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](#_Toc185299503)

[Введение 3](#_Toc185299504)

[Описание работы и инструментов 3](#_Toc185299505)

[Цели и задачи 3](#_Toc185299506)

[Задачи 4](#_Toc185299507)

[Часть 1. Установка Proxmox 5](#_Toc185299508)

[Цель 5](#_Toc185299509)

[Текст задания 5](#_Toc185299510)

[Выполнение 5](#_Toc185299511)

[Часть 2. Конфигурация и выполнение команд LXC/LXD 15](#_Toc185299512)

[Цель 15](#_Toc185299513)

[Текст задания 15](#_Toc185299514)

[Выполнение 15](#_Toc185299515)

[Часть 3. Практическое применение LXD. 25](#_Toc185299516)

[Цель 25](#_Toc185299517)

[Текст задания 25](#_Toc185299518)

[Выполнение 25](#_Toc185299519)

[Заключение 33](#_Toc185299520)

[Список литературы 34](#_Toc185299521)

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 хранилица-My\_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

1. Установка
2. sudo snap install lxd
3. Проверим работу системы командами
4. lxc list
5. lxc --version

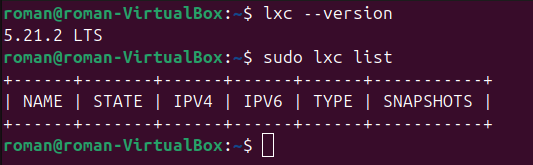


Рисунок 1. Информация о Lxd

1. Сконфигурируем LXC через
2. sudo lxс init

Для конфигурации нужно было ответить на список вопросов касаемо системы

1. 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".

1. 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".

1. 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".

1. 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.

1. 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".

1. 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".

1. 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.

1. 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

1. 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".

1. What should the new bridge be called? [default=lxdbr0]:

Вы можете указать название для сетевого моста; если оставить поле пустым, будет использовано название по умолчанию — "lxdbr0".

1. 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.

1. 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".

1. 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.

1. 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".

1. 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.

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

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

1. 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".

1. 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.

1. Итогом инициализации стал конфиг

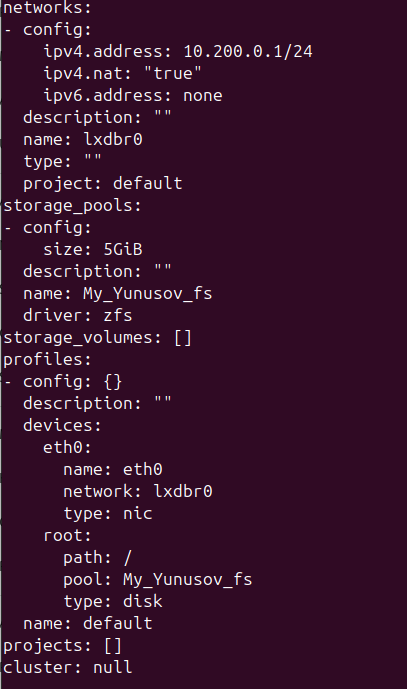


Рисунок 2. Конфиг lxd

1. Получить конфигурации компонентов lxd:
2. lxc profile list
3. lxc network list
4. lxc storage list

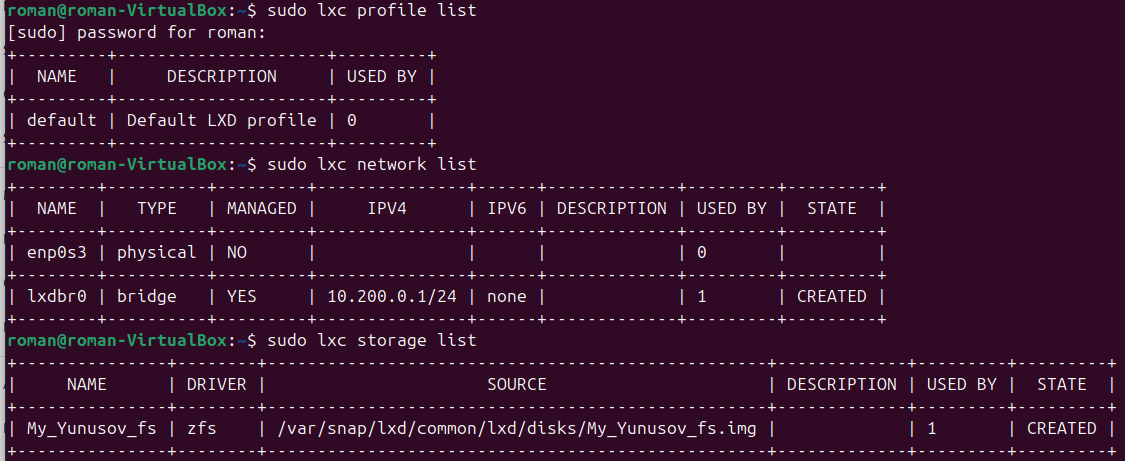


Рисунок 3. Результат конфигурации

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.

