

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 3

Установка виртуальных гипервизоров в Unix

Группа: Р34102

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	5
Установка виртуальных гипервизоров в Unix.....	7
Текст задания.....	7
Выполнение	7
Работа с Gnome Boxes.....	10
Текст задания.....	10
Выполнение	11
Работа с VVM	25
Текст задания.....	25
Выполнение	26
Использование утилиты virt-install.....	64
Текст задания.....	64
Выполнение	64
Заключение	71
Список литературы	75

Введение

Современные технологии виртуализации играют ключевую роль в области информационных технологий, позволяя эффективно управлять ресурсами, упрощать разработку и тестирование программного обеспечения, а также оптимизировать работу серверных систем. Виртуализация позволяет создавать изолированные среды, которые могут работать независимо друг от друга, что особенно важно в условиях растущих требований к масштабируемости и гибкости ИТ-инфраструктуры.

В данной работе рассматриваются основные инструменты и методы виртуализации, доступные в операционных системах Unix-подобных систем. В частности, основное внимание уделено установке и настройке виртуальных гипервизоров, работе с популярными инструментами управления виртуальными машинами, такими как Gnome Boxes, Virt-Manager и VVM, а также использованию утилиты virt-install для автоматизации процесса создания виртуальных машин.

Целью работы является изучение и практическое применение различных подходов к виртуализации, а также анализ возможностей и ограничений каждого из используемых инструментов. В ходе выполнения работы будут рассмотрены основные этапы установки и настройки виртуальных машин, работа с сетевыми настройками, управление ресурсами и создание снимков состояния системы.

Описание работы и инструментов

В данной работе используются различные инструменты и технологии виртуализации, которые позволяют создавать, настраивать и управлять виртуальными машинами в операционных системах Unix-подобных систем. Основные инструменты, которые применяются в работе, включают:

1. VirtualBox:

- Описание: VirtualBox — это кроссплатформенный гипервизор, который позволяет создавать и управлять виртуальными машинами на различных операционных системах. Он поддерживает широкий спектр гостевых операционных систем, включая Windows, Linux, macOS и другие.

- Функциональность: в рамках работы с VirtualBox были выполнены задачи по добавлению жесткого диска, созданию общей папки между хостовой и гостевой ОС.

2. Gnome Boxes:

- Описание: Gnome Boxes — это упрощенный инструмент для управления виртуальными машинами, который использует QEMU/KVM в качестве гипервизора. Он предоставляет графический интерфейс для создания и управления виртуальными машинами.
- Функциональность: в работе с Gnome Boxes были выполнены задачи по установке приложения, созданию виртуальной машины, изучению её свойств, а также подключению к удаленным рабочим столам.

3. Virt-Manager (Virtual Machine Manager):

- Описание: Virt-Manager — это мощный инструмент для управления виртуальными машинами, который использует QEMU/KVM. Он предоставляет графический интерфейс для создания, настройки и мониторинга виртуальных машин.
- Функциональность: в рамках работы с Virt-Manager были выполнены задачи по установке QEMU, созданию виртуальной машины, настройке её параметров, а также управлению снимками состояния системы.

4. QEMU:

- Описание: QEMU — это эмулятор и гипервизор, который позволяет создавать и запускать виртуальные машины. Он поддерживает аппаратную виртуализацию через KVM.
- Функциональность: В работе с QEMU были выполнены задачи по созданию виртуальной машины, настройке параметров загрузки и сетевых настроек.

5. virt-install:

- Описание: virt-install — это утилита командной строки, которая позволяет автоматизировать процесс создания виртуальных машин с использованием QEMU/KVM.
- Функциональность: В рамках работы с virt-install были выполнены

задачи по установке необходимых пакетов, проверке доступных операционных систем, созданию виртуальной машины и подключению к ней по SSH.

6. libvirt:

- Описание: libvirt — это API и инструменты для управления виртуализацией, которые поддерживают различные гипервизоры, включая QEMU/KVM, Xen и другие.
- Функциональность: В работе с libvirt были выполнены задачи по проверке статуса службы libvirtd, добавлению пользователя в группу libvirt, а также управлению виртуальными машинами через командную строку с использованием утилиты virsh.

7. VNC и SSH:

- Описание: VNC (Virtual Network Computing) и SSH (Secure Shell) используются для удаленного доступа к виртуальным машинам. VNC позволяет управлять графическим интерфейсом, а SSH — консолью.
- Функциональность: В работе с VNC и SSH были выполнены задачи по подключению к виртуальным машинам, настройке удаленного доступа и управлению машинами через командную строку.

8. Snapshot-ы:

- Описание: Snapshot-ы — это снимки состояния виртуальной машины, которые позволяют сохранить текущее состояние системы и восстановить его в случае необходимости.
- Функциональность: В работе с snapshot-ами были выполнены задачи по созданию снимков, восстановлению состояния системы и проверке их работоспособности.

Цели и задачи

Целью работы является изучение и практическое применение различных инструментов и технологий виртуализации в Unix-подобных системах. Основные задачи включают:

1. Установку и настройку виртуальных гипервизоров.
2. Работу с VirtualBox для создания и настройки виртуальных машин.

3. Использование Gnome Boxes для управления виртуальными машинами и подключения к удаленным рабочим столам.
4. Работу с Virt-Manager для создания, настройки и управления виртуальными машинами.
5. Использование QEMU и virt-install для автоматизации процесса создания виртуальных машин.
6. Управление виртуальными машинами через командную строку с использованием libvirt и virsh.

Установка виртуальных гипервизоров в Unix

Текст задания

- Добавить Жесткий диск (SATA) объемом 30 Gb в гостевую ОС Ubuntu
- Проверить конфигурацию гостевой ОС:
- Создать общую папку, общую с основной и гостевой ОС, под названием Soft
- Скачать дистрибутив любой ОС (лучше Live, с возможность запуска без установки) с адреса <http://mirror.yandex.ru>

Выполнение

- Добавление жесткого диска (SATA) объемом более 30Гб в гостевую ОС Ubuntu

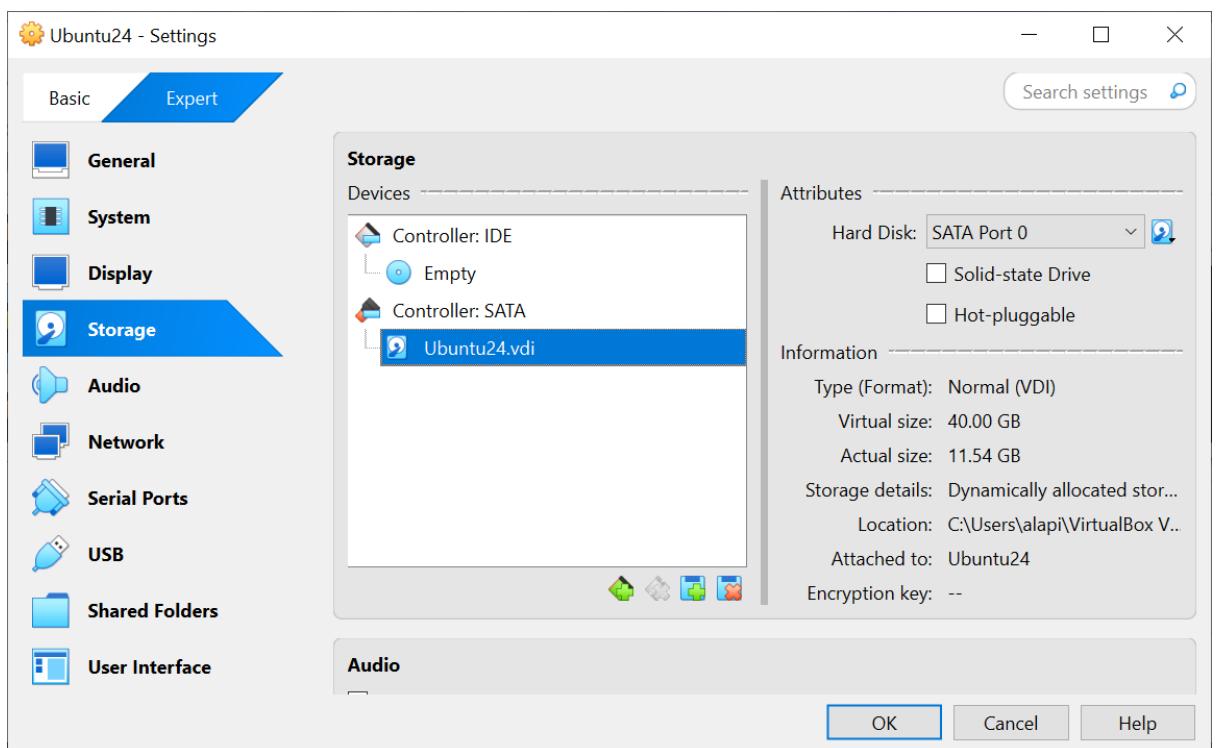


Рисунок 1. Добавление жесткого диска (SATA) объемом более 30Гб в гостевую ОС Ubuntu

Проверка конфигурации гостевой ОС Ubuntu.

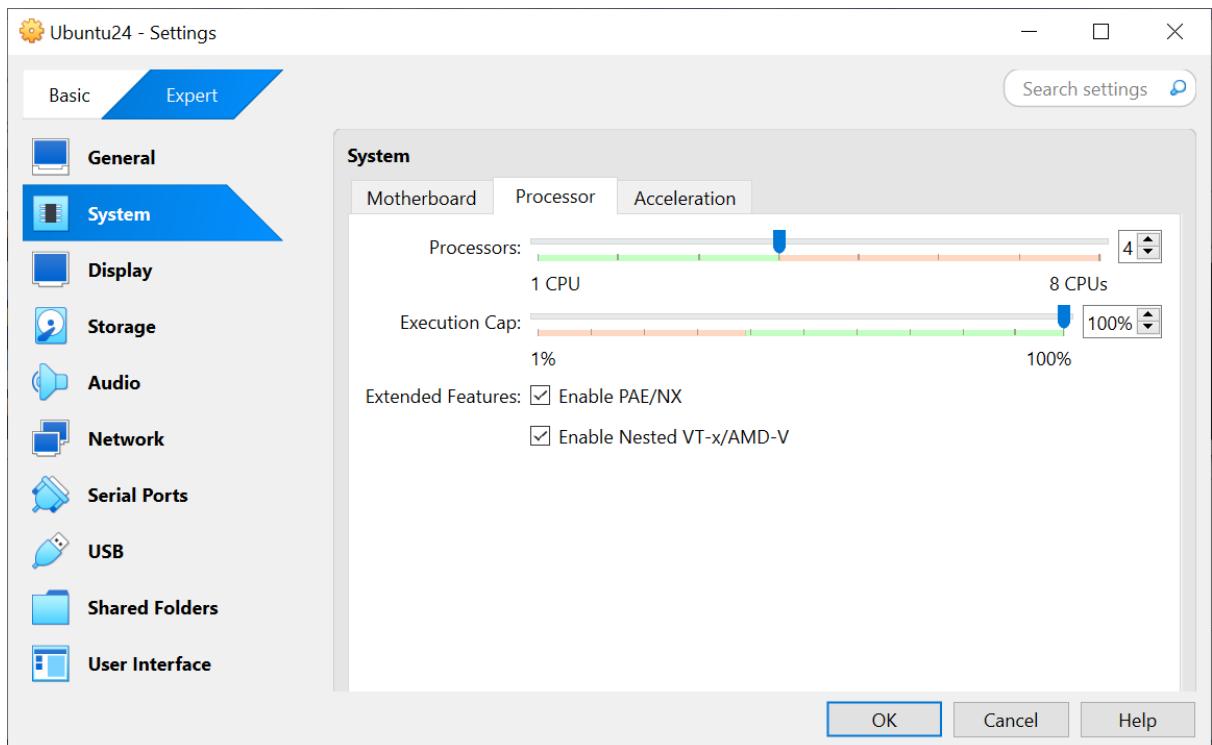


Рисунок 2. Проверка конфигурации процессора гостевой ОС Ubuntu

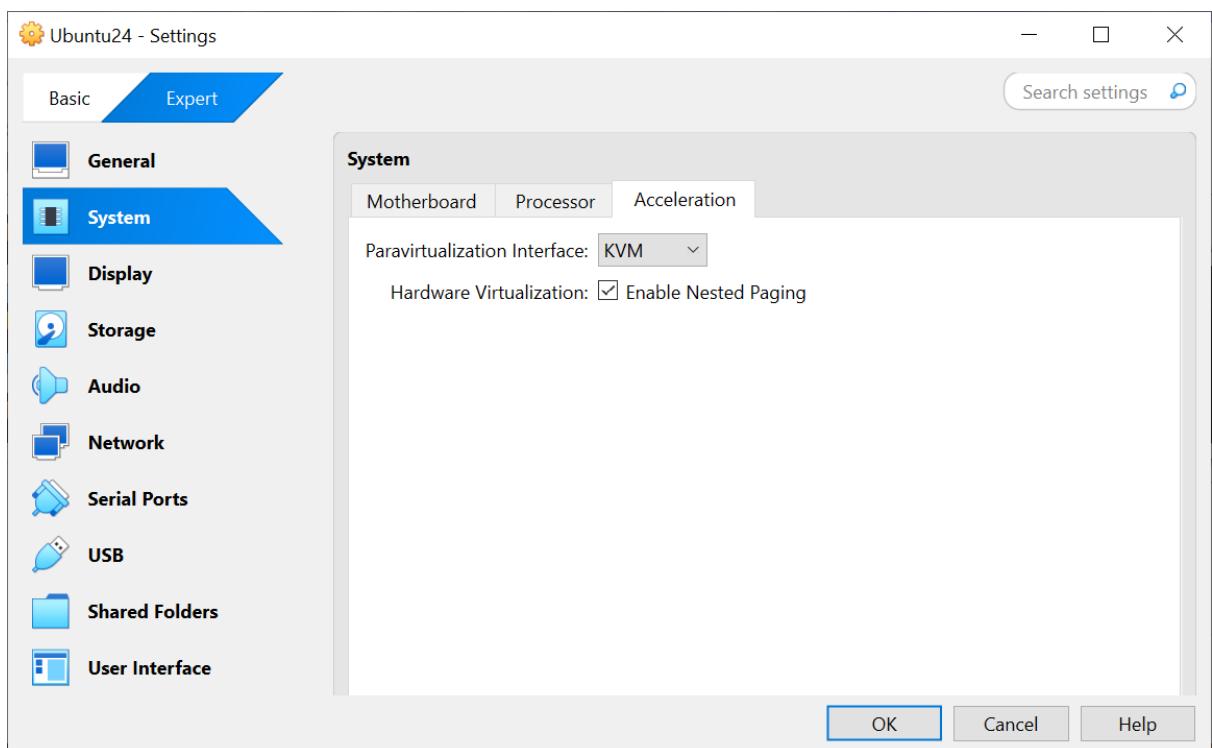


Рисунок 3. Проверка конфигурации ускорения гостевой ОС Ubuntu.

Создадим общую папку Soft между гостевой ОС Ubuntu и хостом.

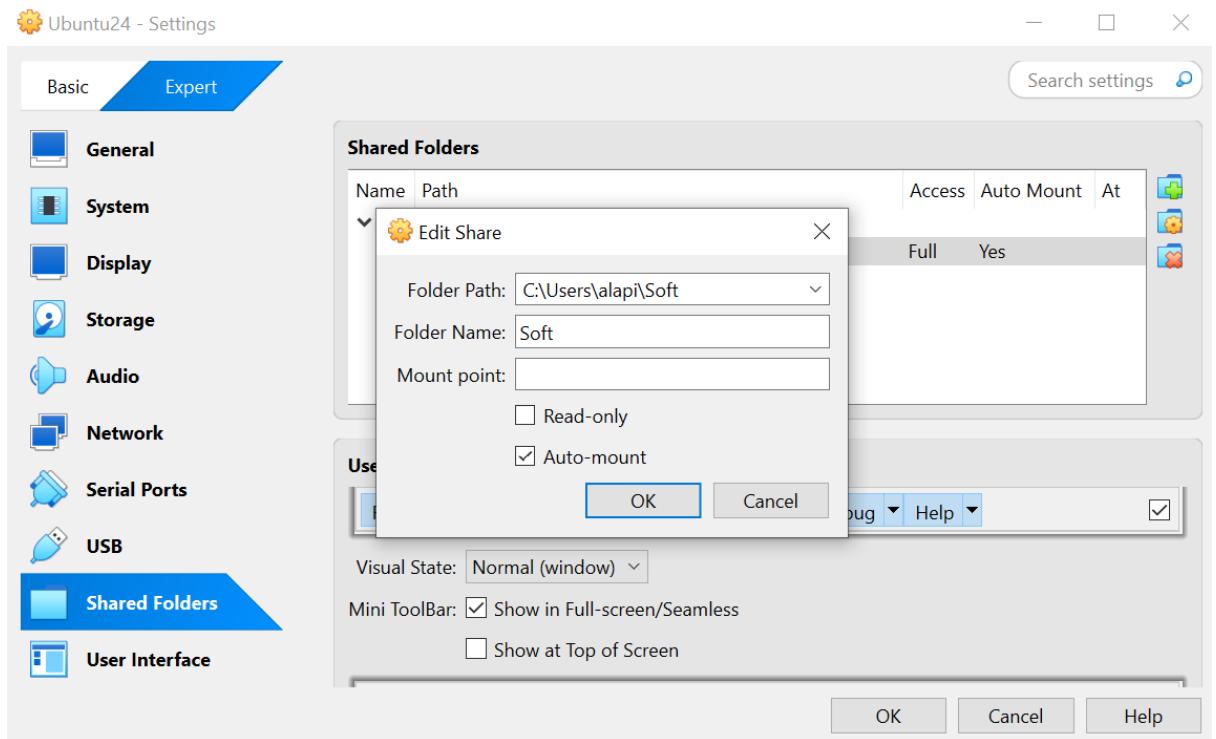


Рисунок 4. Создание общей папки между хостом и гостевой ОС Ubuntu.

Примечание: для работы общих папок необходимо установить образ дополнений гостевой ОС. Кроме того необходимо добавить текущего пользователя в группу vboxsf.

```
sudo usermod -aG vboxsf $USER
```

Скачаем Live образ Ubuntu 24.04 Desktop с официального сайта. Положим его в общую папку.

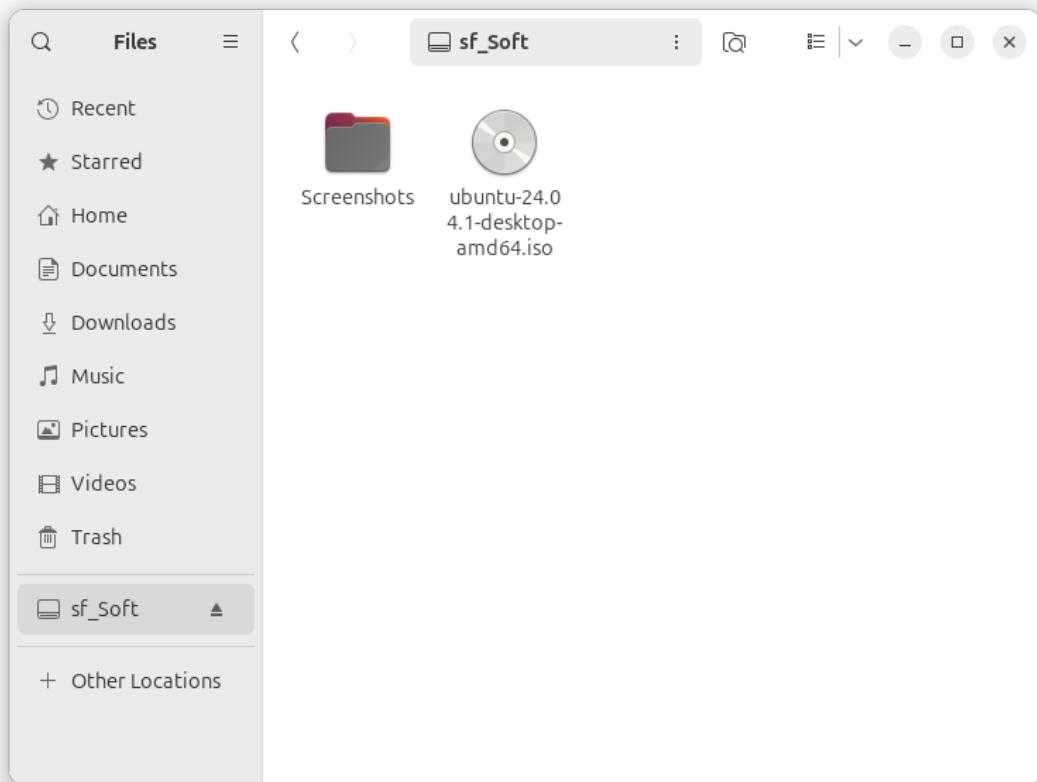


Рисунок 5. Live образ Ubuntu 24.04 Desktop в общей папке Soft.

Работа с Gnome Boxes

Текст задания

1. Установить в гостевой ОС приложение Gnome Boxes
2. Запустите Boxes и создайте виртуальную машину
3. Выберите ISO-образ диска с гостевой ОС
4. Запустите ОС в режиме LiveCD
5. Изучите свойства виртуальной машины, созданной в Boxes, какие изменения в работе виртуальной машины можно производить? Какие параметры можно отслеживать в оперативном режиме?
6. Создайте удаленное подключение к серверу Helios
7. Проверьте подключение
8. Проверьте, что у вашей гостевой системы стоят настройки сетевого подключения Сетевой мост.
9. Попросите вашего соседа или напарника предоставить доступ к

удаленному рабочему столу в ОС Windows (смотри скрин ниже).

Получите у него следующие сведения: IP-адрес, логин и пароль.

10. Подключитесь у удаленному компьютеру
11. Проверьте подключение (если нет возможности подключиться к соседу, настройте подключение к свой Хостовой машине)

Выполнение

- Установим gnome-boxes командой:

```
sudo apt install gnome-boxes
```

- Проверим работу KVM

```
sudo kvm-ok
```

- В GNOME Boxes под капотом используется QEMU/KVM. Проверим используется ли QEMU.

```
ps aux | grep qemu
```

Если QEMU используется, то мы найдем записи о ней.

- Проверим Nested KVM поддержку.

```
cat /sys/module/kvm_(intel|amd)/parameters/nested
```

На выходе должно быть Y или 1, что указывает на то, что Nested KVM включен.

- Проверить работу других компонентов gnome boxes можно командой:

```
gnome-boxes --checks
```

Примечание: Ошибка No KVM не равно проблема в KVM. Эта ошибка ни о чем не говорит, надо искать самим. (NoKVM)

Начать поиск можно с раздела Issues в официальном репозитории GNOME Boxes. (GNO24)

- Запустим и создадим виртуальную машину из скаченного образа.

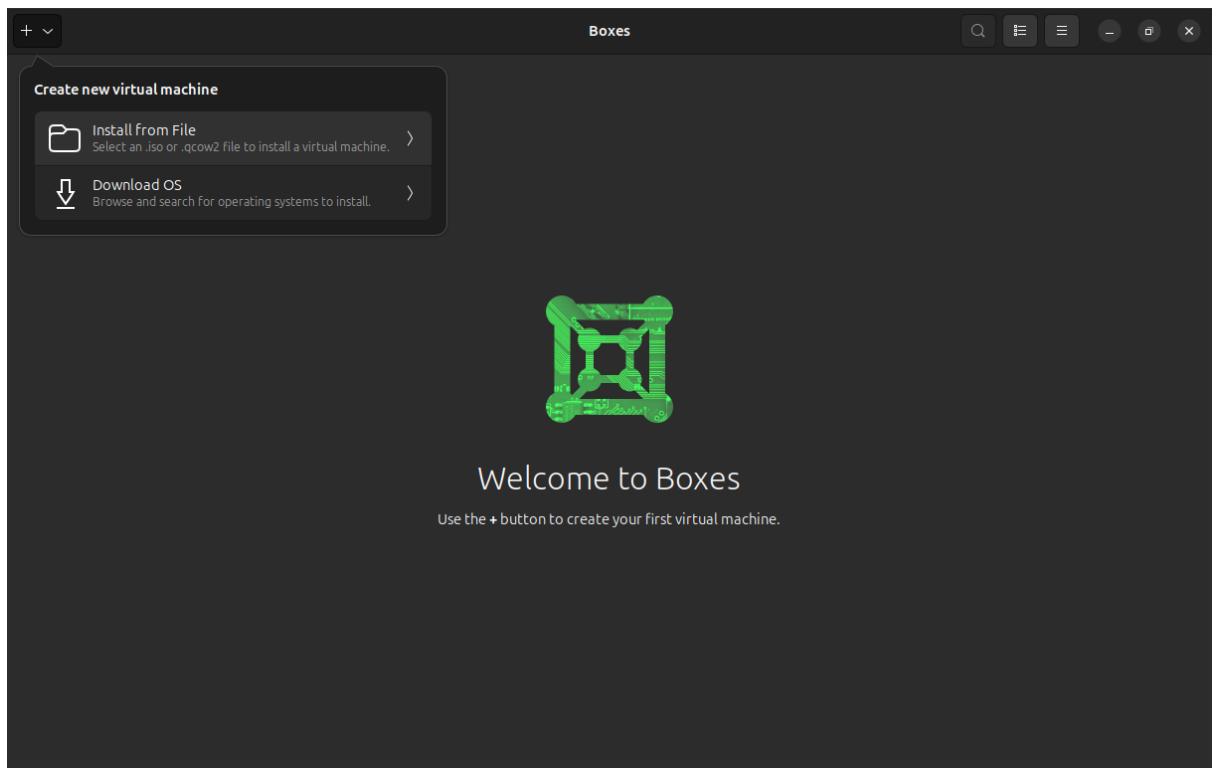


Рисунок 6. Установка виртуальной машины из файла.

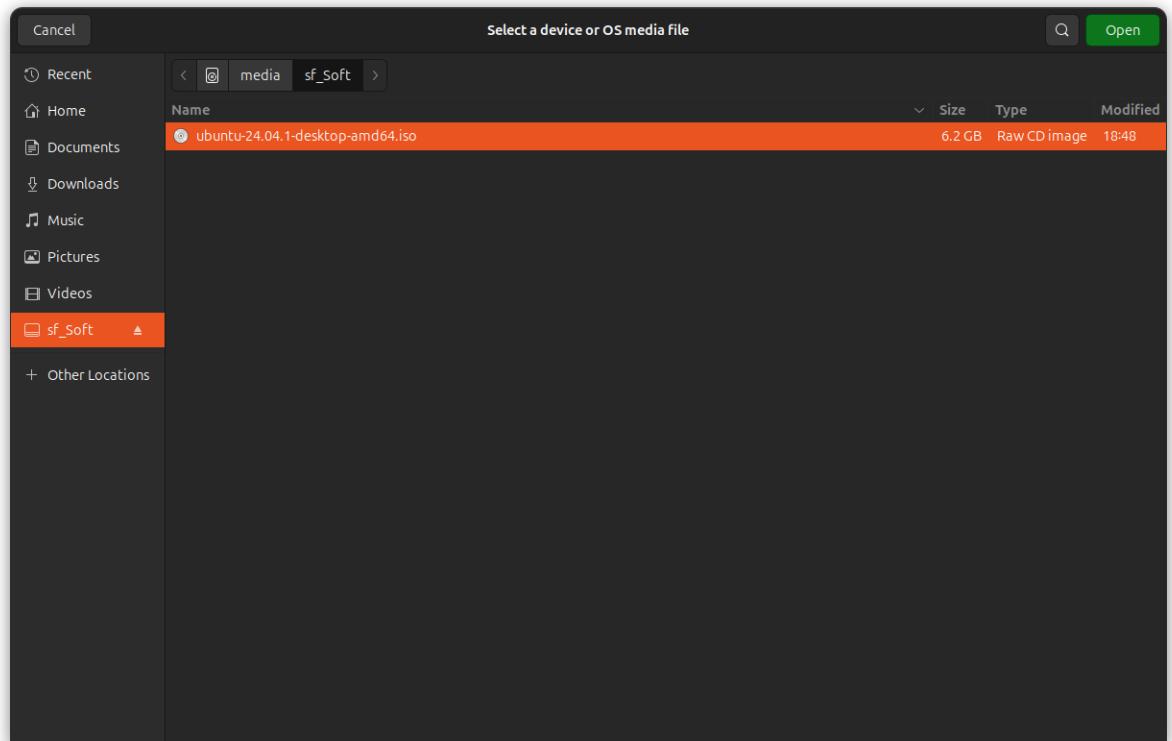


Рисунок 7. Выбор образа виртуальной машины.

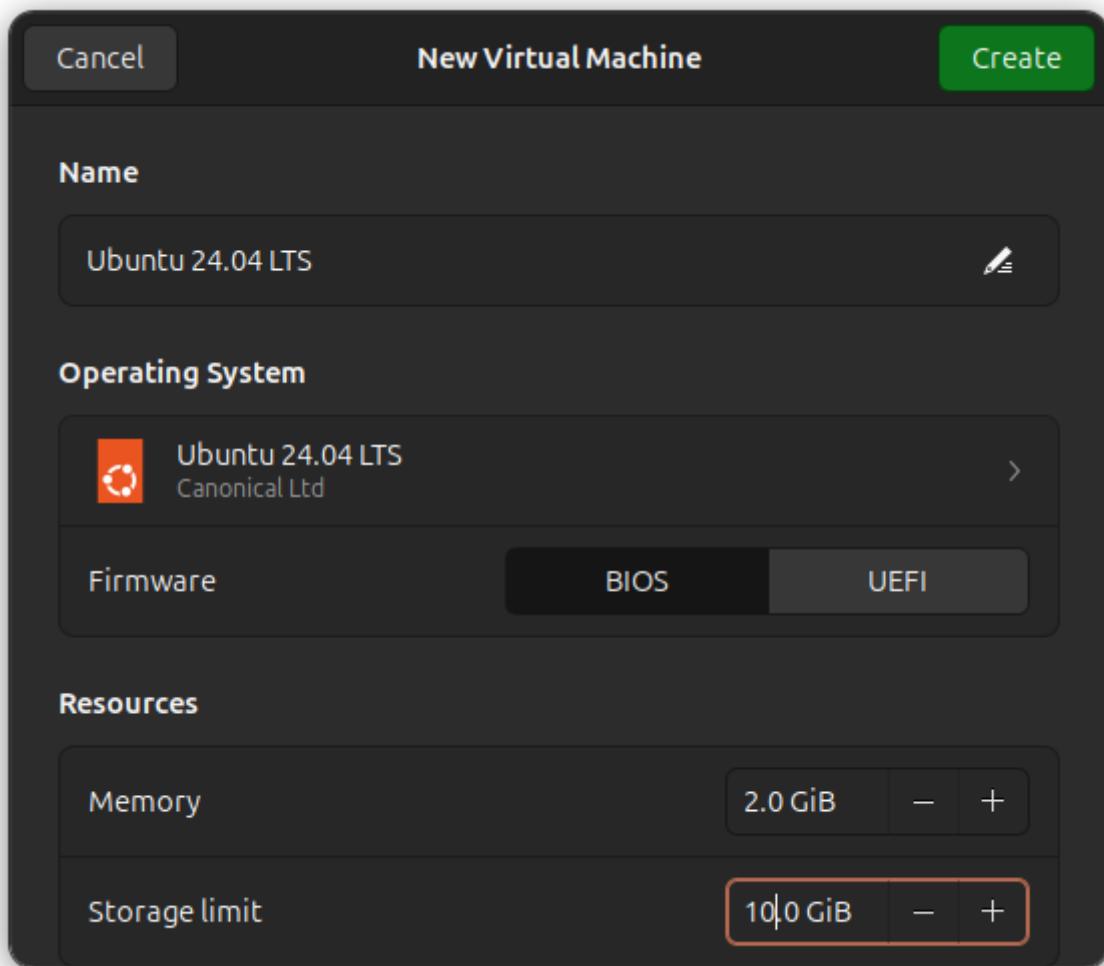


Рисунок 8. Конфигурация создаваемой виртуальной машины.

