

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: P34082

Выполнила: Савельева Д.А.

Проверил:

к.т.н. преподаватель Белозубов А.В.

Санкт-Петербург

2024г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
1 УСТАНОВКА ВИРТУАЛЬНЫХ ГИПЕРВИЗОРОВ В UNIX.....	4
2 РАБОТА С VMM.....	17
3 ИСПОЛЬЗОВАНИЕ УТИЛИТЫ VIRT-INSTALL.....	39
ЗАКЛЮЧЕНИЕ	45

ВВЕДЕНИЕ

В рамках данной лабораторной работы выполняется изучение основ работы с виртуальными гипервизорами на базе Unix-систем. Основной задачей является освоение операций по созданию, настройке и управлению виртуальными машинами с использованием таких инструментов, как Gnome Boxes, QEMU/KVM, Virtual Machine Manager (VMM) и утилита virt-install.

Работа проводится с использованием гостевых операционных систем, для которых настраиваются сетевые подключения, общие папки и дополнительные виртуальные диски. Также исследуются свойства виртуальных машин в процессе их работы, параметры их конфигурации и возможности восстановления системы с использованием снимков состояния. Для взаимодействия с удаленными машинами применяются протоколы SSH и RDP, что позволяет наладить работу между основными и гостевыми системами.

Цель работы — изучение принципов работы гипервизоров, методов управления виртуальными машинами через графический интерфейс и командную строку, а также исследование возможностей виртуальных сред для выполнения различных задач.

Задачи лабораторной работы включают:

- Установку и настройку гипервизоров и виртуальных машин.
- Настройку сетевых взаимодействий между системами.
- Использование снимков состояния для восстановления систем.
- Управление виртуальными машинами через командную строку.

Работа с виртуализацией позволяет лучше понять процессы, связанные с созданием и управлением изолированными вычислительными средами, что важно для дальнейшего применения в области системного администрирования и разработки программного обеспечения.

1 УСТАНОВКА ВИРТУАЛЬНЫХ ГИПЕРВИЗОРОВ В UNIX

Вся работа производится в гостевой ОС в Unix-системе.

Ниже представлено задание, которое необходимо выполнить в рамках данного раздела:

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

Далее будем работать с Gnome Boxes. Последовательно выполняем задачи.

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 к удаленному компьютеру.

11. Проверьте подключение (если нет возможности подключиться к соседу, настройте подключение к своей Хостовой машине).

Далее поэтапно будем выполнять каждое из заданий.

1. Добавить Жесткий диск (SATA) объемом 30 Gb в гостевую ОС Ubuntu.

