземпляры LXC используют один экземпляр ядра операционной системы.

В ходе работы будет представлена информация по установке системы, базовой работой и методами конфигурации контейнер, также мы запустим несколько программ из контейнеров, изолируя их от нашей системы.

Описание работы и инструментов

Используемое программное обеспечение:

1. Virtualbox с установленной машиной на ubuntu 24.04
2. Apache2
3. NetCloud
4. Mozilla Firefox
5. MidnightCommander

Цели и задачи

Получение практических навыков работы с системой контейнеров LXC, включая установку системы, создание и конфигурацию контейнеров, запуск приложений.

Задачи

1. Основы работы с LXC/LXD:

- Установка LXD
- Инициализация хранилища
- Создание и запуск нескольких контейнеров
- Получить списки работающих контейнеров с фильтрами

2. Конфигурация и выполнение команд LXC/LXD:

- Создать несколько контейнеров
- Вывести информацию по контейнерам
- Настроить контейнеры с разными ограничениями
- Запуск утилит в контейнере
- Переименование контейнеров

3. Практическое применение LXD:

- Создать 4 контейнера с ограничением по памяти в 500 МиБ
- Установить и запустить Apache внутри контейнера
- Установить и запустить Nextcloud внутри контейнера
- Установить и запустить Mozilla Firefox в графическом режиме внутри контейнера
- Установить и запустить MidnightCommander в графическом режиме внутри контейнера

Часть 1. Установка LXD

Цель

Познакомиться с установкой LXD.

Текст задания

1. Установить LXD при помощи
2. Проверить работу LXD командой
3. Выполнить инициализацию LXD командой `lxd init` со следующими параметрами:
 - Кластеринг-нет
 - Название хранилища-Mu_ФИО_fs
 - Размер 5GB
 - MAAS-Нет
 - Настроить сеть –Сетевой мост
 - Включить-NAT
4. Получить конфигурации компонентов lxd:
 - `lxc profile list`
 - `lxc network list`
 - `lxc storage list`
5. Создать 3 контейнера с названиями: Apache, Chrome, MC
6. Запустить контейнеры Apache и MC.
7. Получить список контейнеров.
8. Вывести список, содержащий только IP-адреса контейнеров.
9. Вывести список, содержащий только остановленные контейнеры.
10. Удалить контейнер MC.
11. Получить список, содержащий только тип контейнеров.

Выполнение

6. Установка

1. <code>sudo snap install lxd</code>

7. Проверим работу системы командами

2. `lxc list`
3. `lxc --version`

```
roman@roman-VirtualBox:~$ lxc --version
5.21.2 LTS
roman@roman-VirtualBox:~$ sudo lxc list
+-----+-----+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS |
+-----+-----+-----+-----+-----+-----+
roman@roman-VirtualBox:~$
```

Рисунок 1. Информация о Lxd

8. Сконфигурируем LXC через

4. `sudo lxc init`

Для конфигурации нужно было ответить на список вопросов касемо системы

5. `would you like to use LXD clustering? (yes/no) [default=no]: no`

Этот параметр позволяет включить кластеризацию для LXD, что полезно, если вы планируете управлять несколькими экземплярами LXD вместе. По умолчанию установлено "нет".

6. `Do you want to configure a new storage pool? (yes/no) [default=yes]: yes`

Этот параметр спрашивает, хотите ли вы создать новый пул хранения. Пул хранения необходим для управления хранилищем ваших контейнеров. По умолчанию ответ "да".

7. `Name of the new storage pool [default=default]: my_yunusov_fs`

Вы можете указать собственное название для вашего пула хранения. Если оставить поле пустым, будет использовано название по умолчанию — "default".

8. `Name of the storage backend to use (powerflex, zfs, btrfs, ceph, dir, lvm) [default=zfs]: zfs`

Эта опция позволяет выбрать тип бэкенда для хранения, такой как ZFS, Btrfs, LVM и т.д. По умолчанию используется ZFS, который популярен благодаря своим продвинутым функциям.

9. Create a new ZFS pool? (yes/no) [default=yes]: yes

Если вы выбрали ZFS в качестве бэкенда, этот параметр позволяет решить, хотите ли вы создать новый пул ZFS. По умолчанию ответ "да".

10. would you like to use an existing empty block device (e.g. a disk or partition)? (yes/no) [default=no]: no

Этот параметр спрашивает, хотите ли вы использовать существующий диск или раздел для вашего пула хранения. По умолчанию ответ "нет".

11. Size in GiB of the new loop device (1GiB minimum) [default=7GiB]: 5GiB

Если создается устройство в виде файла (loop device), здесь указывается его размер. Минимальный размер — 1 GiB, по умолчанию — 7 GiB.

12. would you like to connect to a MAAS server? (yes/no) [default=no]: no

Эта опция определяет, хотите ли вы подключиться к серверу Metal as a Service (MAAS) для управления физическими машинами. По умолчанию ответ "нет".

С помощью MAAS можно настроить автоматическую установку любых подготовленных образов систем на голое железо

13. would you like to create a new local network bridge? (yes/no) [default=yes]: yes

Эта опция позволяет создать сетевой мост для ваших контейнеров, позволяя им общаться друг с другом и с внешним миром. По умолчанию ответ "да".

14. what should the new bridge be called? [default=lxdb0]:

Вы можете указать название для сетевого моста; если оставить поле пустым, будет использовано название по умолчанию — "lxdb0".

15. what IPv4 address should be used? (CIDR subnet notation, "auto" or "none")
[default=auto]: 10.200.0.1/24

Здесь вы указываете IPv4 адрес в формате CIDR для мостовой сети. По умолчанию установлено "auto", что автоматически назначает адрес.

16. would you like LXD to NAT IPv4 traffic on your bridge? [default=yes]: yes

Эта опция включает NAT (преобразование сетевых адресов) для IPv4 трафика на мосту, позволяя контейнерам получать доступ к внешним сетям. По умолчанию ответ "да".

17. what IPv6 address should be used? (CIDR subnet notation, "auto" or "none")
[default=auto]: none

Аналогично IPv4, этот параметр спрашивает о наличии IPv6 адреса в формате CIDR. По умолчанию установлено "auto", но можно выбрать "none", если он не нужен.

18. would you like the LXD server to be available over the network? (yes/no)
[default=no]: yes

Эта опция позволяет обеспечить доступ к серверу LXD извне. По умолчанию установлено "нет".

19. Address to bind LXD to (not including port) [default=all]: all

Укажите, к какому адресу должен привязываться LXD (например, ко всем интерфейсам). По умолчанию привязка осуществляется ко всем доступным адресам.

20. Port to bind LXD to [default=8443]: 8443

Этот параметр задает номер порта, на котором LXD будет слушать подключения; порт по умолчанию — 8443.

21. would you like stale cached images to be updated automatically? (yes/no)
[default=yes]: yes

Эта опция определяет, должен ли LXD автоматически обновлять устаревшие кэшированные образы. По умолчанию установлен ответ "да".


```
22. would you like a YAML "lxd init" preseed to be printed? (yes/no)
[default=no]: yes
```

Эта опция позволяет сгенерировать YAML файл конфигурации, который можно использовать позже для автоматизированных настроек. По умолчанию ответ "нет", но вы можете выбрать "да", если это необходимо.

Чуть подробнее разберем типы хранилищ.

ZFS

Описание: Мощная файловая система с функциями управления томами.

Преимущества: Поддержка снапшотов, высокая производительность и надежность.

Btrfs

Описание: Современная файловая система с возможностями управления массивами.

Преимущества: Снапшоты, динамическое управление пространством, поддержка подтомов.

LVM

Описание: Менеджер логических томов для управления физическими дисками.

Преимущества: Гибкость в управлении пространством, возможность создания снимков.

Ceph

Описание: Распределенное хранилище с высокой отказоустойчивостью.

Преимущества: Масштабируемость, поддержка блочных и файловых хранилищ.

Dir

Описание: Использование обычных директорий как хранилища.