- Конфигурация дистрибутива.

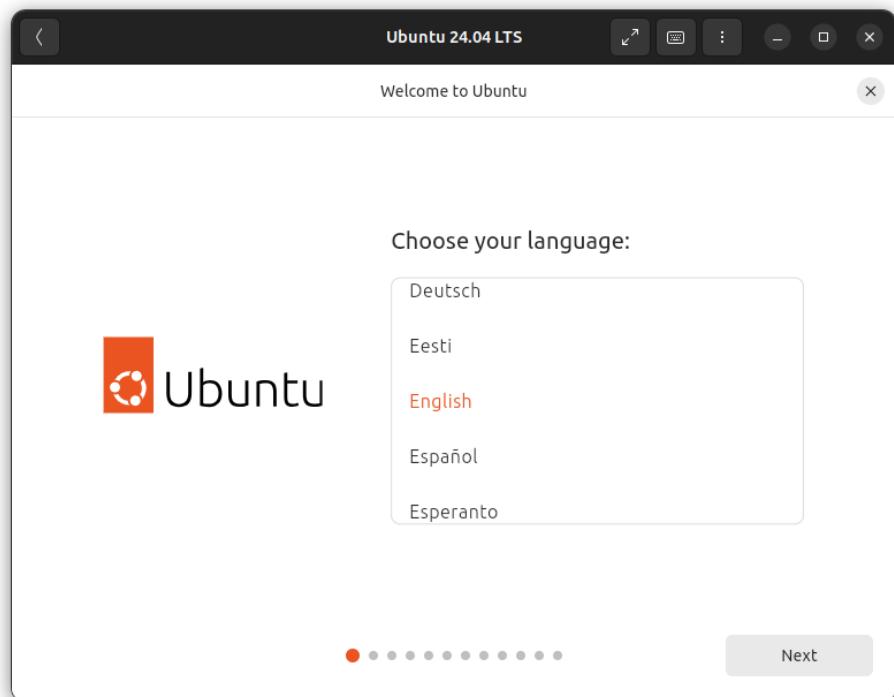


Рисунок 9. Выбираем язык английский (с ним точно проблем не будет).

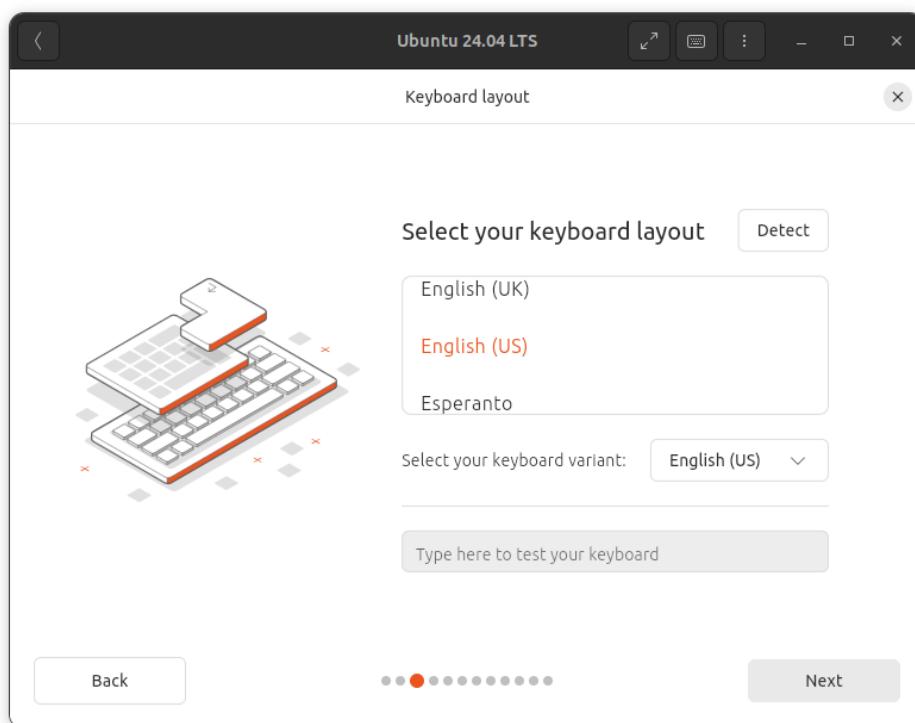


Рисунок 10. Выбираем раскладку клавиатуры.

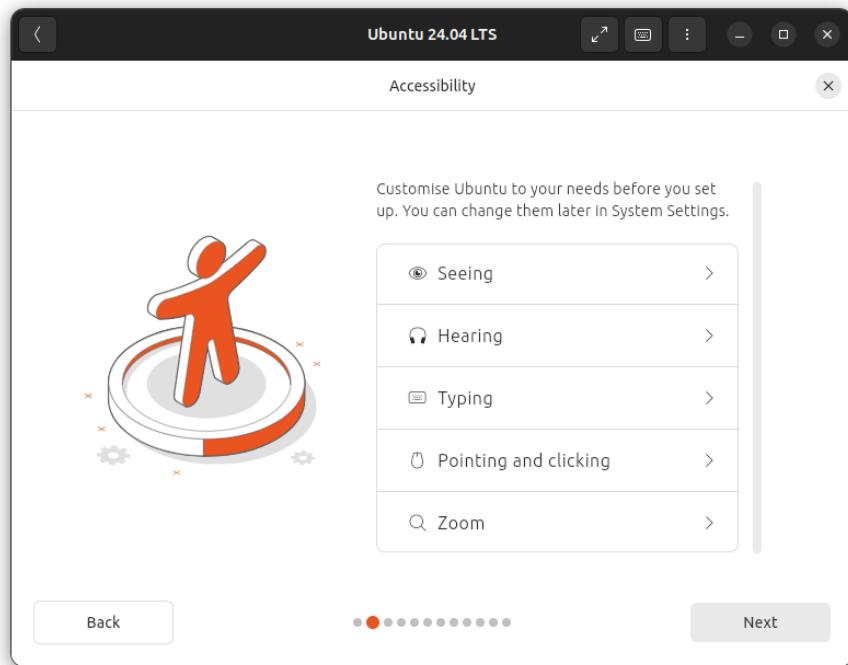


Рисунок 11. Выбираем под себя раздел доступности.

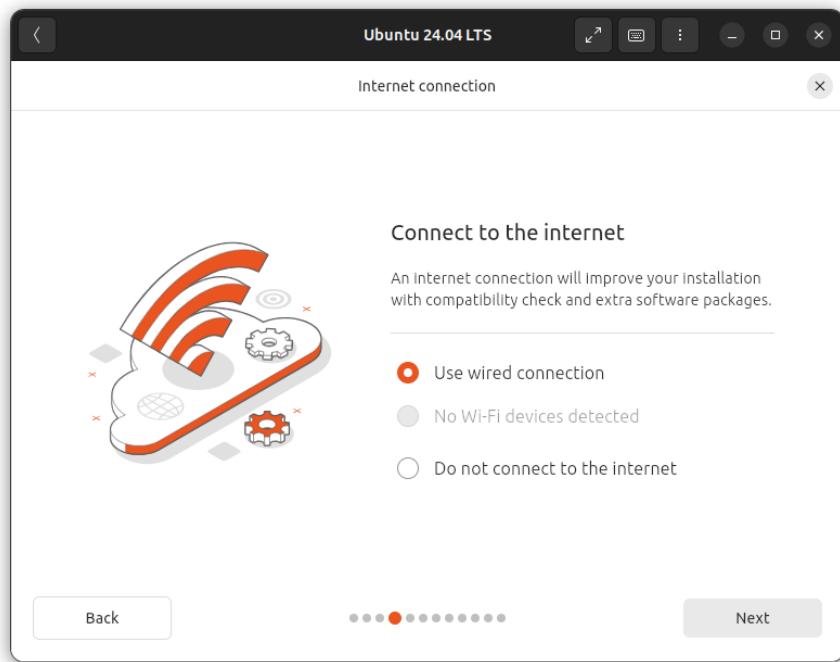


Рисунок 12. Подключаемся к интернету.

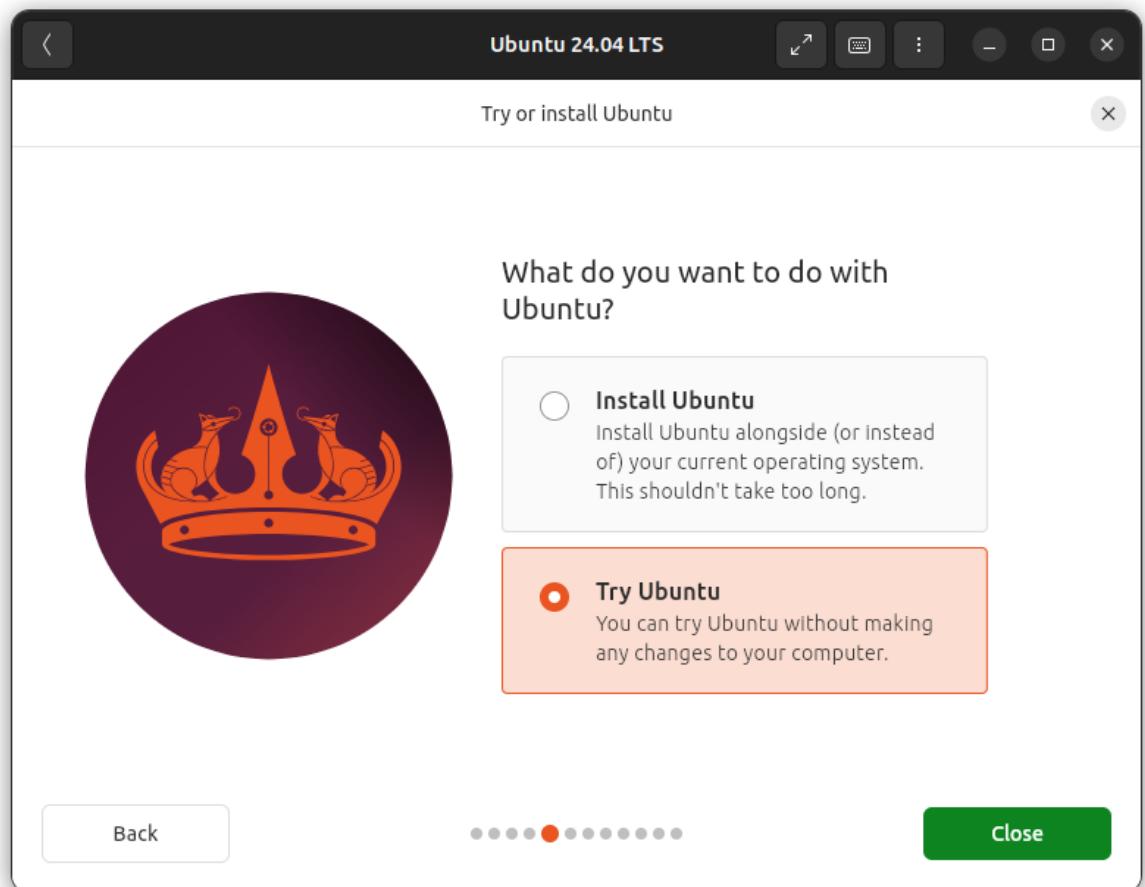


Рисунок 13. Устанавливать Ubuntu не будем, только попробуем.

- Проверим работоспособность.

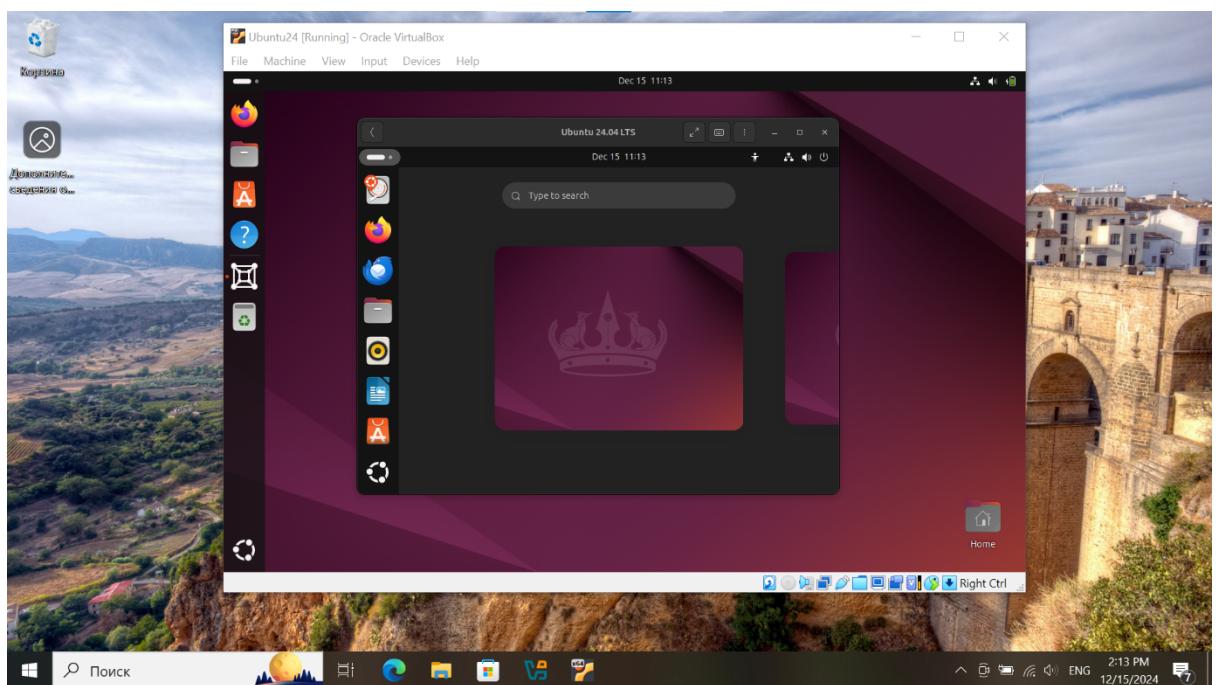


Рисунок 14. Проверка работоспособности.

- Свойства виртуальной машины. Изменения и доступные параметры виртуальной машины.

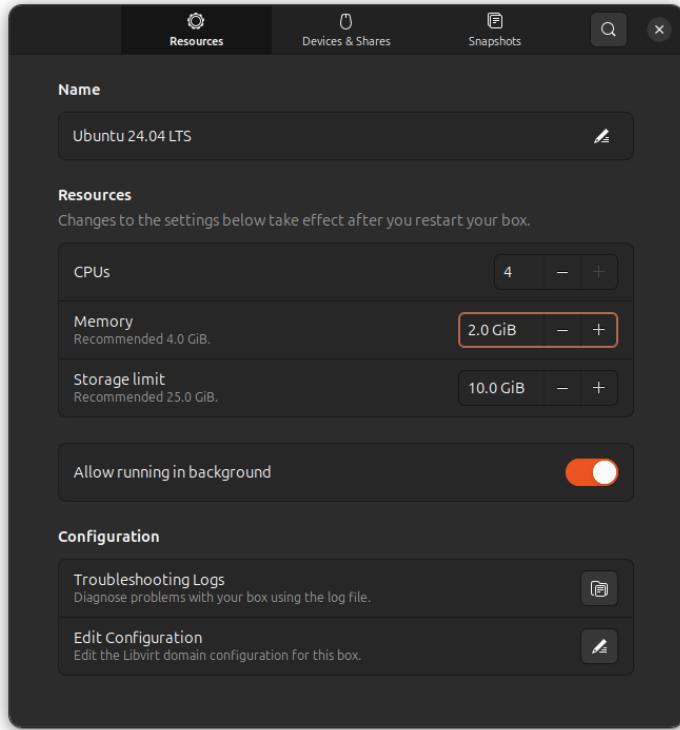


Рисунок 15. Свойства ресурсов виртуальной машины в Gnome Boxes

Для изменения доступны:

- Имя виртуальной машины
- Количество процессоров
- Количество памяти
- Максимальный объем хранилища
- Разрешить работать в фоновом режиме
- Конфигурация Libvirt для умных пользователей.

Эти настройки можно менять во время работы виртуальной машины.

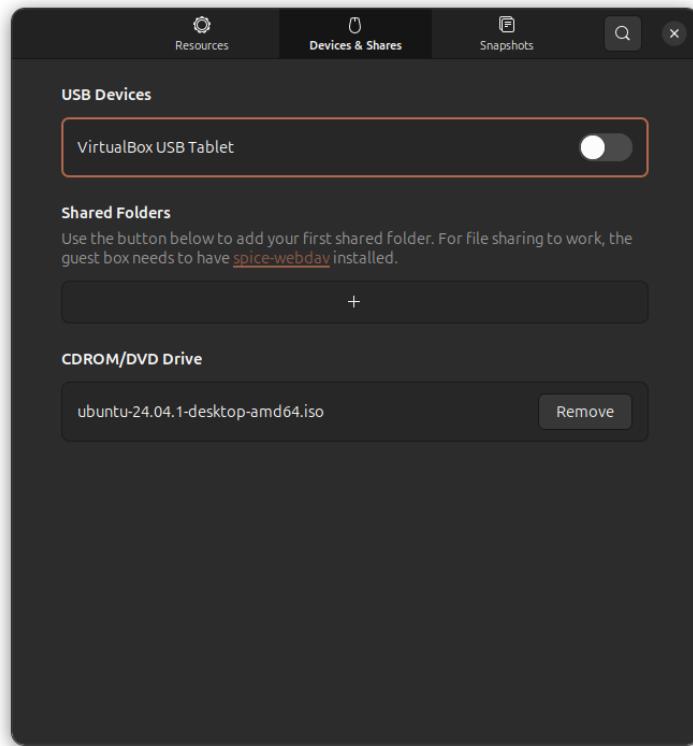


Рисунок 16. Устройства и общие папки

В этом разделе можно настроить USB девайсы, общие папки и управлять CDROM/DVD дисками.

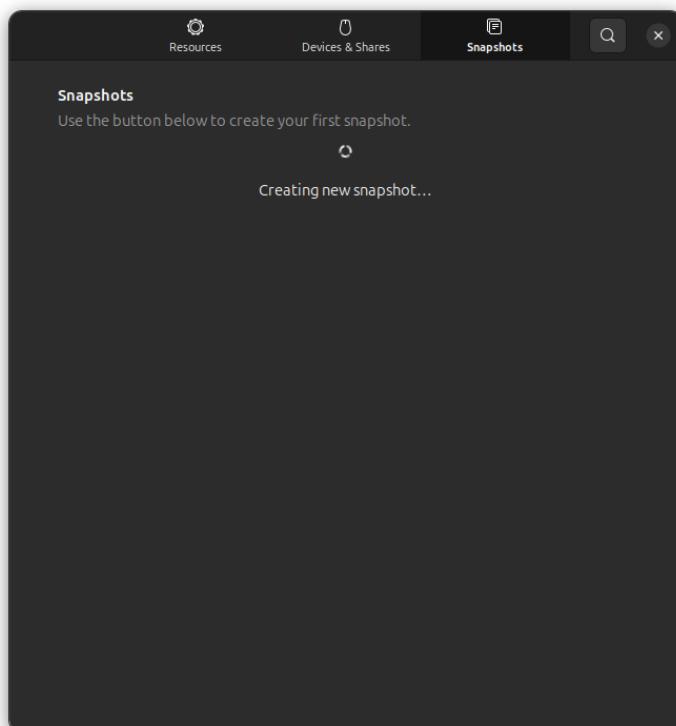


Рисунок 17. Раздел Snapshots.

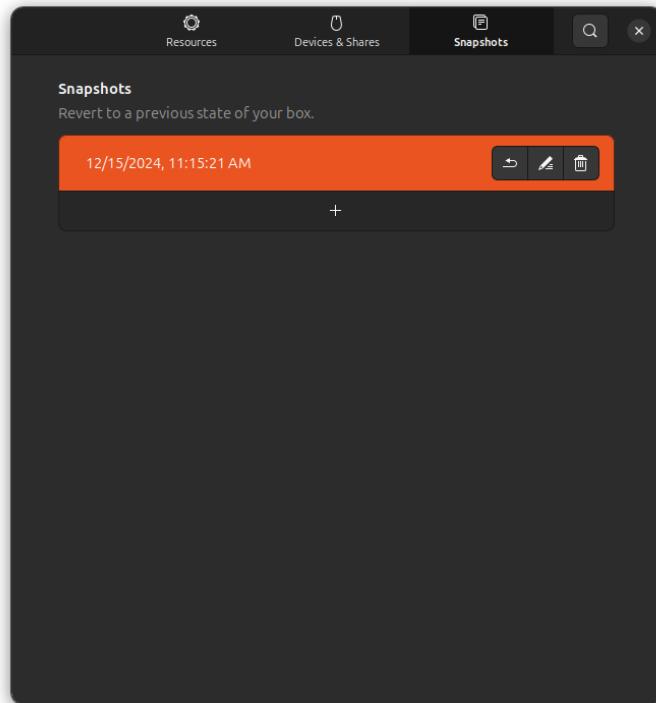


Рисунок 18. Созданный snapshot виртуальной машины.

В данном разделе можно создавать Snapshot-ы виртуальной машины.

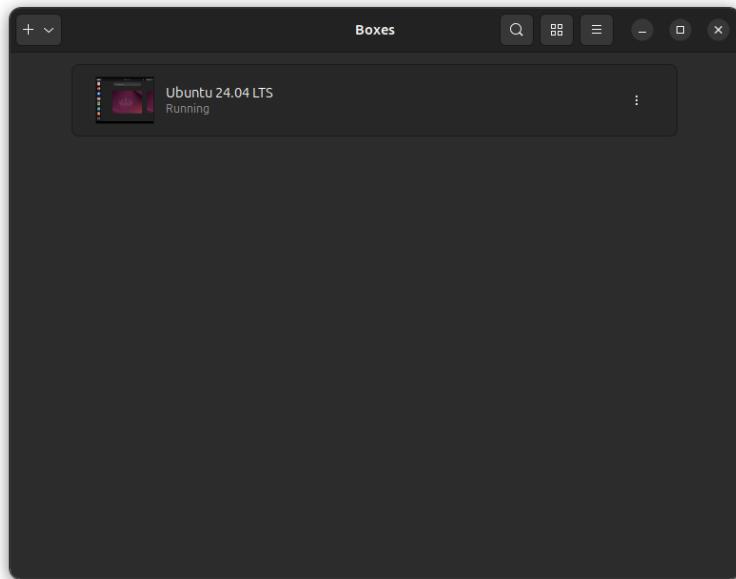


Рисунок 19. Dashboard виртуальных машин Gnome boxes.

- Подключимся по ssh к helios.

Такой функции нет в Gnome Boxes и Gnome Connections, так что подключимся к helios из виртуальной машины внутри Boxes, внутри VirtualBox.

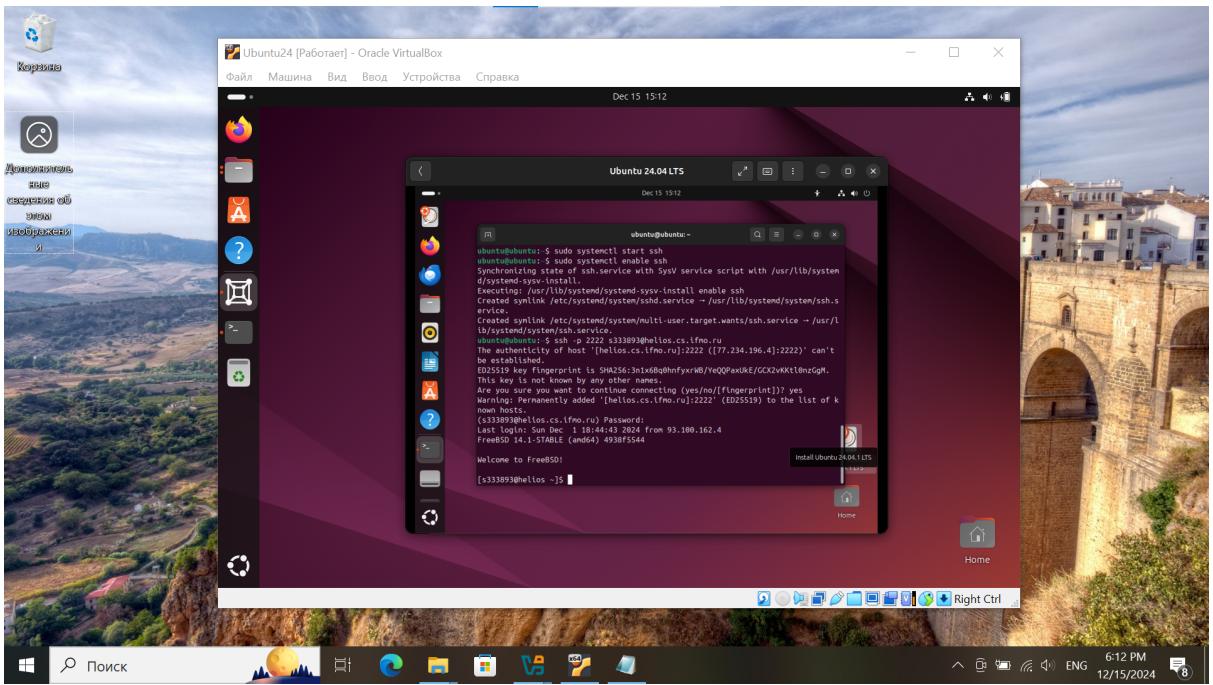


Рисунок 20. Подключение к helios по ssh.

- Проверим, что у нашей виртуальной машины настроен Сетевой мост.

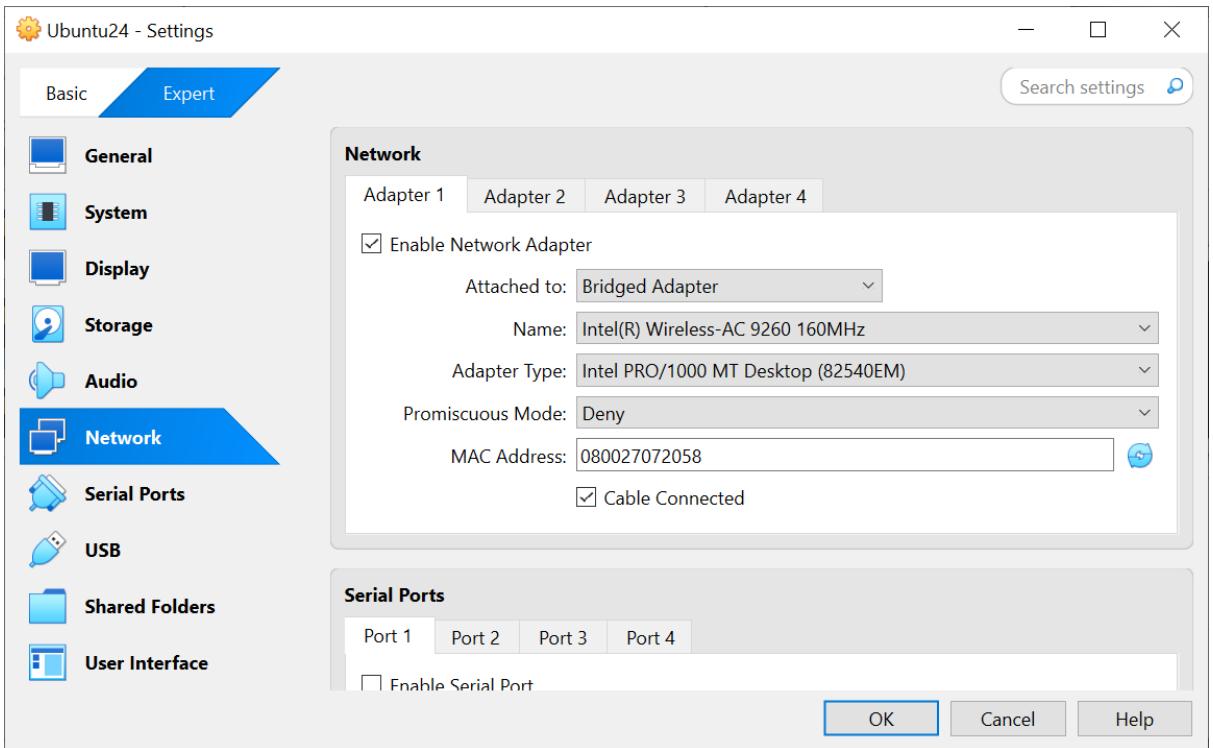


Рисунок 21. Сетевые настройки виртуальной машины.

- Включим удаленный рабочий стол в Window 10.