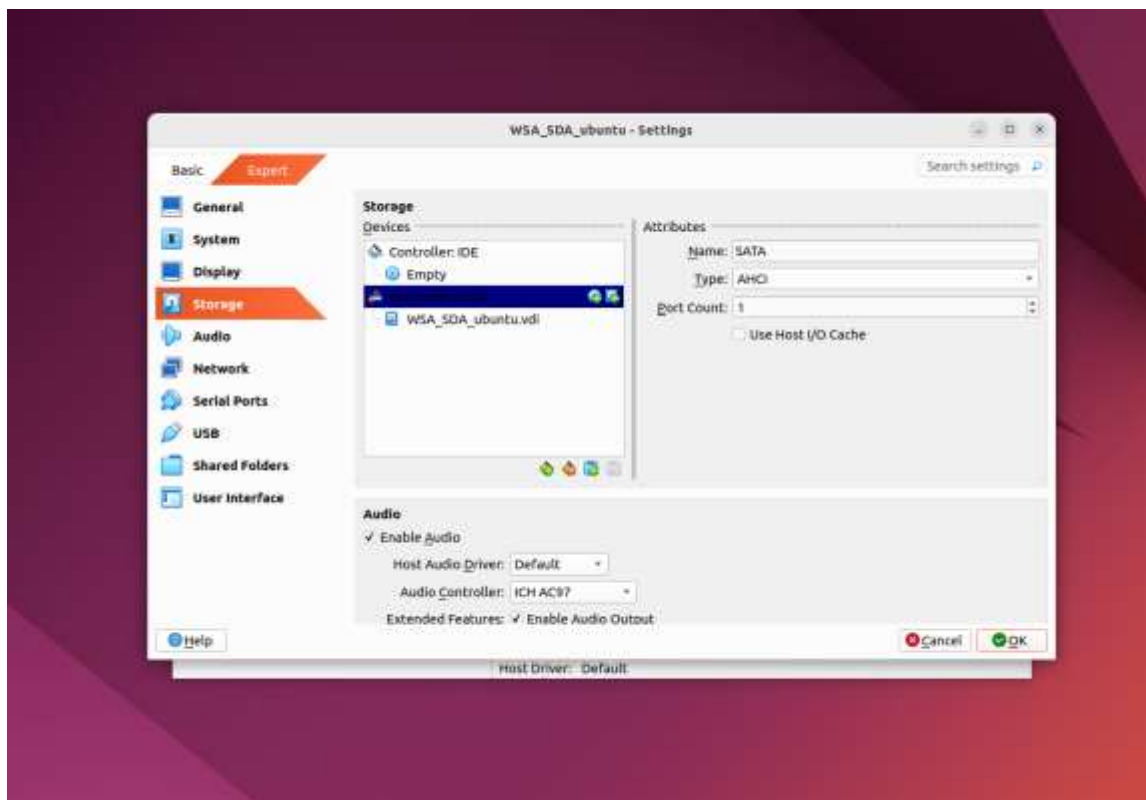


Рисунок 1 – Добавление жесткого диска в гостевую ОС

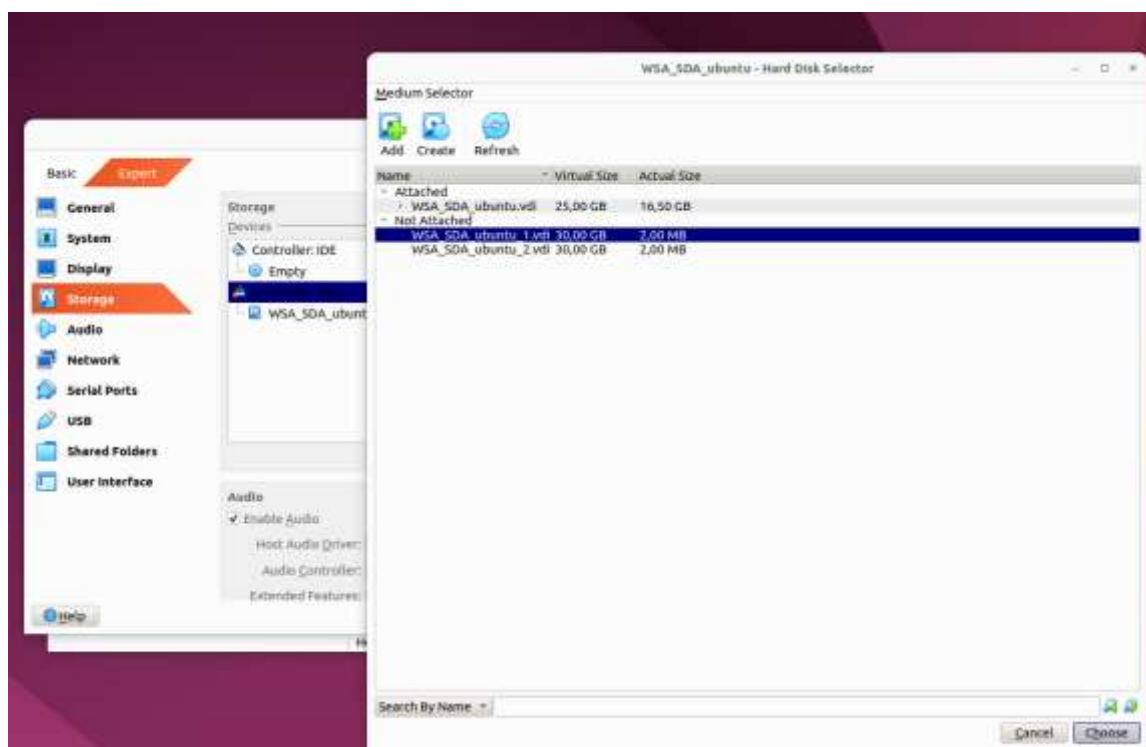


Рисунок 2 – Добавление жесткого диска в гостевую ОС

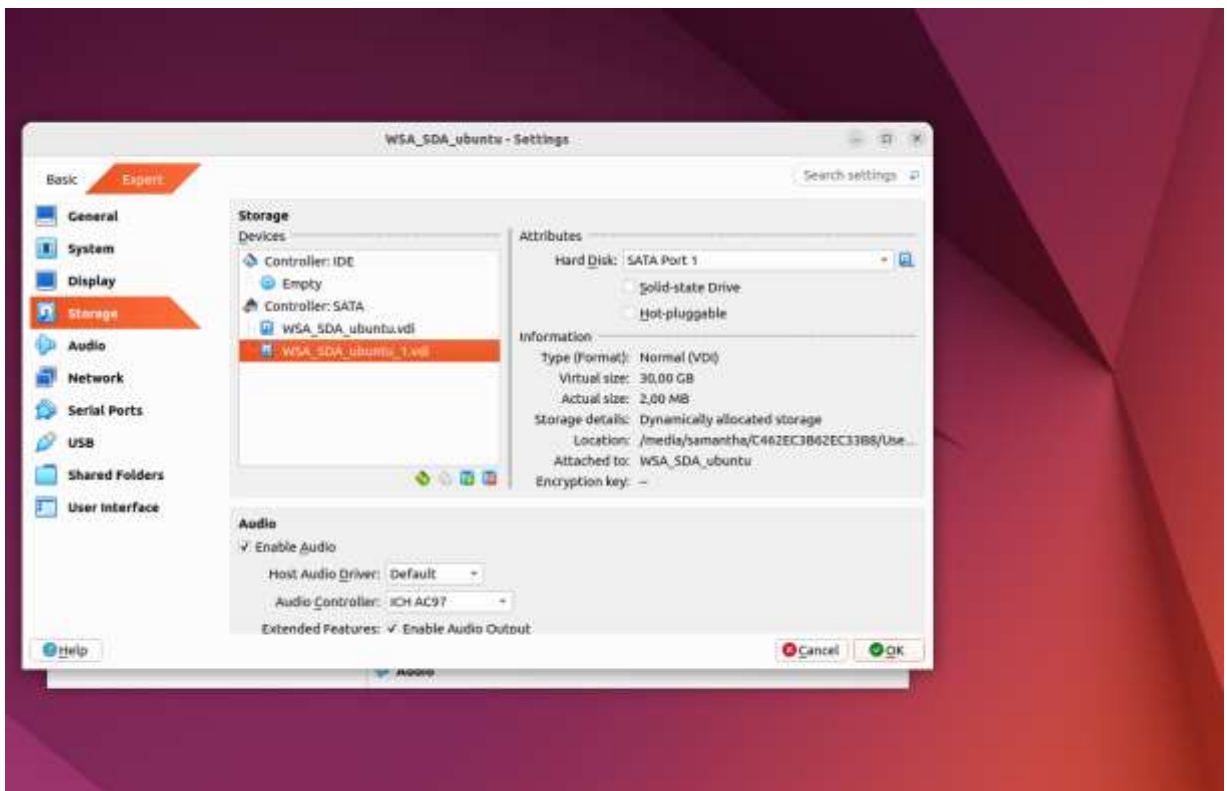


Рисунок 3 – Добавление жесткого диска в гостевую ОС

2. Проверить конфигурацию гостевой ОС.

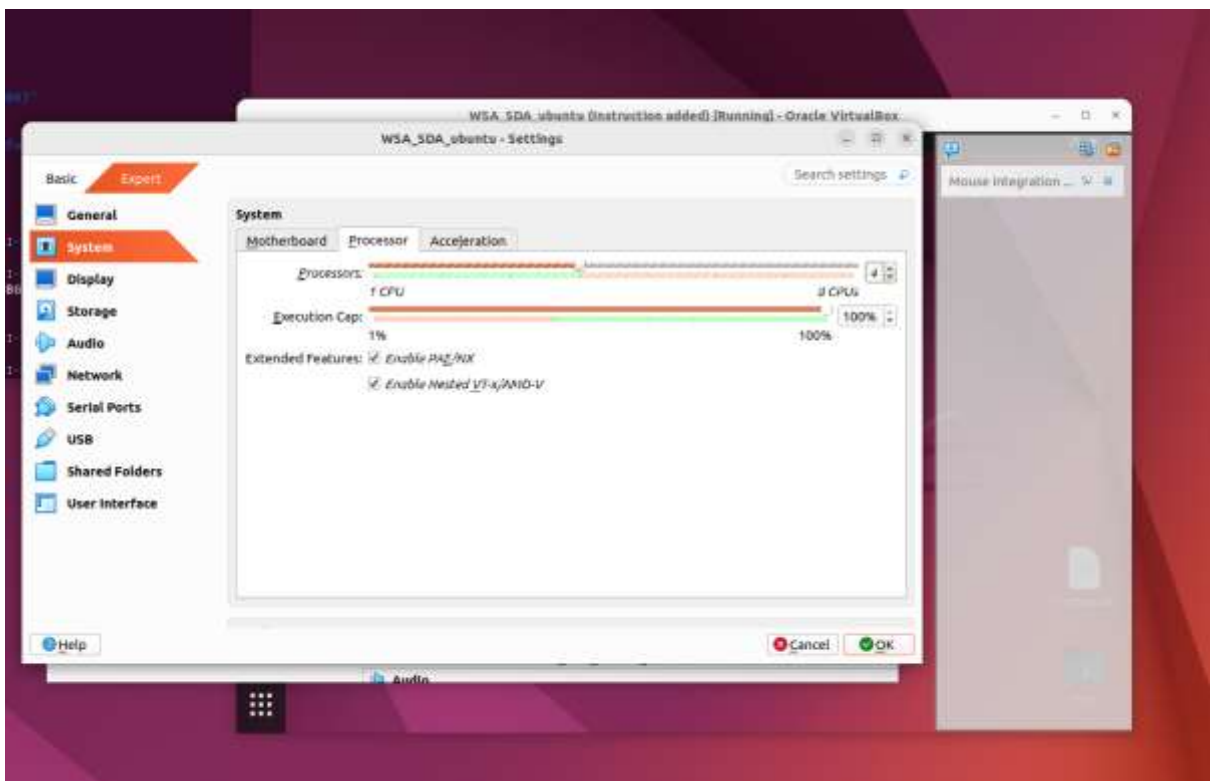


Рисунок 4 – Конфигурация гостевой ОС Ubuntu

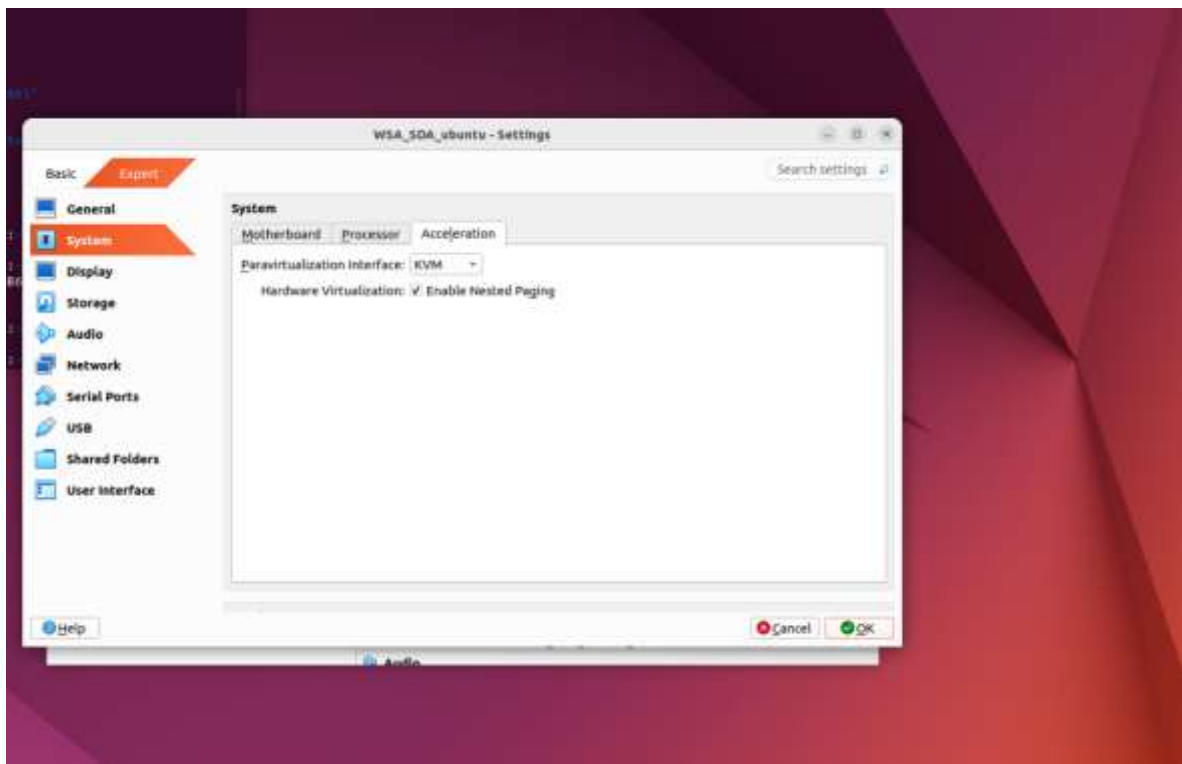


Рисунок 5 – Конфигурация гостевой ОС Ubuntu

3. Создать общую папку, общую с основной и гостевой ОС, под названием Soft.