1. 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

1. sudo lxc init ubuntu:20.04 name
2. sudo lxc start name

Также можно одновременно создать и запустить контейнер при помощи команды launch

1. lxc launch ubuntu:20.04 my-container
2. Посмотрим список контейнеров

Смотрим список через команду

1. sudo lxc list

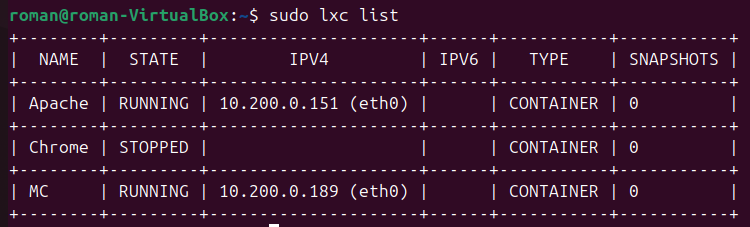


Рисунок 4. Просмотр контейнеров

1. Введем разные фильтры на список контейнеров

Только ip адреса

1. sudo lxc list -c 4

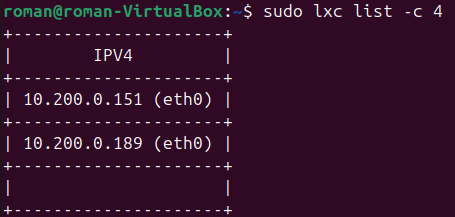


Рисунок 5. Только Ip адреса

Только остановленные

1. sudo lxc list status=stopped

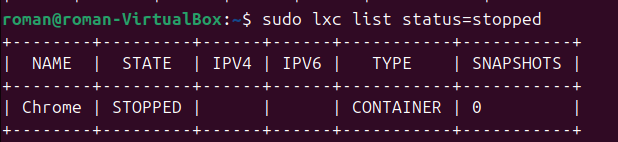


Рисунок 6. Только остановленные

Только контейнеры

1. sudo lxc list -c t

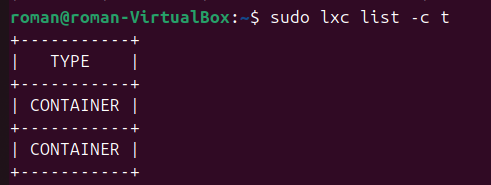


Рисунок 7. Только контейнеры

1. Удаление контейнера

Сначала нужно остановить контейнер, а только потом удалять

1. sudo lxc stop MC

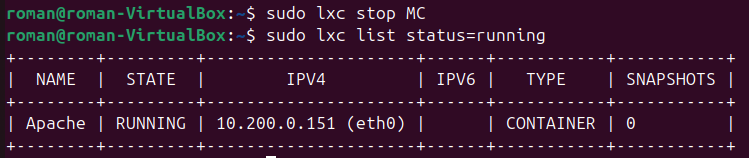


Рисунок 8. Остановка контейнера

1. lxc delete MC

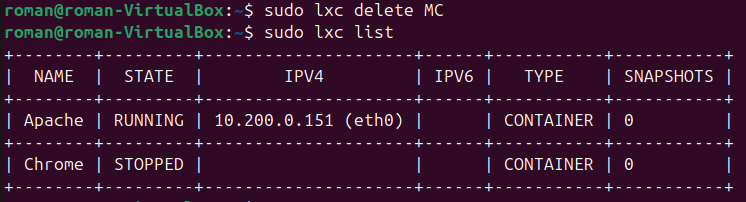


Рисунок 8. Удаление контейнера

**Анализ результатов**

В ходе выполнения задания были достигнуты следующие результаты:

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

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

Часть 2. Конфигурация и выполнение команд LXC/LXD

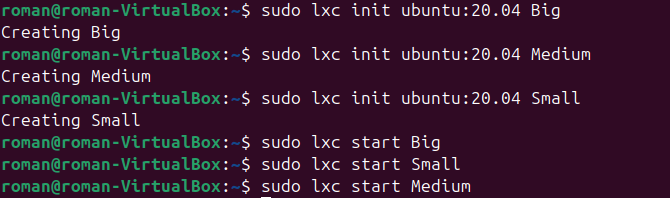
Цель

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  
   Рисунок 9. Создание контейнеров для работы
2. Дальше получаем информацию о каждом через команду
3. Sudo lxc info name