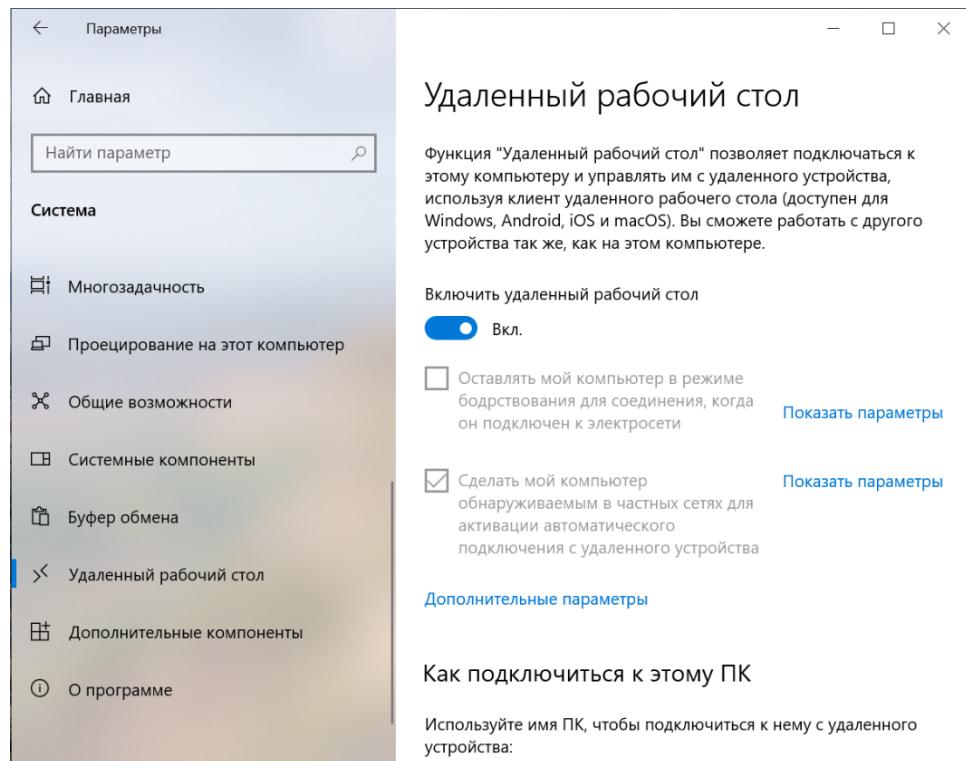


Рисунок 22. Настройка удаленного рабочего стола в Windows 10.

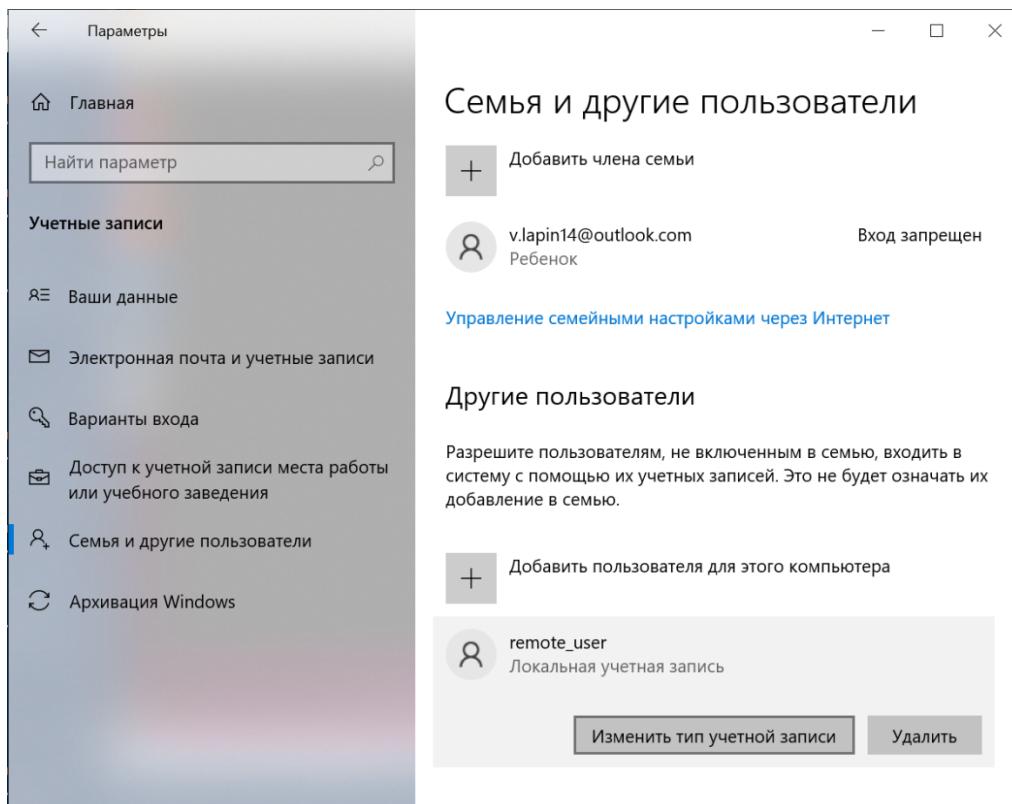


Рисунок 23. Добавим пользователя, к которому мы будем подключаться.

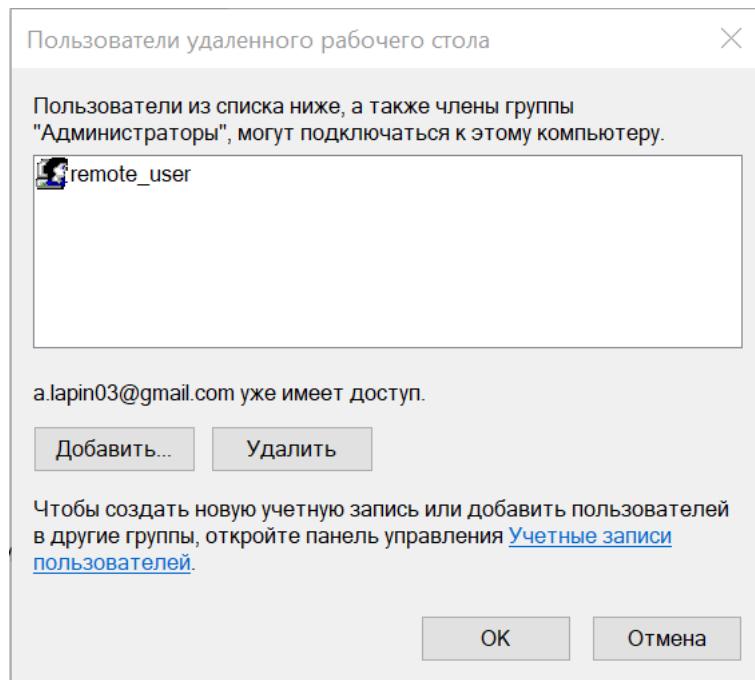


Рисунок 24. Предоставим этому пользователю подключаться к удаленному рабочему столу.

Узнаем IP адрес Windows 10.

```
Выбрать Windows PowerShell
Адаптер Ethernet Ethernet:
Состояние среды. . . . . : Среда передачи недоступна.
DNS-суффикс подключения . . . . . :
Адаптер Ethernet Ethernet 2:
Состояние среды. . . . . : Среда передачи недоступна.
DNS-суффикс подключения . . . . . :
Адаптер Ethernet Ethernet 3:
DNS-суффикс подключения . . . . . :
Локальный IPv6-адрес канала . . . . . : fe80::8215:877a:351:e6a7%9
IPv4-адрес . . . . . : 192.168.56.1
Маска подсети . . . . . : 255.255.255.0
Основной шлюз. . . . . :
Адаптер беспроводной локальной сети Подключение по локальной сети* 1:
Состояние среды. . . . . : Среда передачи недоступна.
DNS-суффикс подключения . . . . . :
Адаптер беспроводной локальной сети Подключение по локальной сети* 2:
Состояние среды. . . . . : Среда передачи недоступна.
DNS-суффикс подключения . . . . . :
Адаптер беспроводной локальной сети Беспроводная сеть:
DNS-суффикс подключения . . . . . :
Локальный IPv6-адрес канала . . . . . : fe80::d490:5c83:13c6:c92d%2
IPv4-адрес . . . . . : 192.168.0.127
Маска подсети . . . . . : 255.255.255.0
Основной шлюз. . . . . : 192.168.0.1
Адаптер Ethernet Сетевое подключение Bluetooth:
Состояние среды. . . . . : Среда передачи недоступна.
DNS-суффикс подключения . . . . . :
PS C:\Users\alapi>
```

Рисунок 25. IP адрес компьютера с Windows 10.

Gnome Boxes больше не поддерживает подключение к удаленным рабочим столам с версии 40.2. Этот функционал вынесен в отдельной приложение Gnome Connections. (issues735)

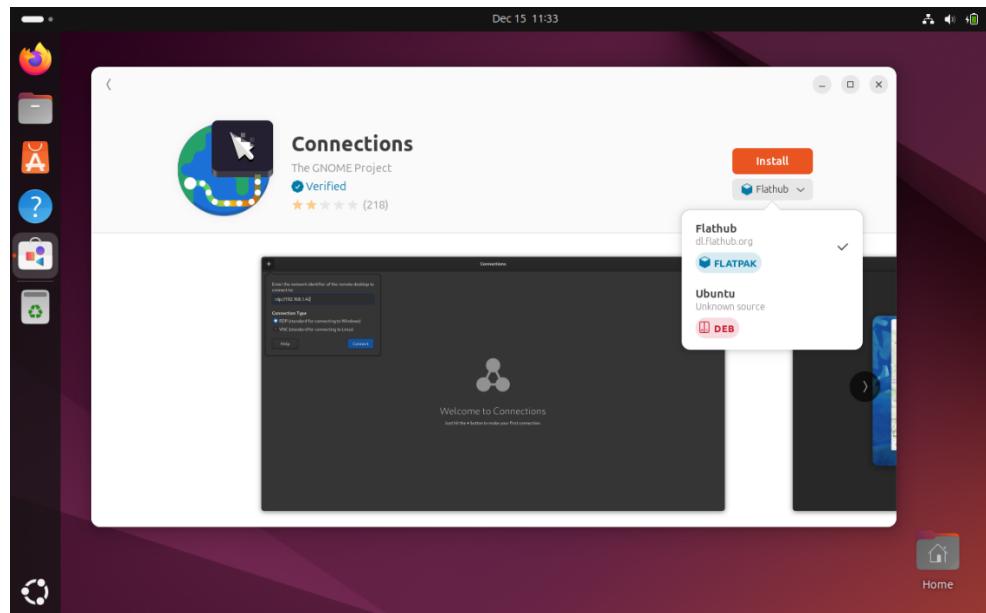


Рисунок 26. Приложение Gnome Connections.

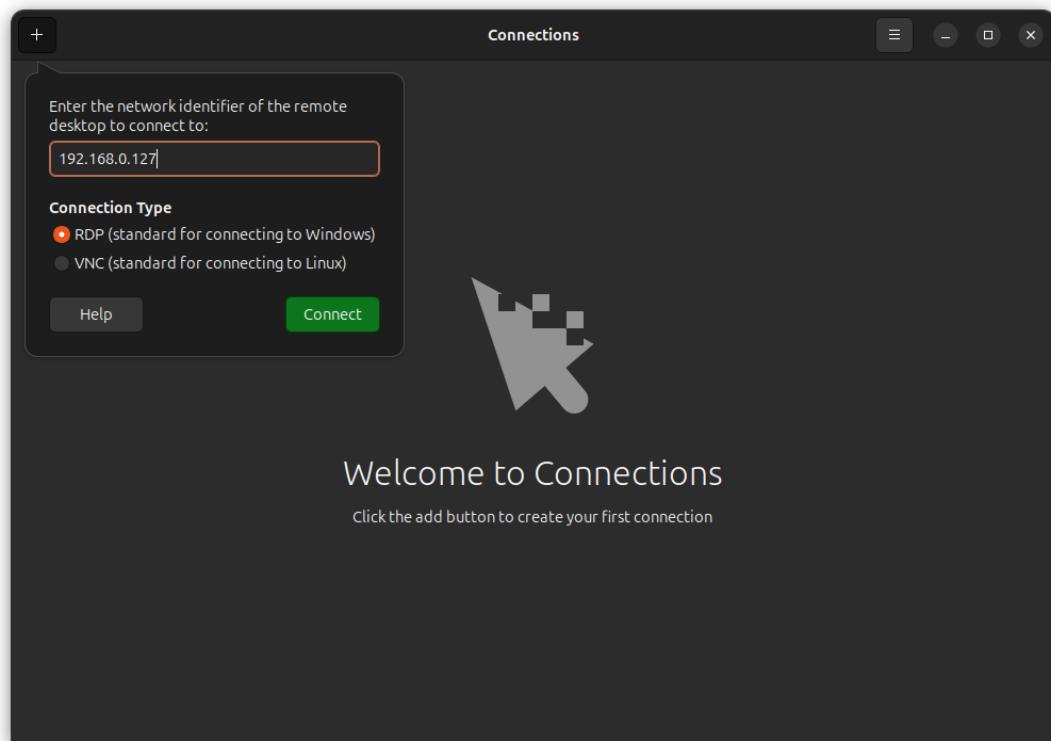


Рисунок 27. Подключение к рабочему столу Windows 10.

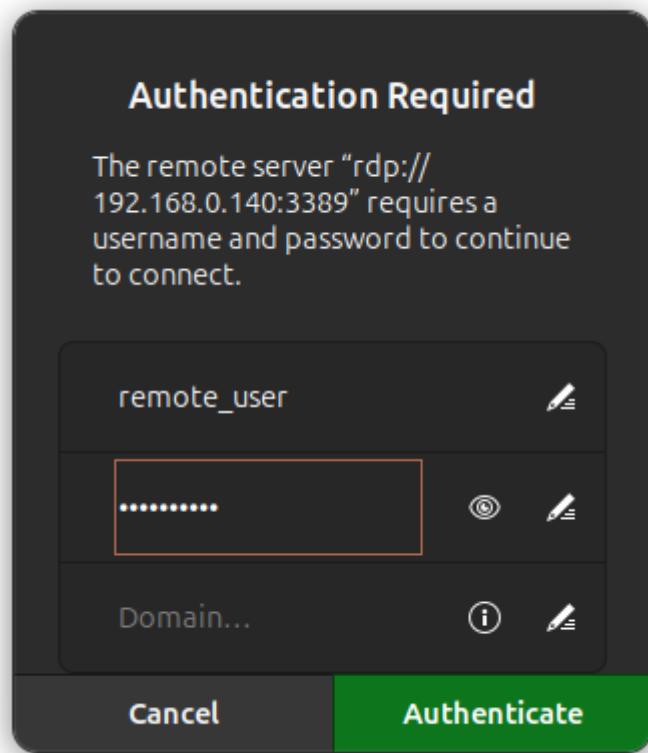


Рисунок 28. Аутентификация при подключении к удаленному рабочему столу.

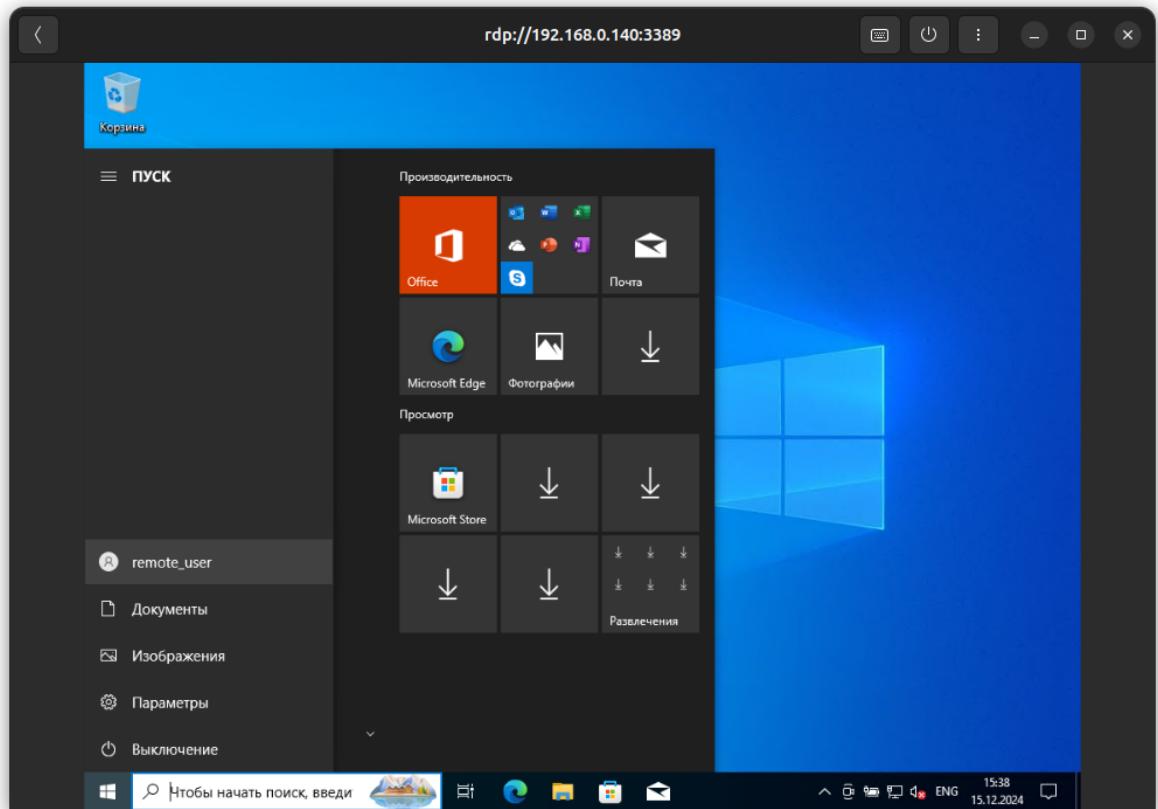


Рисунок 29. Проверка работоспособности удаленного рабочего стола Windows 10.

Работа с VVM

Текст задания

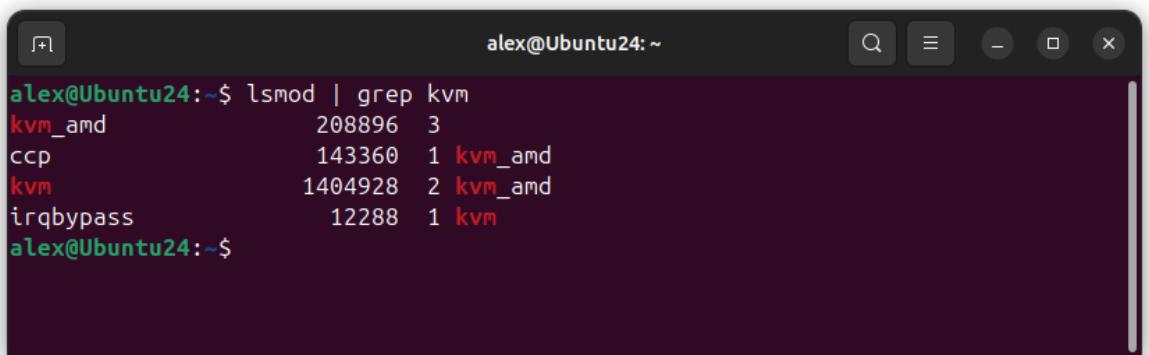
1. Установить QEMU
2. Проверить, поддерживает ваша система KVM
3. Создайте виртуальную машину
4. Запустите виртуальную машину
5. Запустите виртуальную машину с установкой операционной системы.
(Уточните полный путь до дистрибутива. Можно переименовать дистрибутив с более коротким именем fedora.iso)
6. После установки системы создайте команду запуска гостевой ОС с помощью команды qemu со следующими параметрами (воспользуйтесь командой qemu-system-x86_64):
 - a. Загрузка в полноэкранном режиме
 - b. Установка mac-адреса 17:10:20:22:20:09
7. Установка Virtual Manger
8. Установите пакеты приложений для нормальной работы Virtual-Manger
9. Добавить своего пользователя в группу libvirt
10. Проверка что libvirt запущен и работает
11. Перезапустите гостевую систему
12. Проверка на правильность работы системы и все ли правильно установлено.
13. После перезапуска найдите в приложениях программу Virtual Machine Manager
14. Создайте виртуальную машину с именем FIO-ваша_группа
15. Проверьте настройки системы.
16. Добавьте еще один жесткий диск объемом 10МиБ
17. Изучите свойства виртуальной машины, созданной в VMM, какие настройки виртуальной машины можно выставлять? Какие параметры можно отслеживать в оперативном режиме?
18. Создать снимок состояния системы
19. Установить приложение Yandex Browser
20. Восстановить состояние системы в первоначальное состояние.
21. Проверьте работу снимков состояния
22. Какие сетевые настройки можно устанавливать гостевой машине?

23. Управление виртуальной машиной из командной строки virsh

Выполнение

- Проверим, поддерживает ли система KVM.

```
lsmod | grep kvm
```



```
alex@Ubuntu24:~$ lsmod | grep kvm
kvm_amd           208896  3
ccp                143360  1 kvm_amd
kvm                1404928 2 kvm_amd
irqbypass         12288   1 kvm
alex@Ubuntu24:~$
```

Рисунок 30. Проверяем KVM

- Создадим виртуальную машину.

```
mkdir VM
qemu-img create -f qcow2 ./VM/ubuntu24.img 10G
```

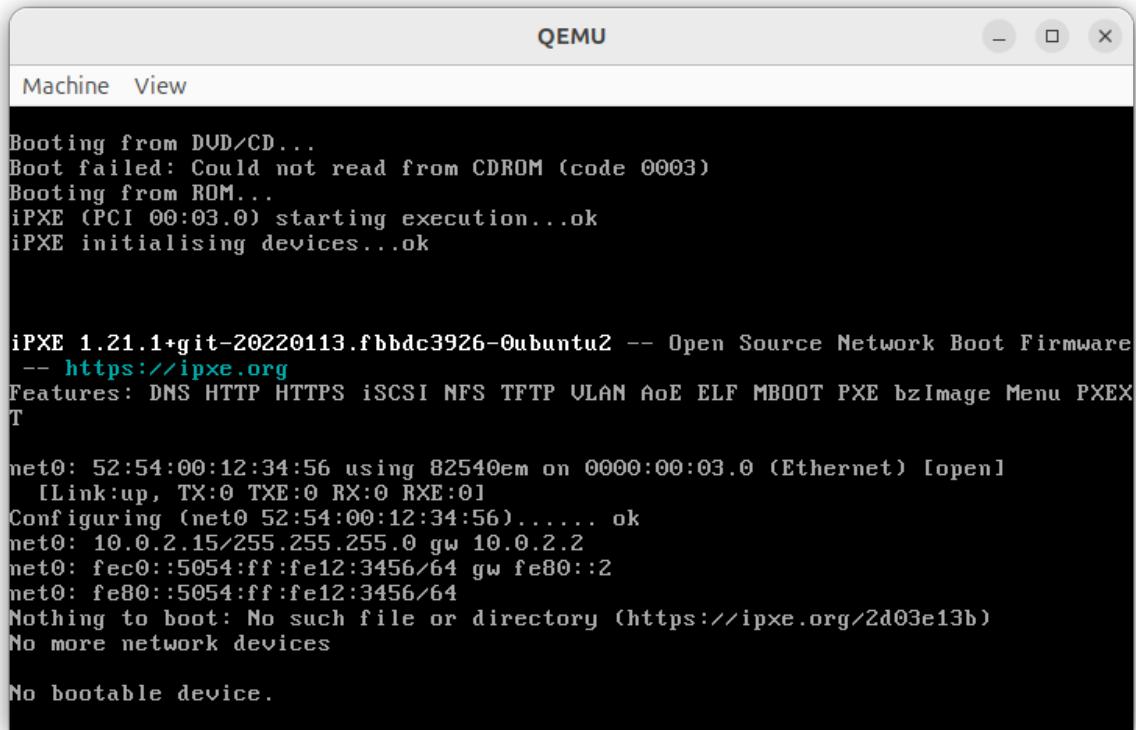
Команда qemu-img create используется для создания виртуального диска в различных форматах. В данном случае команда создает образ диска в формате qcow2 с именем Ubuntu24.img и размером 10 гигабайт. Давайте разберем каждый аргумент:

- qemu-img: это утилита командной строки, которая входит в состав QEMU (Quick Emulator). Она используется для управления образами дисков виртуальных машин.
- create: это команда, которая указывает qemu-img создать новый образ диска.
- -f: флаг -f (или --format) указывает формат создаваемого образа диска. В данном случае используется формат
- qcow2 (QEMU Copy-On-Write version 2): это современный и эффективный формат, который поддерживает такие функции, как динамическое выделение места, сжатие, шифрование и моментальные снимки.
- ./VM/Ubuntu24.img: это путь и имя файла создаваемого образа диска.

В данном случае образ будет создан в директории ./VM/ с именем Ubuntu24.img.

- 10G: это размер создаваемого образа диска. В данном случае размер указан как 10 гигабайт (10G). Для формата qcow2 это максимальный размер диска, а фактический размер файла будет увеличиваться по мере записи данных на диск.

3. Запустим виртуальную машину без ОС



The screenshot shows the QEMU interface with the title "QEMU". The window is titled "Machine View". The log output is as follows:

```
Booting from DVD/CD...
Boot failed: Could not read from CDROM (code 0003)
Booting from ROM...
iPXE (PCI 00:03.0) starting execution...ok
iPXE initialising devices...ok

iPXE 1.21.1+git-20220113.fbdbdc3926-0ubuntu2 -- Open Source Network Boot Firmware
-- https://ipxe.org
Features: DNS HTTP HTTPS iSCSI NFS TFTP VLAN AoE ELF MBOOT PXE bzImage Menu PXEX
T

net0: 52:54:00:12:34:56 using 82540em on 0000:00:03.0 (Ethernet) [open]
[Link:up, TX:0 RX:0 RXE:0]
Configuring (net0 52:54:00:12:34:56)..... ok
net0: 10.0.2.15/255.255.255.0 gw 10.0.2.2
net0: fec0::5054:ff:fe12:3456/64 gw fe80::2
net0: fe80::5054:ff:fe12:3456/64
Nothing to boot: No such file or directory (https://ipxe.org/2d03e13b)
No more network devices

No bootable device.
```

Рисунок 31. Запуск QEMU вирт. машины без ОС.

4. Установим ОС Ubuntu на виртуальную машину

```
qemu-system-x86_64 -m 2048 -enable-kvm ./VM/Ubuntu24.img -cdrom /media/sf_soft/ubuntu-24.04.1-desktop-amd64.iso
```

Устанавливаем ОС Ubuntu.

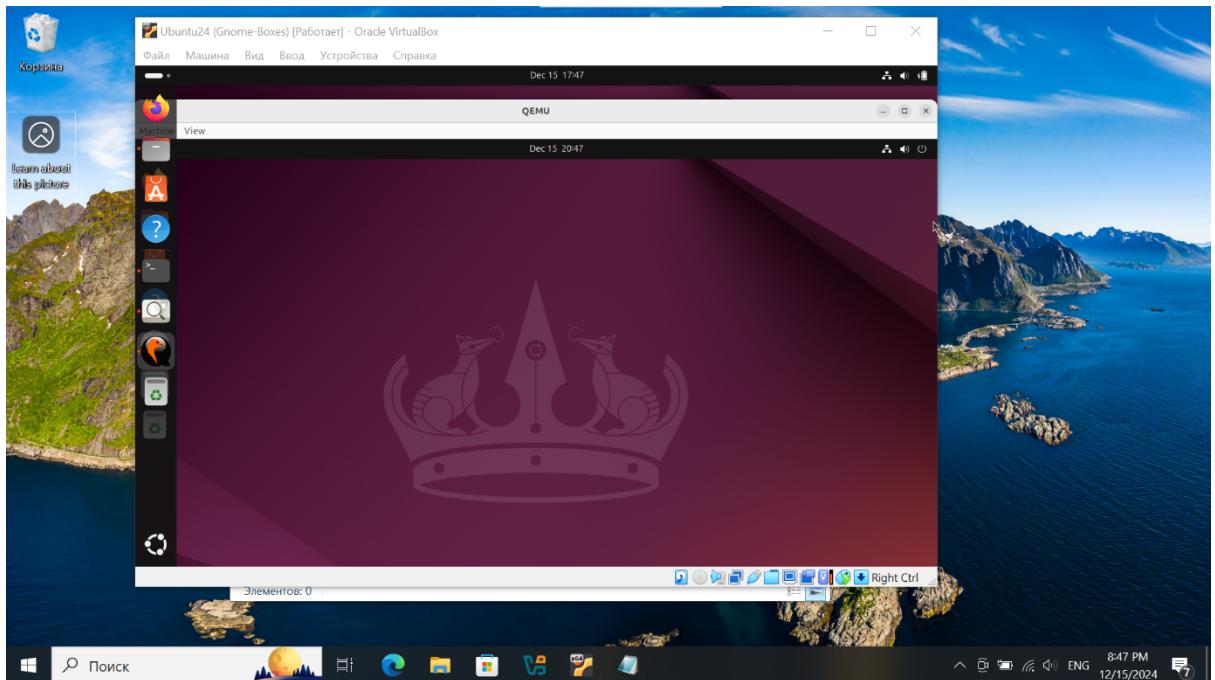


Рисунок 32. Установленная Ubuntu в QEMU.

5. Команда для запуска виртуальной машины в полноэкранном режиме с mac адресом 17:10:20:22:20:09.

```
qemu-system-x86_64 -m 2048 -enable-kvm -full-screen -netdev user,id=net0 -device e1000,netdev=net0,mac=17:10:20:22:20:09 -hda Ubuntu24.img
```

- `qemu-system-x86_64`: это исполняемый файл QEMU, который запускает виртуальную машину для архитектуры x86_64. Он эмулирует процессор, память, устройства и другие компоненты, необходимые для работы операционной системы.
- `-m 2048`: Флаг `-m` задает объем оперативной памяти (RAM), выделяемый для виртуальной машины. В данном случае это 2048 мегабайт (2 ГБ).
- `-enable-kvm`: Этот флаг включает поддержку KVM (Kernel-based Virtual Machine), которая позволяет использовать аппаратную виртуализацию на Linux. KVM значительно ускоряет работу виртуальной машины, так как позволяет гостевой ОС напрямую использовать процессор хоста.
- `-full-screen`: Этот флаг запускает виртуальную машину в полноэкранном режиме. После запуска QEMU переключится в полноэкранный режим, чтобы отображать экран гостевой ОС на весь

экран хоста.

- **-netdev user,id=net0:** Этот флаг настраивает сетевой интерфейс для виртуальной машины.
- **user:** указывает, что используется пользовательская сетевая модель QEMU. В этой модели QEMU предоставляет виртуальный сетевой стек, который позволяет гостевой ОС выходить в интернет через хост, но без поддержки прямого доступа к сети хоста.
- **id=net0:** указывает идентификатор сетевого устройства, который будет использоваться для связи с сетевым адаптером виртуальной машины.
- **-device e1000,netdev=net0,mac=17:10:20:22:20:09** Этот флаг добавляет сетевой адаптер в виртуальную машину.
- **e1000:** указывает, что используется сетевой адаптер Intel e1000. Это один из наиболее распространенных виртуальных сетевых адаптеров, который поддерживается большинством операционных систем.
- **netdev=net0:** связывает этот сетевой адаптер с сетевым устройством, настроенным ранее (net0).
- **mac=17:10:20:22:20:09:** Указывает MAC-адрес для этого сетевого адаптера. MAC-адрес — это уникальный идентификатор сетевого устройства, который используется для идентификации устройства в сети.
- **-hda Ubuntu24.img:** Этот флаг указывает, что образ диска Ubuntu24.img будет использоваться как основной жесткий диск (Primary Hard Drive) виртуальной машины. В данном случае это образ диска, созданный ранее с помощью qemu-img.