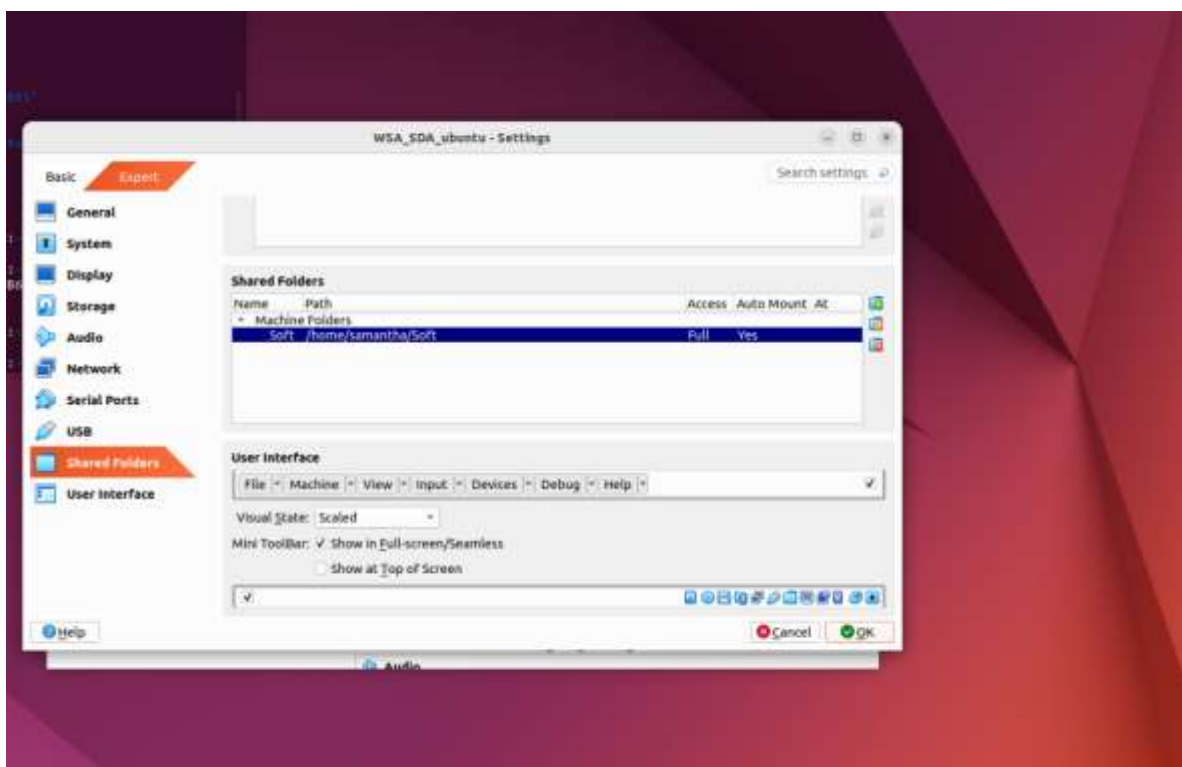


Рисунок 6 – Добавление общей папки

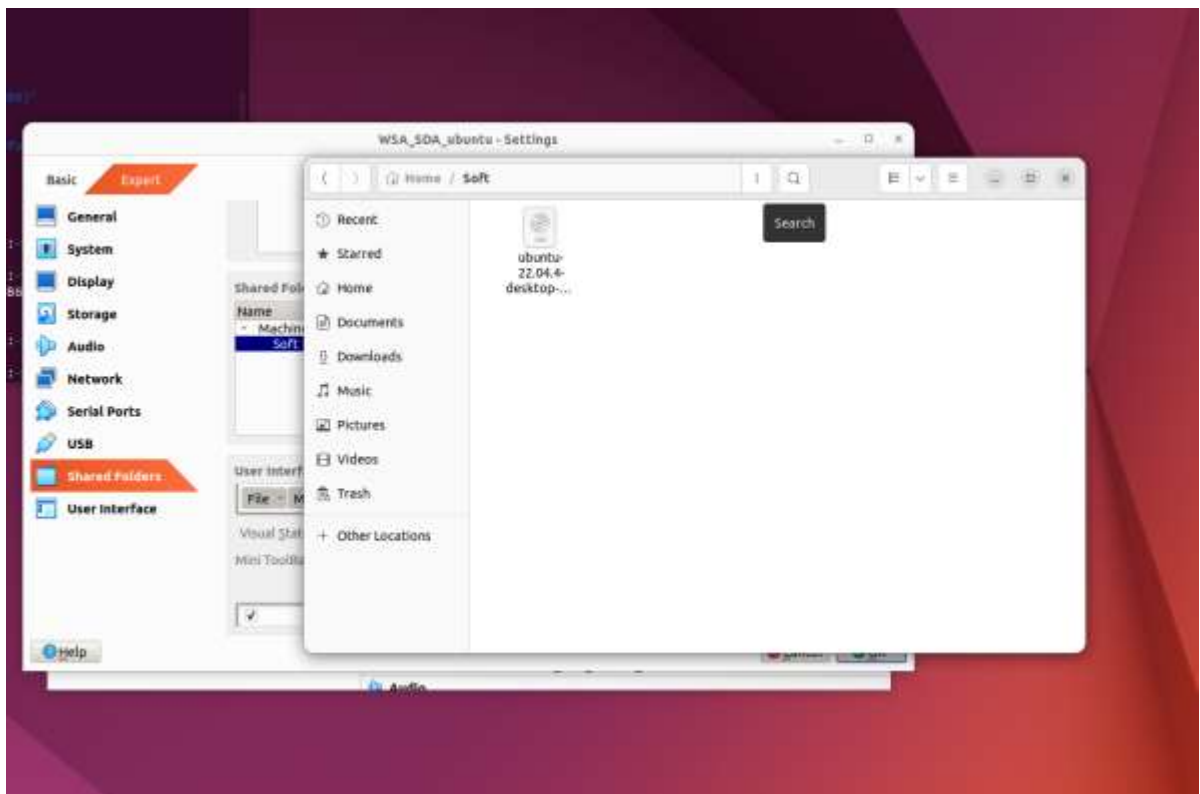


Рисунок 7 – Общая папка

4. Установить в гостевой ОС приложение Gnome Boxes.

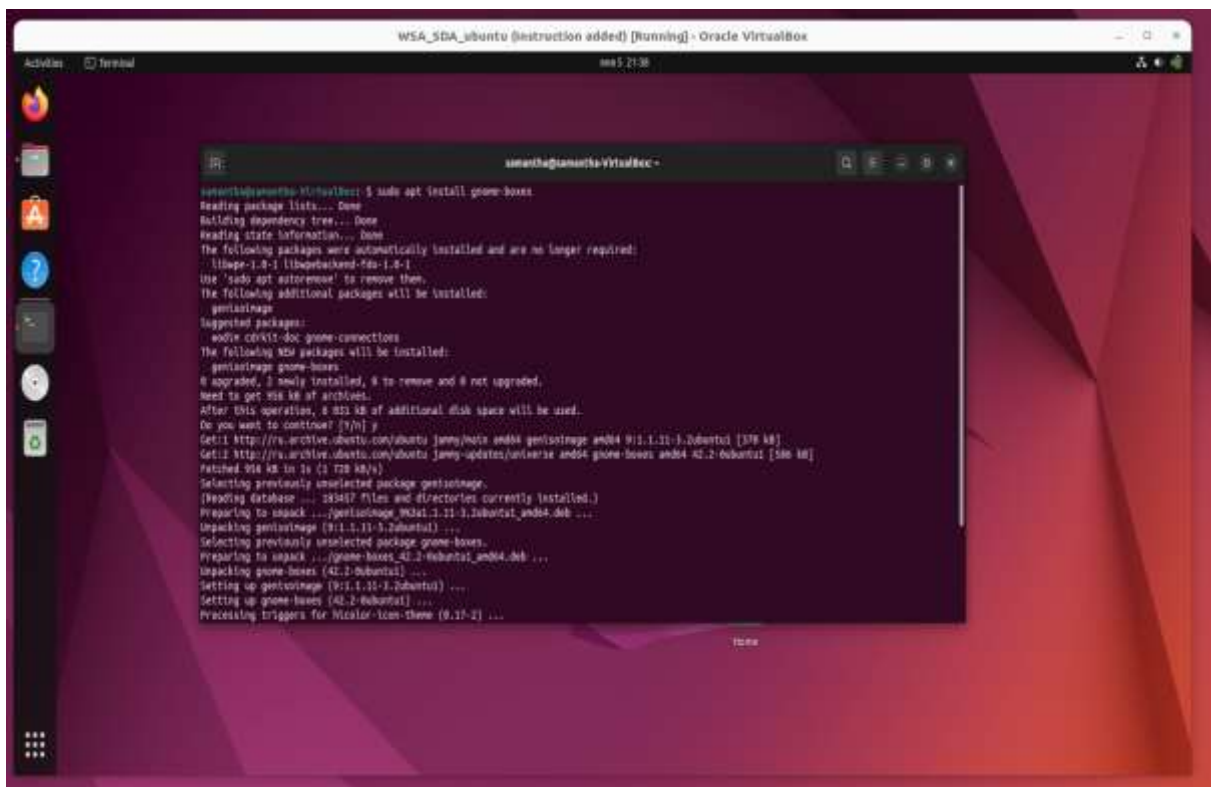


Рисунок 8 – Установка Gnome Boxes

5. Запустите Boxes и создайте виртуальную машину.