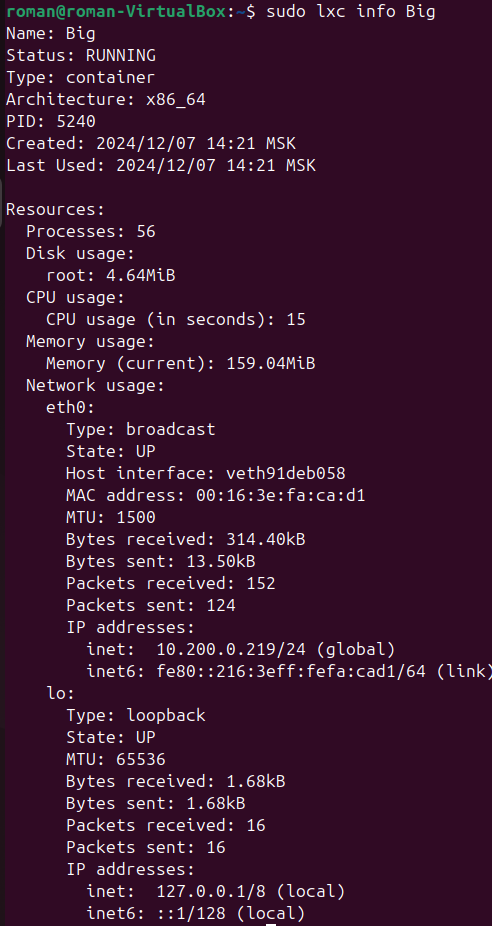


Рисунок 10. Информация по бигу

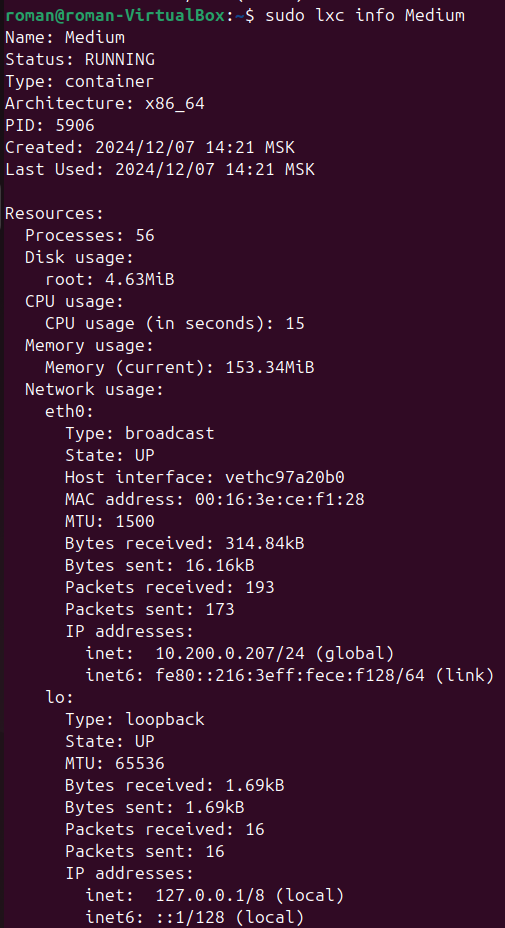


Рисунок 11. Информация по Medium

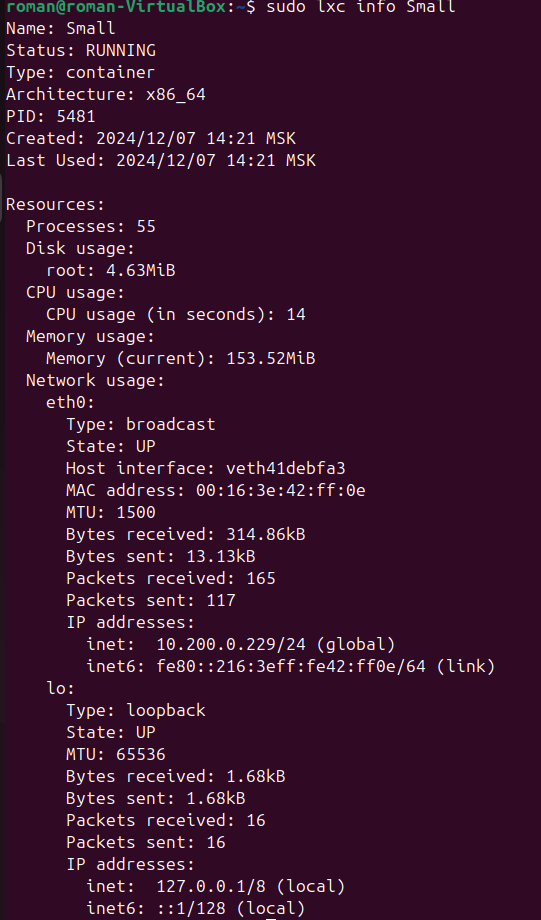


Рисунок 12. Информация по Small

1. Дальше конфигурируем контейнер Big

Сначала его необходимо остановить через stop

* 1. sudo lxc stop Big

Потом производим конфигурацию через лимит используемых процессоров и оперативной памяти

* 1. sudo lxc config set Big limits.cpu 2
  2. sudo lxc config set Big limits.memory 300MiB

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

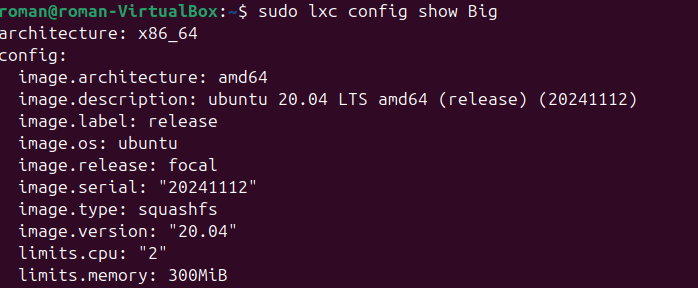


Рисунок 13. Результат нового конфига

После отправим утилиту free в контейнер через команду

* 1. sudo lxc exec Big – free -h

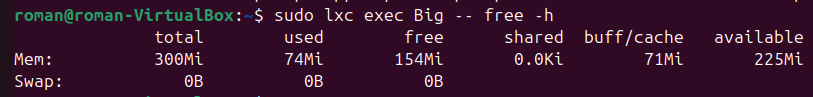


Рисунок 14. Результат free

Видно, что количество доступной памяти тоже сократилось.