6. Установка Virtual Manger

7. Установим необходимые пакеты

```
sudo apt install qemu-system qemu-kvm libvirt-daemon libvirt-clients bridge-utils  
virt-manager
```

8. Добавим своего пользователя в группу libvirt

```
sudo gpasswd -a $USER libvirt
```

9. Проверка что libvirt запущен и работает

```
sudo systemctl status libvирtd
```

```
alex@Ubuntu24:~$ sudo systemctl status libvирtd
● libvирtd.service - libvirt legacy monolithic daemon
  Loaded: loaded (/usr/lib/systemd/system/libvирtd.service; enabled; preset:>)
  Active: active (running) since Sun 2024-12-15 18:38:56 UTC; 8s ago
TriggeredBy: ● libvирtd.socket
              ● libvирtd-admin.socket
              ● libvирtd-ro.socket
    Docs: man:libvирtd(8)
          https://libvirt.org/
   Main PID: 4463 (libvирtd)
     Tasks: 22 (limit: 32768)
    Memory: 18.7M (peak: 20.3M)
       CPU: 460ms
      CGroup: /system.slice/libvирtd.service
              └─1510 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.a
                ├─1511 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.a
                └─4463 /usr/sbin/libvирtd --timeout 120

Dec 15 18:38:56 Ubuntu24 systemd[1]: Starting libvирtd.service - libvirt legacy>
Dec 15 18:38:56 Ubuntu24 systemd[1]: Started libvирtd.service - libvirt legacy >
Dec 15 18:38:57 Ubuntu24 dnsmasq[1510]: read /etc/hosts - 8 names
Dec 15 18:38:57 Ubuntu24 dnsmasq[1510]: read /var/lib/libvirt/dnsmasq/default.a
Dec 15 18:38:57 Ubuntu24 dnsmasq-dhcp[1510]: read /var/lib/libvirt/dnsmasq/default.a
lines 1-22/22 (END)
```

Рисунок 33. Проверка статуса libvирtd

10. Проверка на правильность работы системы и все ли правильно установлено.

```
kvm-ok
```

```
alex@Ubuntu24:~$ kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
alex@Ubuntu24:~$
```

Рисунок 34. Проверка работы KVM

11. Запустим Virtual Machine Manager

12. Создадим новую виртуальную машину

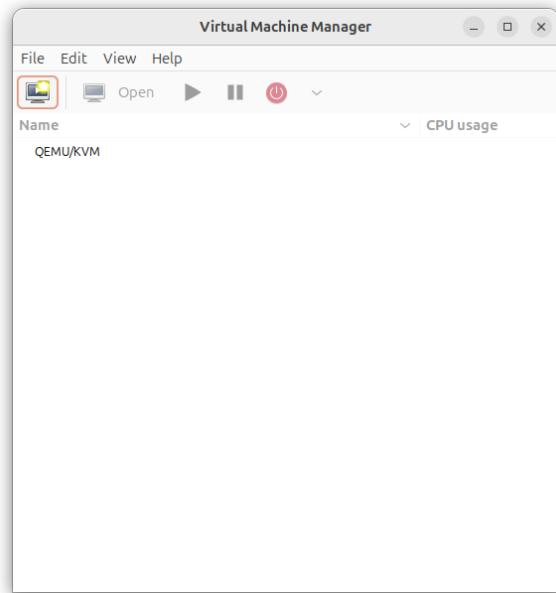


Рисунок 35. Кнопка создания виртуальной машины

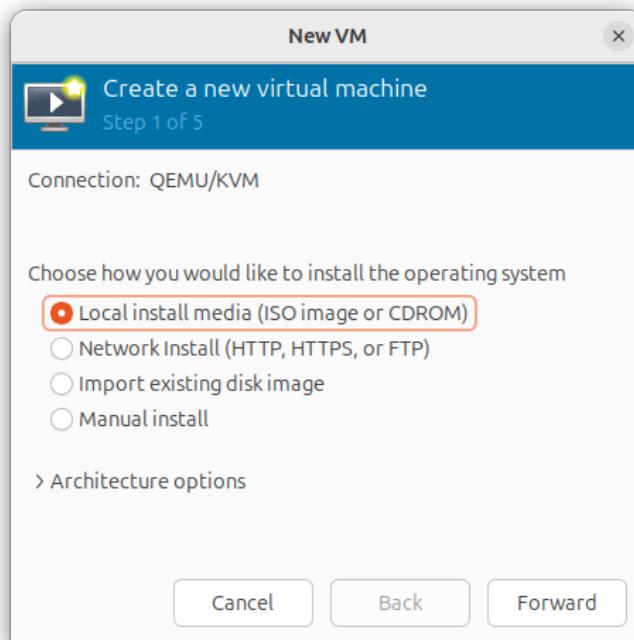


Рисунок 36. Выбираем создать из ISO образа

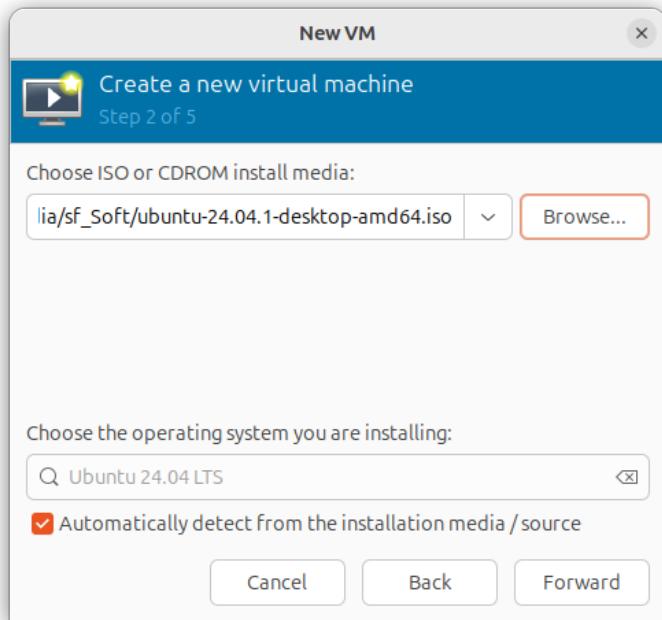


Рисунок 37. Выбираем путь до ISO образа

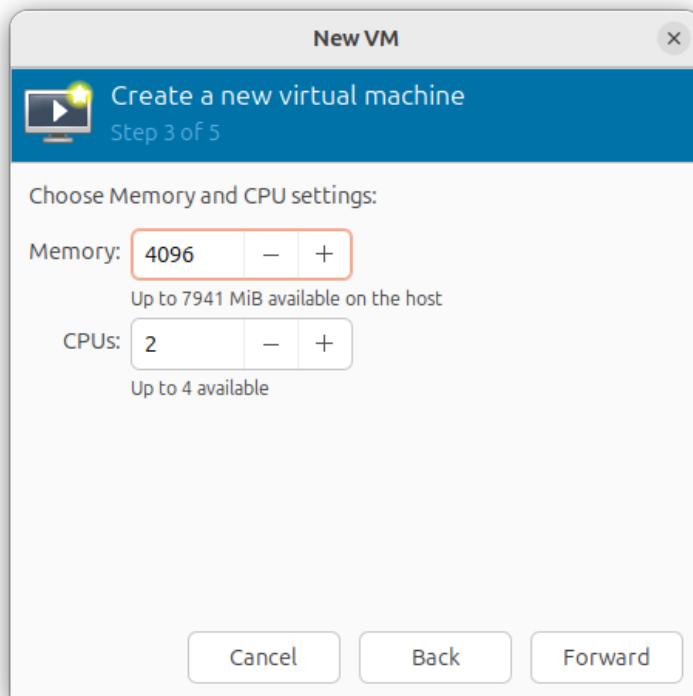


Рисунок 38. Выбираем кол-во памяти и CPU

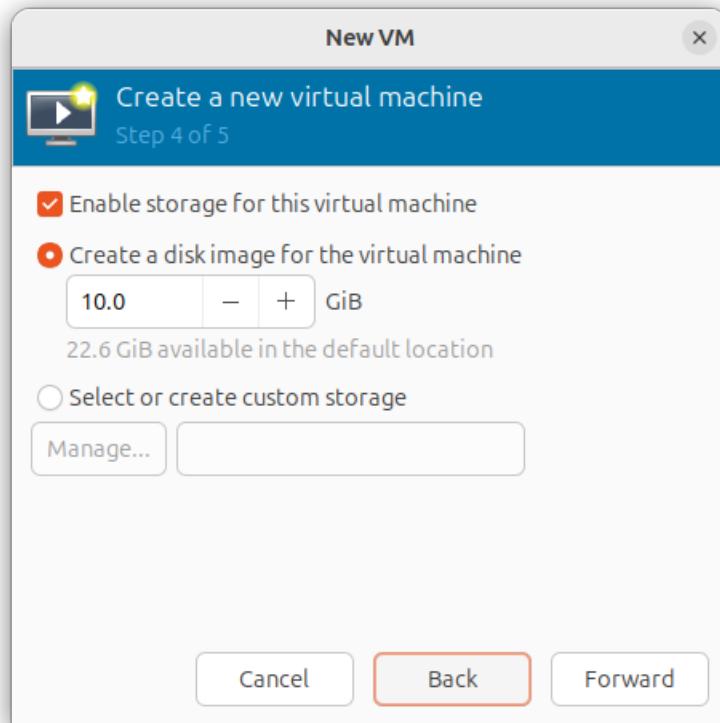


Рисунок 39. Создаем жесткий диск

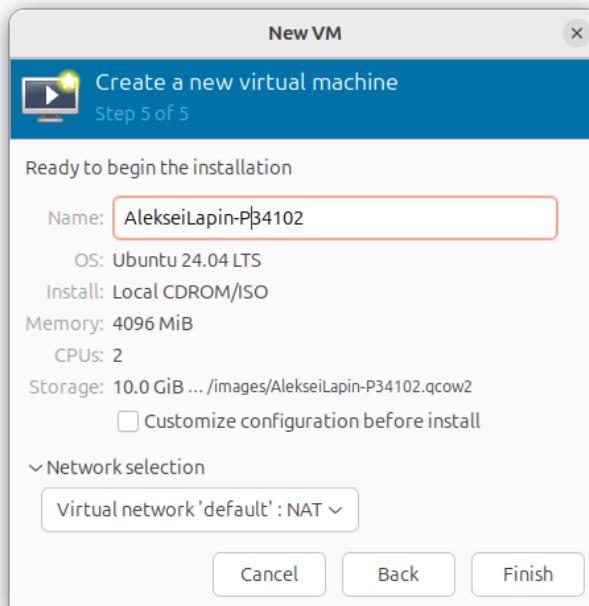


Рисунок 40. Конфигурация виртуальной машины

13. Настройки виртуальной машины.

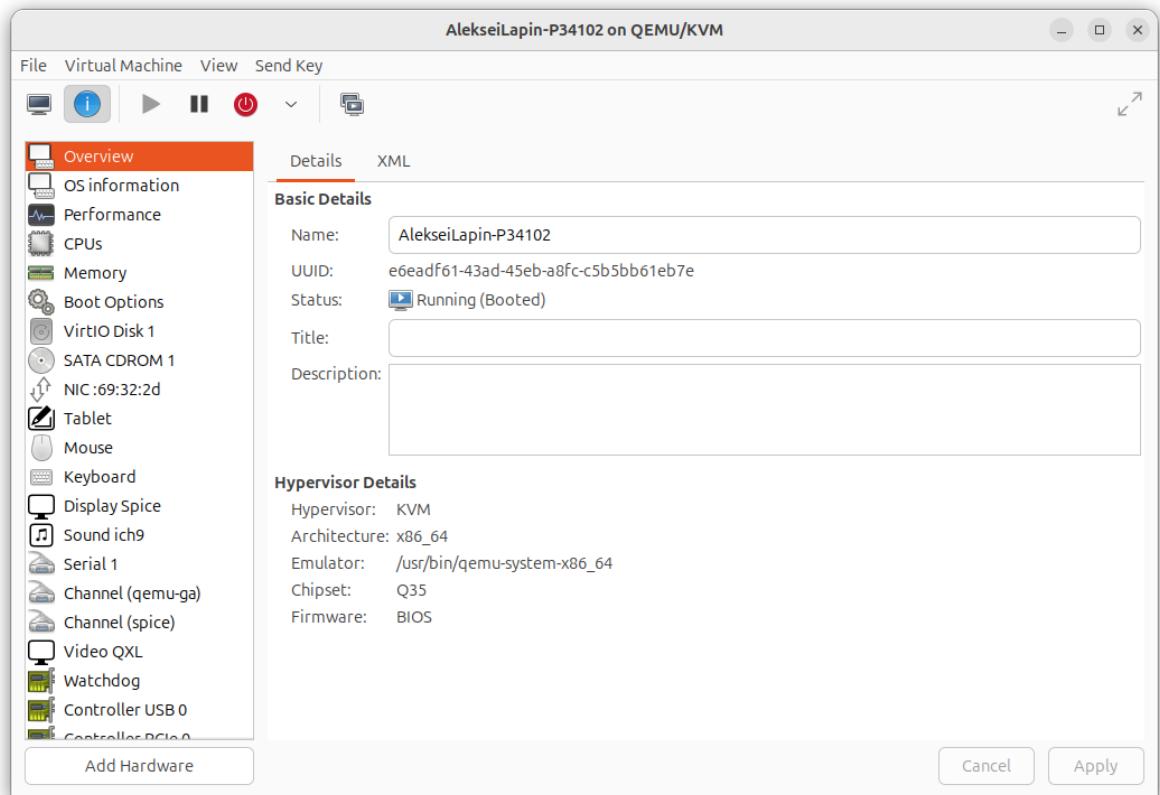


Рисунок 41. Overview

В данном разделе Virtual Machine Manager отображаются общие параметры виртуальной машины. Вот краткий обзор ключевых параметров:

Идентификация виртуальной машины:

- Name: AlekseiLapin-P34102 - имя виртуальной машины.
- UUID: e6eadf61-43ad-45eb-a8fc-c5b5bb61eb7e - уникальный идентификатор виртуальной машины.

Статус и общие сведения:

- Status: Running (Booted) - виртуальная машина запущена и загружена.
- Hypervisor: KVM - гипервизор, используемый для виртуализации.
- Architecture: x86_64 - 64-битная архитектура процессора.
- Chipset: Q35 - чипсет, эмулируемый в виртуальной машине.
- Emulator: /usr/bin/qemu-system-x86_64 - путь к эмулятору, используемому для запуска виртуальной машины.
- Firmware: BIOS (тип прошивки для загрузки виртуальной машины).

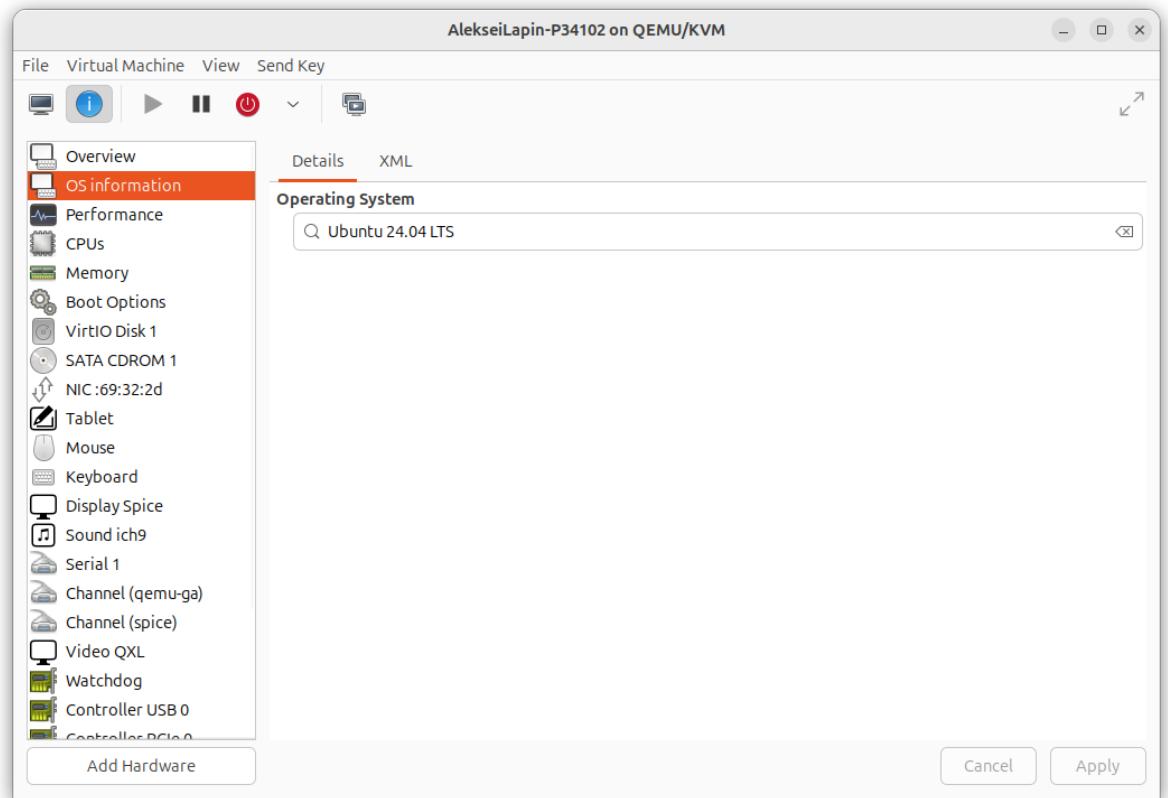


Рисунок 42. OS information

В данном разделе Virtual Machine Manager отображается информация об операционной системе виртуальной машины:

- Операционная система: Ubuntu 24.04 LTS (долгосрочная поддержка версии Ubuntu).

Этот раздел позволяет выбрать тип и версию операционной системы, используемой в виртуальной машине.

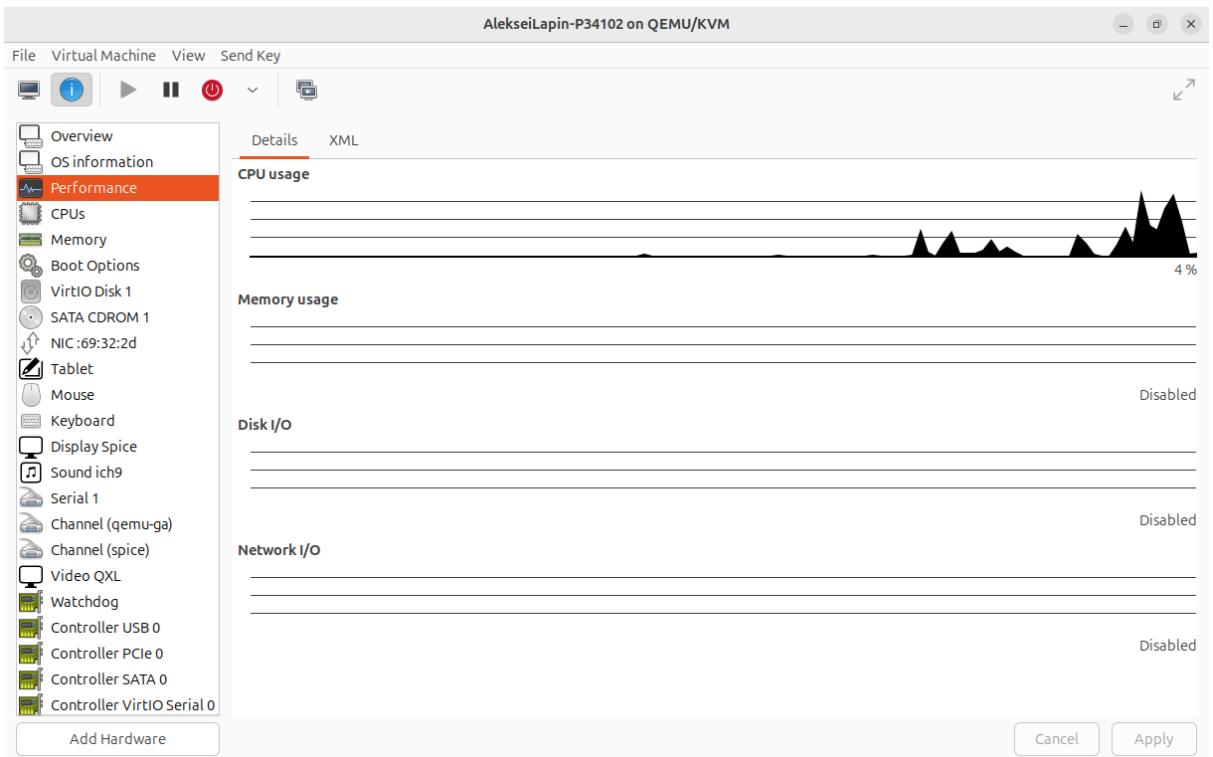


Рисунок 43. Performance

В данном разделе Virtual Machine Manager отображаются различные параметры производительности виртуальной машины:

- CPU usage (Использование процессора) - показывает загрузку CPU
- Memory usage (Использование памяти) - отображает использование оперативной памяти
- Disk I/O (Дисковый ввод/вывод) - показывает активность работы с диском
- Network I/O (Сетевой ввод/вывод) - отображает сетевую активность

Эти параметры помогают оценить текущую нагрузку на систему.

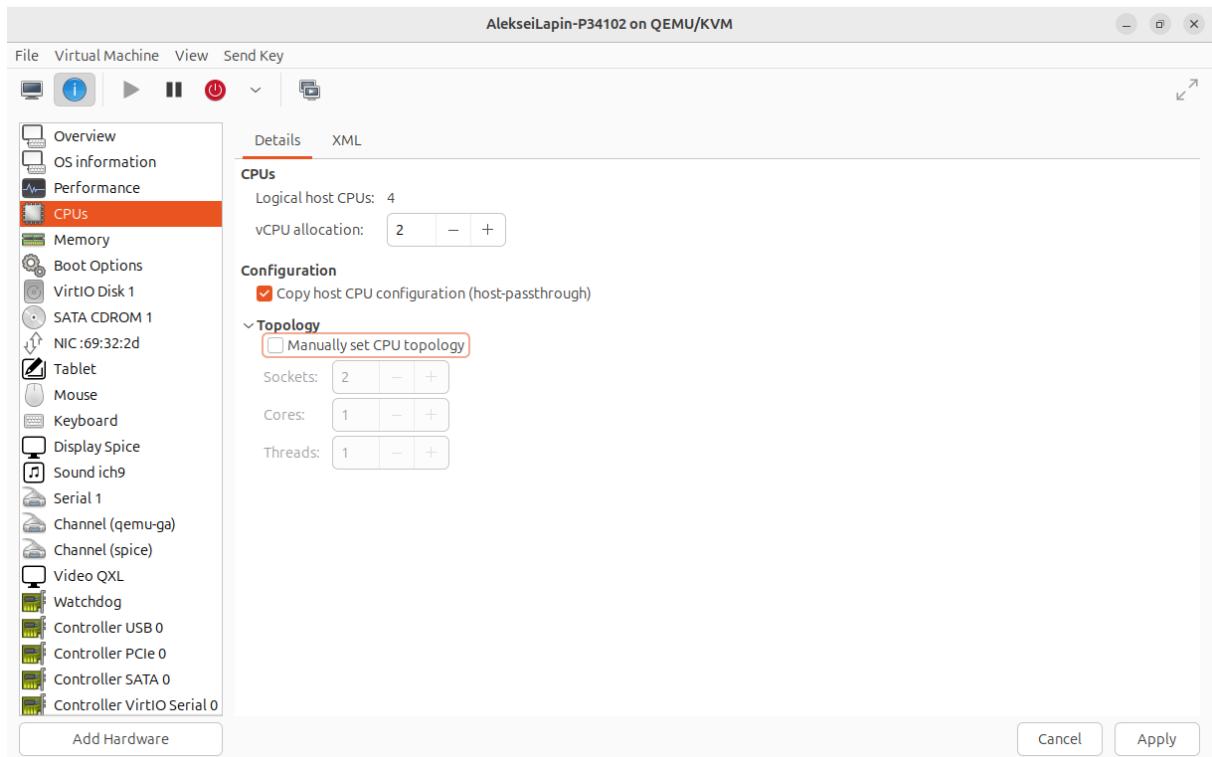


Рисунок 44. CPUs

В данном разделе Virtual Machine Manager отображаются параметры виртуальных процессоров (vCPU) для виртуальной машины.

CPUs:

- Logical host CPUs: Общее количество логических ядер процессора на хосте (в данном случае 4).
- vCPU allocation: Количество виртуальных CPU, выделяемых для виртуальной машины (установлено 2).

Configuration:

- Copy host CPU configuration: Флаг для копирования конфигурации процессора хоста (режим host-passthrough).

Topology:

- Manually set CPU topology: Позволяет вручную задать топологию виртуального процессора (сокеты, ядра, потоки). В текущем состоянии флаг не активирован.
- Sockets, Cores, Threads: Если флаг будет активирован, можно будет вручную задать количество сокетов, ядер и потоков для vCPU.

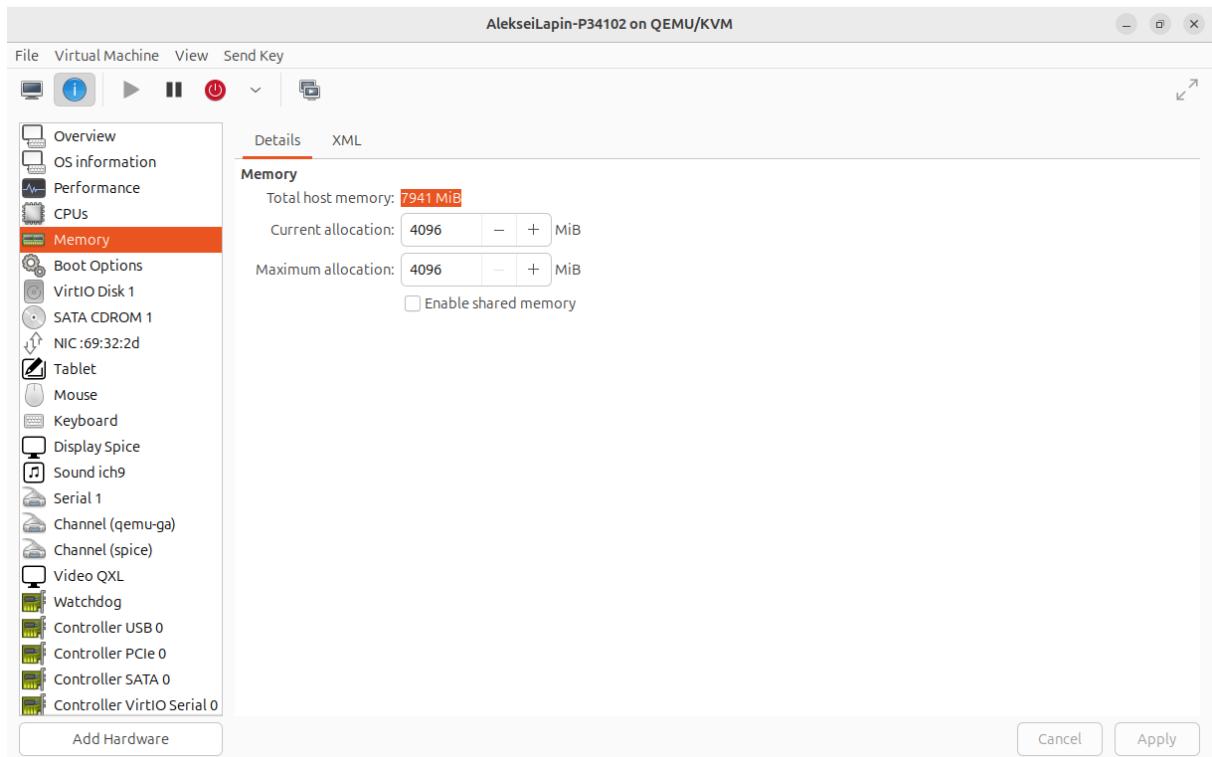


Рисунок 45. Memory

В данном разделе Virtual Machine Manager отображаются настройки памяти виртуальной машины:

- Общая память хоста (Total host memory) составляет 7941 MiB
- Текущее выделение памяти (Current allocation) установлено на 4096 MiB
- Максимальное выделение памяти (Maximum allocation) также составляет 4096 MiB
- Есть опция включения общей памяти (Enable shared memory), которая в данный момент отключена

Current allocation (Текущее выделение) - это количество памяти, которое сейчас выделено виртуальной машине и может использоваться гостевой операционной системой. В вашем случае это 4096 MiB.

Maximum allocation (Максимальное выделение) - это верхний предел памяти, который может быть выделен виртуальной машине. Даже если гостевой системе потребуется больше памяти, она не сможет превысить это значение. Это своего рода «потолок» для динамического выделения памяти.

Shared memory (Общая память) — это механизм, который позволяет нескольким виртуальным машинам совместно использовать одни и те же страницы памяти, если в

них находятся идентичные данные.

Это помогает:

- Экономить физическую память хоста
- Запускать больше виртуальных машин на одном хосте
- Повысить эффективность использования памяти, особенно когда запущено несколько однотипных ВМ

Например, если у вас запущено несколько виртуальных машин с одинаковой операционной системой, многие системные библиотеки и файлы будут идентичны. При включенной shared memory эти идентичные страницы памяти будут храниться в одном экземпляре, а не дублироваться для каждой ВМ.

Это основной раздел управления оперативной памятью для виртуальной машины, где администратор может настраивать объем выделяемой памяти в пределах доступных ресурсов хоста.