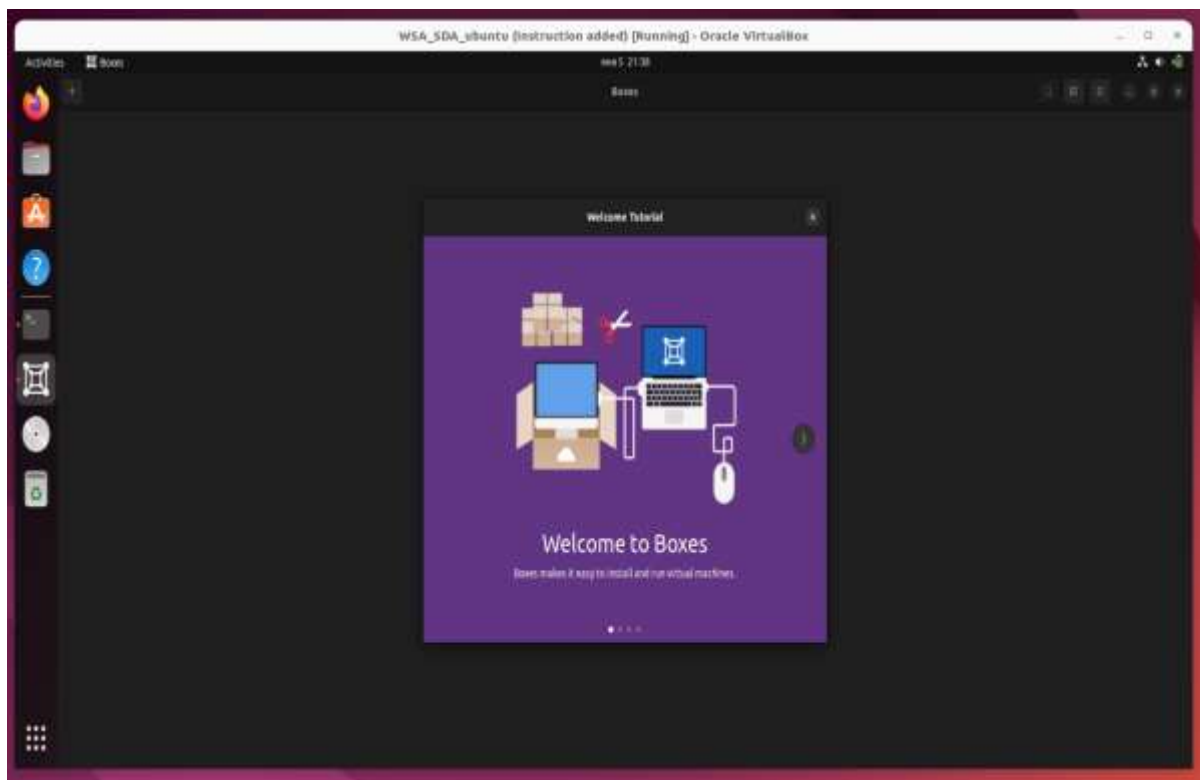


Рисунок 9 – Запуск Gnome Boxes

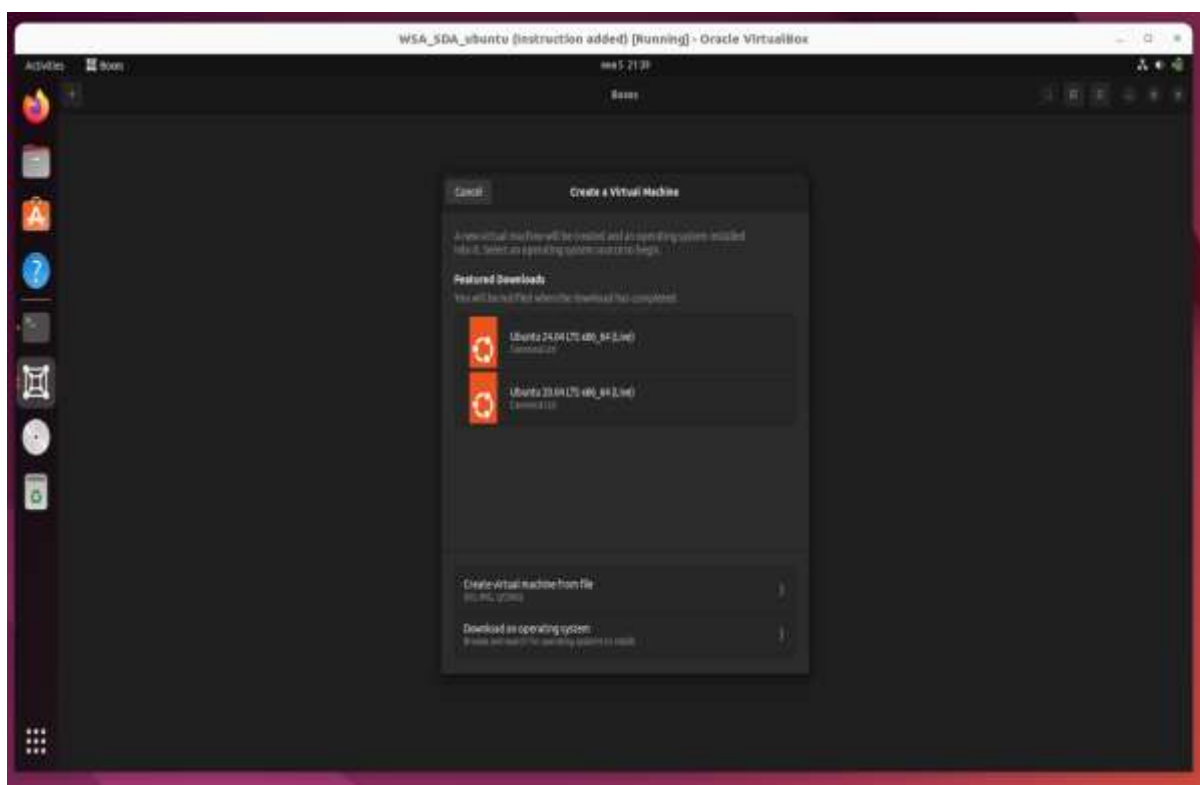


Рисунок 10 – Создание виртуальной машины Fedora

6. Запустите ОС в режиме LiveCD.



Рисунок 11 – Демонстрация отказа в работе вложенной ВМ

При запуске возникает критическая ошибка гостевой ОС Ubuntu, которую не получается исправить.

Поэтому дальнейшие действия будут осуществляться на хост-системе.