1. Конфигурируем контейнер Medium

Запустили внутри контейнера утилиту top для отслеживания параметров системы

* 1. sudo lxc exec Medium -- top

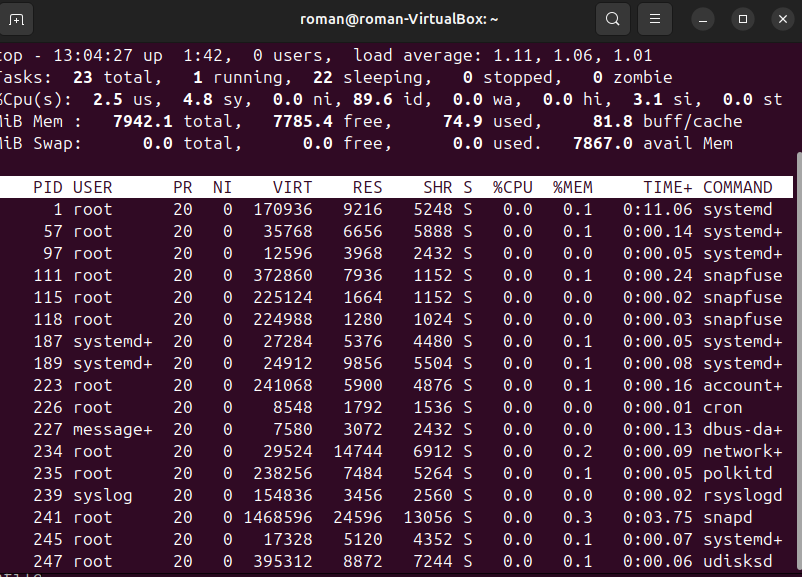


Рисунок 15. Процесс работы утилиты top

Меняем в терминале данные

1. sudo lxc config set Medium limits.memory 200MiB

Теперь лимит в 200 MiB, что автоматически отразилось на top

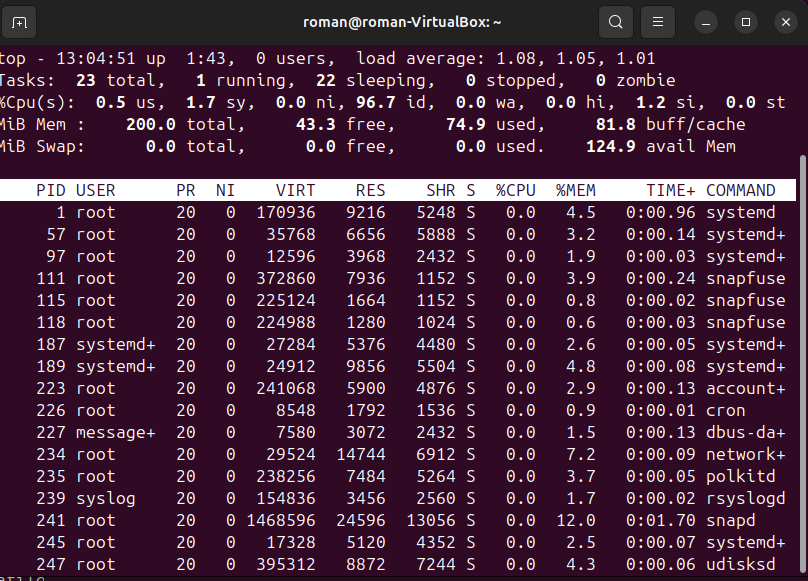
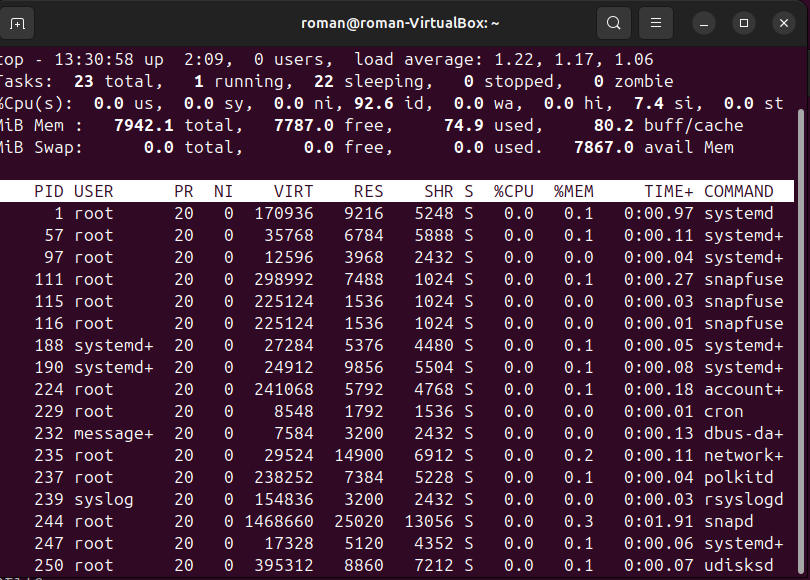