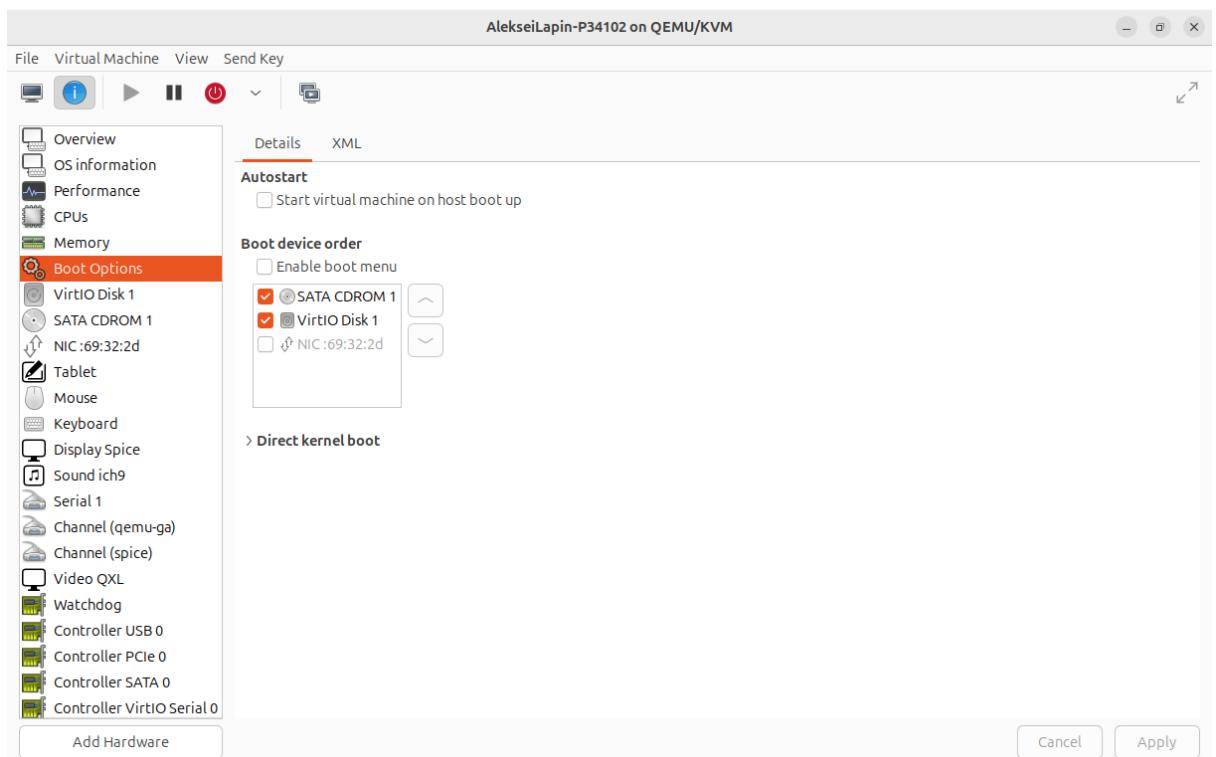


Рисунок 46. Boot Options

В данном разделе Virtual Machine Manager отображаются настройки загрузки (Boot Options) виртуальной машины:

Autostart - опция автоматического запуска виртуальной машины при загрузке хоста.

Boot device order - порядок загрузочных устройств.

Enable boot menu - опция включения загрузочного меню.

Direct kernel boot - опция прямой загрузки ядра.

Этот раздел позволяет настроить параметры запуска виртуальной машины и определить последовательность устройств, с которых будет производиться загрузка системы.

Direct kernel boot (Прямая загрузка ядра) — это механизм в QEMU/KVM, который позволяет напрямую загружать ядро Linux и initial ramdisk (initrd/initramfs) в виртуальную машину, минуя загрузчик операционной системы (например, GRUB). Этот метод используется разработчиками ядра Linux и системными администраторами для тестирования и отладки.

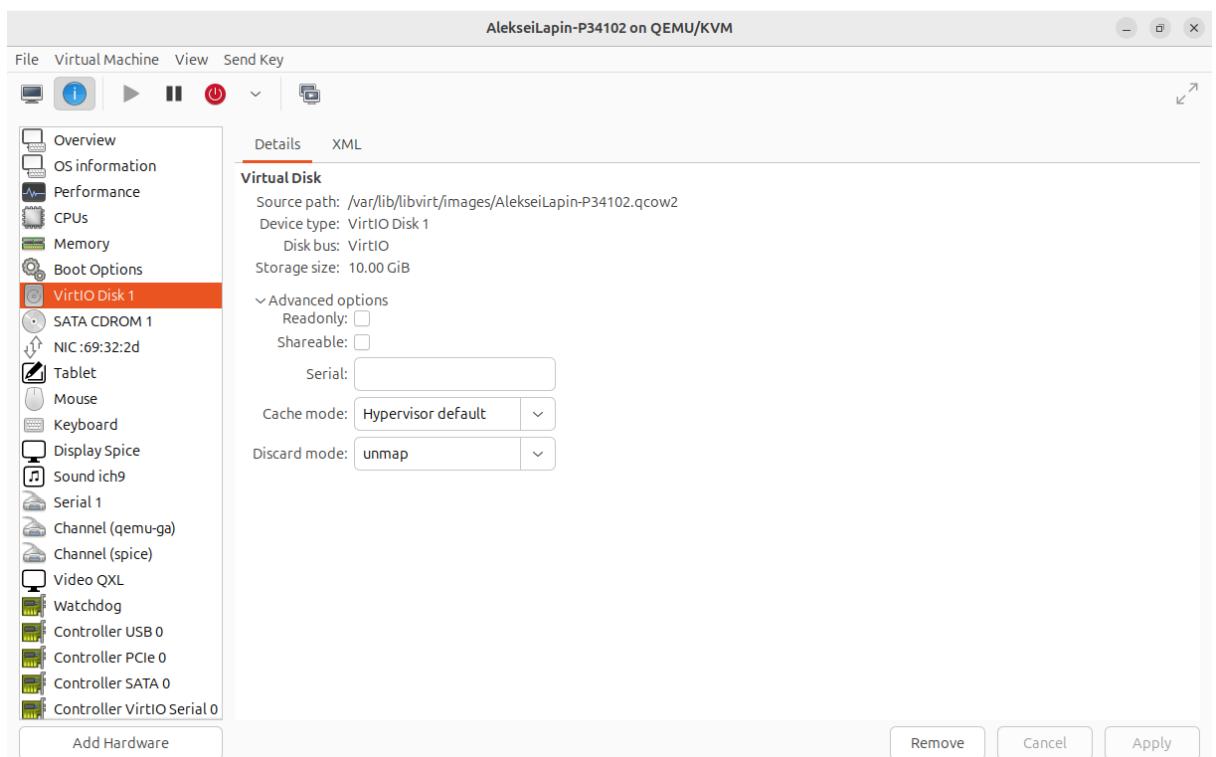


Рисунок 47. VirtIO Disk 1

В данном разделе Virtual Machine Manager отображаются параметры виртуального диска:

- Source path: путь к образу диска - /var/lib/libvirt/images/AlekseiLapin-P34102.qcow2
- Device type: тип устройства - VirtIO Disk 1
- Disk bus: используемая шина – VirtIO
- Storage size: размер хранилища - 10.00 GiB

Расширенные опции (Advanced options):

- Readonly: возможность сделать диск только для чтения
- Shareable: возможность общего доступа к диску
- Serial: серийный номер диска
- Cache mode: режим кэширования (установлен Hypervisor default)
- Discard mode: режим освобождения неиспользуемого пространства (установлен unmap)

Это основные настройки виртуального диска, которые определяют его характеристики и режим работы в виртуальной машине.

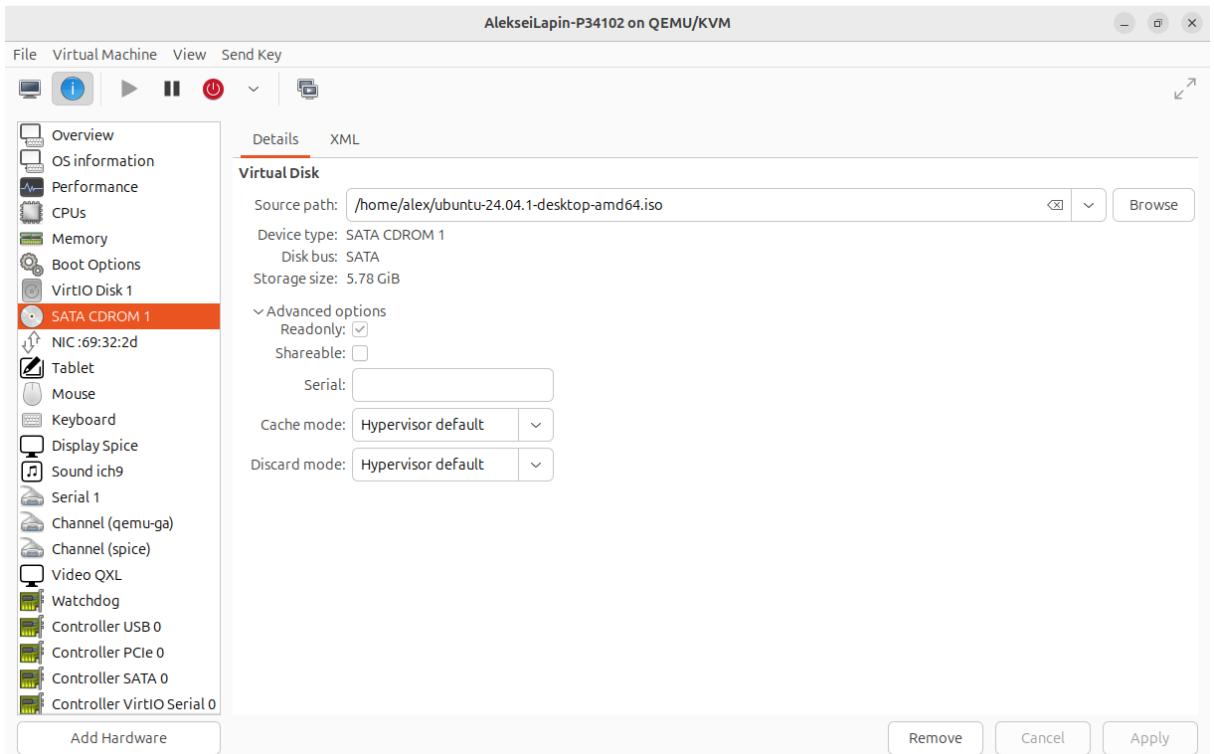


Рисунок 48. SATA CDROM 1

В данном разделе Virtual Machine Manager отображаются настройки виртуального CD-ROM:

- Source path: путь к ISO-образу - /home/alex/ubuntu-24.04.1-desktop-amd64.iso
- Device type: тип устройства - SATA CDROM 1
- Disk bus: используемая шина – SATA
- Storage size: размер образа - 5.78 GiB

Расширенные опции (Advanced options):

- Readonly: включен режим только для чтения
- Shareable: возможность общего доступа (отключена)

- Serial: серийный номер устройства
- Cache mode: режим кэширования (Hypervisor default)
- Discard mode: режим освобождения памяти (Hypervisor default)

Это настройки виртуального CD-ROM привода, который используется для установки операционной системы или доступа к данным с ISO-образов.

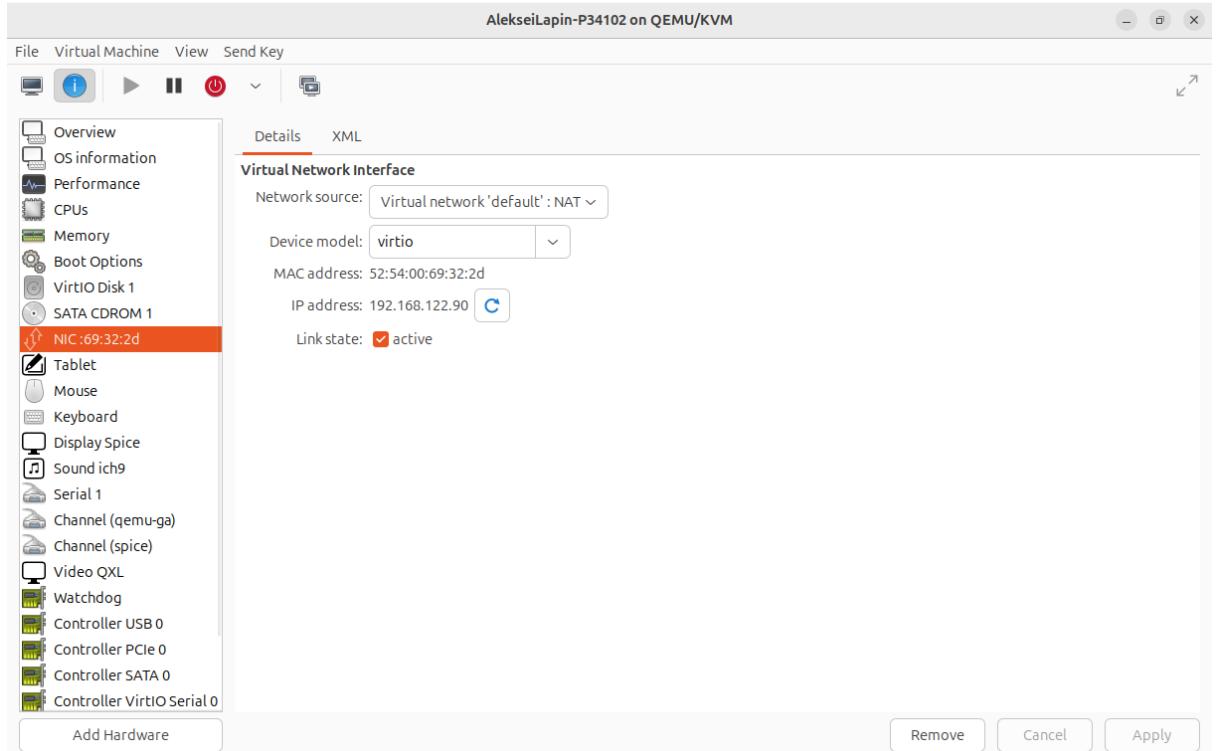


Рисунок 49. NIC:69:32:2d

В данном разделе Virtual Machine Manager отображаются настройки сетевого интерфейса (Virtual Network Interface):

- Network source: источник сети - Virtual network 'default' : NAT
- Device model: модель сетевого устройства – virtio
- MAC address: физический адрес устройства - 52:54:00:69:32:2d
- IP address: IP-адрес виртуальной машины - 192.168.122.90
- Link state: состояние подключения - active (активно)

Эти параметры определяют настройки сетевого подключения виртуальной машины, включая тип сети, модель устройства и сетевые адреса, что позволяет настроить доступ виртуальной машины к сети.

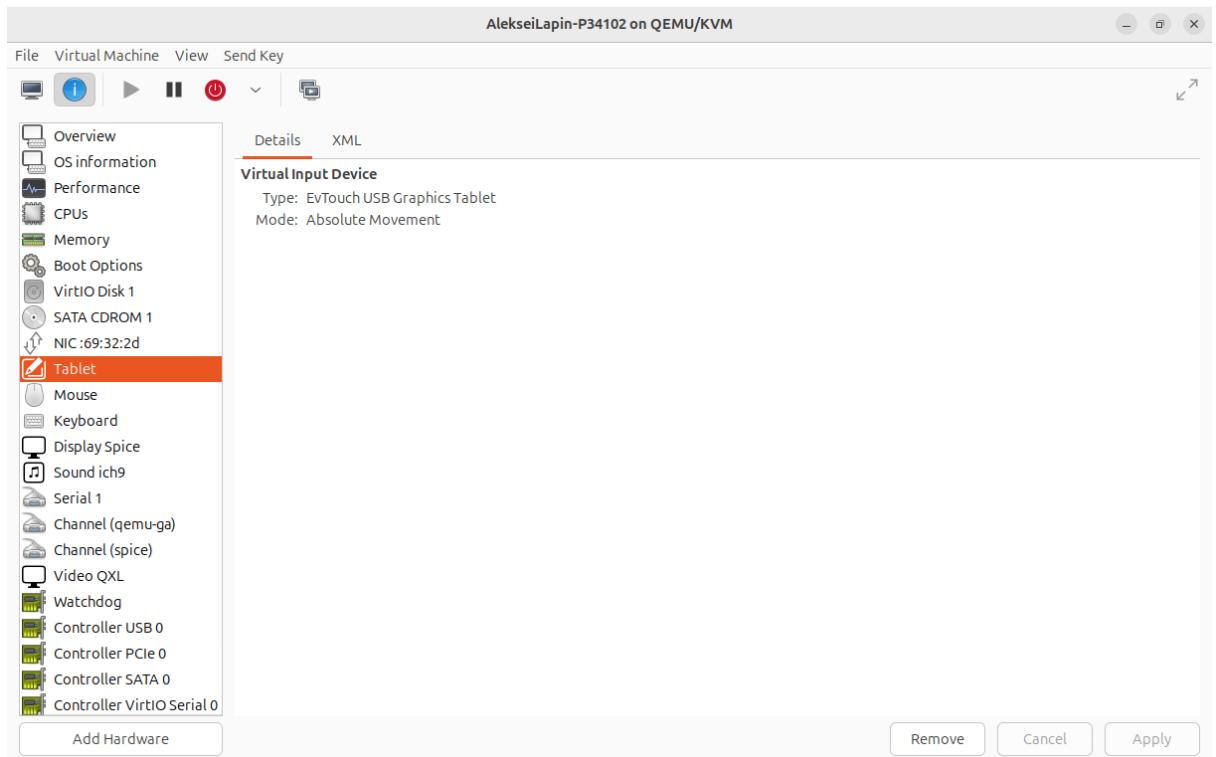


Рисунок 50. Tablet

В данном разделе Virtual Machine Manager отображаются параметры виртуального устройства ввода, например, графического планшета EvTouch USB Graphics Tablet. Указан режим работы устройства - Absolute Movement. Этот раздел позволяет управлять настройками виртуальных устройств ввода, таких как графические планшеты.

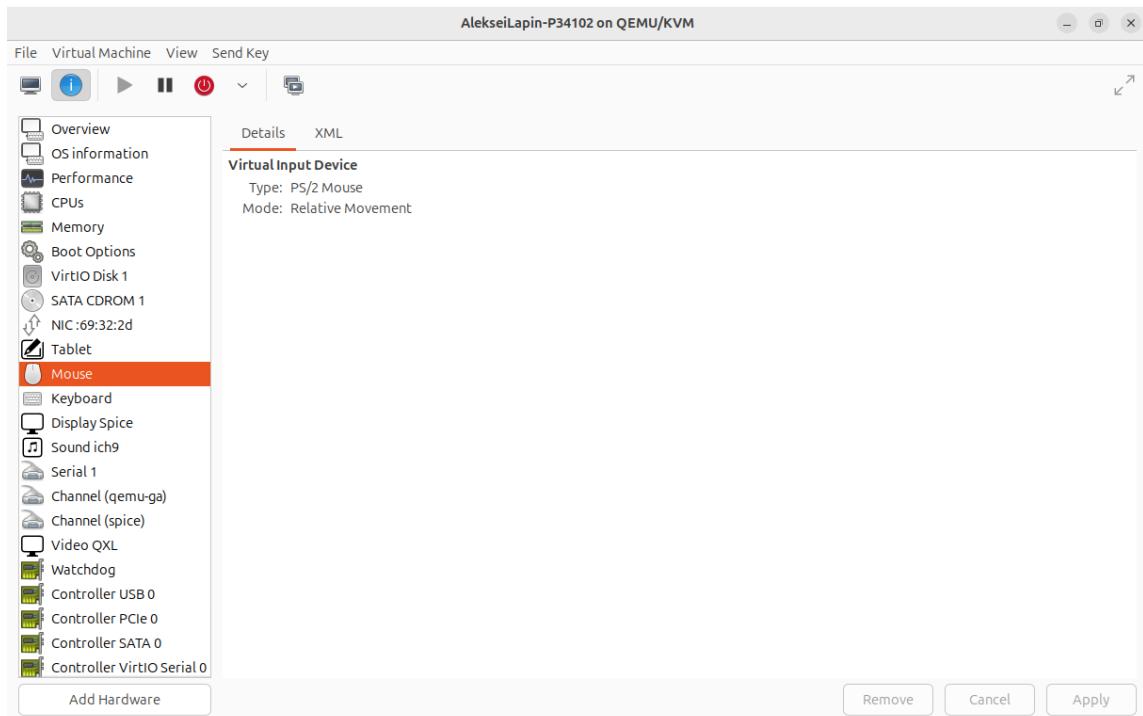


Рисунок 51. Mouse

В данном разделе Virtual Machine Manager отображаются параметры виртуальных устройств ввода. В частности, здесь показаны настройки виртуальной мыши типа PS/2 Mouse, работающей в режиме Relative Movement. Этот раздел позволяет пользователю управлять настройками виртуальных устройств ввода, таких как мыши.

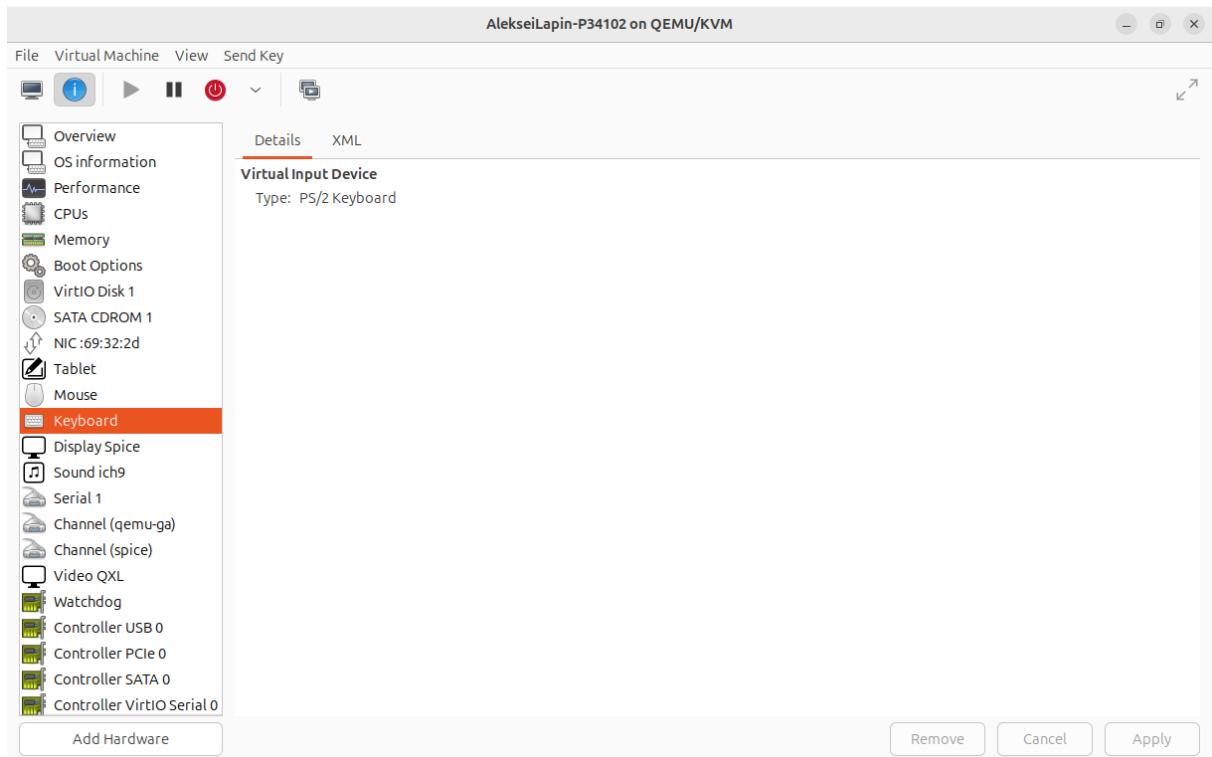


Рисунок 52. Keyboard

В данном разделе Virtual Machine Manager отображаются параметры виртуальных устройств ввода. В частности, здесь показаны настройки виртуальной мыши типа PS/2 Keyboard. Этот раздел позволяет пользователю управлять настройками виртуальных устройств ввода, таких как клавиатуры.

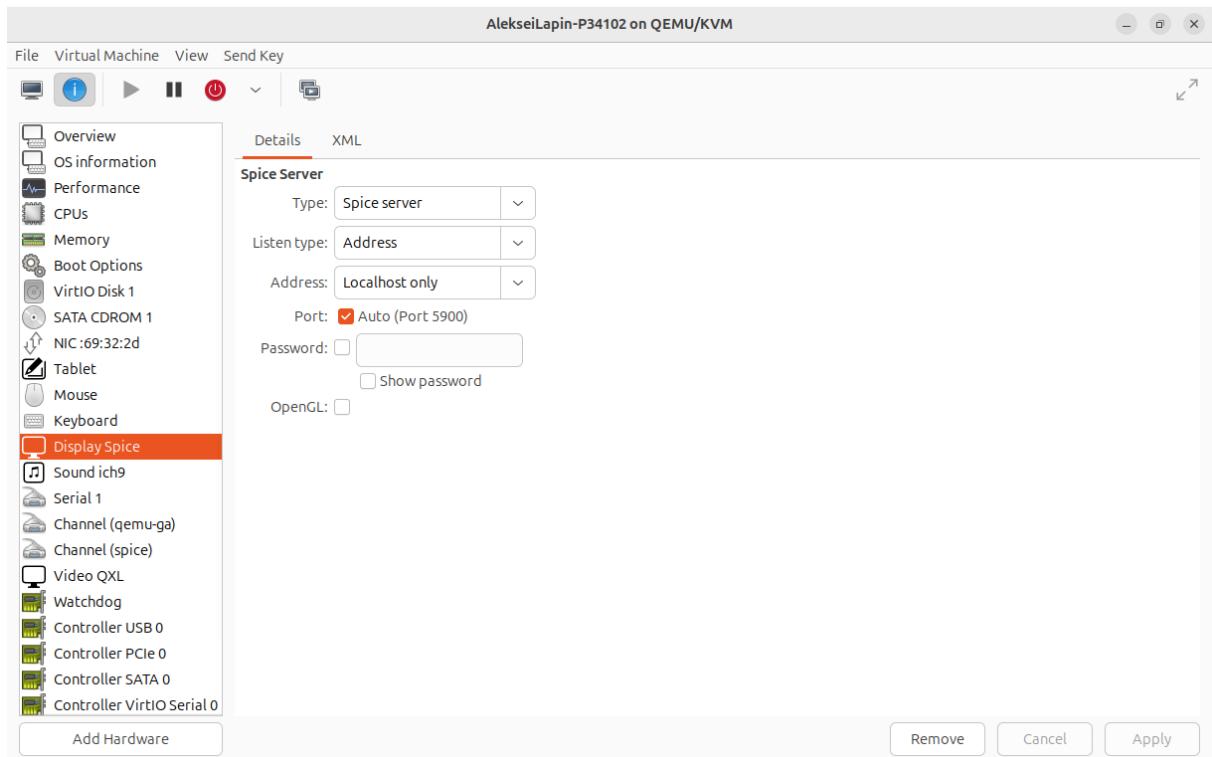


Рисунок 53. Display Spice

В данном разделе показаны настройки дисплея SPICE для виртуальной машины в Virtual Machine Manager (virt-manager). SPICE (Simple Protocol for Independent Computing Environments) - это протокол для удаленного доступа к виртуальным рабочим столам.

Основные параметры, которые можно настроить:

- Type: Тип SPICE сервера
- Listen type: Тип прослушивания (установлен как Address)
- Address: Настройка доступа (установлен Localhost only для безопасности)
- Port: Порт подключения (автоматически установлен 5900)
- Password: Возможность установить пароль для подключения
- OpenGL: Опция включения поддержки OpenGL

Эти настройки позволяют:

- Получить удаленный доступ к графическому интерфейсу виртуальной машины
- Настроить безопасное подключение через локальную сеть
- Управлять разрешениями доступа к виртуальной машине

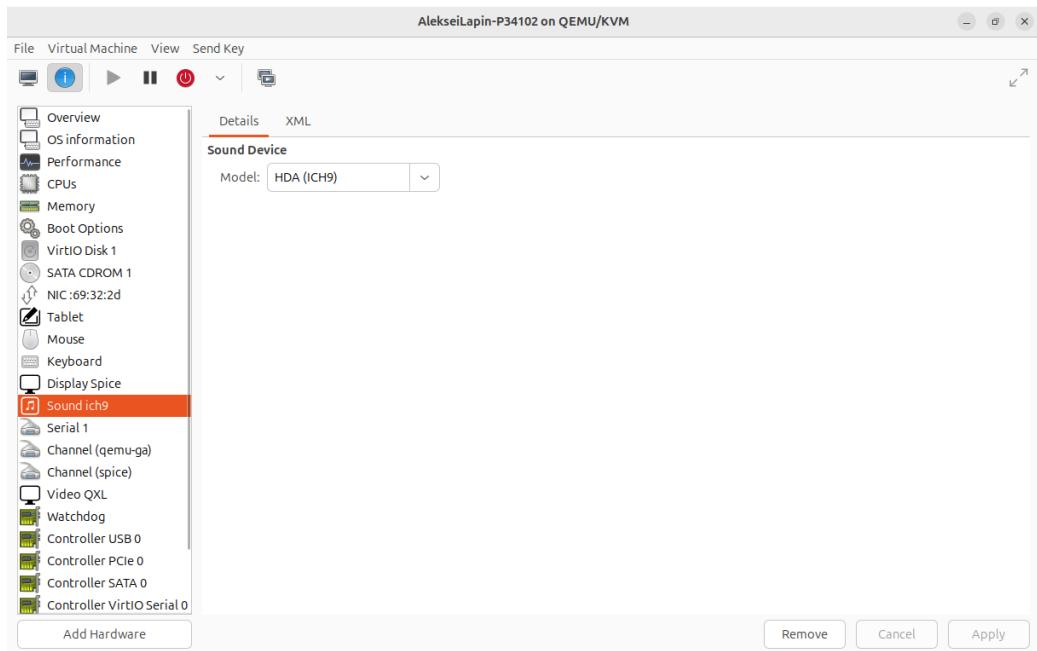


Рисунок 54. Sound ich9

В данном разделе показаны настройки звука (Sound Device) для виртуальной машины. В частности, используется модель HDA (ICH9) — это эмуляция звуковой карты Intel High Definition Audio.

Основные возможности этой настройки:

- Обеспечение звукового вывода в виртуальной машине
- Передача звука между хостовой и гостевой системой
- Эмуляция стандартной звуковой карты, совместимой с большинством операционных систем

ICH9 - это более современная версия звукового контроллера по сравнению с AC97, которая обеспечивает:

- Лучшее качество звука
- Более широкую совместимость с современными операционными системами
- Поддержку множества аудиоканалов
- Автоматическое определение и настройку звуковых устройств

Эта настройка особенно важна, если вы планируете использовать виртуальную машину для задач, требующих воспроизведения звука, например, для мультимедийных приложений или видеоконференций.