Преимущества: Простота настройки и использования, подходит для небольших проектов.

9. Итогом инициализации стал конфиг

```
networks:
- config:
  ipv4.address: 10.200.0.1/24
  ipv4.nat: "true"
  ipv6.address: none
  description: ""
  name: lxdbr0
  type: ""
  project: default
storage_pools:
- config:
  size: 5GiB
  description: ""
  name: My_Yunusov_fs
  driver: zfs
storage_volumes: []
profiles:
- config: {}
  description: ""
  devices:
    eth0:
      name: eth0
      network: lxdbr0
      type: nic
    root:
      path: /
      pool: My_Yunusov_fs
      type: disk
  name: default
projects: []
cluster: null
```

Рисунок 2. Конфиг lxd

10. Получить конфигурации компонентов lxd:

23. `lxc profile list`
24. `lxc network list`
25. `lxc storage list`

```

roman@roman-VirtualBox:~$ sudo lxc profile list
[sudo] password for roman:
+-----+-----+-----+
| NAME | DESCRIPTION | USED BY |
+-----+-----+-----+
| default | Default LXD profile | 0 |
+-----+-----+-----+
roman@roman-VirtualBox:~$ sudo lxc network list
+-----+-----+-----+-----+-----+-----+-----+-----+
| NAME | TYPE | MANAGED | IPV4 | IPV6 | DESCRIPTION | USED BY | STATE |
+-----+-----+-----+-----+-----+-----+-----+-----+
| enp0s3 | physical | NO | | | | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| lxdbr0 | bridge | YES | 10.200.0.1/24 | none | | 1 | CREATED |
+-----+-----+-----+-----+-----+-----+-----+-----+
roman@roman-VirtualBox:~$ sudo lxc storage list
+-----+-----+-----+-----+-----+-----+-----+-----+
| NAME | DRIVER | SOURCE | DESCRIPTION | USED BY | STATE |
+-----+-----+-----+-----+-----+-----+-----+-----+
| My_Yunusov_fs | zfs | /var/snap/lxd/common/lxd/disks/My_Yunusov_fs.img | | 1 | CREATED |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Рисунок 3. Результат конфигурации

lxc profile list

Описание: Выводит список всех доступных профилей в LXD.

Использование: Профили позволяют задавать общие настройки и ограничения для контейнеров, такие как лимиты по ресурсам и параметры сети.

lxc network list

Описание: Показывает список всех сетевых мостов и сетей, доступных в LXD.

Использование: Сетевые мосты позволяют контейнерам взаимодействовать друг с другом и с внешними сетями.

lxc storage list

Описание: Отображает список всех пулов хранения, настроенных в LXD.

Использование: Пулы хранения управляют местом, где располагаются данные контейнеров, и позволяют выбрать подходящий бэкенд для хранения.

11. Создание и запуск контейнеров

Создадим 3 контейнера с именами Apache, Chrome, MC. Из них запустим только Apache и MC

Создание и запуск выполняются двумя командами

```

26. sudo lxc init ubuntu:20.04 name
27. sudo lxc start name

```

Также можно одновременно создать и запустить контейнер при помощи

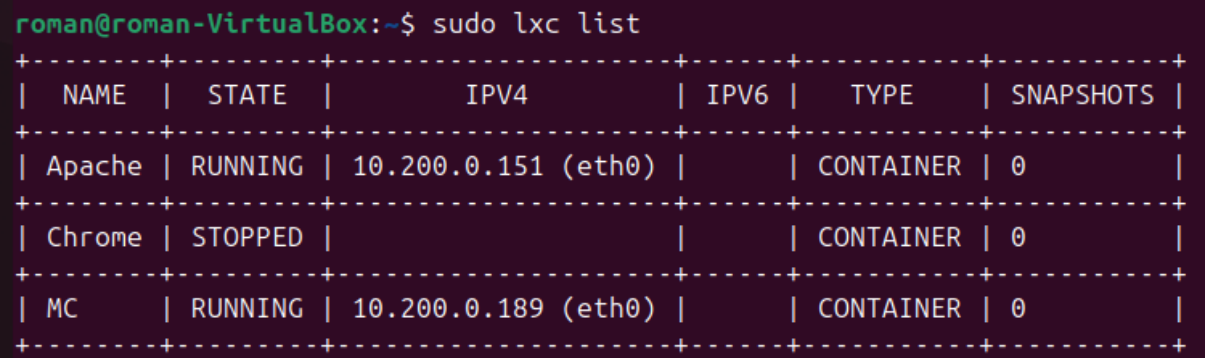
команды launch

```
28. lxc launch ubuntu:20.04 my-container
```

12. Посмотрим список контейнеров

Смотрим список через команду

```
29. sudo lxc list
```



```
roman@roman-VirtualBox:~$ sudo lxc list
```

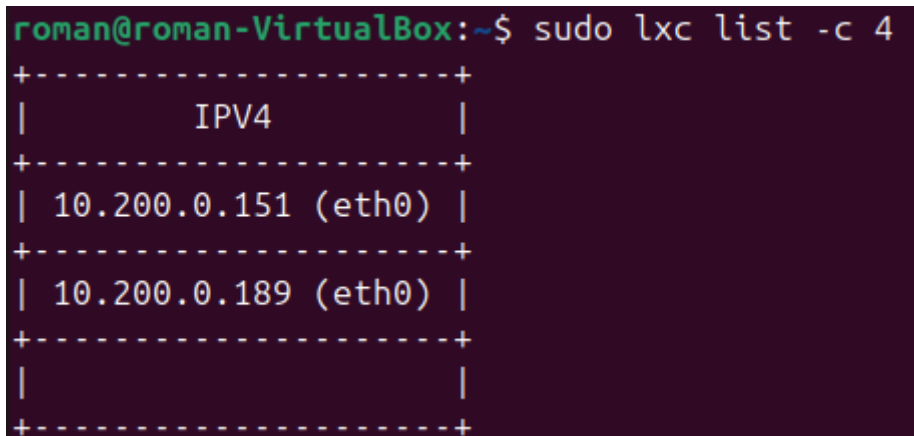
NAME	STATE	IPV4	IPV6	TYPE	SNAPSHOTS
Apache	RUNNING	10.200.0.151 (eth0)		CONTAINER	0
Chrome	STOPPED			CONTAINER	0
MC	RUNNING	10.200.0.189 (eth0)		CONTAINER	0

Рисунок 4. Просмотр контейнеров

13. Введем разные фильтры на список контейнеров

Только ip адреса

```
30. sudo lxc list -c 4
```



```
roman@roman-VirtualBox:~$ sudo lxc list -c 4
```

IPV4
10.200.0.151 (eth0)
10.200.0.189 (eth0)

Рисунок 5. Только Ip адреса

Только остановленные

```
1. sudo lxc list status=stopped
```

```
roman@roman-VirtualBox:~$ sudo lxc list status=stopped
```

NAME	STATE	IPV4	IPV6	TYPE	SNAPSHOTS
Chrome	STOPPED			CONTAINER	0

Рисунок 6. Только остановленные

Только контейнеры

```
1. sudo lxc list -c t
```

```
roman@roman-VirtualBox:~$ sudo lxc list -c t
```

TYPE
CONTAINER
CONTAINER

Рисунок 7. Только контейнеры

14. Удаление контейнера

Сначала нужно остановить контейнер, а только потом удалять

```
2. sudo lxc stop MC
```

```
roman@roman-VirtualBox:~$ sudo lxc stop MC
roman@roman-VirtualBox:~$ sudo lxc list status=running
```

NAME	STATE	IPV4	IPV6	TYPE	SNAPSHOTS
Apache	RUNNING	10.200.0.151 (eth0)		CONTAINER	0

Рисунок 8. Остановка контейнера