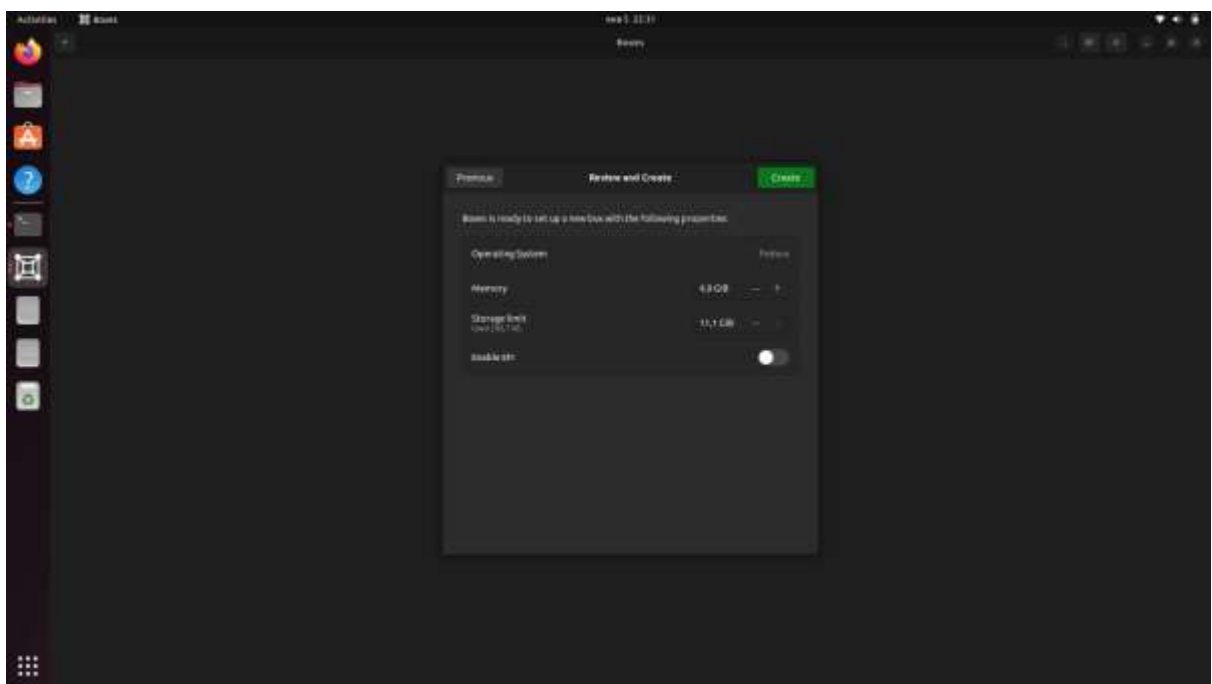


Рисунок 12 – Создание новой виртуальной машины

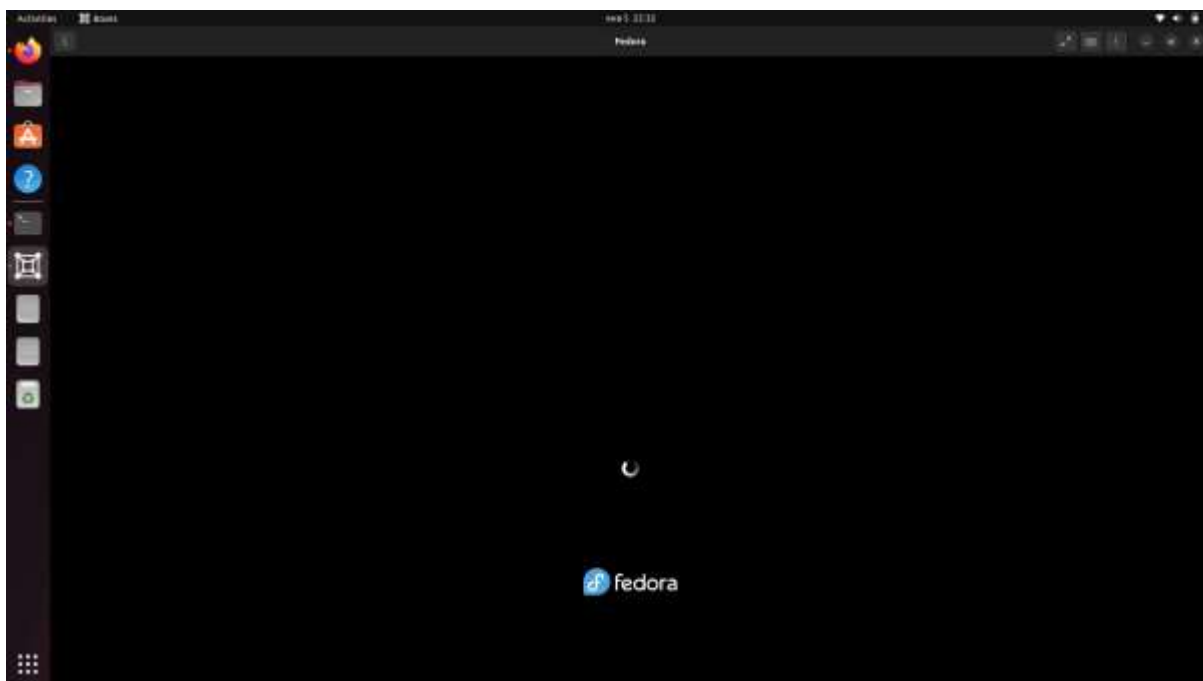


Рисунок 13 – Запуск машины в режиме Fedora Live

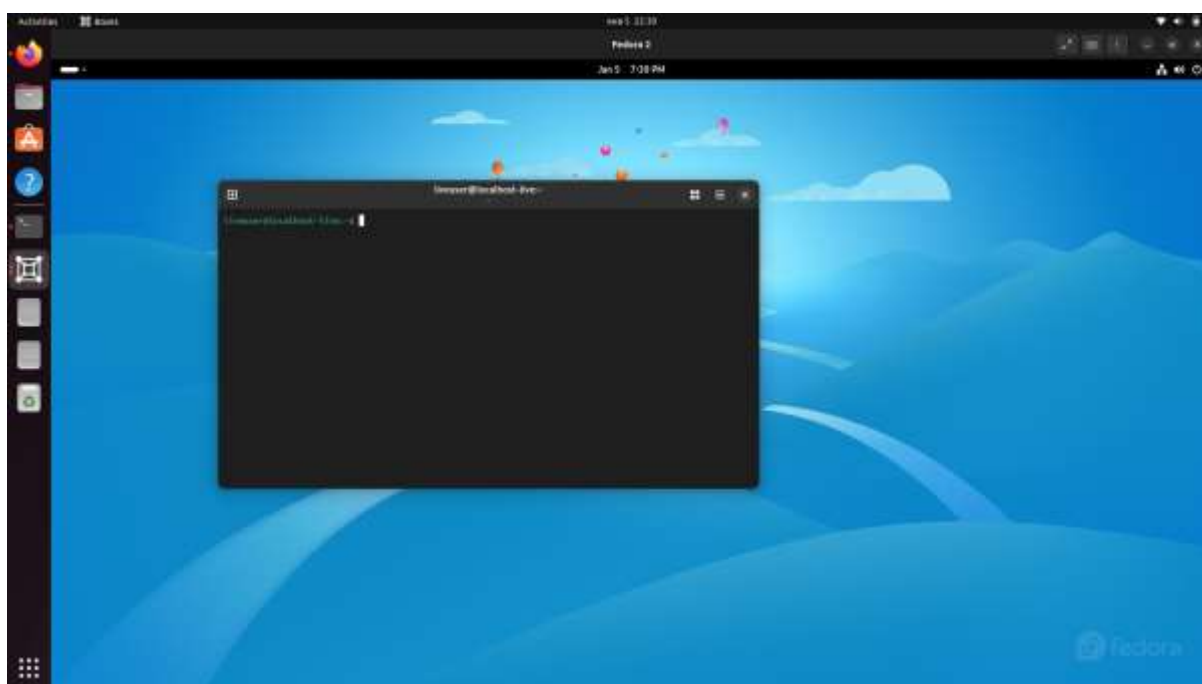


Рисунок 14 – Демонстрация работы Fedora Live

7. Изучите свойства виртуальной машины созданной в Vboxes, какие изменения в работе виртуальной машины можно производить? Какие параметры можно отслеживать в оперативном режиме?

The screenshot displays a Kali Linux desktop with a blue background. On the left is a vertical dock containing icons for the Dash application, Firefox, a file manager, a terminal, and other utilities. At the top, a panel shows the date and time as 'Wed 5 21:44' and the hostname 'Pedro's PC'. Below this, a window titled 'Terminal' is open, showing the output of an SSH session. The terminal text includes the SSH banner, a warning about the authenticity of the host, a confirmation of the host's fingerprint, and a successful login for the user 'root'.

```

root@kali:~# ssh root@192.168.1.101 -p 2222
Warning: Permanently added '192.168.1.101' (ssh-rsa) to the list of known hosts.
root@192.168.1.101:~#

```

9. Подключитесь у удаленному компьютеру.