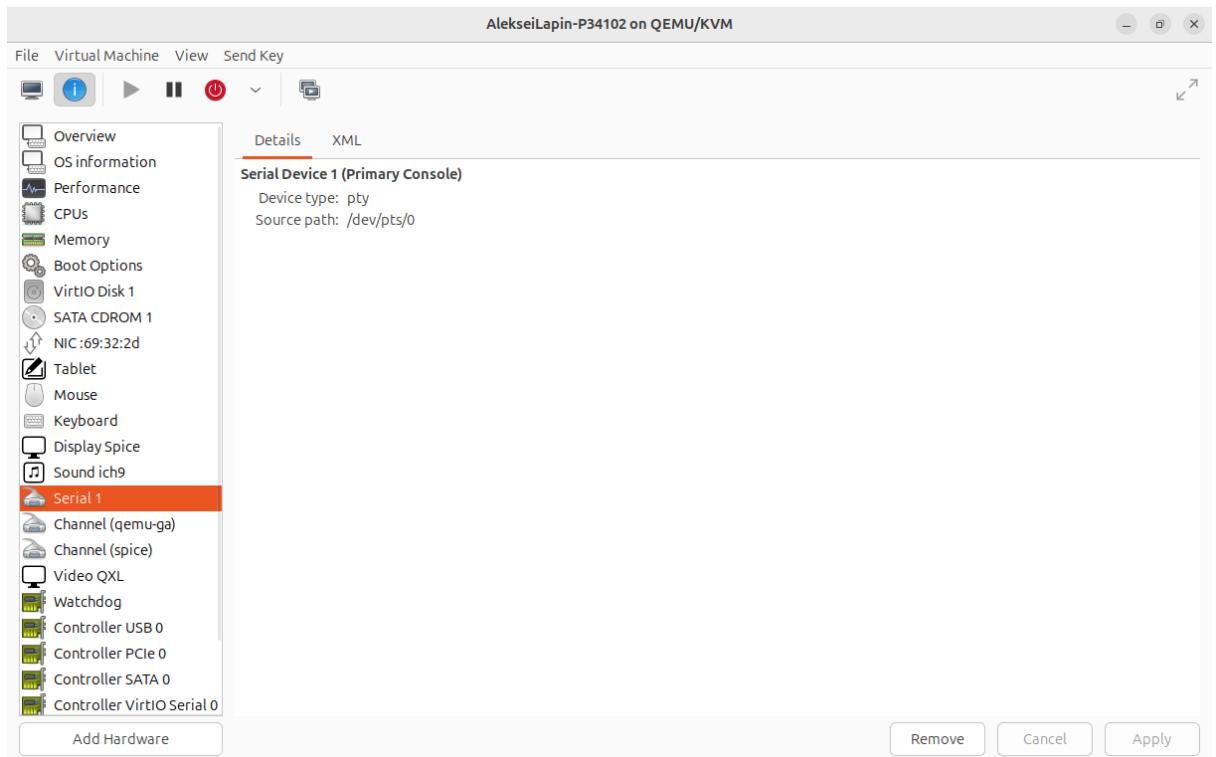


Рисунок 55. Serial 1

В данном разделе показаны настройки последовательного порта (Serial Device) виртуальной машины, который настроен как основная консоль (Primary Console).

Основные параметры:

- Device type: pty - псевдотерминал, виртуальное устройство для эмуляции терминала
- Source path: /dev/pts/0 - путь к устройству в системе

Этот раздел позволяет:

- Получить доступ к консоли виртуальной машины через последовательный порт
- Проводить отладку системы через последовательный интерфейс
- Взаимодействовать с виртуальной машиной даже при отсутствии графического интерфейса

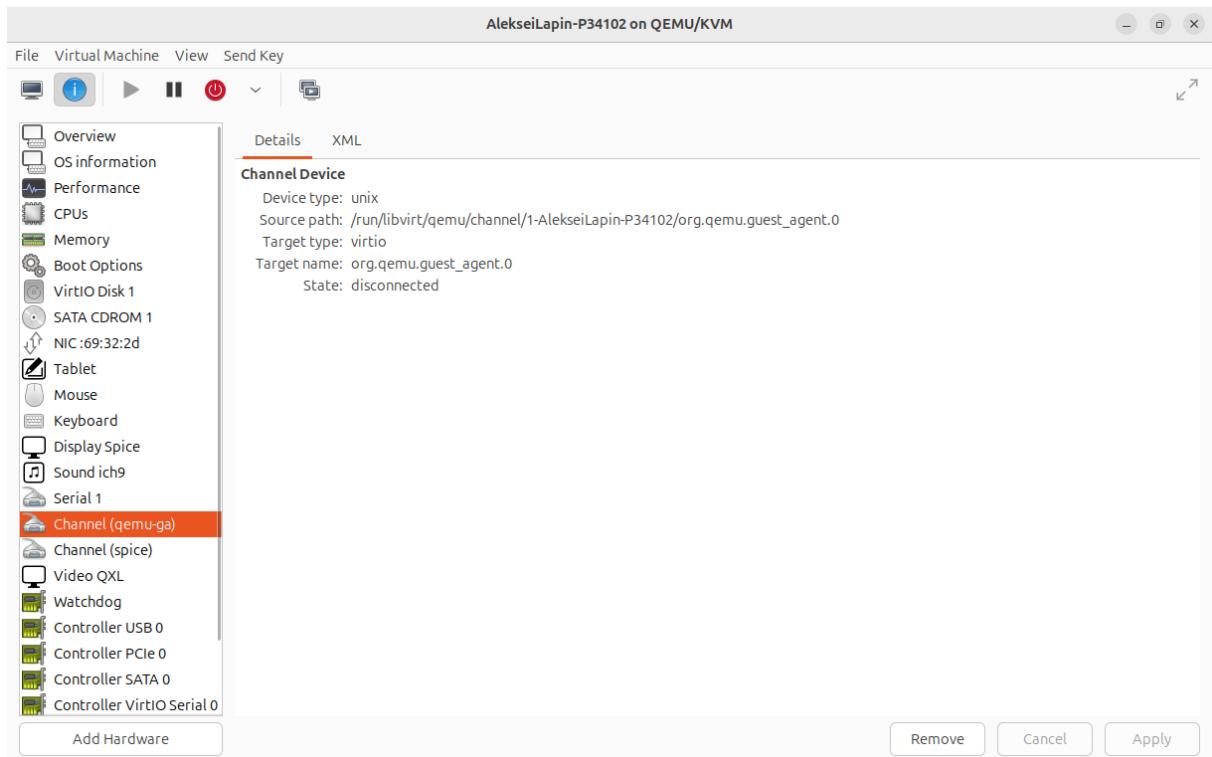


Рисунок 56. Channel (qemu-ga)

В данном разделе показаны настройки канала связи QEMU Guest Agent, который обеспечивает взаимодействие между хостовой системой и гостевой виртуальной машиной.

Основные параметры:

- Device type: unix - тип устройства для локального взаимодействия
- Source path: путь к сокету QEMU Guest Agent
- Target type: virtio - использует драйвер virtio для эффективного взаимодействия
- Target name: org.qemu.guest_agent.0 - идентификатор агента
- State: disconnected - текущее состояние соединения

QEMU Guest Agent позволяет:

- Выполнять команды внутри гостевой системы из хоста
- Получать информацию о состоянии гостевой системы
- Синхронизировать время между хостом и гостем

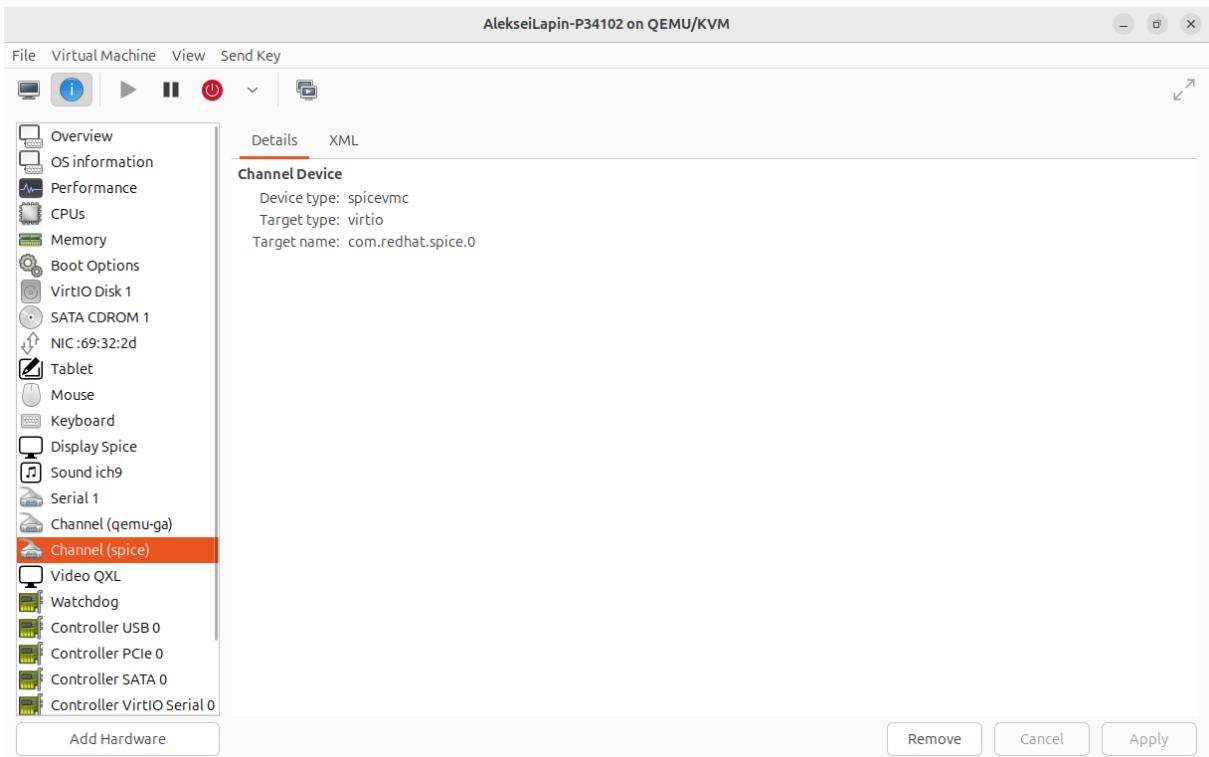


Рисунок 57. Channel (spice)

В данном разделе Virtual Machine Manager отображаются параметры канала SPICE для виртуальной машины. Конкретно показаны:

- Device type: spicevmc
- Target type: virtio
- Target name: com.redhat.spice.0

SPICE (Simple Protocol for Independent Computing Environments) - это протокол удаленного доступа к виртуальной машине , который позволяет:

- Получить доступ к графическому интерфейсу виртуальной машины
- Передавать ввод с клавиатуры и мыши
- Обеспечивать передачу аудио
- Поддерживать буфер обмена между хостом и гостевой ОС
- Обеспечивать USB-редирект (проброс USB-устройств в виртуальную машину)

Channel (spice) в данном случае - это канал связи между хостовой и гостевой системами, который обеспечивает эти функции. Использование virtio в качестве target type говорит о том, что используется оптимизированный драйвер для виртуализации, что обеспечивает лучшую производительность.

Этот компонент является важной частью системы виртуализации, так как

обеспечивает удобное взаимодействие пользователя с виртуальной машиной через графический интерфейс.

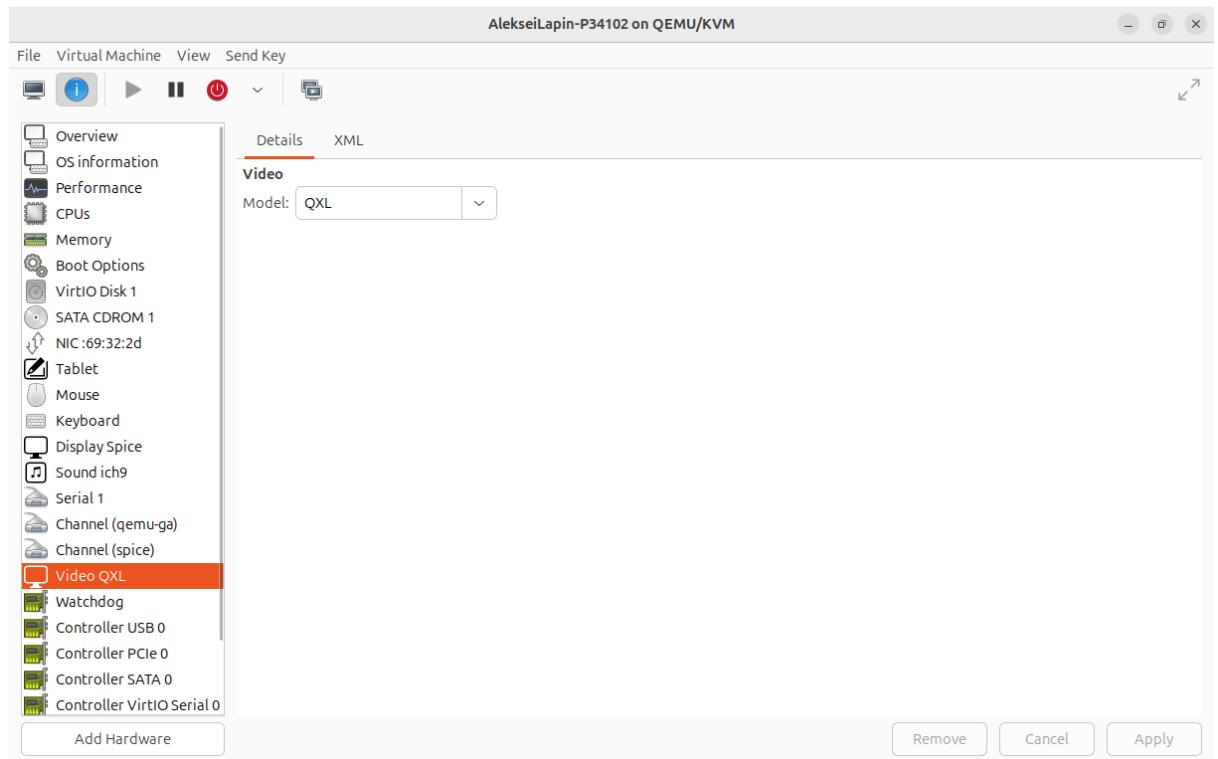


Рисунок 58. Video QXL

В данном разделе Virtual Machine Manager отображаются настройки видеоадаптера виртуальной машины:

Model: QXL - это специализированный видеодрайвер QEMU, который обеспечивает:

1. Улучшенную производительность графической подсистемы
 - a. Аппаратное ускорение 2D-графики
 - b. Поддержку высоких разрешений
 - c. Оптимизированную работу с протоколом SPICE
2. Расширенные возможности
 - a. Динамическое изменение разрешения экрана
 - b. Поддержка множества мониторов
 - c. Улучшенное качество отображения
3. Преимущества использования
 - a. Снижение нагрузки на CPU
 - b. Более плавная работа графического интерфейса
 - c. Лучшая совместимость с современными операционными системами

QXL является рекомендуемым драйвером для использования с протоколом SPICE и обеспечивает наилучший пользовательский опыт при работе с графическим интерфейсом виртуальной машины.

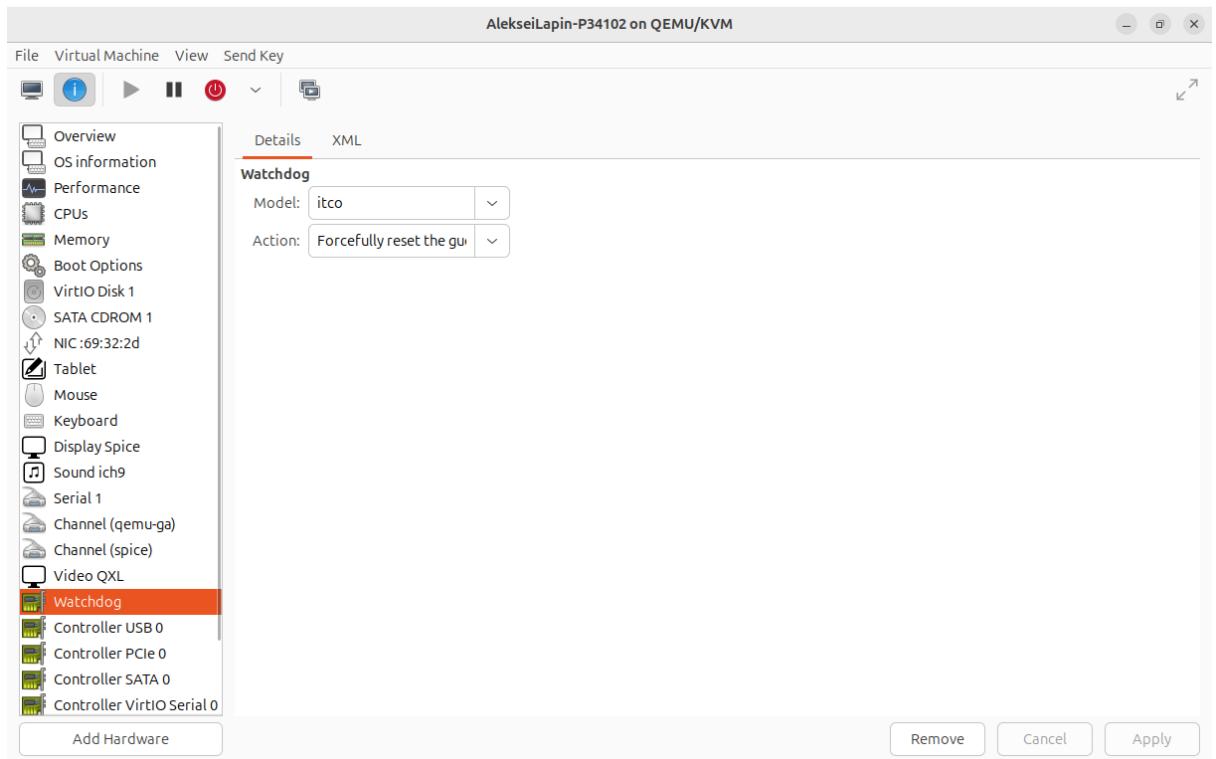


Рисунок 59. Watchdog

В данном разделе Virtual Machine Manager отображаются настройки Watchdog (сторожевого таймера) для виртуальной машины:

1. Основные параметры:
 - a. Model: i6t0 (модель сторожевого таймера)
 - b. Action: Forcefully reset the guest (принудительная перезагрузка гостевой системы)
2. Назначение Watchdog:
 - a. Мониторинг работоспособности виртуальной машины
 - b. Автоматическое восстановление работы при зависании
 - c. Защита от сбоев и зацикливания системы
3. Принцип работы:
 - a. Периодическая проверка отклика от гостевой системы
 - b. Автоматическое выполнение заданного действия при отсутствии отклика

- c. Принудительная перезагрузка системы при обнаружении проблем
4. Преимущества использования:

- a. Повышение отказоустойчивости виртуальной машины
- b. Автоматическое восстановление после сбоев
- c. Минимизация времени простоя системы

Watchdog является важным компонентом для обеспечения стабильной работы виртуальной машины, особенно в производственных средах, где требуется высокая доступность системы.

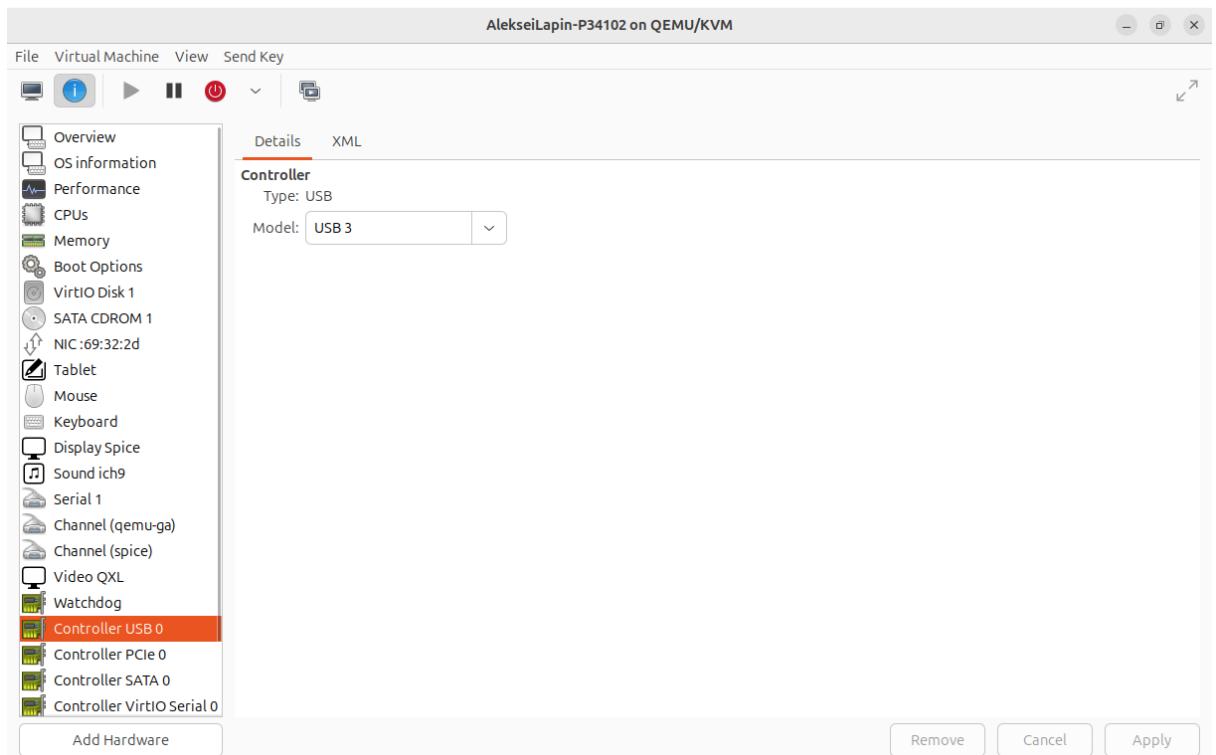


Рисунок 60. Controller USB 0

В данном разделе Virtual Machine Manager отображаются настройки USB-контроллера виртуальной машины:

1. Основные параметры:
 - a. Type: USB (тип контроллера)
 - b. Model: USB 3 (версия USB-протокола)
2. Функциональные возможности:
 - a. Поддержка подключения USB-устройств к виртуальной машине

- b. Обеспечение высокоскоростной передачи данных (до 5 Гбит/с для USB 3.0)
 - c. Возможность горячего подключения устройств
3. Преимущества USB 3:
- a. Повышенная скорость передачи данных
 - b. Обратная совместимость с USB 2.0 и 1.1

USB-контроллер является важным компонентом виртуальной машины, обеспечивающим взаимодействие с физическими USB-устройствами хост-системы.