```
15. lxc delete MC
```

```

roman@roman-VirtualBox:~$ sudo lxc delete MC
roman@roman-VirtualBox:~$ sudo lxc list
+-----+-----+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS |
+-----+-----+-----+-----+-----+-----+
| Apache | RUNNING | 10.200.0.151 (eth0) | | CONTAINER | 0 |
+-----+-----+-----+-----+-----+-----+
| Chrome | STOPPED | | | CONTAINER | 0 |
+-----+-----+-----+-----+-----+-----+

```

Рисунок 8. Удаление контейнера

Анализ результатов

В ходе выполнения задания были достигнуты следующие результаты:

Успешно выполнена установка и настройка LXD в виртуальной машине в VirtualBox:

16. Выполнена и проверена базовая конфигурация системы (файловая система, сетевые мосты, MAAS)
17. Были выполнены базовые функции системы контейнеров – создание, запуск, удаление
18. Были изучены фильтры вывода информации о действующих контейнерах

Часть 2. Конфигурация и выполнение команд LXC/LXD

Цель

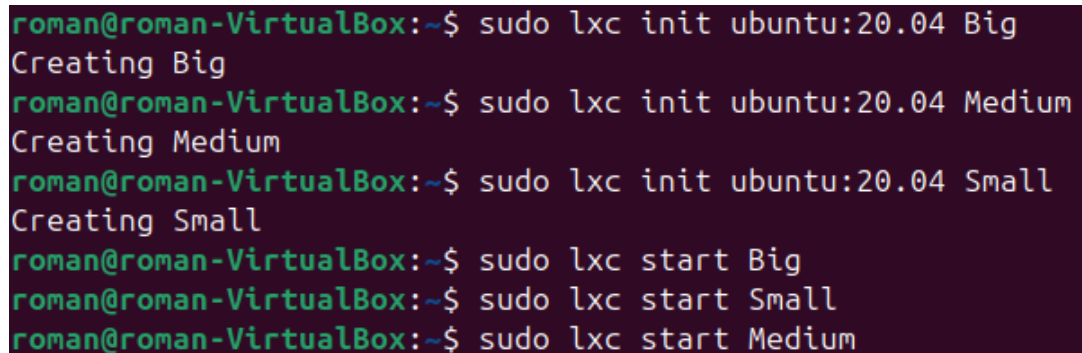
Получить навыки конфигурации контейнеров и запуска команд в них

Текст задания

1. Создать и запустить 3 контейнера: Big, Medium, Small.
2. Получить конфигурацию каждого контейнера.
3. Остановить контейнер Big.
4. Произвести настройку контейнера Big
5. Запустить команду free в контейнере Big
6. Выполнить конфигурирование контейнеров Small и Medium во время работы утилиты top
7. Остановить контейнер Small, а потом поменять его имя

Выполнение

1. Аналогично прошлому пункту создаём и запускаем три контейнера

A screenshot of a terminal window with a dark purple background. The prompt is 'roman@roman-VirtualBox:~\$'. The user enters 'sudo lxc init ubuntu:20.04 Big', followed by 'Creating Big'. Then 'sudo lxc init ubuntu:20.04 Medium', followed by 'Creating Medium'. Then 'sudo lxc init ubuntu:20.04 Small', followed by 'Creating Small'. Finally, the user enters 'sudo lxc start Big', 'sudo lxc start Small', and 'sudo lxc start Medium' in sequence.

```
roman@roman-VirtualBox:~$ sudo lxc init ubuntu:20.04 Big
Creating Big
roman@roman-VirtualBox:~$ sudo lxc init ubuntu:20.04 Medium
Creating Medium
roman@roman-VirtualBox:~$ sudo lxc init ubuntu:20.04 Small
Creating Small
roman@roman-VirtualBox:~$ sudo lxc start Big
roman@roman-VirtualBox:~$ sudo lxc start Small
roman@roman-VirtualBox:~$ sudo lxc start Medium
```

Рисунок 9. Создание контейнеров для работы

2. Далее получаем информацию о каждом через команду

19. Sudo lxc info name

```
roman@roman-VirtualBox:~$ sudo lxc info Big
Name: Big
Status: RUNNING
Type: container
Architecture: x86_64
PID: 5240
Created: 2024/12/07 14:21 MSK
Last Used: 2024/12/07 14:21 MSK

Resources:
  Processes: 56
  Disk usage:
    root: 4.64MiB
  CPU usage:
    CPU usage (in seconds): 15
  Memory usage:
    Memory (current): 159.04MiB
  Network usage:
    eth0:
      Type: broadcast
      State: UP
      Host interface: veth91deb058
      MAC address: 00:16:3e:fa:ca:d1
      MTU: 1500
      Bytes received: 314.40kB
      Bytes sent: 13.50kB
      Packets received: 152
      Packets sent: 124
      IP addresses:
        inet: 10.200.0.219/24 (global)
        inet6: fe80::216:3eff:fefa:cad1/64 (link)
    lo:
      Type: loopback
      State: UP
      MTU: 65536
      Bytes received: 1.68kB
      Bytes sent: 1.68kB
      Packets received: 16
      Packets sent: 16
      IP addresses:
        inet: 127.0.0.1/8 (local)
        inet6: ::1/128 (local)
```

Рисунок 10. Информация по бигу


```
roman@roman-VirtualBox:~$ sudo lxc info Medium
Name: Medium
Status: RUNNING
Type: container
Architecture: x86_64
PID: 5906
Created: 2024/12/07 14:21 MSK
Last Used: 2024/12/07 14:21 MSK

Resources:
  Processes: 56
  Disk usage:
    root: 4.63MiB
  CPU usage:
    CPU usage (in seconds): 15
  Memory usage:
    Memory (current): 153.34MiB
  Network usage:
    eth0:
      Type: broadcast
      State: UP
      Host interface: vethc97a20b0
      MAC address: 00:16:3e:ce:f1:28
      MTU: 1500
      Bytes received: 314.84kB
      Bytes sent: 16.16kB
      Packets received: 193
      Packets sent: 173
      IP addresses:
        inet: 10.200.0.207/24 (global)
        inet6: fe80::216:3eff:fece:f128/64 (link)
    lo:
      Type: loopback
      State: UP
      MTU: 65536
      Bytes received: 1.69kB
      Bytes sent: 1.69kB
      Packets received: 16
      Packets sent: 16
      IP addresses:
        inet: 127.0.0.1/8 (local)
        inet6: ::1/128 (local)
```

Рисунок 11. Информация по Medium

```
roman@roman-VirtualBox:~$ sudo lxc info Small
Name: Small
Status: RUNNING
Type: container
Architecture: x86_64
PID: 5481
Created: 2024/12/07 14:21 MSK
Last Used: 2024/12/07 14:21 MSK

Resources:
  Processes: 55
  Disk usage:
    root: 4.63MiB
  CPU usage:
    CPU usage (in seconds): 14
  Memory usage:
    Memory (current): 153.52MiB
  Network usage:
    eth0:
      Type: broadcast
      State: UP
      Host interface: veth41debfa3
      MAC address: 00:16:3e:42:ff:0e
      MTU: 1500
      Bytes received: 314.86kB
      Bytes sent: 13.13kB
      Packets received: 165
      Packets sent: 117
      IP addresses:
        inet: 10.200.0.229/24 (global)
        inet6: fe80::216:3eff:fe42:ff0e/64 (link)
    lo:
      Type: loopback
      State: UP
      MTU: 65536
      Bytes received: 1.68kB
      Bytes sent: 1.68kB
      Packets received: 16
      Packets sent: 16
      IP addresses:
        inet: 127.0.0.1/8 (local)
        inet6: ::1/128 (local)
```

Рисунок 12. Информация по Small

3. Далее конфигурируем контейнер Big

Сначала его необходимо остановить через stop

```
a. sudo lxc stop Big
```

Потом производим конфигурацию через лимит используемых процессоров и оперативной памяти

```
b. sudo lxc config set Big limits.cpu 2  
c. sudo lxc config set Big limits.memory 300MiB
```

проверим, как мы видим информация о количестве доступных процессоров, успешно добавилась в информацию контейнера.