Рисунок 16. Изменение top

1. Работа с контейнером Small

Аналогично запустим топ и будем менять размер памяти

 Рисунок 17. Начальный top

После введем команду

* 1. sudo lxc config set Small limits.memory 150MiB

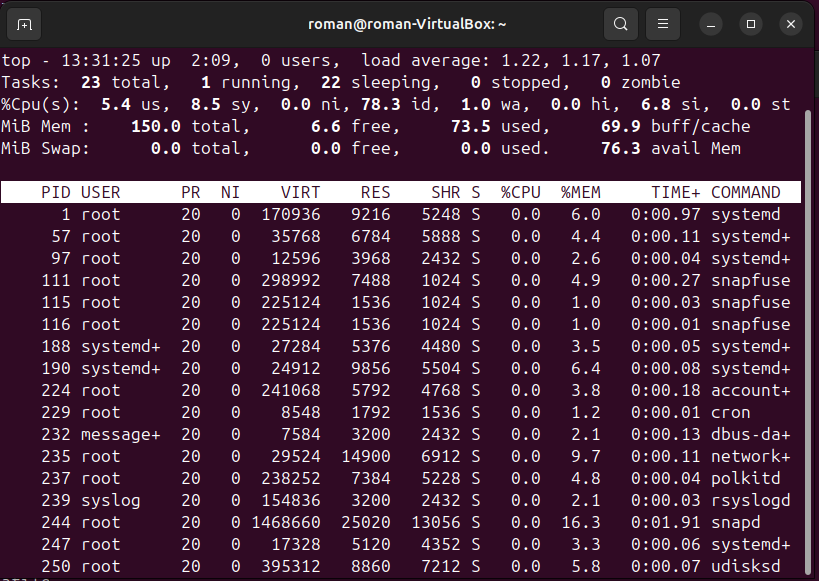


Рисунок 18. Top после первого изменения конфига

1. sudo lxc config set Small limits.memory 800MiB

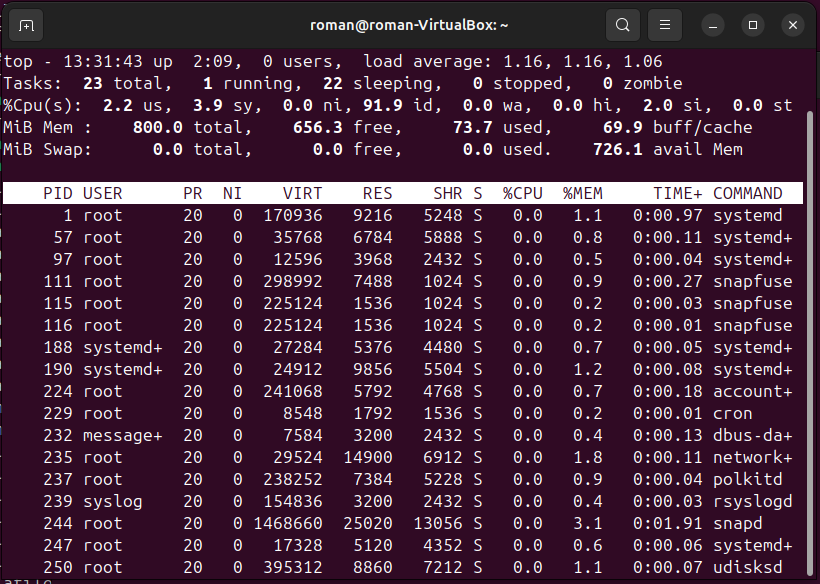


Рисунок 19. Top после второго изменения конфига

Добавим описание к контейнеру и выведем его

1. lxc config set Small user.description "Future not Small"
2. sudo lxc config show Small --expanded

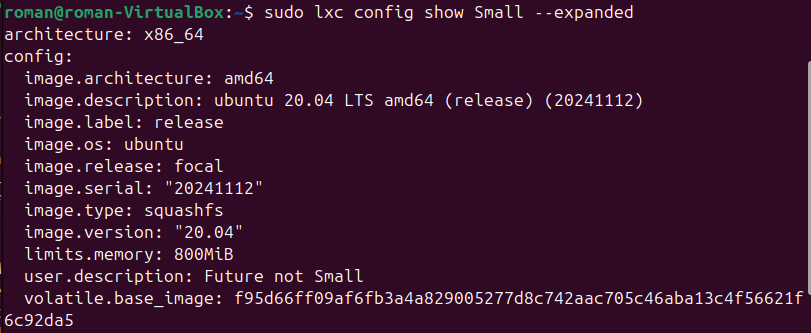


Рисунок 20. Добавили описание

Дальше остановим контейнер через команду stop и выведем начальный список контейнеров

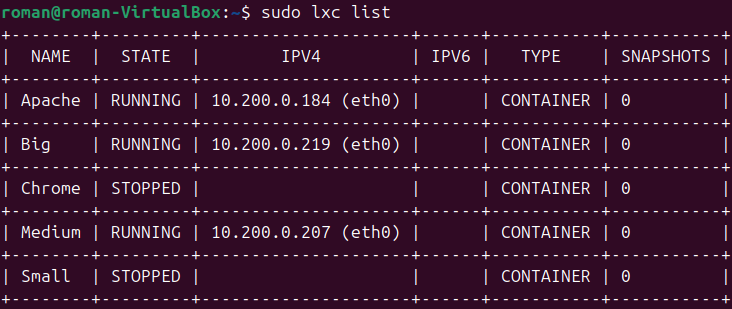


Рисунок 21. Начальный список контейнеров

Переименуем наш контейнер Small в BigBig

1. lxc config set Small user.description "Future not Small"

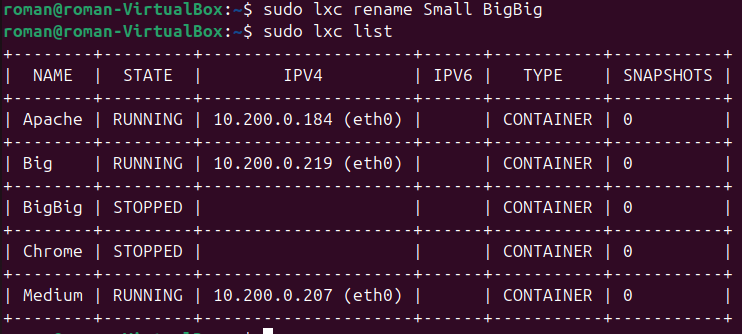


Рисунок 22. Новый список контейнеров

**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

1. Создать и запустить 4 контейнера c ограничением на 500 MiB
2. В контейнере Apache запустить сервер Apache, где в главной странице будет страница с моим фио
3. Прокинуть порты для контейнера nextcloud и запустить его
4. 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