14

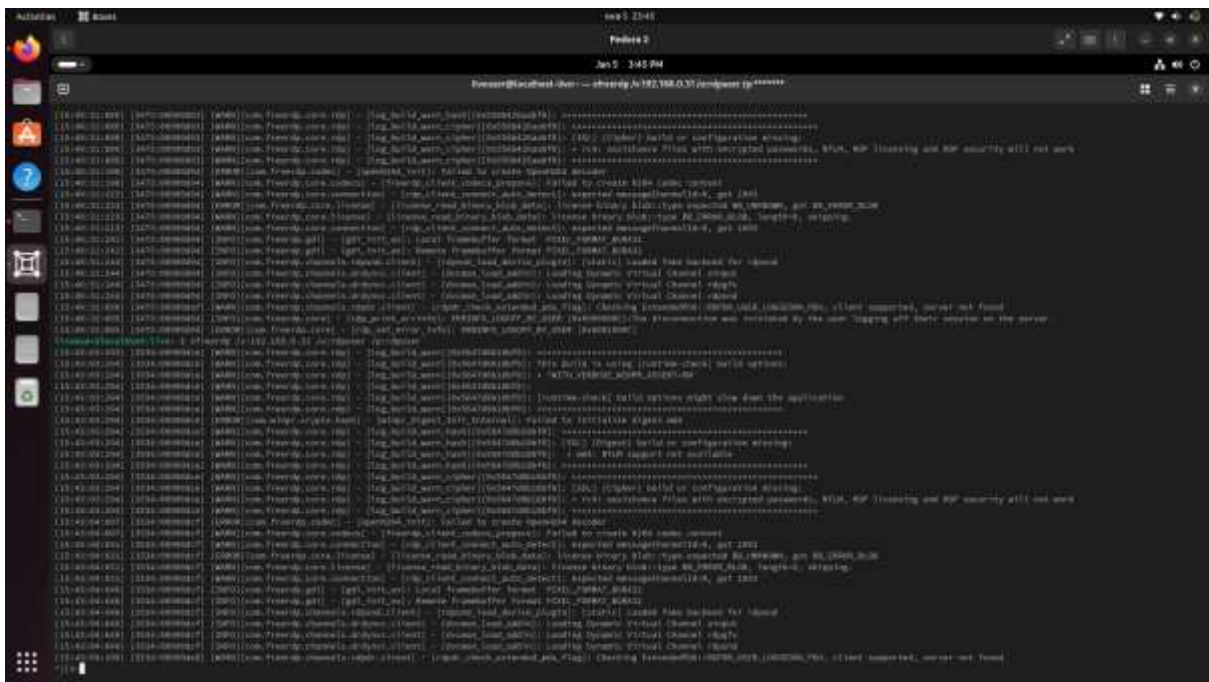


Рисунок 21 – Подключение по RDP к хосту

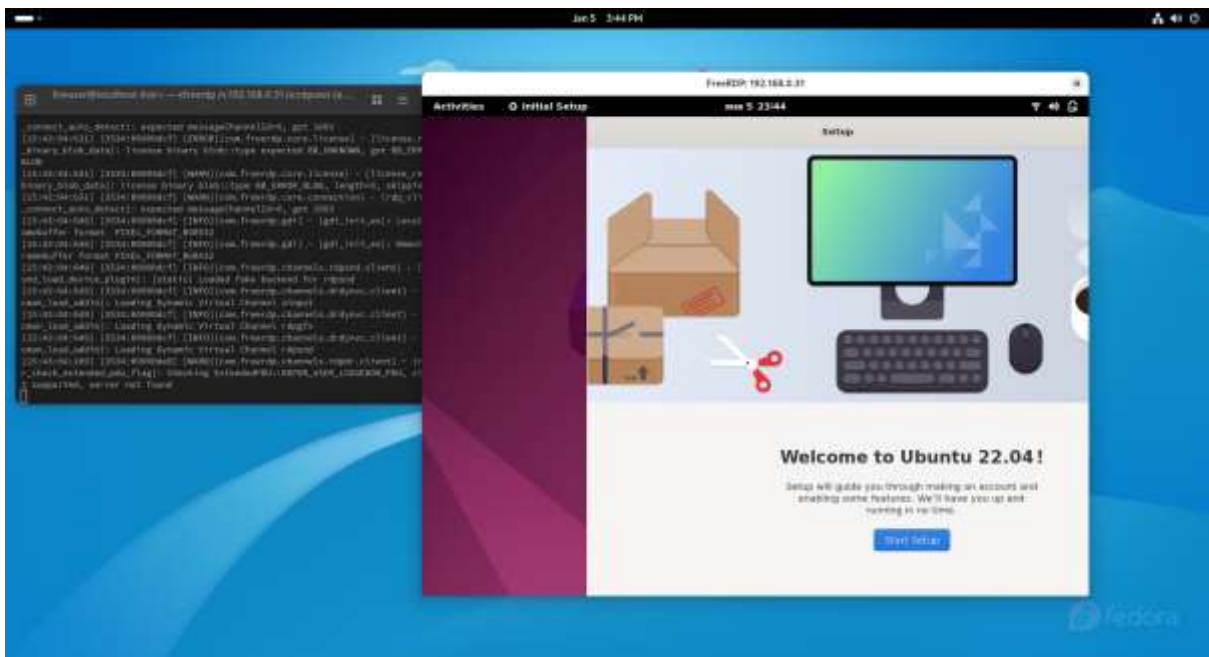


Рисунок 22 – Работоспособность подключения по RDP к хосту

2 РАБОТА С VMM

Ниже представлено задание, которое необходимо выполнить в рамках данного раздела:

1. Установить QEMU.
2. Проверить, поддерживает ваша система KVM
3. Создайте виртуальную машину
4. Выполните следующие команды

```
$mkdir VM
```

```
$qemu-img create -f qcow2 ./VM/Fedora-36.img 10G
```

5. Запустите виртуальную машину

```
$qemu-system-x86_64 ./VM/Fedora-36.img
```

6. Запустите виртуальную машину с установкой операционной системы. (Уточните полный путь до дистрибутива. Можно переименовать дистрибутив с более коротким именем fedora.iso)

```
$qemu-system-x86_64 -m 2048 -enable-kvm Fedora-36.img -cdrom fedora.iso
```

7. После установки системы создайте команду запуска гостевой ОС с помощью команды qemu с следующими параметрами (воспользуйтесь командой qemu-system-x86_64):

- Загрузка в полноэкранном режиме
- Установка mac-адреса 17:10:20:22:20:09

8. Установка Virtual Man.
9. Установите пакеты приложений для нормальной работы Virtual-Manger.

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

10. Добавить своего пользователя в группу libvirt.

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

11. Проверка что libvirt запущен и работает

\$sudo systemctl status libvirtd

12. Перезапустите гостевую систему

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

\$kvm-ok

14. После перезапуска найдите в приложениях программу Virtual Machine Manager

15. Создайте виртуальную машину с именем FIO-ваша_группа.

16. Проверьте настройки системы.

17. Добавьте еще один жесткий диск объемом 10МиБ.

18. Изучите свойства виртуальной машины созданной в VMM, какие настройки виртуальной машины можно выставлять? Какие параметры можно отслеживать в оперативном режиме?

19. Создать снимок состояния системы

20. Установить приложение Yandex Browser

21. Восстановить состояние системы в первоначальное состояние.

22. Проверьте работу снимков состояния

23. Какие сетевые настройки можно устанавливать гостевой машине?

24. Управление виртуальной машиной из командной строки virsh

25. 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

Далее поэтапно будем выполнять каждое из заданий.

1. Установить QEMU.



Рисунок 23 – Установка QEMU

2. Проверить, поддерживает ли ваша система KVM.

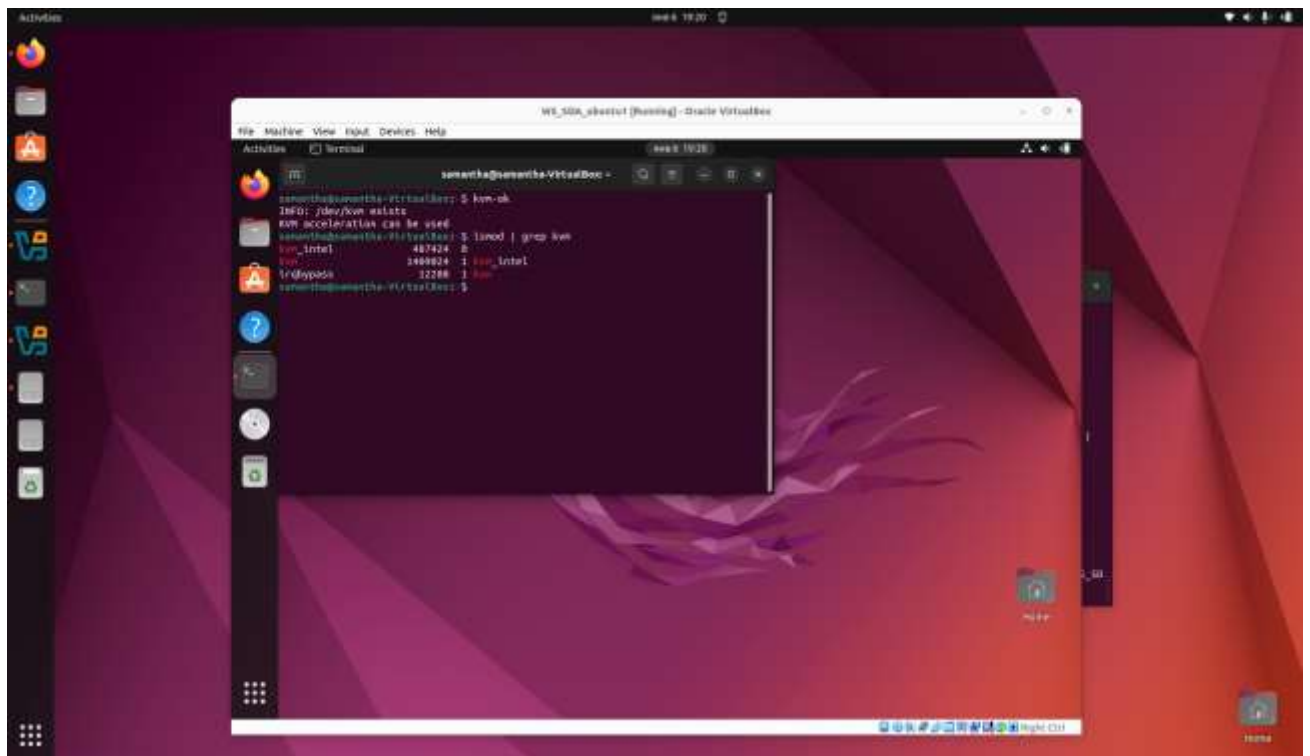


Рисунок 24 – Проверка поддержки KVM

3. Создайте виртуальную машину. Выполните следующие команды:

\$mkdir VM

\$qemu-img create -f qcow2 ./VM/Fedora-36.img 10G

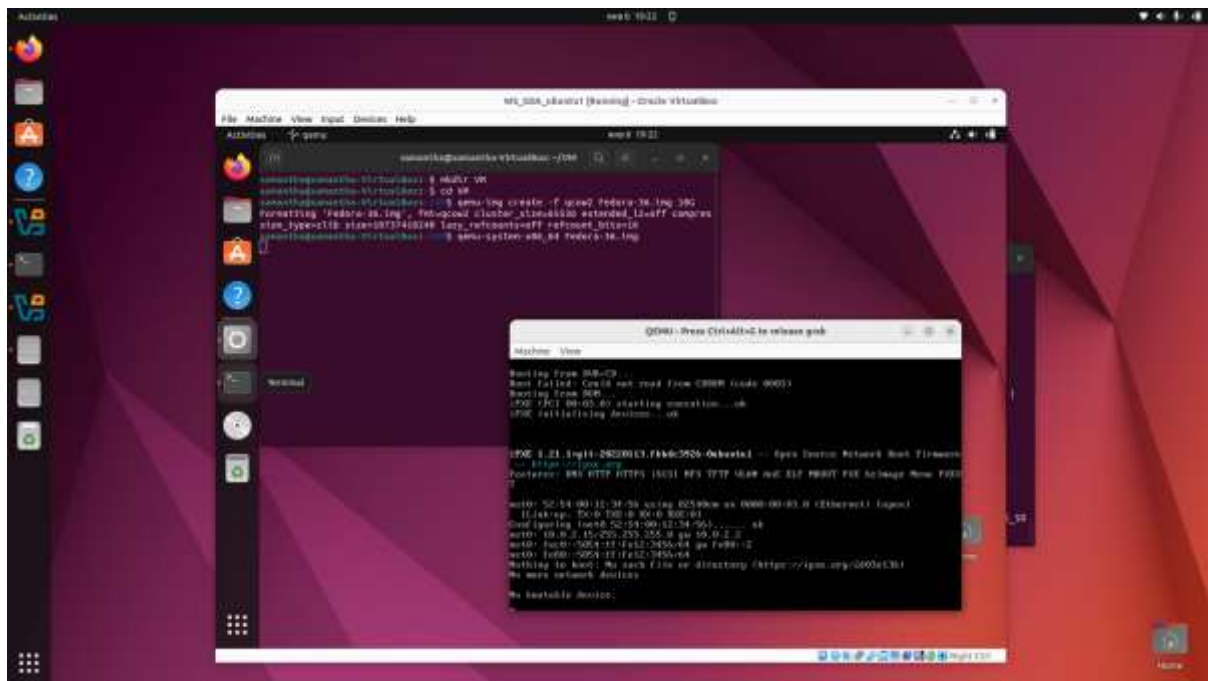


Рисунок 25 – Создание виртуальной машины

4. Запустите виртуальную машину.

\$qemu-system-x86_64 ./VM/Fedora-36.img

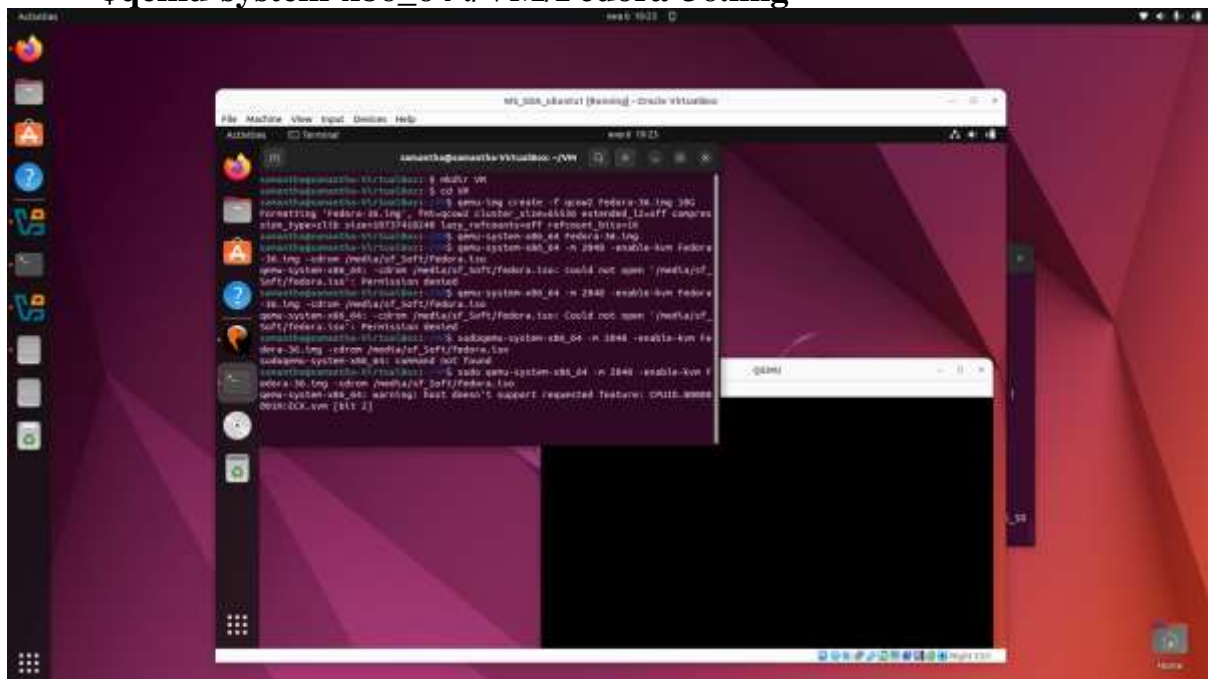


Рисунок 26 – Запуск виртуальной машины

5. Запустите виртуальную машину с установкой операционной системы.