```
roman@roman-VirtualBox:~$ sudo lxc config show Big  
architecture: x86_64  
config:  
  image.architecture: amd64  
  image.description: ubuntu 20.04 LTS amd64 (release) (20241112)  
  image.label: release  
  image.os: ubuntu  
  image.release: focal  
  image.serial: "20241112"  
  image.type: squashfs  
  image.version: "20.04"  
  limits.cpu: "2"  
  limits.memory: 300MiB
```

Рисунок 13. Результат нового конфига

После отправим утилиту free в контейнер через команду

```
d. sudo lxc exec Big -- free -h
```

```
roman@roman-VirtualBox:~$ sudo lxc exec Big -- free -h  
              total        used        free      shared  buff/cache   available  
Mem:           300Mi         74Mi        154Mi         0.0Ki         71Mi        225Mi  
Swap:           0B           0B           0B
```

Рисунок 14. Результат free

Видно, что количество доступной памяти тоже сократилось.

4. Конфигурируем контейнер Medium

Запустили внутри контейнера утилиту top для отслеживания параметров системы

```
a. sudo lxc exec Medium -- top
```

```
roman@roman-VirtualBox: ~  
top - 13:04:27 up 1:42, 0 users, load average: 1.11, 1.06, 1.01  
Tasks: 23 total, 1 running, 22 sleeping, 0 stopped, 0 zombie  
Cpu(s): 2.5 us, 4.8 sy, 0.0 ni, 89.6 id, 0.0 wa, 0.0 hi, 3.1 si, 0.0 st  
Mem: 7942.1 total, 7785.4 free, 74.9 used, 81.8 buff/cache  
Swap: 0.0 total, 0.0 free, 0.0 used, 7867.0 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	170936	9216	5248	S	0.0	0.1	0:11.06	systemd
57	root	20	0	35768	6656	5888	S	0.0	0.1	0:00.14	systemd+
97	root	20	0	12596	3968	2432	S	0.0	0.0	0:00.05	systemd+
111	root	20	0	372860	7936	1152	S	0.0	0.1	0:00.24	snapfuse
115	root	20	0	225124	1664	1152	S	0.0	0.0	0:00.02	snapfuse
118	root	20	0	224988	1280	1024	S	0.0	0.0	0:00.03	snapfuse
187	systemd+	20	0	27284	5376	4480	S	0.0	0.1	0:00.05	systemd+
189	systemd+	20	0	24912	9856	5504	S	0.0	0.1	0:00.08	systemd+
223	root	20	0	241068	5900	4876	S	0.0	0.1	0:00.16	account+
226	root	20	0	8548	1792	1536	S	0.0	0.0	0:00.01	cron
227	message+	20	0	7580	3072	2432	S	0.0	0.0	0:00.13	dbus-da+
234	root	20	0	29524	14744	6912	S	0.0	0.2	0:00.09	network+
235	root	20	0	238256	7484	5264	S	0.0	0.1	0:00.05	polkitd
239	syslog	20	0	154836	3456	2560	S	0.0	0.0	0:00.02	rsyslogd
241	root	20	0	1468596	24596	13056	S	0.0	0.3	0:03.75	snappd
245	root	20	0	17328	5120	4352	S	0.0	0.1	0:00.07	systemd+
247	root	20	0	395312	8872	7244	S	0.0	0.1	0:00.06	udisksd

Рисунок 15. Процесс работы утилиты top

Меняем в терминале данные

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

Теперь лимит в 200 MiB, что автоматически отразилось на top

```
roman@roman-VirtualBox: ~  
top - 13:04:51 up 1:43, 0 users, load average: 1.08, 1.05, 1.01  
Tasks: 23 total, 1 running, 22 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 0.5 us, 1.7 sy, 0.0 ni, 96.7 id, 0.0 wa, 0.0 hi, 1.2 si, 0.0 st  
MiB Mem : 200.0 total, 43.3 free, 74.9 used, 81.8 buff/cache  
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 124.9 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	170936	9216	5248	S	0.0	4.5	0:00.96	systemd
57	root	20	0	35768	6656	5888	S	0.0	3.2	0:00.14	systemd+
97	root	20	0	12596	3968	2432	S	0.0	1.9	0:00.03	systemd+
111	root	20	0	372860	7936	1152	S	0.0	3.9	0:00.24	snapfuse
115	root	20	0	225124	1664	1152	S	0.0	0.8	0:00.02	snapfuse
118	root	20	0	224988	1280	1024	S	0.0	0.6	0:00.03	snapfuse
187	systemd+	20	0	27284	5376	4480	S	0.0	2.6	0:00.05	systemd+
189	systemd+	20	0	24912	9856	5504	S	0.0	4.8	0:00.08	systemd+
223	root	20	0	241068	5900	4876	S	0.0	2.9	0:00.13	account+
226	root	20	0	8548	1792	1536	S	0.0	0.9	0:00.01	cron
227	message+	20	0	7580	3072	2432	S	0.0	1.5	0:00.13	dbus-da+
234	root	20	0	29524	14744	6912	S	0.0	7.2	0:00.09	network+
235	root	20	0	238256	7484	5264	S	0.0	3.7	0:00.05	polkitd
239	syslog	20	0	154836	3456	2560	S	0.0	1.7	0:00.02	rsyslogd
241	root	20	0	1468596	24596	13056	S	0.0	12.0	0:01.70	snappd
245	root	20	0	17328	5120	4352	S	0.0	2.5	0:00.07	systemd+
247	root	20	0	395312	8872	7244	S	0.0	4.3	0:00.06	udisksd

Рисунок 16. Изменение top

5. Работа с контейнером Small

Аналогично запустим топ и будем менять размер памяти

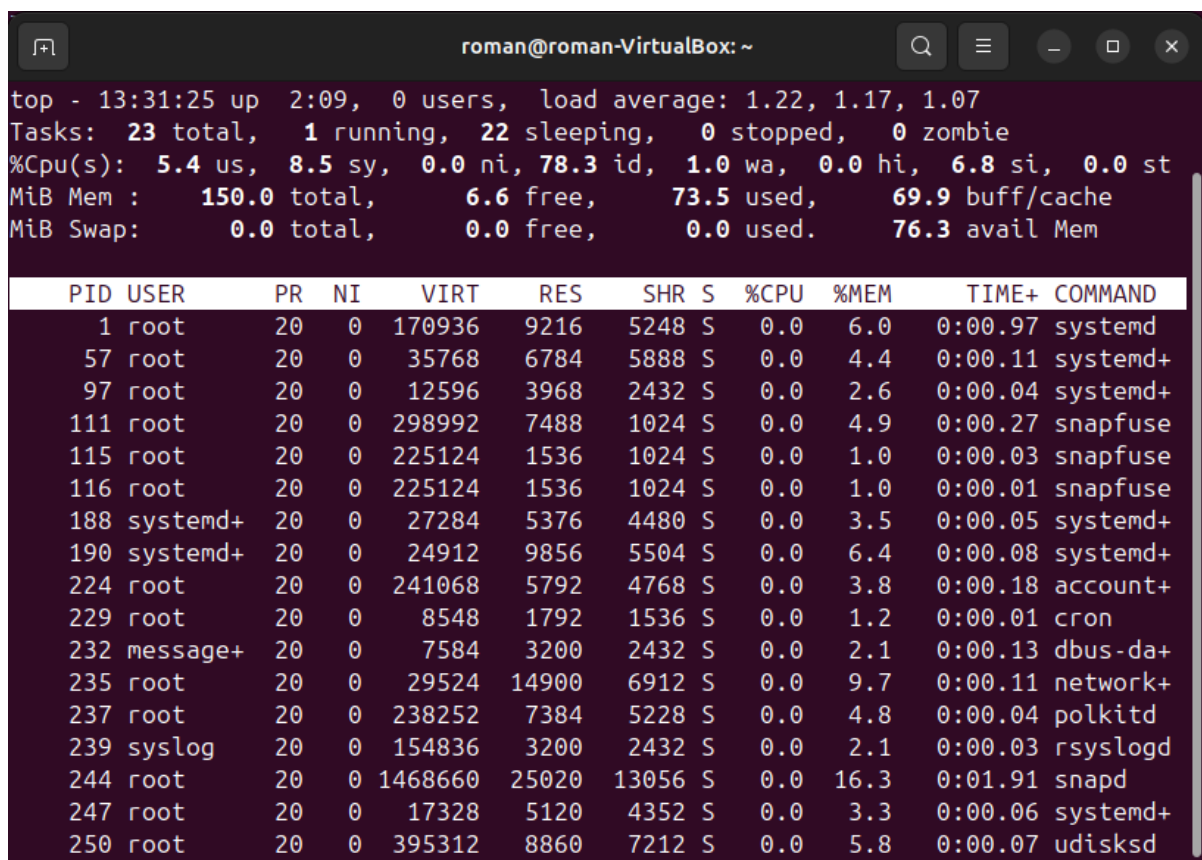
```
roman@roman-VirtualBox: ~  
top - 13:30:58 up 2:09, 0 users, load average: 1.22, 1.17, 1.06  
Tasks: 23 total, 1 running, 22 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 92.6 id, 0.0 wa, 0.0 hi, 7.4 si, 0.0 st  
MiB Mem : 7942.1 total, 7787.0 free, 74.9 used, 80.2 buff/cache  
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 7867.0 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	170936	9216	5248	S	0.0	0.1	0:00.97	systemd
57	root	20	0	35768	6784	5888	S	0.0	0.1	0:00.11	systemd+
97	root	20	0	12596	3968	2432	S	0.0	0.0	0:00.04	systemd+
111	root	20	0	298992	7488	1024	S	0.0	0.1	0:00.27	snapfuse
115	root	20	0	225124	1536	1024	S	0.0	0.0	0:00.03	snapfuse
116	root	20	0	225124	1536	1024	S	0.0	0.0	0:00.01	snapfuse
188	systemd+	20	0	27284	5376	4480	S	0.0	0.1	0:00.05	systemd+
190	systemd+	20	0	24912	9856	5504	S	0.0	0.1	0:00.08	systemd+
224	root	20	0	241068	5792	4768	S	0.0	0.1	0:00.18	account+
229	root	20	0	8548	1792	1536	S	0.0	0.0	0:00.01	cron
232	message+	20	0	7584	3200	2432	S	0.0	0.0	0:00.13	dbus-da+
235	root	20	0	29524	14900	6912	S	0.0	0.2	0:00.11	network+
237	root	20	0	238252	7384	5228	S	0.0	0.1	0:00.04	polkitd
239	syslog	20	0	154836	3200	2432	S	0.0	0.0	0:00.03	rsyslogd
244	root	20	0	1468660	25020	13056	S	0.0	0.3	0:01.91	snappd
247	root	20	0	17328	5120	4352	S	0.0	0.1	0:00.06	systemd+
250	root	20	0	395312	8860	7212	S	0.0	0.1	0:00.07	udisksd