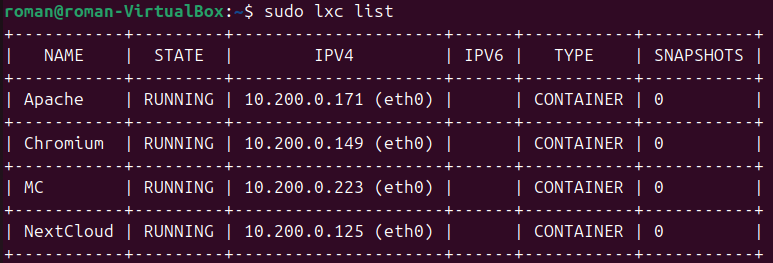


Рисунок 23. Список контейнеров в начале выполнения задания

1. 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

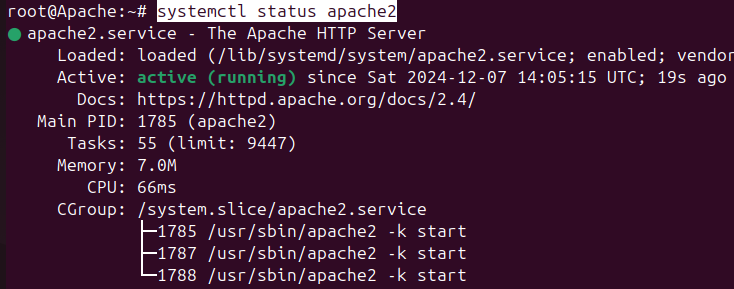


Рисунок 24. Статус апач сервера

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

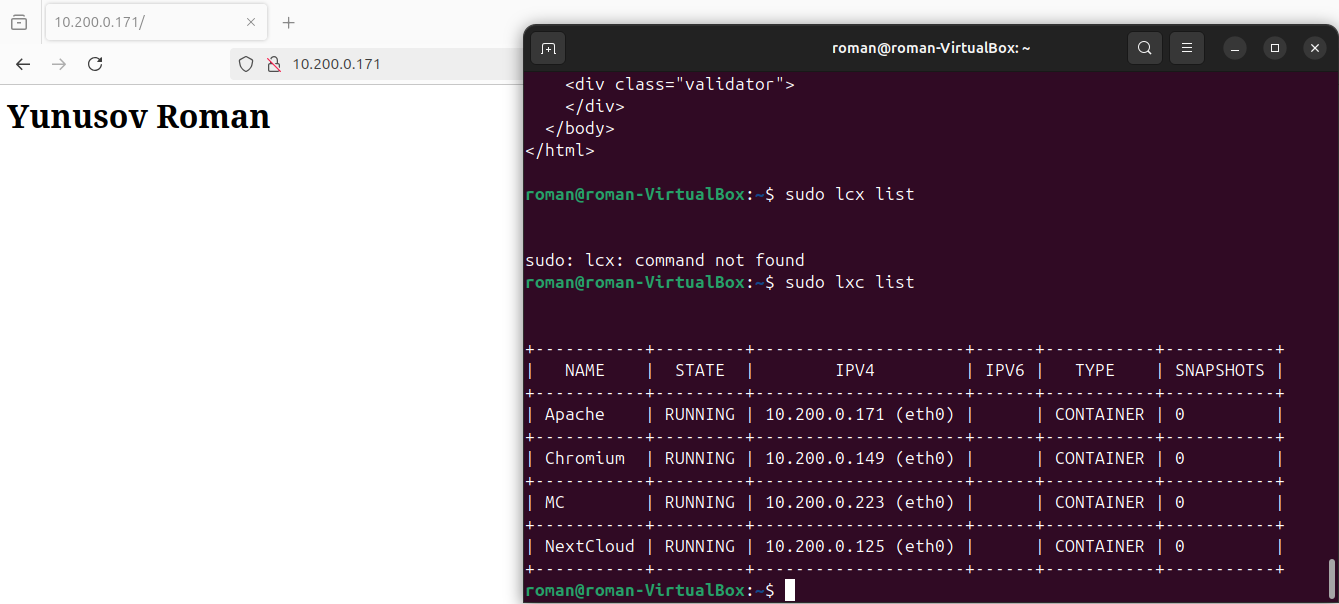


Рисунок 25. Подключение выполнено

1. 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

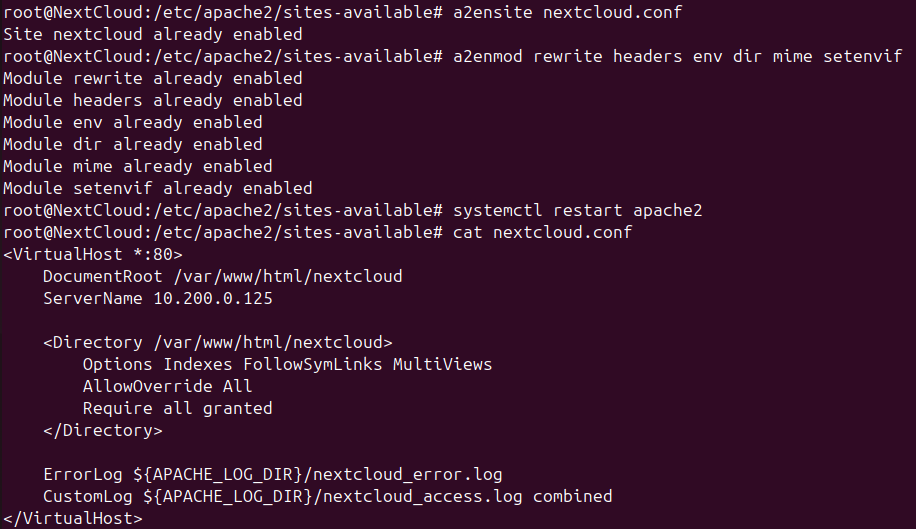


Рисунок 26. Результат конфигурация Nextcloud

Дальше нужно перезапустить новую апач с новой конфигурацией