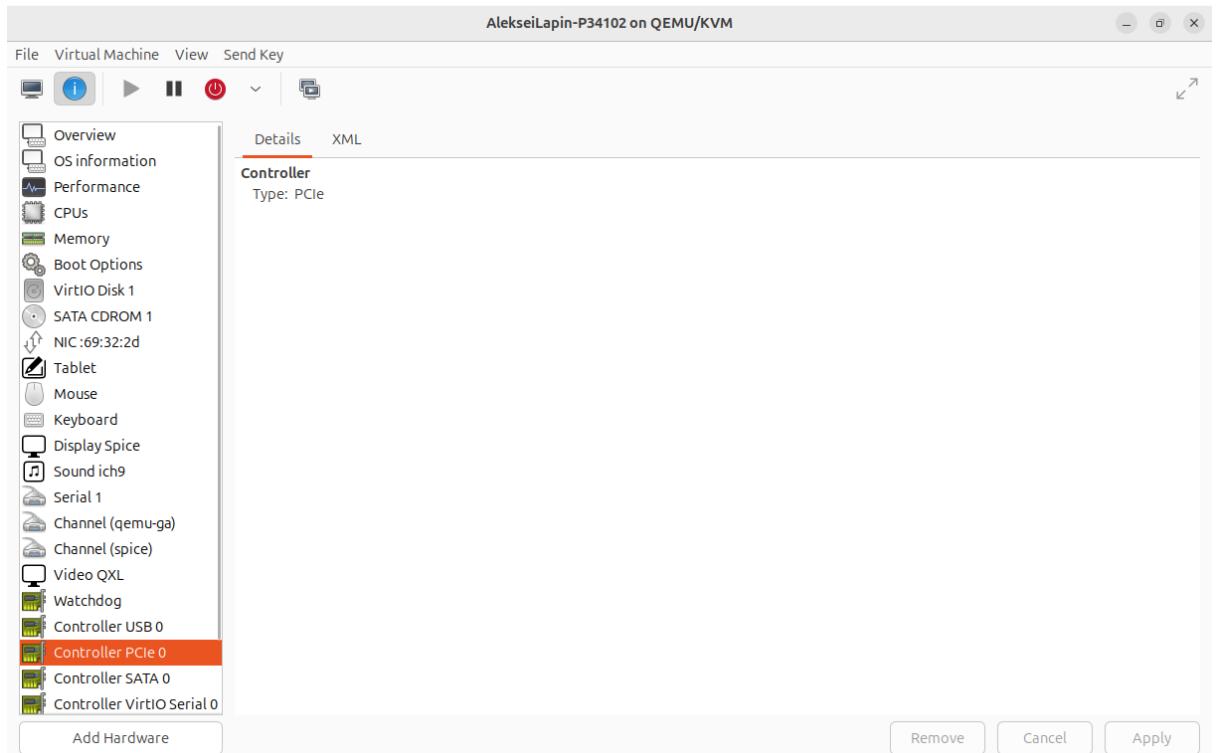


Рисунок 61. Controller PCIe 0

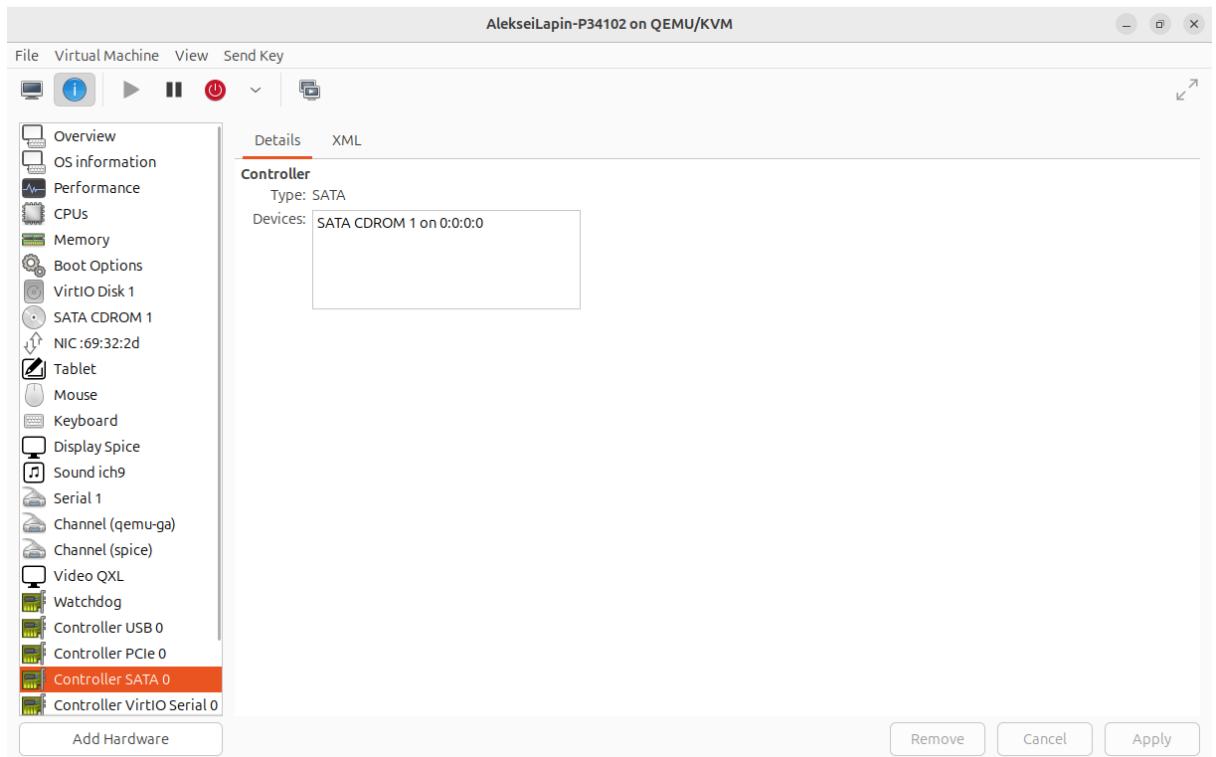


Рисунок 62. Controller SATA 0

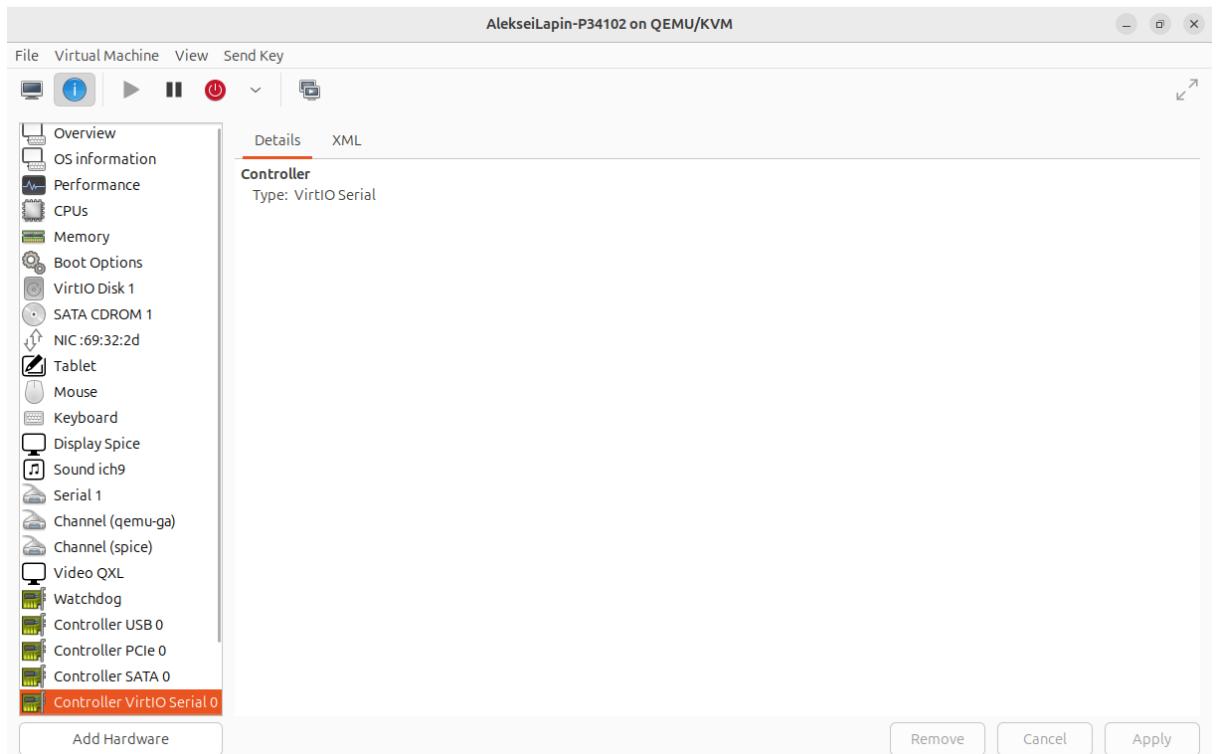


Рисунок 63. Controller VirtIO Serial 0

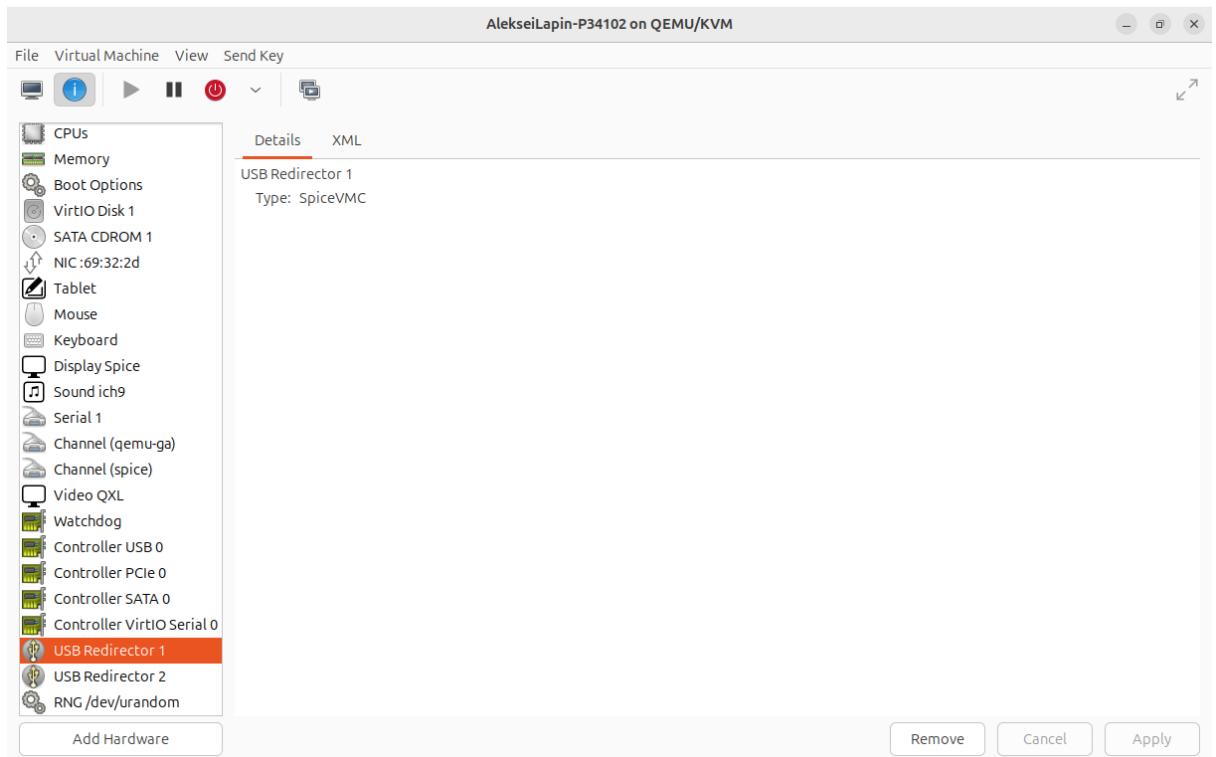


Рисунок 64. USB Redirector 1

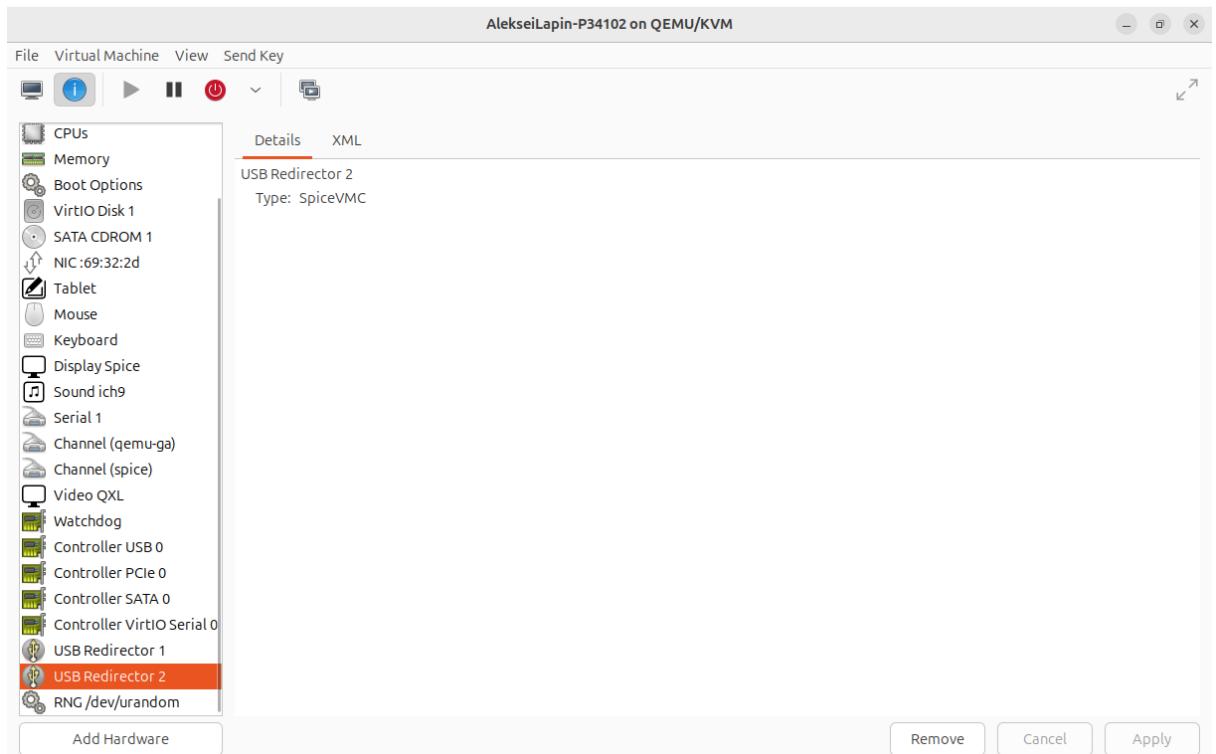


Рисунок 65. USB Redirector

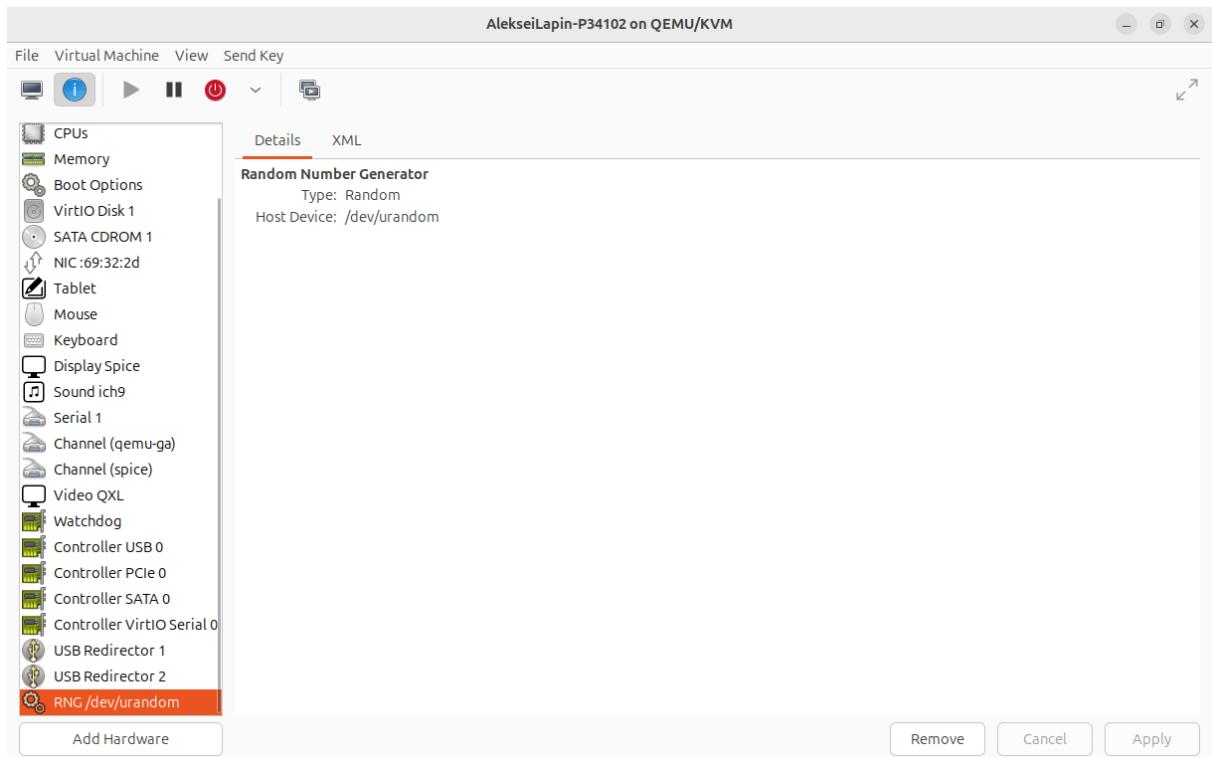


Рисунок 66. RNG

Подведем итоги:

Virtual Machine Manager отображаются параметры виртуальной машины, которые позволяют настроить и управлять её работой. Вот краткое описание некоторых из этих параметров:

1. Overview: Общие сведения о виртуальной машине, включая её состояние и основные характеристики.
2. OS information: Информация о операционной системе, установленной на виртуальной машине.
3. Performance (Производительность): Информация о производительности виртуальной машины, включая использование CPU, памяти, Disk IO и Network IO.
4. CPUs (Процессоры): Настройка процессоров, выделенных для виртуальной машины.
5. Memory (Память): Настройка объема оперативной памяти, выделенной для виртуальной машины.
6. Boot Options (Параметры загрузки): Настройка порядка загрузки устройств и других параметров загрузки.
7. Virtio Disk1, SATA CDROM 1, NIC:69:32:2d (Виртуальные диски, CD-ROM и

- сетевая карта): Настройки виртуальных дисков, оптических приводов и сетевых интерфейсов.
- 8. Tablet, Mouse, Keyboard (Графический планшет, мышь, клавиатура): Настройки устройств ввода, таких как графический планшет, мышь и клавиатура.
 - 9. Display Spice, Sound idb9 (Дисплей и звук): Настройки дисплея и звука для виртуальной машины.
 - 10. Serial1, Channel (qemu-ga), Channel (spice) (Последовательный порт, каналы): Настройки последовательного порта и каналов для управления и мониторинга виртуальной машины.
 - 11. Video QXL, Watchdog (Видео и сторожевой таймер): Настройки видеодрайвера и сторожевого таймера для виртуальной машины.
 - 12. Controller USB0, Controller PCIe0, Controller SATA0, Controller Virtio Serial0 (Контроллеры): Настройки различных контроллеров, таких как USB, PCIe, SATA и Virtio Serial.

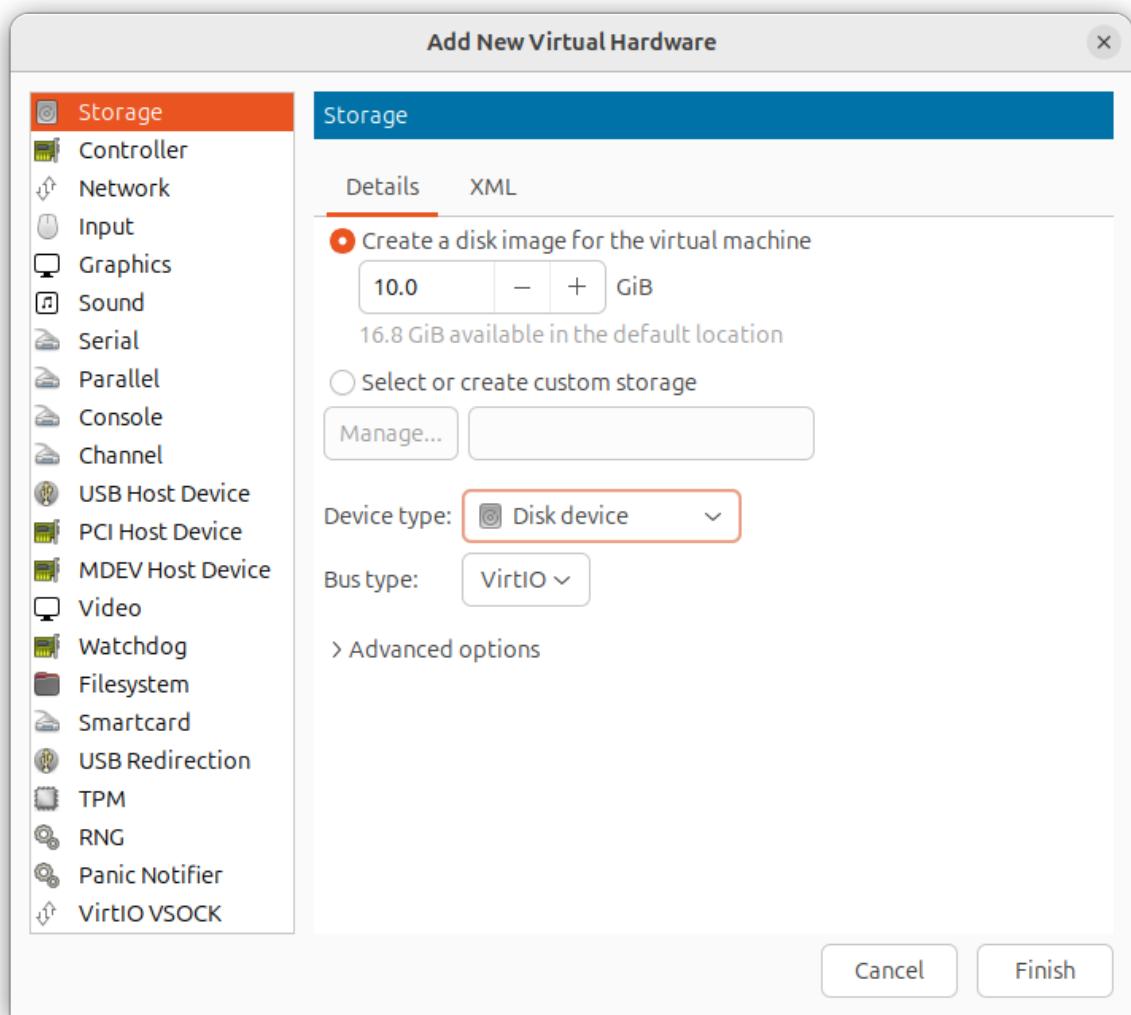


Рисунок 67. Добавление оборудования

Добавим еще один жесткий диск.

В данном разделе Virtual Machine Manager отображаются параметры для добавления нового виртуального оборудования, а именно настройки хранилища (Storage). Этот раздел позволяет создать образ диска для виртуальной машины, выбрать или настроить пользовательское хранилище, а также указать тип устройства (в данном случае — Disk device).

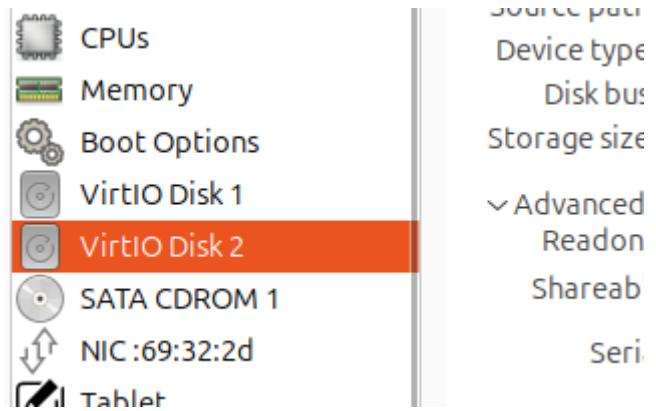


Рисунок 68. Добавленный диск

Создадим Snapshot системы. Для этого переходим в View > Snapshots

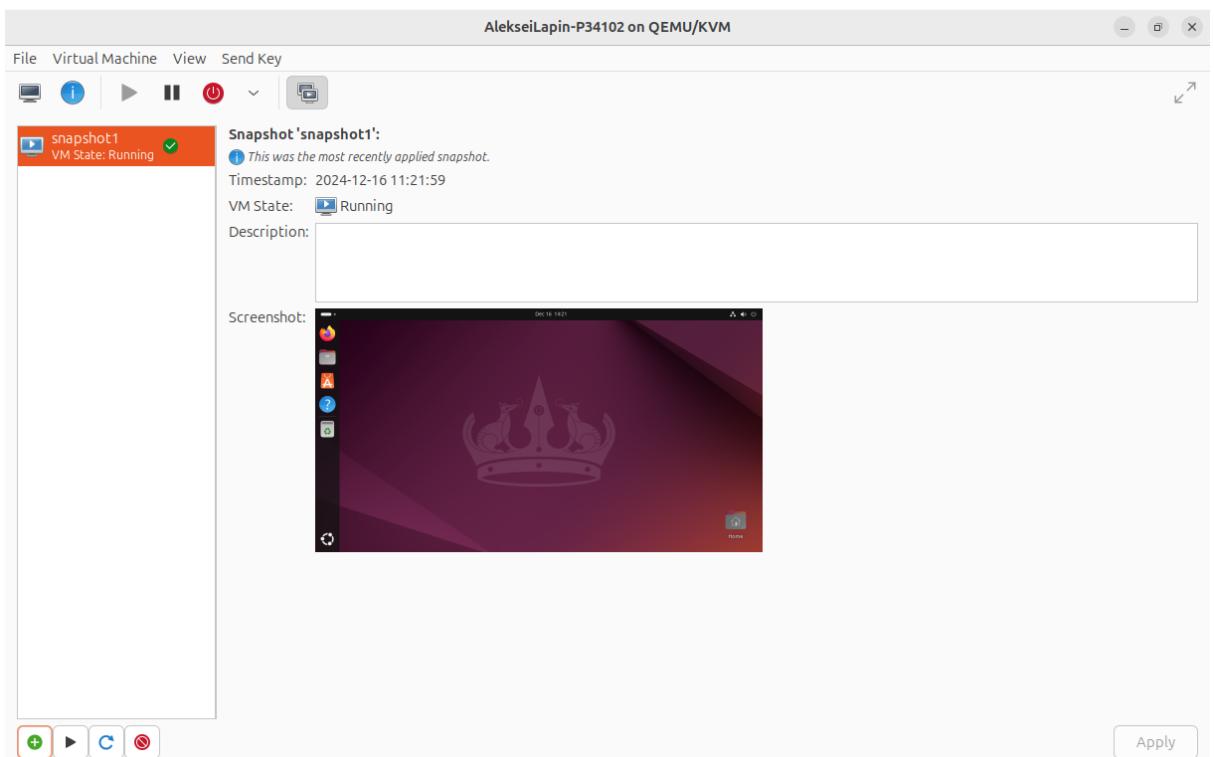


Рисунок 69. Snapshot системы

Скачаем Yandex Browser. Сделаем снимок системы. Вернемся к предыдущему состоянию.

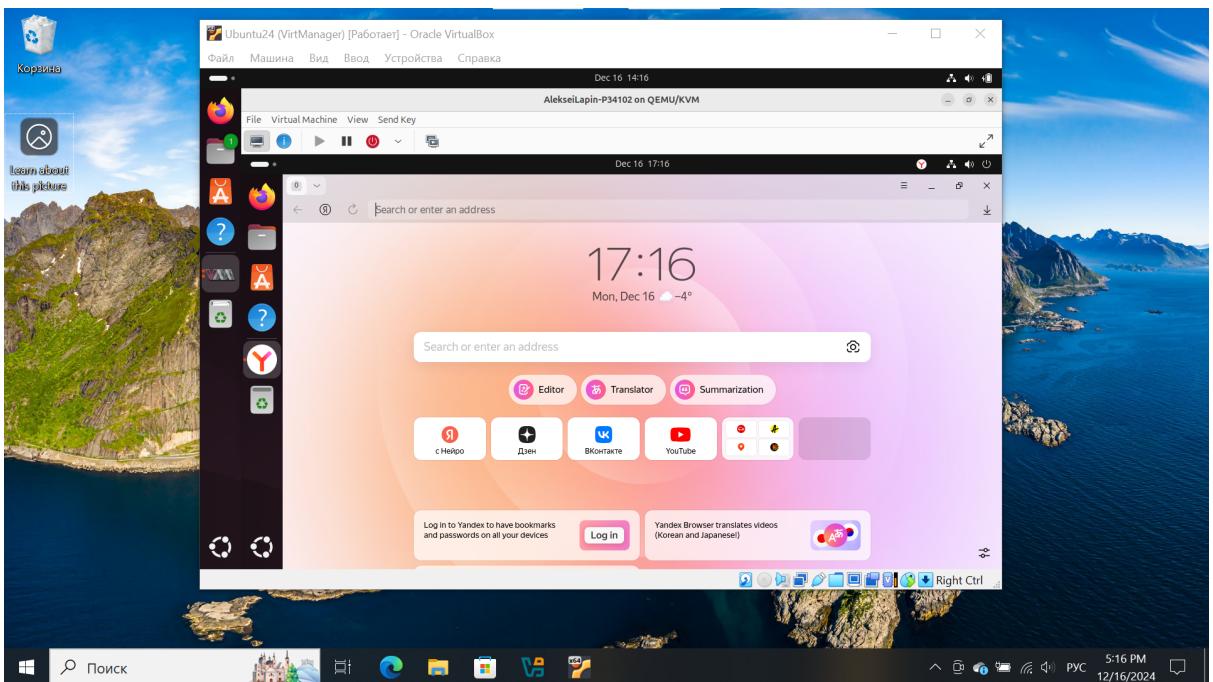


Рисунок 70. Установленный Яндекс Браузер

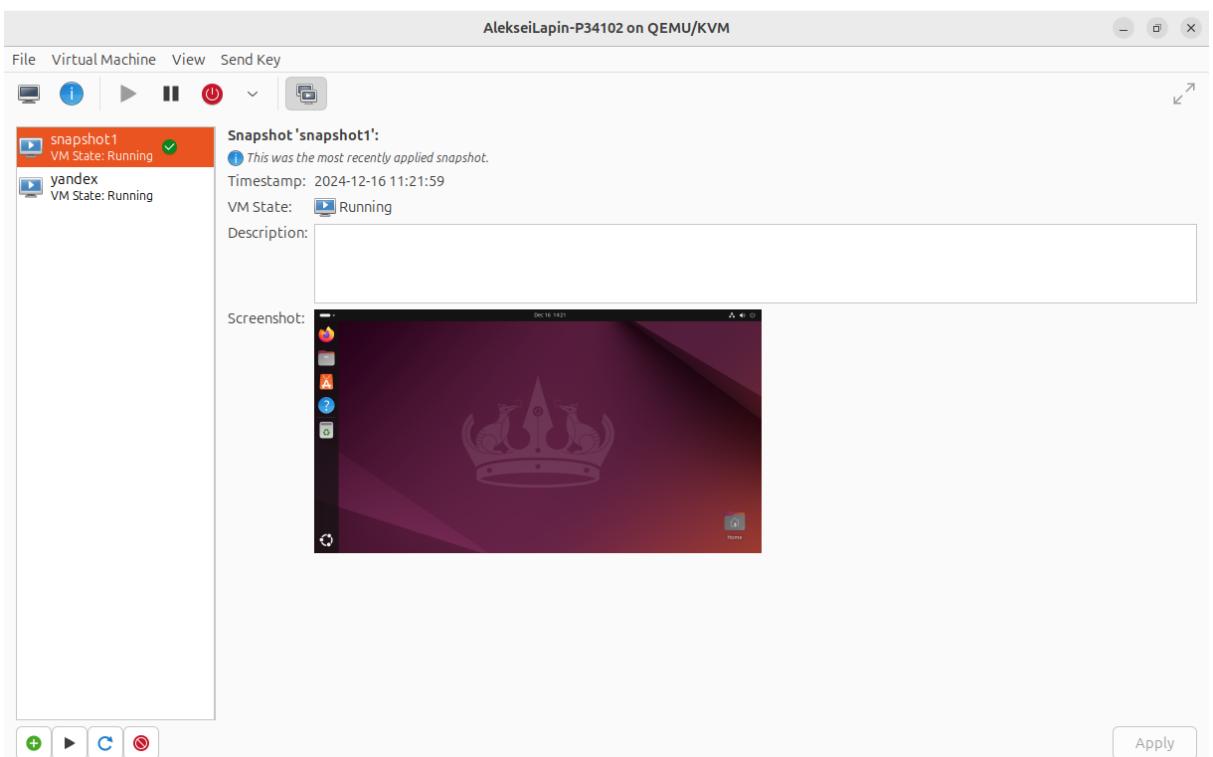


Рисунок 71. Восстанавливаем состояние без Яндекса

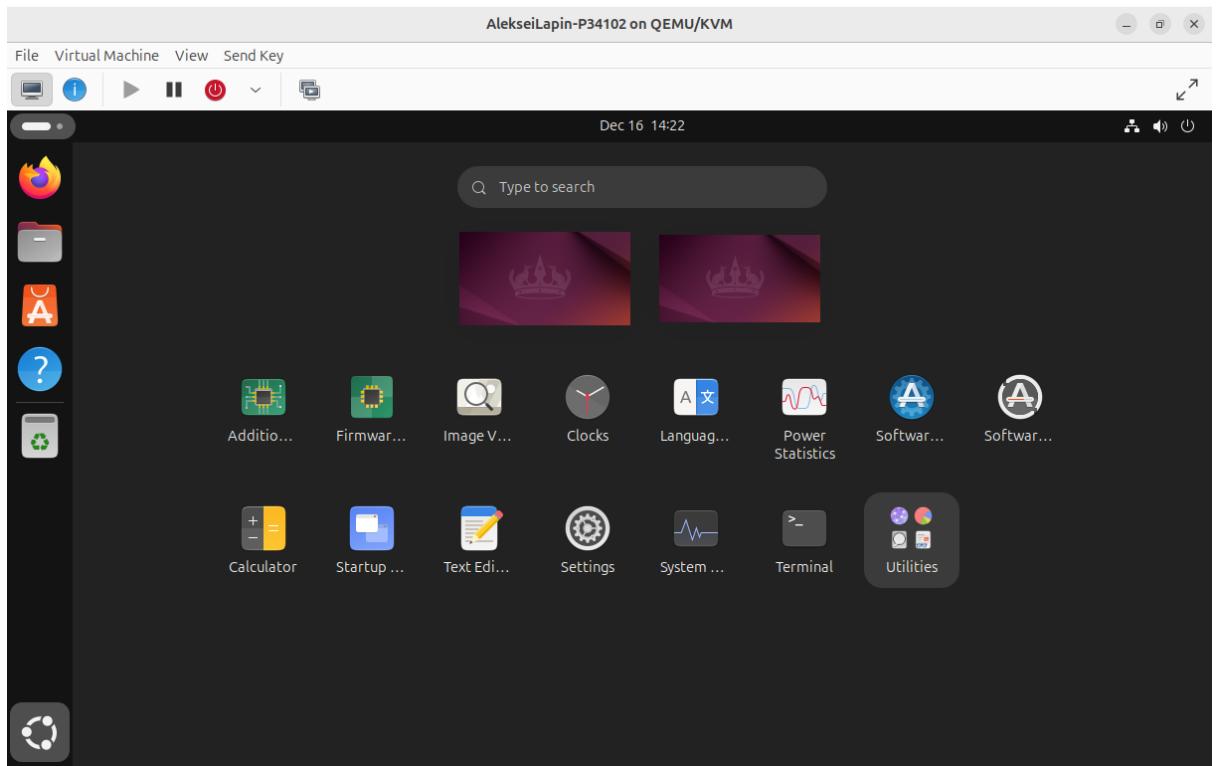


Рисунок 72. Яндекса нет

Сетевые настройки гостевой машины:

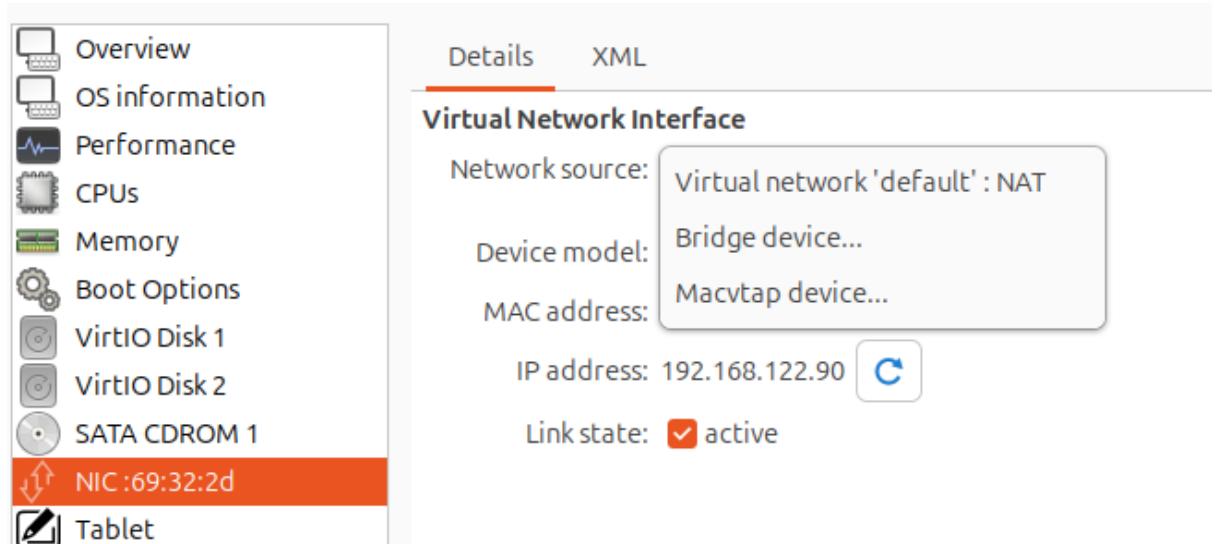


Рисунок 73. Сетевые настройки гостевой машины

Управление виртуальной машиной из командной строки virsh

Посмотреть запущенные виртуальные хосты (все доступные --all):

virsh list

Перезагрузить хост можно:

virsh reboot \$VM_NAME

Остановить виртуальную машину:

virsh stop \$VM_NAME

Выполнить halt:

virsh destroy \$VM_NAME

Запуск:

virsh start \$VM_NAME

Отключение:

virsh shutdown \$VM_NAME

Добавить в автозапуск:

virsh autostart \$VM_NAME

Склонировать систему, чтобы в будущем использовать её как основу для других виртуальных ОС

virt-clone –help

The screenshot shows a terminal window titled "alex@Ubuntu24:~". The terminal displays the following commands and their output:

```
alex@Ubuntu24:~$ virsh list
 Id  Name          State
 --  --
 4   AlekseiLapin-P34102  running

alex@Ubuntu24:~$ virsh shutdown AlekseiLapin-P34102
Domain 'AlekseiLapin-P34102' is being shutdown

alex@Ubuntu24:~$ virsh list
 Id  Name      State
 --  --

```

Рисунок 74. Выключение виртуальной машины из командной строки

Использование утилиты virt-install

Текст задания

1. Установите в системы следующие пакеты:

```
$sudo apt install virt-install libosinfo-bin
```

2. Проверьте какие ОС доступны из репозитория
3. Отсортируйте все ОС Windows и сохраните список
4. Отсортируйте все ОС Fedora и сохраните список
5. Добавить виртуальную машину с помощью virt-install
6. Объясните значение параметров используемых при вводе команды.
7. Проверьте запуск виртуальной машины.
8. Как можно запустить машину в фоновом режиме?
9. Подключиться к виртуальной машине по ssh. Что для этого нужно сделать?