Рисунок 17. Начальный top

После введем команду

```
a. sudo lxc config set Small limits.memory 150MiB
```



The screenshot shows a terminal window titled 'roman@roman-VirtualBox: ~'. The 'top' command has been executed, displaying system statistics and a list of running processes. The statistics show 23 total tasks, with 1 running and 22 sleeping. CPU usage is 5.4% user, 8.5% system, and 78.3% idle. Memory usage is 150.0 MiB total, with 6.6 MiB free and 73.5 MiB used. Swap usage is 0.0 MiB total, with 0.0 MiB free and 0.0 MiB used. The process list shows various system processes, including systemd, snapfuse, cron, dbus-daemon, networkd, polkitd, rsyslogd, snapd, and udisksd.

```
top - 13:31:25 up 2:09, 0 users, load average: 1.22, 1.17, 1.07
Tasks: 23 total, 1 running, 22 sleeping, 0 stopped, 0 zombie
%Cpu(s): 5.4 us, 8.5 sy, 0.0 ni, 78.3 id, 1.0 wa, 0.0 hi, 6.8 si, 0.0 st
MiB Mem : 150.0 total, 6.6 free, 73.5 used, 69.9 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 76.3 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
    1 root        20   0 170936   9216  5248  S   0.0   6.0   0:00.97 systemd
   57 root        20   0  35768   6784  5888  S   0.0   4.4   0:00.11 systemd+
   97 root        20   0  12596   3968  2432  S   0.0   2.6   0:00.04 systemd+
  111 root        20   0 298992   7488  1024  S   0.0   4.9   0:00.27 snapfuse
  115 root        20   0 225124   1536  1024  S   0.0   1.0   0:00.03 snapfuse
  116 root        20   0 225124   1536  1024  S   0.0   1.0   0:00.01 snapfuse
  188 systemd+    20   0  27284   5376  4480  S   0.0   3.5   0:00.05 systemd+
  190 systemd+    20   0  24912   9856  5504  S   0.0   6.4   0:00.08 systemd+
  224 root        20   0 241068   5792  4768  S   0.0   3.8   0:00.18 account+
  229 root        20   0   8548   1792  1536  S   0.0   1.2   0:00.01 cron
  232 message+    20   0   7584   3200  2432  S   0.0   2.1   0:00.13 dbus-da+
  235 root        20   0  29524  14900  6912  S   0.0   9.7   0:00.11 network+
  237 root        20   0 238252   7384  5228  S   0.0   4.8   0:00.04 polkitd
  239 syslog      20   0 154836   3200  2432  S   0.0   2.1   0:00.03 rsyslogd
  244 root        20   0 1468660 25020 13056  S   0.0  16.3   0:01.91 snapd
  247 root        20   0  17328   5120  4352  S   0.0   3.3   0:00.06 systemd+
  250 root        20   0 395312   8860  7212  S   0.0   5.8   0:00.07 udisksd
```

Рисунок 18. Топ после первого изменения конфига

```
a. sudo lxc config set Small limits.memory 800MiB
```



```
roman@roman-VirtualBox: ~  
top - 13:31:43 up 2:09, 0 users, load average: 1.16, 1.16, 1.06  
Tasks: 23 total, 1 running, 22 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 2.2 us, 3.9 sy, 0.0 ni, 91.9 id, 0.0 wa, 0.0 hi, 2.0 si, 0.0 st  
MiB Mem : 800.0 total, 656.3 free, 73.7 used, 69.9 buff/cache  
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 726.1 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	170936	9216	5248	S	0.0	1.1	0:00.97	systemd
57	root	20	0	35768	6784	5888	S	0.0	0.8	0:00.11	systemd+
97	root	20	0	12596	3968	2432	S	0.0	0.5	0:00.04	systemd+
111	root	20	0	298992	7488	1024	S	0.0	0.9	0:00.27	snapfuse
115	root	20	0	225124	1536	1024	S	0.0	0.2	0:00.03	snapfuse
116	root	20	0	225124	1536	1024	S	0.0	0.2	0:00.01	snapfuse
188	systemd+	20	0	27284	5376	4480	S	0.0	0.7	0:00.05	systemd+
190	systemd+	20	0	24912	9856	5504	S	0.0	1.2	0:00.08	systemd+
224	root	20	0	241068	5792	4768	S	0.0	0.7	0:00.18	account+
229	root	20	0	8548	1792	1536	S	0.0	0.2	0:00.01	cron
232	message+	20	0	7584	3200	2432	S	0.0	0.4	0:00.13	dbus-da+
235	root	20	0	29524	14900	6912	S	0.0	1.8	0:00.11	network+
237	root	20	0	238252	7384	5228	S	0.0	0.9	0:00.04	polkitd
239	syslog	20	0	154836	3200	2432	S	0.0	0.4	0:00.03	rsyslogd
244	root	20	0	1468660	25020	13056	S	0.0	3.1	0:01.91	snapt
247	root	20	0	17328	5120	4352	S	0.0	0.6	0:00.06	systemd+
250	root	20	0	395312	8860	7212	S	0.0	1.1	0:00.07	udisksd

Рисунок 19. Тор после второго изменения конфига

Добавим описание к контейнеру и выведем его

- a. `lxc config set Small user.description "Future not Small"`
 - b. `sudo lxc config show Small --expanded`