a2ensite nextcloud.conf

a2enmod rewrite headers env dir mime setenvif

systemctl restart apache2

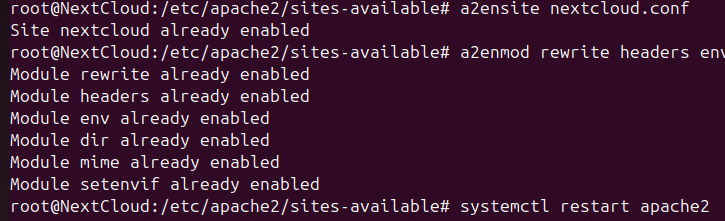


Рисунок 27. Результат перезапуска

Готово теперь можно подключаться по нашему адресу

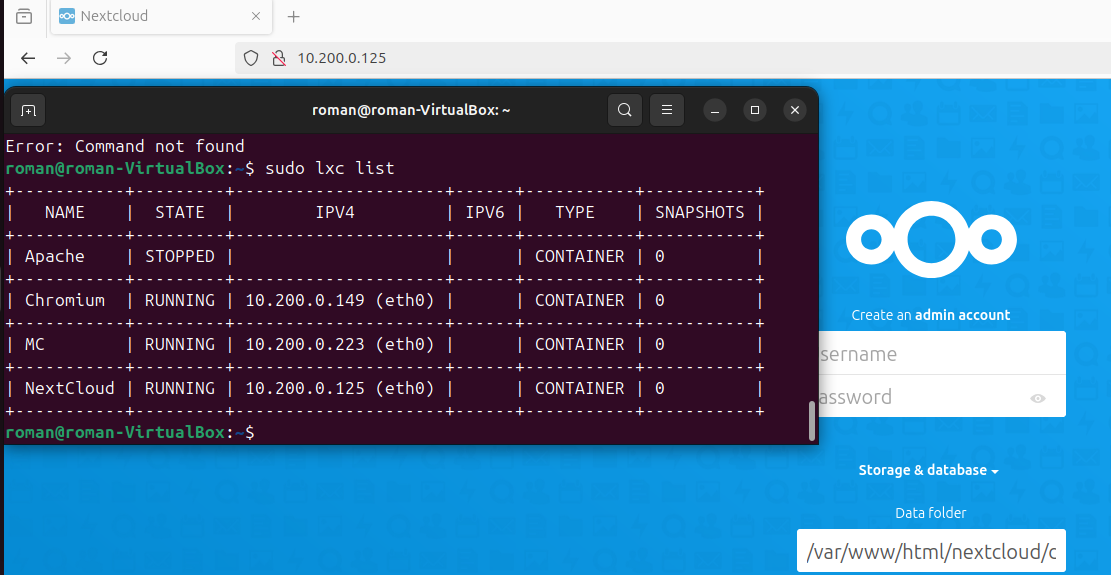


Рисунок 28. Подключение по Ip адресу контейнера

1. 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

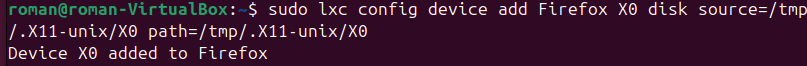


Рисунок 29. Добавление девайса, который подменяет адреса Xserverа на контейнере на Xserver хоста

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)