Выполнение

1. Установка необходимых пакетов

Для работы с утилитой virt-install и проверки доступных операционных систем, необходимо установить следующие пакеты:

```
sudo apt install virtinst libosinfo-bin
```

virtinst - Утилита для создания виртуальных машин.

libosinfo-bin: Инструмент для запроса информации о поддерживаемых операционных системах.

2. Проверка доступных операционных систем

Чтобы узнать, какие операционные системы доступны для использования с virt-install, выполните следующую команду:

```
osinfo-query os
```

Эта команда выведет список всех поддерживаемых операционных систем, которые можно использовать при создании виртуальных машин.

Short ID	Name	Version	ID
almalinux8	AlmaLinux 8	8	http://almalinux.org/almalinux/8
almalinux9	AlmaLinux 9	9	http://almalinux.org/almalinux/9
alpinelinux3.10	Alpine Linux 3.10	3.10	http://alpinelinux.org/alpinelinux/3.10
alpinelinux3.11	Alpine Linux 3.11	3.11	http://alpinelinux.org/alpinelinux/3.11
alpinelinux3.12	Alpine Linux 3.12	3.12	http://alpinelinux.org/alpinelinux/3.12
alpinelinux3.13	Alpine Linux 3.13	3.13	http://alpinelinux.org/alpinelinux/3.13
alpinelinux3.14	Alpine Linux 3.14	3.14	http://alpinelinux.org/alpinelinux/3.14
alpinelinux3.15	Alpine Linux 3.15	3.15	http://alpinelinux.org/alpinelinux/3.15
alpinelinux3.16	Alpine Linux 3.16	3.16	http://alpinelinux.org/alpinelinux/3.16
alpinelinux3.17	Alpine Linux 3.17	3.17	http://alpinelinux.org/alpinelinux/3.17
alpinelinux3.18	Alpine Linux 3.18	3.18	http://alpinelinux.org/alpinelinux/3.18
alpinelinux3.19	Alpine Linux 3.19	3.19	http://alpinelinux.org/alpinelinux/3.19
alpinelinux3.5	Alpine Linux 3.5	3.5	http://alpinelinux.org/alpinelinux/3.5
alpinelinux3.6	Alpine Linux 3.6	3.6	http://alpinelinux.org/alpinelinux/3.6
alpinelinux3.7	Alpine Linux 3.7	3.7	http://alpinelinux.org/alpinelinux/3.7
alpinelinux3.8	Alpine Linux 3.8	3.8	http://alpinelinux.org/alpinelinux/3.8
alpinelinux3.9	Alpine Linux 3.9	3.9	http://alpinelinux.org/alpinelinux/3.9
alt.p10	ALT p10 StarterKits	p10	http://altlinux.org/alt/p10.starterkits
alt.p8	ALT p8 StarterKits	p8	http://altlinux.org/alt/p8.starterkits
alt.p9	ALT p9 StarterKits	p9	http://altlinux.org/alt/p9.starterkits
alt.sisyphus	ALT regular	sisyphus	http://altlinux.org/alt/sisyphus
alt10.0	ALT 10.0	10.0	http://altlinux.org/alt/10.0
alt10.1	ALT 10.1	10.1	http://altlinux.org/alt/10.1
alt8.0	ALT 8 Education	8.0	http://altlinux.org/alt/8.0
alt8.1	ALT 8.1	8.1	http://altlinux.org/alt/8.1
alt8.2	ALT 8.2	8.2	http://altlinux.org/alt/8.2
alt9.0	ALT 9.0	9.0	http://altlinux.org/alt/9.0
alt9.1	ALT 9.1	9.1	http://altlinux.org/alt/9.1
alt9.2	ALT 9.2	9.2	http://altlinux.org/alt/9.2
altlinux1.0	Mandrake RE Spring 2001	1.0	http://altlinux.org/altlinux/1.0
altlinux2.0	ALT Linux 2.0	2.0	http://altlinux.org/altlinux/2.0
altlinux2.2	ALT Linux 2.2	2.2	http://altlinux.org/altlinux/2.2
altlinux2.4	ALT Linux 2.4	2.4	http://altlinux.org/altlinux/2.4
altlinux3.0	ALT Linux 3.0	3.0	http://altlinux.org/altlinux/3.0

Рисунок 75. Вывод команды osinfo-query

3. Отсортированные списки ОС

a. Отсортированный список ОС Windows

Чтобы отсортировать и сохранить список всех операционных систем Windows,

выполните:

```
osinfo-query os | grep -i windows > windows_os_list.txt
```

win1.0	Microsoft Windows 1.0	1.0	http://microsoft.com/win/1.0
win10	Microsoft Windows 10	10.0	http://microsoft.com/win/10
win11	Microsoft Windows 11	11.0	http://microsoft.com/win/11
win2.0	Microsoft Windows 2.0	2.0	http://microsoft.com/win/2.0
win2.1	Microsoft Windows 2.1	2.1	http://microsoft.com/win/2.1
win2k	Microsoft Windows 2000	5.0	http://microsoft.com/win/2k
win2k12	Microsoft Windows Server 2012	6.3	http://microsoft.com/win/2k12
win2k12r2	Microsoft Windows Server 2012 R2	6.3	http://microsoft.com/win/2k12r2
win2k16	Microsoft Windows Server 2016	10.0	http://microsoft.com/win/2k16
win2k19	Microsoft Windows Server 2019	10.0	http://microsoft.com/win/2k19
win2k22	Microsoft Windows Server 2022	10.0	http://microsoft.com/win/2k22
win2k3	Microsoft Windows Server 2003	5.2	http://microsoft.com/win/2k3
win2k3r2	Microsoft Windows Server 2003 R2	5.2	http://microsoft.com/win/2k3r2
win2k8	Microsoft Windows Server 2008	6.0	http://microsoft.com/win/2k8
win2k8r2	Microsoft Windows Server 2008 R2	6.1	http://microsoft.com/win/2k8r2
win3.1	Microsoft Windows 3.1	3.1	http://microsoft.com/win/3.1
win7	Microsoft Windows 7	6.1	http://microsoft.com/win/7
win8	Microsoft Windows 8	6.2	http://microsoft.com/win/8
win8.1	Microsoft Windows 8.1	6.3	http://microsoft.com/win/8.1
win95	Microsoft Windows 95	4.0	http://microsoft.com/win/95
win98	Microsoft Windows 98	4.1	http://microsoft.com/win/98
winme	Microsoft Windows Millennium Edition	4.9	http://microsoft.com/win/me
winnt3.1	Microsoft Windows NT Server 3.1	3.1	http://microsoft.com/winnt/3.1
winnt3.5	Microsoft Windows NT Server 3.5	3.5	http://microsoft.com/winnt/3.5
winnt3.51	Microsoft Windows NT Server 3.51	3.51	http://microsoft.com/winnt/3.51
winnt4.0	Microsoft Windows NT Server 4.0	4.0	http://microsoft.com/winnt/4.0
winvista	Microsoft Windows Vista	6.0	http://microsoft.com/win/vista
winxp	Microsoft Windows XP	5.1	http://microsoft.com/win/xp

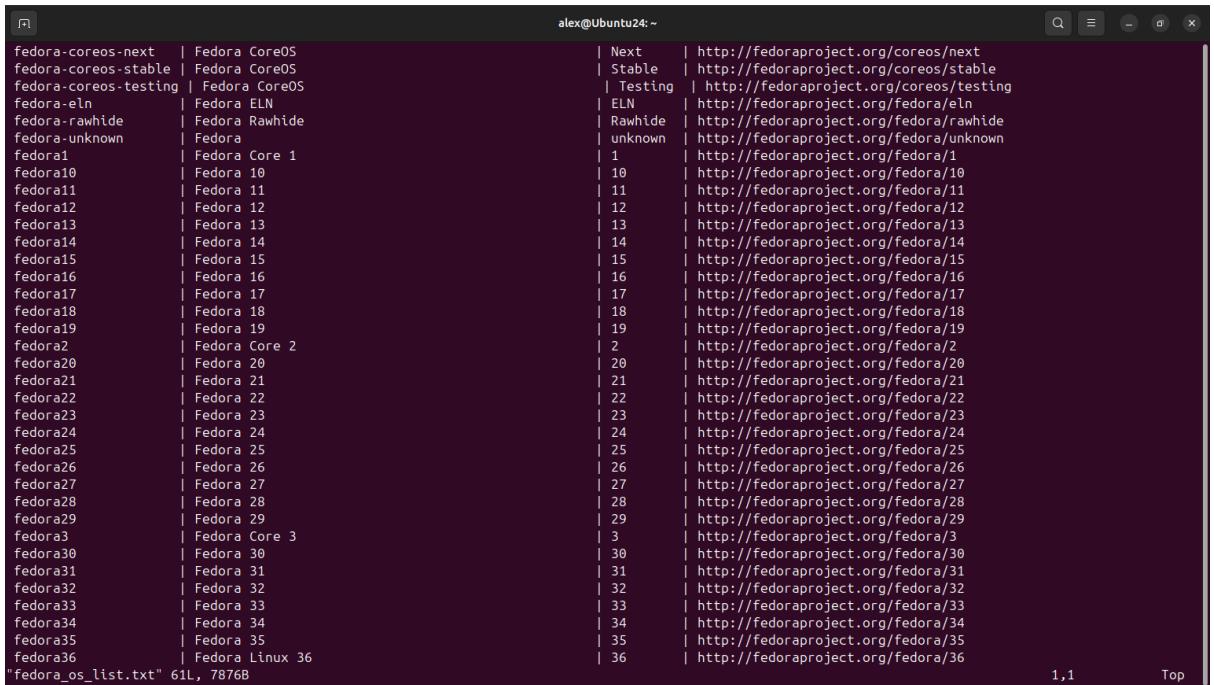
Рисунок 76. Отсортированный список Windows

b. Отсортированный список ОС Fedora

Чтобы отсортировать и сохранить список всех операционных систем Fedora,

выполните:

```
osinfo-query os | grep -i fedora > fedora_os_list.txt
```



The screenshot shows a terminal window titled "alex@Ubuntu24: ~". It displays a sorted list of Fedora versions, each with its name and a corresponding URL. The list starts with "fedora-coreos-next" and ends with "fedora36". The URLs follow a pattern like "http://fedoraproject.org/coreos/next", "http://fedoraproject.org/coreos/stable", etc., up to "http://fedoraproject.org/fedora/36". The terminal window has a dark background with light-colored text. At the bottom right, there are status icons for battery, signal, and volume.

Fedora Version	URL
fedora-coreos-next	http://fedoraproject.org/coreos/next
fedora-coreos-stable	http://fedoraproject.org/coreos/stable
fedora-coreos-testing	http://fedoraproject.org/coreos/testing
fedora-eln	http://fedoraproject.org/fedora/eln
fedora-rawhide	http://fedoraproject.org/fedora/rawhide
fedora-unknown	http://fedoraproject.org/fedora/unknown
fedora1	http://fedoraproject.org/fedora/1
fedora10	http://fedoraproject.org/fedora/10
fedora11	http://fedoraproject.org/fedora/11
fedora12	http://fedoraproject.org/fedora/12
fedora13	http://fedoraproject.org/fedora/13
fedora14	http://fedoraproject.org/fedora/14
fedora15	http://fedoraproject.org/fedora/15
fedora16	http://fedoraproject.org/fedora/16
fedora17	http://fedoraproject.org/fedora/17
fedora18	http://fedoraproject.org/fedora/18
fedora19	http://fedoraproject.org/fedora/19
fedora2	http://fedoraproject.org/fedora/2
fedora20	http://fedoraproject.org/fedora/20
fedora21	http://fedoraproject.org/fedora/21
fedora22	http://fedoraproject.org/fedora/22
fedora23	http://fedoraproject.org/fedora/23
fedora24	http://fedoraproject.org/fedora/24
fedora25	http://fedoraproject.org/fedora/25
fedora26	http://fedoraproject.org/fedora/26
fedora27	http://fedoraproject.org/fedora/27
fedora28	http://fedoraproject.org/fedora/28
fedora29	http://fedoraproject.org/fedora/29
fedora3	http://fedoraproject.org/fedora/3
fedora30	http://fedoraproject.org/fedora/30
fedora31	http://fedoraproject.org/fedora/31
fedora32	http://fedoraproject.org/fedora/32
fedora33	http://fedoraproject.org/fedora/33
fedora34	http://fedoraproject.org/fedora/34
fedora35	http://fedoraproject.org/fedora/35
fedora36	http://fedoraproject.org/fedora/36

Рисунок 77. Отсортированный список Fedora

4. Создание виртуальной машины с помощью virt-install

Для создания виртуальной машины с именем AlekseiLapin_P34102, выполните следующую команду:

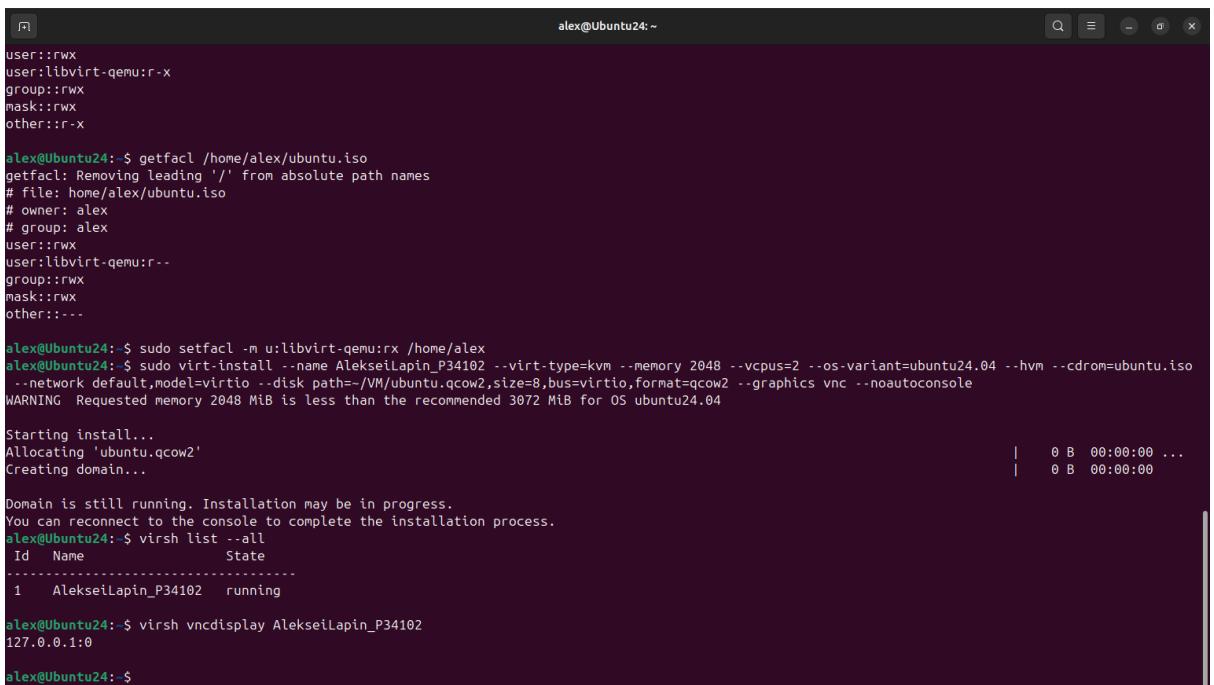
```
virt-install \
--name AlekseiLapin_P34102 \
--virt-type=kvm \
--memory 2048 \
--vcpus=2 \
--os-variant=ubuntu24.04 \
--hvm \
--cdrom=ubuntu.iso \
--network default,model=virtio \
--disk path=~/VM/ubuntu.qcow2,size=8,bus=virtio,format=qcow2 \
--graphics vnc \
--noautoconsole
```

5. Объяснение параметров команды

- **--name FIO_Group:** устанавливает имя виртуальной машины как

FIO_Group.

- `--virt-type=kvm`: указывает, что виртуальная машина будет использовать гипервизор KVM.
- `--memory 2048`: выделяет 2048 МБ оперативной памяти для виртуальной машины.
- `--vcpus=2`: устанавливает количество виртуальных процессоров равным 2.
- `--os-variant=fedora31`: указывает, что операционная система будет Fedora 31, что позволяет оптимизировать настройки для этой ОС.
- `--hvm`: указывает, что виртуальная машина будет полностью виртуализированной.
- `--cdrom=fedora.iso`: указывает путь к ISO-образу Fedora для установки.
- `--network default,model=virtio`: Подключает виртуальную машину к сети по умолчанию, используя модель сетевого адаптера virtio.
- `--disk path=~/VM/fedora31.qcow2,size=8,bus=virtio,format=qcow2`: Создает виртуальный диск размером 8 ГБ в формате qcow2 с использованием шины virtio.
- `--graphics vnc`: устанавливает графический интерфейс через VNC.
- `--noautoconsole`: отключает автоматическое подключение к консоли виртуальной машины.



```
alex@Ubuntu24:~$ getfacl /home/alex/ubuntu.iso
getfacl: Removing leading '/' from absolute path names
# file: home/alex/ubuntu.iso
# owner: alex
# group: alex
user::rwx
user:libvirt-qemu:r-x
group::rwx
mask::rwx
other::r-x

alex@Ubuntu24:~$ sudo setfacl -m u:libvirt-qemu:rwx /home/alex
alex@Ubuntu24:~$ sudo virt-install --name AlekseiLapin_P34102 --virt-type=kvm --memory 2048 --vcpus=2 --os-variant=ubuntu24.04 --hvm --cdrom=ubuntu.iso
--network default,model=virtio --disk path=~/VM/ubuntu.qcow2,size=8,bus=virtio,format=qcow2 --graphics vnc --noautoconsole
WARNING Requested memory 2048 MiB is less than the recommended 3072 MiB for OS ubuntu24.04

Starting install...
Allocating 'ubuntu.qcow2'
Creating domain... | 0 B 00:00:00 ...
| 0 B 00:00:00 ...

Domain is still running. Installation may be in progress.
You can reconnect to the console to complete the installation process.
alex@Ubuntu24:~$ virsh list --all
 Id  Name           State
 -----
 1   AlekseiLapin_P34102  running

alex@Ubuntu24:~$ virsh vncdisplay AlekseiLapin_P34102
127.0.0.1:0

alex@Ubuntu24:~$
```

Рисунок 78. Вывод команды `virt-install`

Установка началась, теперь подключимся к VNC, чтобы завершить установку. Для этого воспользуемся командой Connections.

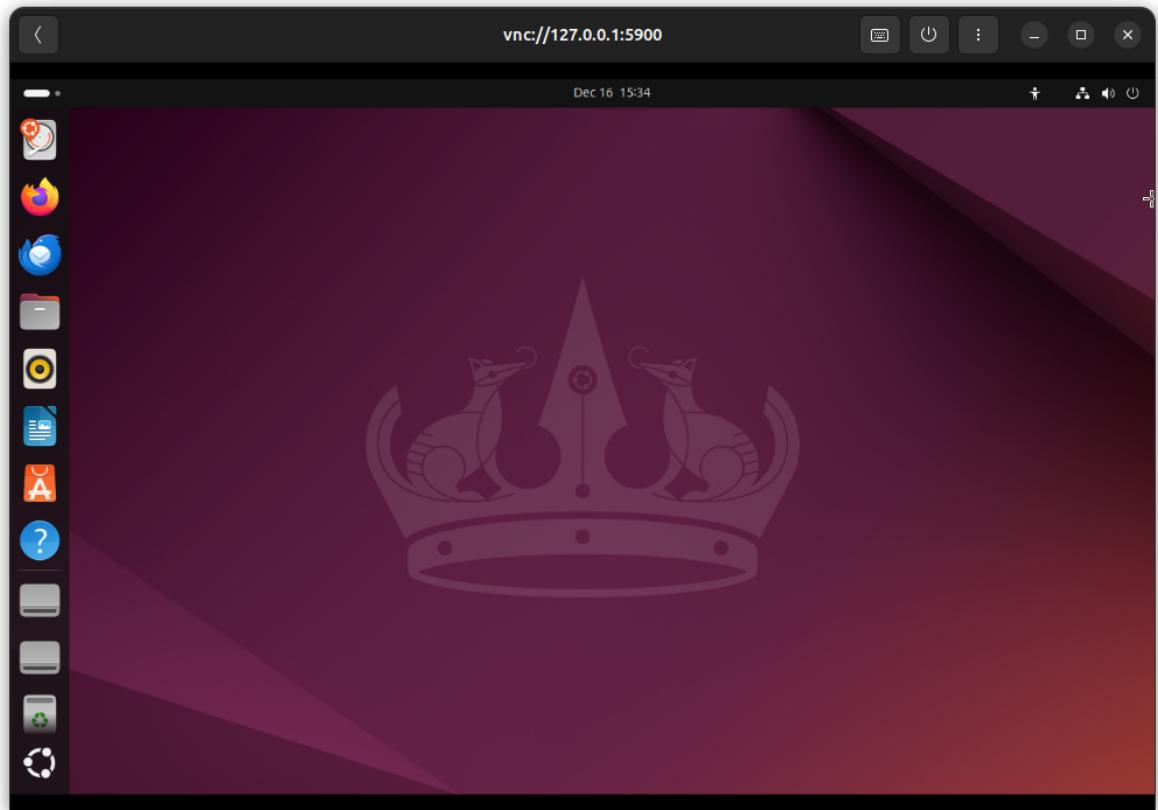


Рисунок 79. Подключение к виртуальной машине по VNC

5. Запуск машины в фоновом режиме

```
virsh start AlekseiLapin_P34102
```

6. Подключение к виртуальной машине по SSH

Для подключения к виртуальной машине по SSH, выполним следующие шаги:

1. Убедимся, что SSH-сервер установлен и запущен на виртуальной машине.

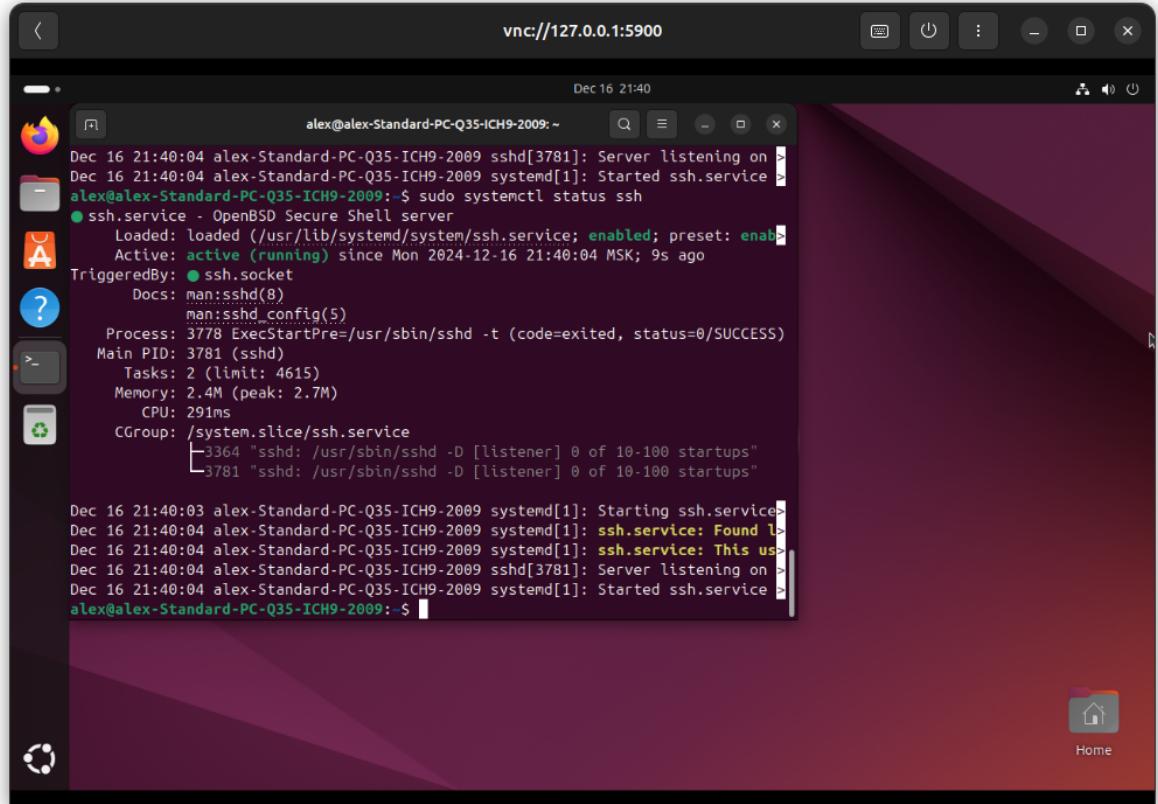


Рисунок 80. Проверка статуса ssh на виртуальной машине

2. Получим IP-адрес виртуальной машины:

```
virsh domifaddr AlekseiLapin_P34102
```

Подключимся по SSH:

```
ssh alex@ip_address
```

```
Ubuntu24 (QEMU) [Работает] - Oracle VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Dec 16 18:42
alex@alex-Standard-PC-Q35-ICH9-2009:~>

alex@Ubuntu24: $ virsh domifaddr AlekseiLapin_P34102
Name      MAC address   Protocol  Address
vnet6    52:54:00:d7:cd:03  ipv4     192.168.122.252/24

alex@Ubuntu24: $ ssh alex@192.168.122.252
The authenticity of host '192.168.122.252 (192.168.122.252)' can't be established.
ED25519 key fingerprint is SHA256:hbjfre55K8TK7ugM28nv98w+gWbj6YUf0fsf+Q449vs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.122.252' (ED25519) to the list of known hosts.
alex@192.168.122.252's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-51-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

124 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

alex@alex-Standard-PC-Q35-ICH9-2009: ~
```

Рисунок 81. Подключение к виртуальной машине по ssh

Анализ результатов лабораторной работы

В данном разделе проводится анализ результатов выполнения лабораторной работы по виртуализации в Unix-подобных системах. Целью анализа является оценка эффективности использования различных инструментов и методов виртуализации, а также выявление возможностей и ограничений каждого из них.

Анализ использования инструментов виртуализации

1. VirtualBox

- Были выполнены задачи по добавлению жесткого диска, созданию общей папки между хостовой и гостевой ОС, а также установке Live-образа операционной системы.
- Результаты: VirtualBox показал себя как удобный и надежный инструмент для создания виртуальных машин с широкими возможностями настройки.

9. Gnome Boxes/Connections:

- Были выполнены задачи по установке приложения, созданию виртуальной машины, изучению её свойств, а также подключению к удаленным рабочим столам.
- Результаты: Gnome Boxes оказался простым и удобным инструментом для быстрого создания виртуальных машин. Однако его функциональность по настройке крайне ограничена. Для подключения к удаленным рабочим столам потребовалось использовать отдельное приложение Gnome Connections.

10. Virt-Manager:

- Были выполнены задачи по установке QEMU, созданию виртуальной машины, настройке её параметров, а также управлению снимками состояния системы.
- Результаты: Virt-Manager показал себя как мощный инструмент для управления виртуальными машинами с широкими возможностями настройки.

11. QEMU и virt-install

- Были выполнены задачи по созданию виртуальной машины, настройке параметров загрузки и сетевых настроек, а также автоматизации

процесса создания виртуальных машин.

- Результаты: QEMU и virt-install предоставили низкоуровневый контроль над созданием и запуском виртуальных машин. Однако работа с командной строкой требовала более глубоких знаний о параметрах и настройках.

Сравнение инструментов виртуализации

1. Эффективность инструментов:

- a. VirtualBox и Virt-Manager показали себя как наиболее функциональные инструменты для управления виртуальными машинами. Они предоставляют широкие возможности настройки и управления, включая работу с общими папками, снимками и сетевыми настройками.
- b. Gnome Boxes удобен для быстрого создания виртуальных машин, но его функциональность ограничена.
- c. QEMU и virt-install эффективны для автоматизации создания виртуальных машин, но требуют более глубоких знаний о параметрах командной строки и настройках.

2. Удобство использования:

- a. Графические интерфейсы (VirtualBox, Virt-Manager, Gnome Boxes) оказались более удобными для начинающих пользователей, так как они предоставляют интуитивно понятный интерфейс.
- b. Командная строка (QEMU, virt-install, libvirt, virsh) требует более высокой квалификации, но позволяет автоматизировать процессы и управлять виртуальными машинами в больших масштабах.

3. Ограничения:

- a. Некоторые инструменты, такие как Gnome Boxes, имеют ограниченный функционал, что может быть недостатком в сложных сценариях.
- b. Работа с командной строкой требует более глубоких знаний, что может быть препятствием для начинающих пользователей.

Итог анализа

Рекомендации:

Для быстрого создания и управления виртуальными машинами в домашних условиях рекомендуется использовать VirtualBox или Gnome Boxes.

Для профессионального использования в средах с большим количеством виртуальных машин рекомендуется использовать Virt-Manager.

Для автоматизации процессов создания виртуальных машин и управления ими через командную строку рекомендуется использовать QEMU и virt-install (virsh).

Заключение

В ходе выполнения данной лабораторной работы были изучены и применены различные инструменты и технологии виртуализации в Unix-подобных системах. Основное внимание было уделено установке и настройке виртуальных гипервизоров, работе с популярными инструментами управления виртуальными машинами, такими как VirtualBox, Gnome Boxes, Virt-Manager, а также использованию утилиты virt-install для автоматизации процесса создания виртуальных машин. Это позволило оценить их возможности и ограничения, а также выработать рекомендации по их использованию в различных сценариях.

Список литературы

GNOME Boxes Issues [Online] // GitLab Gnome Boxes. - 12 15, 2024. - <https://gitlab.gnome.org/GNOME/gnome-boxes/-/issues>.

No option to connect to remote VM in the Flathub version [Online] // GitLab GNOME Boxes. - GNOME. - 12 15, 2024. - <https://gitlab.gnome.org/GNOME/gnome-boxes/-/issues/735>.

Proxmox VE Administration Guide [Online]. - Proxmox Server Solutions GmbH. - <https://pve.proxmox.com/pve-docs/pve-admin-guide.html>.

Resizing/Extending Logical Volumes (LVM) in Proxmox / auth. i12breto. - [s.l.] : Youtube, oct 2022.

The "No KVM" should be more helpful [Online] // GNOME Boxes Issues. - 12 15, 2024. - <https://gitlab.gnome.org/GNOME/gnome-boxes/-/issues/1094>.

Как установить статический IP-адрес и настроить сеть в Linux [Online] // Блог SEDICOMM. - <https://blog.sedicomm.com/2017/11/20/kak-ustanovit-staticeskij-ip-adres-i-nastroit-set-v-linux/>.