```
roman@roman-VirtualBox:~$ sudo lxc config show Small --expanded  
architecture: x86_64  
config:  
  image.architecture: amd64  
  image.description: ubuntu 20.04 LTS amd64 (release) (20241112)  
  image.label: release  
  image.os: ubuntu  
  image.release: focal  
  image.serial: "20241112"  
  image.type: squashfs  
  image.version: "20.04"  
  limits.memory: 800MiB  
  user.description: Future not Small  
  volatile.base_image: f95d66ff09af6fb3a4a829005277d8c742aac705c46aba13c4f56621f6c92da5
```

Рисунок 20. Добавили описание

Дальше остановим контейнер через команду `stop` и выведем начальный список контейнеров

```
roman@roman-VirtualBox:~$ sudo lxc list
```

NAME	STATE	IPV4	IPV6	TYPE	SNAPSHOTS
Apache	RUNNING	10.200.0.184 (eth0)		CONTAINER	0
Big	RUNNING	10.200.0.219 (eth0)		CONTAINER	0
Chrome	STOPPED			CONTAINER	0
Medium	RUNNING	10.200.0.207 (eth0)		CONTAINER	0
Small	STOPPED			CONTAINER	0

Рисунок 21. Начальный список контейнеров

Переименуем наш контейнер Small в BigBig

```
a. lxc config set Small user.description "Future not Small"
```

```
roman@roman-VirtualBox:~$ sudo lxc rename Small BigBig
roman@roman-VirtualBox:~$ sudo lxc list
```

NAME	STATE	IPV4	IPV6	TYPE	SNAPSHOTS
Apache	RUNNING	10.200.0.184 (eth0)		CONTAINER	0
Big	RUNNING	10.200.0.219 (eth0)		CONTAINER	0
BigBig	STOPPED			CONTAINER	0
Chrome	STOPPED			CONTAINER	0
Medium	RUNNING	10.200.0.207 (eth0)		CONTAINER	0

Рисунок 22. Новый список контейнеров

Анализ результатов

В ходе выполнения задания были достигнуты следующие результаты:

1. Успешно созданы несколько контейнеров
2. Были протестированы возможности конфигурации разных параметров контейнеров, как во время работы, так и выключенных
3. Были изучены методы по заданию описания и переименования контейнеров

Все поставленные задачи выполнены успешно и можно приступать к следующему этапу

Часть 3. Практическое применение LXD.

Цель

Запустить несколько приложений внутри контейнера и получить к ним доступ со стороны хост машины

Текст задания

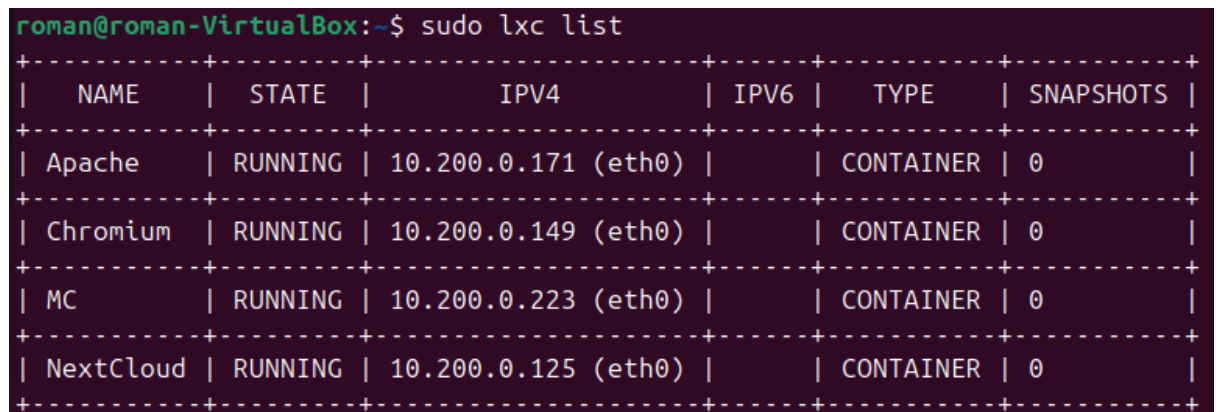
4. Создать и запустить 4 контейнера с ограничением на 500 MiB
5. В контейнере Apache запустить сервер Apache, где в главной странице будет страница с моим фио
6. Прокинуть порты для контейнера nextcloud и запустить его
7. Запустить графические версии браузера Firefox и MidnightCommander

Выполнение

1. Создадим контейнеры и добавим ограничение по памяти, пример команды для MC на экране

```
sudo lxc init ubuntu:20.04 MC
sudo lxc start MC
sudo lxc config set MC limits.memory 500MiB
```

Выведем список работающих контейнеров для проверки



```
roman@roman-VirtualBox:~$ sudo lxc list
```

NAME	STATE	IPV4	IPV6	TYPE	SNAPSHOTS
Apache	RUNNING	10.200.0.171 (eth0)		CONTAINER	0
Chromium	RUNNING	10.200.0.149 (eth0)		CONTAINER	0
MC	RUNNING	10.200.0.223 (eth0)		CONTAINER	0
NextCloud	RUNNING	10.200.0.125 (eth0)		CONTAINER	0

Рисунок 23. Список контейнеров в начале выполнения задания

2. Запуск и настройка апач сервера внутри контейнера

Для начала зайдем в bash контейнера

```
sudo lxc exec Apache -- /bin/bash
```

Далее установим апач

```
apt update && install apache2 -y
```

Далее нужно его запустить

```
systemctl start apache2
```

Проверим то что он работает

```
systemctl status apache2
```

```
root@Apache:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor
   Active: active (running) since Sat 2024-12-07 14:05:15 UTC; 19s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 1785 (apache2)
    Tasks: 55 (limit: 9447)
   Memory: 7.0M
      CPU: 66ms
   CGroup: /system.slice/apache2.service
           └─1785 /usr/sbin/apache2 -k start
             └─1787 /usr/sbin/apache2 -k start
               └─1788 /usr/sbin/apache2 -k start
```

Рисунок 24. Статус апач сервера

Дальше необходимо сделать базовую страницу

```
cd /var/www/html
echo "<html><body><h1>Yunusov Roman</h1></body></html>" > index.html
```

Дальше подключаемся по IPшнику контейнера на стандартный порт Апача через браузер на хост системе

The image shows a web browser window on the left and a terminal window on the right. The browser window displays the text "Yunusov Roman" at the URL "10.200.0.171/". The terminal window shows the command "sudo lxc list" and its output, which is a table listing containers.

NAME	STATE	IPV4	IPV6	TYPE	SNAPSHOTS
Apache	RUNNING	10.200.0.171 (eth0)		CONTAINER	0
Chromium	RUNNING	10.200.0.149 (eth0)		CONTAINER	0
MC	RUNNING	10.200.0.223 (eth0)		CONTAINER	0
NextCloud	RUNNING	10.200.0.125 (eth0)		CONTAINER	0

Рисунок 25. Подключение выполнено

3. Запуск и настройка сервиса Netcloud внутри контейнера

Вообще лучше делать порты при создании контейнера, но можно пробросить уже в рабочий

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

Объяснение команды.

Мы добавляем к контейнеру устройства типа прокси с именем myport80, который будет передавать все запросы по порту 8080 на хосте, и будет передавать их на локальный порт 80 контейнера

Далее действуем стандартно, зайдем в bash контейнера и установим все зависимости

```
sudo lxc exec nextCloud -- /bin/bash
apt install apache2 php libapache2-mod-php php-mysql php-xml php-mbstring php-curl php-zip php-gd -y
```

Скачаем и распакуем последнюю версию NextCloud