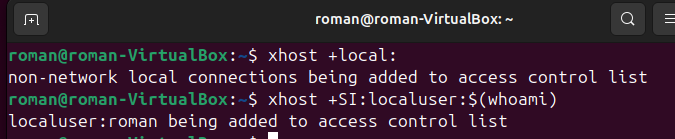


Рисунок 30. Разрешаем работу Xserveра

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

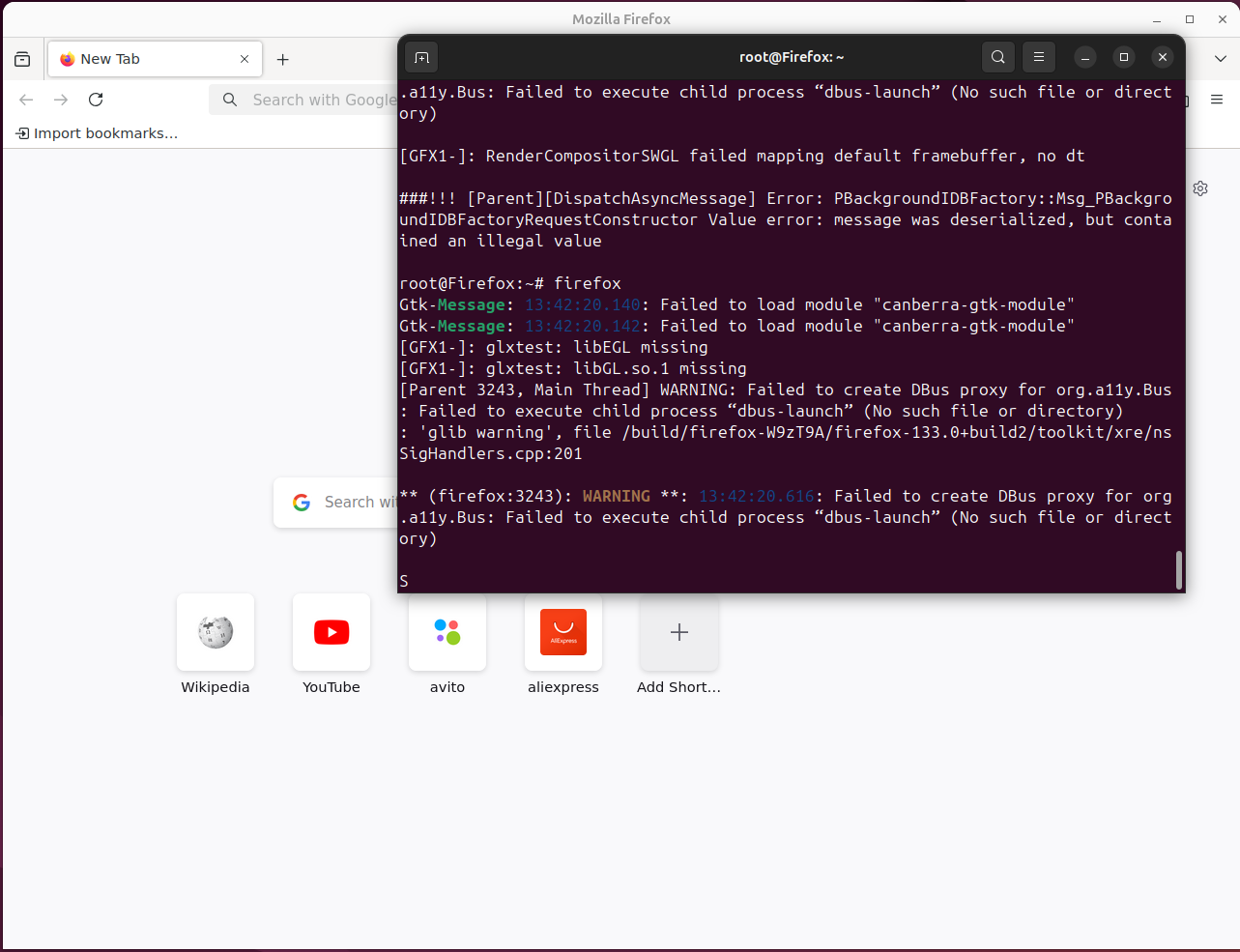


Рисунок 31. Запуск firefox

1. 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

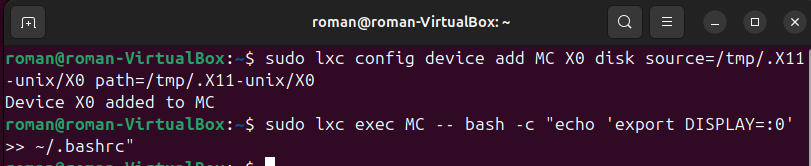


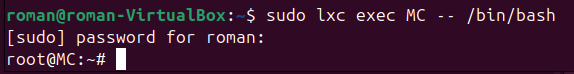
Рисунок 32. Результат установки переменных

Дальше заходим внутрь контейнера, устанавливаем и запускаем midnightcommander

sudo lxc exec MC -- /bin/bash

sudo apt update && sudo apt install mc

mc

 Рисунок 33. Заход в bash на контейнере

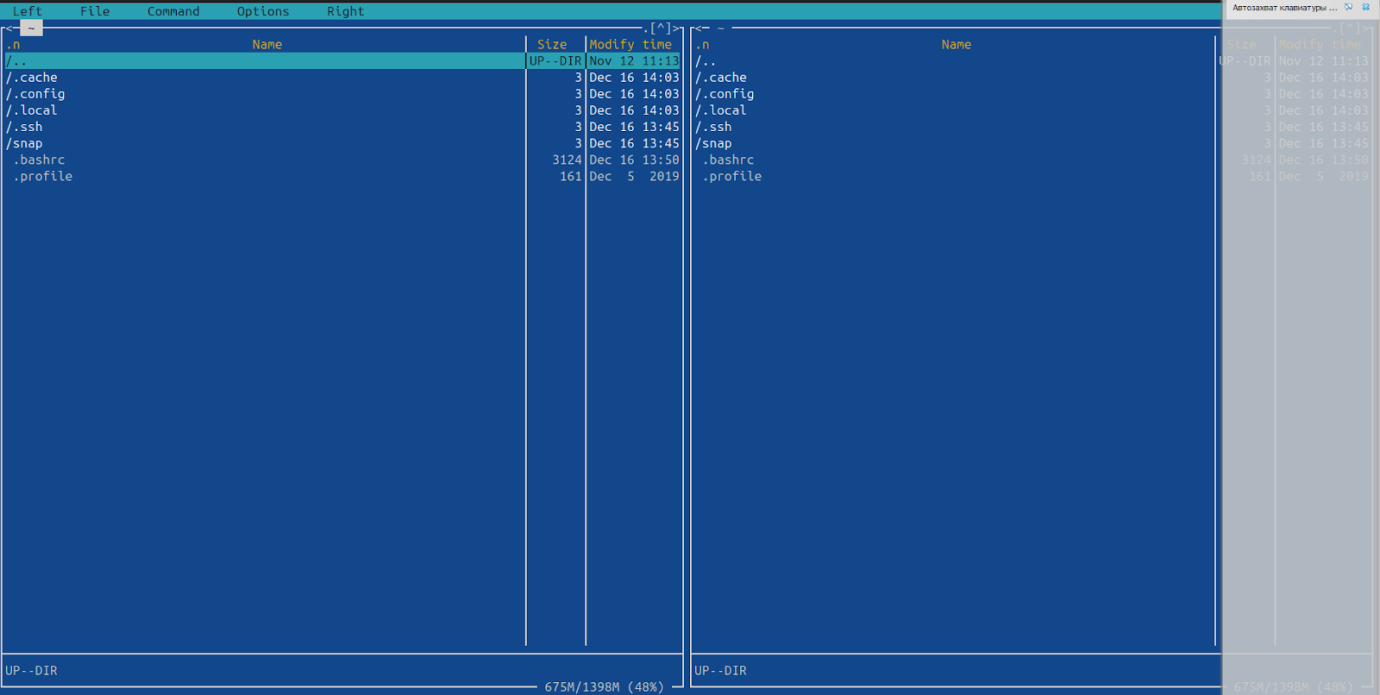


Рисунок 34. Результат установки Mc

**Анализ результатов**

В результате выполнения задания были достигнуты следующие результаты:

1. Успешно запущен сервер Apache в контейнере:
2. Успешно запущена программа Nextcloud:
3. Настроен проброс портов с хост-машины (порт 8080) внутрь контейнера (порт 80)
4. Успешно запущены графические программы Apache и MC:
5. Был смонтирован адрес апач сервера на хосте внутрь контейнера
6. 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 произведенена настройка контейнеров по разным параметром, 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.