```
wget https://download.nextcloud.com/server/releases/nextcloud-24.0.4.zip
unzip nextcloud-24.0.4.zip
mv nextcloud /var/www/html/
chown -R www-data:www-data /var/www/html/nextcloud
```

Дальше создадим конфигурацию

```
nano /etc/apache2/sites-available/nextcloud.conf
```

```

root@NextCloud:/etc/apache2/sites-available# a2ensite nextcloud.conf
Site nextcloud already enabled
root@NextCloud:/etc/apache2/sites-available# a2enmod rewrite headers env dir mime setenvif
Module rewrite already enabled
Module headers already enabled
Module env already enabled
Module dir already enabled
Module mime already enabled
Module setenvif already enabled
root@NextCloud:/etc/apache2/sites-available# systemctl restart apache2
root@NextCloud:/etc/apache2/sites-available# cat nextcloud.conf
<VirtualHost *:80>
    DocumentRoot /var/www/html/nextcloud
    ServerName 10.200.0.125

    <Directory /var/www/html/nextcloud>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/nextcloud_error.log
    CustomLog ${APACHE_LOG_DIR}/nextcloud_access.log combined
</VirtualHost>

```

Рисунок 26. Результат конфигурация Nextcloud

Дальше нужно перезапустить новую апач с новой конфигурацией

```

a2ensite nextcloud.conf
a2enmod rewrite headers env dir mime setenvif
systemctl restart apache2

```

```

root@NextCloud:/etc/apache2/sites-available# a2ensite nextcloud.conf
Site nextcloud already enabled
root@NextCloud:/etc/apache2/sites-available# a2enmod rewrite headers env
Module rewrite already enabled
Module headers already enabled
Module env already enabled
Module dir already enabled
Module mime already enabled
Module setenvif already enabled
root@NextCloud:/etc/apache2/sites-available# systemctl restart apache2

```

Рисунок 27. Результат перезапуска

Готово теперь можно подключаться по нашему адресу

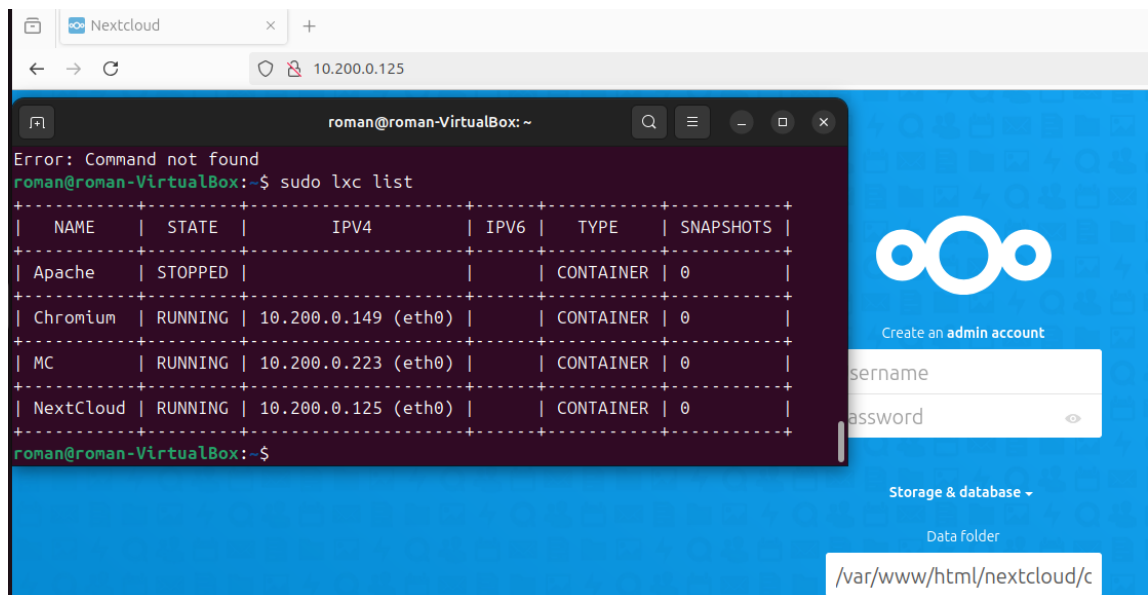


Рисунок 28. Подключение по Ip адресу контейнера

4. Запуск Firefox

Для запуска графических приложений в linux используется отдельное графическое приложение, не связанное системой, так называемый X сервер. Чтобы контейнер мог запустить такое приложение, нужно прокинуть адрес хост X сервера в контейнер.

```
lxc config device add Firefox X0 disk source=/tmp/.X11-unix/X0 path=/tmp/.X11-unix/X0
```

```
roman@roman-VirtualBox:~$ sudo lxc config device add Firefox X0 disk source=/tmp/.X11-unix/X0 path=/tmp/.X11-unix/X0
Device X0 added to Firefox
```

Рисунок 29. Добавление девайса, который подменяет адреса Xservera на контейнере на Xserver хоста

Также нужно настроить переменную окружения внутри контейнера

```
sudo lxc exec Firefox -- bash -c "echo 'export DISPLAY=:0' >> ~/.bashrc"
sudo lxc exec Firefox -- bash -c "export DISPLAY=:0"
```

Разрешаем локальные подключения к Xserverу, в частности для нашего активного пользователя

```
xhost +local:
xhost +SI:localuser:$(whoami)
```

```
roman@roman-VirtualBox: ~  
roman@roman-VirtualBox:~$ xhost +local:  
non-network local connections being added to access control list  
roman@roman-VirtualBox:~$ xhost +SI:localuser:$(whoami)  
localuser:roman being added to access control list
```

Рисунок 30. Разрешаем работу Xservera

Дальше как обычно заходим в контейнер и устанавливаем firefox

```
sudo lxc exec Firefox -- /bin/bash  
sudo apt update  
lxc exec Firefox -- apt install -y Firefox
```

После этого можно запустить firefox в контейнере

```
firefox
```

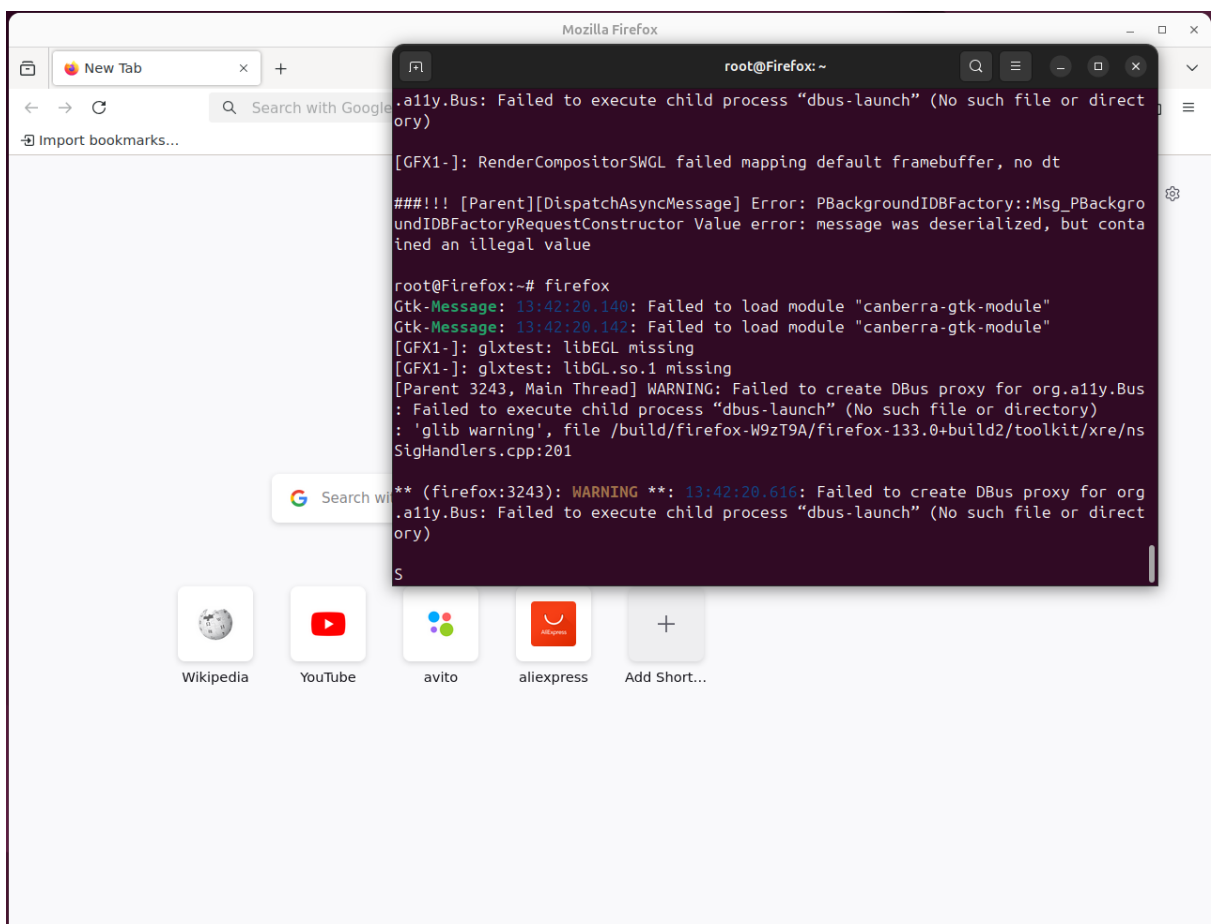


Рисунок 31. Запуск firefox

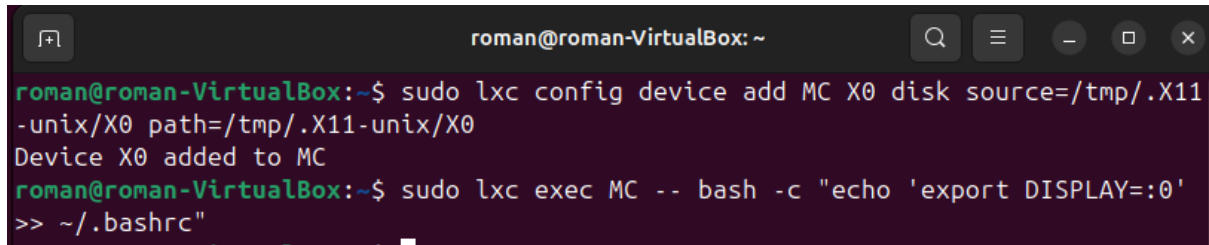
5. MidnightCommander

Делаем операции аналогичные Firefox, монтируем диск и добавляем переменную

окружения

```
lxc config device add MC X0 disk source=/tmp/.X11-unix/X0 path=/tmp/.X11-unix/X0 sudo  
lxc exec MC -- bash -c "echo 'export DISPLAY=:0' >> ~/.bashrc"
```

Но в данном случае уже не нужно делать локальные разрешения для хост сервера, поскольку они были настроены в прошлом пункте

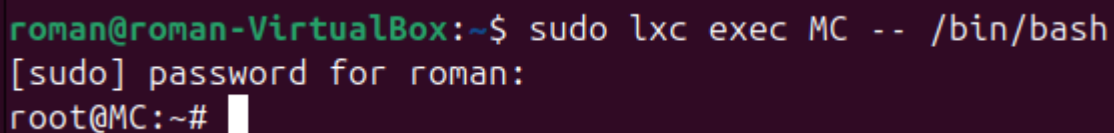


```
roman@roman-VirtualBox: ~  
roman@roman-VirtualBox:~$ sudo lxc config device add MC X0 disk source=/tmp/.X11-unix/X0 path=/tmp/.X11-unix/X0  
Device X0 added to MC  
roman@roman-VirtualBox:~$ sudo lxc exec MC -- bash -c "echo 'export DISPLAY=:0' >> ~/.bashrc"
```

Рисунок 32. Результат установки переменных

Дальше заходим внутрь контейнера, устанавливаем и запускаем midnightcommander

```
sudo lxc exec MC -- /bin/bash  
sudo apt update && sudo apt install mc  
mc
```



```
roman@roman-VirtualBox:~$ sudo lxc exec MC -- /bin/bash  
[sudo] password for roman:  
root@MC:~#
```

Рисунок 33. Заход в bash на контейнере

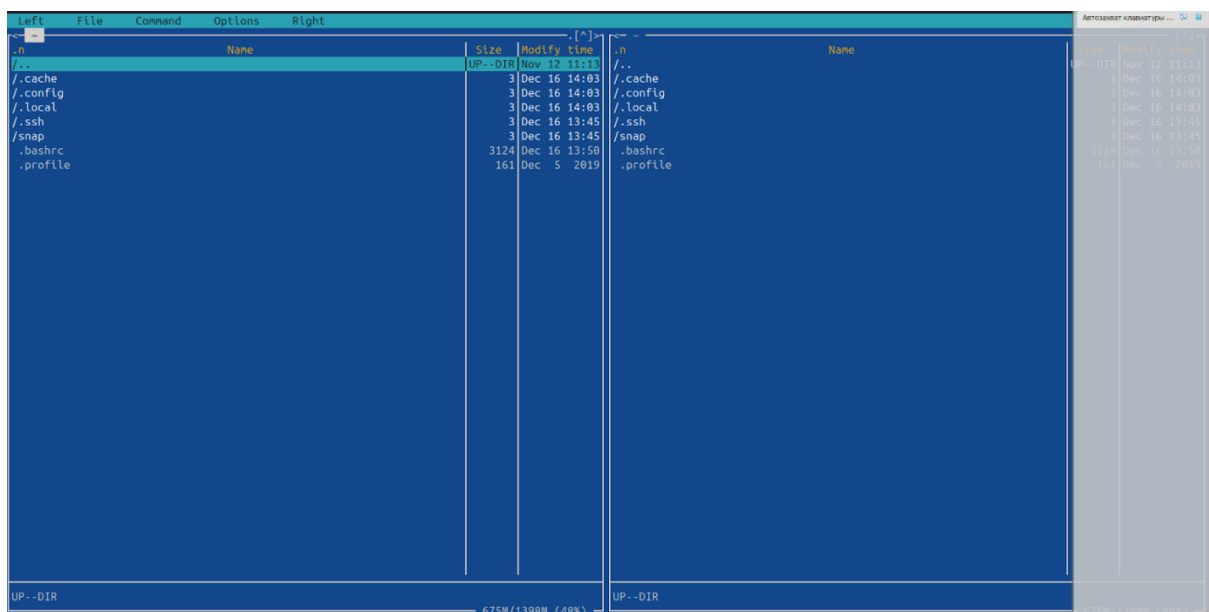


Рисунок 34. Результат установки Mc

Анализ результатов

В результате выполнения задания были достигнуты следующие результаты:

1. Успешно запущен сервер Apache в контейнере:
2. Успешно запущена программа Nextcloud:
 1. Настроен проброс портов с хост-машины (порт 8080) внутрь контейнера (порт 80)
3. Успешно запущены графические программы Apache и MS:
 1. Был смонтирован адрес апач сервера на хосте внутрь контейнера
 2. Настроен прием локальных подключений на Xserver со стороны конкретного пользователя

Все поставленные задачи выполнены успешно.

Заключение

В ходе выполнения лабораторной работы были успешно достигнуты следующие результаты:

1. Был установлен и настроен LXD, получены начальные навыки работы с контейнерами:
2. Была произведена настройка контейнеров по разным параметрам, настройка производилась как выключенных контейнеров, так и в режиме работы:
3. Были успешно запущены 4 приложения внутри контейнерам:

Лабораторная работа позволила получить практический опыт работы с системой контейнеров LXC, что является важным навыком для современного IT-специалиста.

Список литературы

LXD [Online]. - linuxcontainer.org.