**\$qemu-system-x86_64 -m 2048 -enable-kvm Fedora-36.img -
cdrom fedora.iso**



Рисунок 27 – Запуск виртуальной машины Fedora

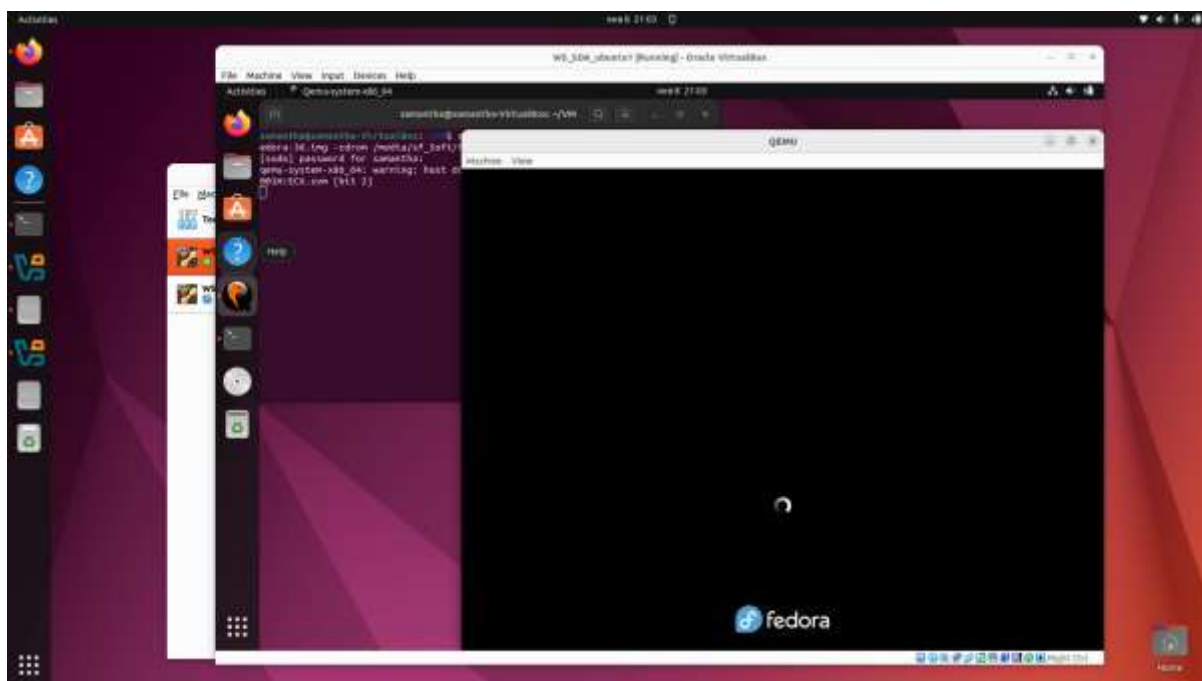


Рисунок 28 – Запуск виртуальной машины Fedora

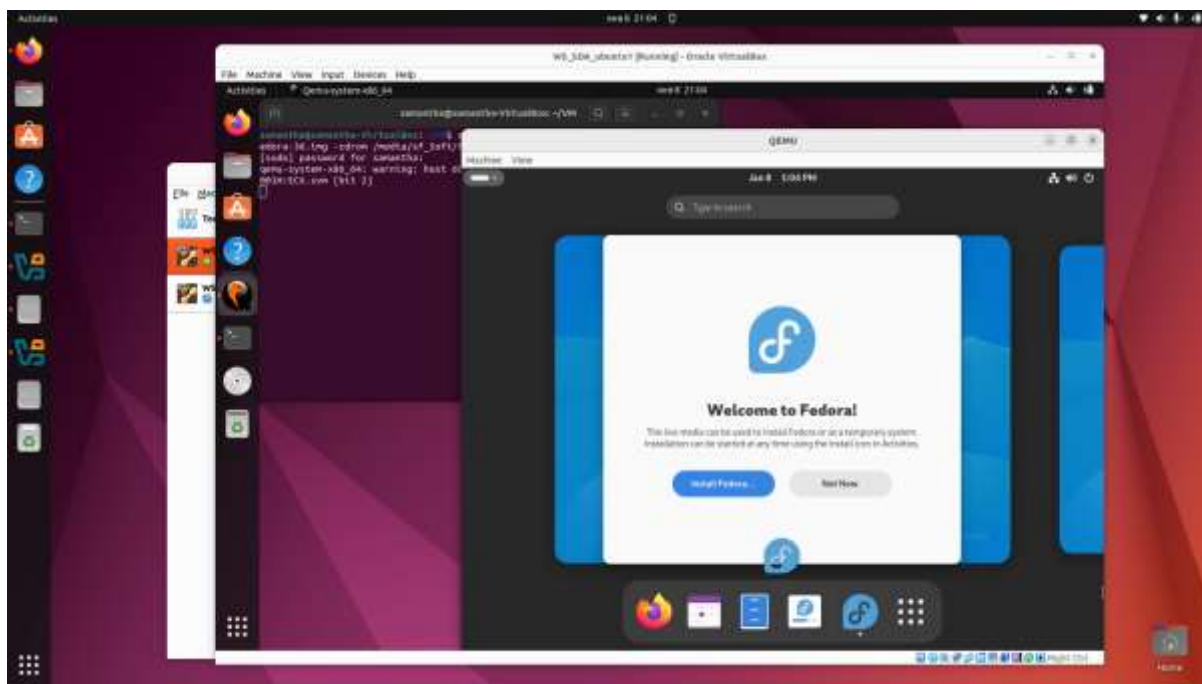


Рисунок 29 – Установка Fedora

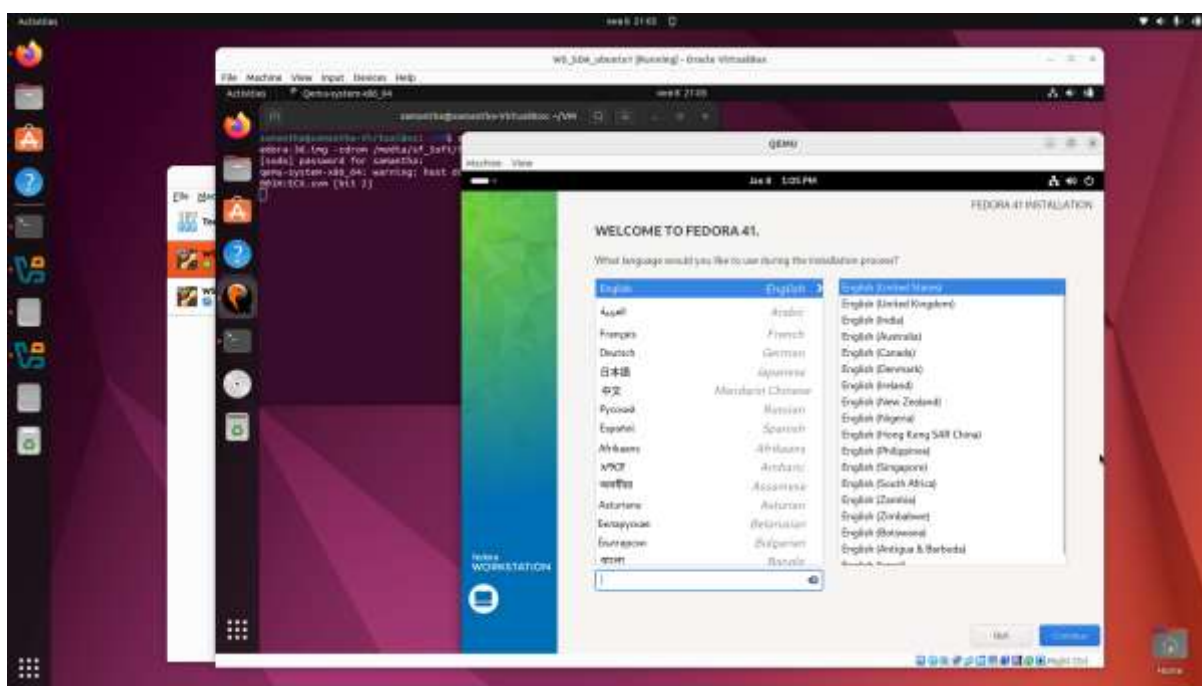


Рисунок 30 – Установка Fedora

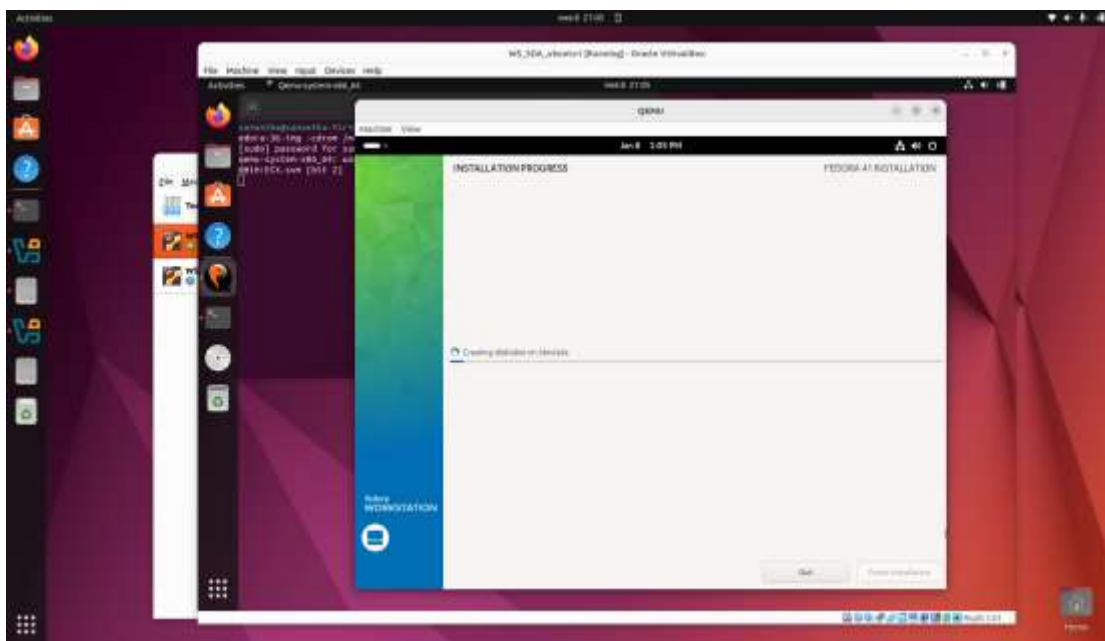


Рисунок 31 – Установка Fedora

6. После установки системы создайте команду запуска гостевой ОС с помощью команды `qemu` с следующими параметрами (воспользуйтесь командой `qemu-system-x86_64`):

6.1. Загрузка в полноэкранном режиме

6.2. Установка mac-адреса 17:10:20:22:20:09

```
sudo      qemu-system-x86_64      -full-screen      -m      2048      -net  
nic,macaddr=18:10:20:22:20:09      -netdev      user,id=net0      -enable-kvm  
./VM/Fedora-36.img
```

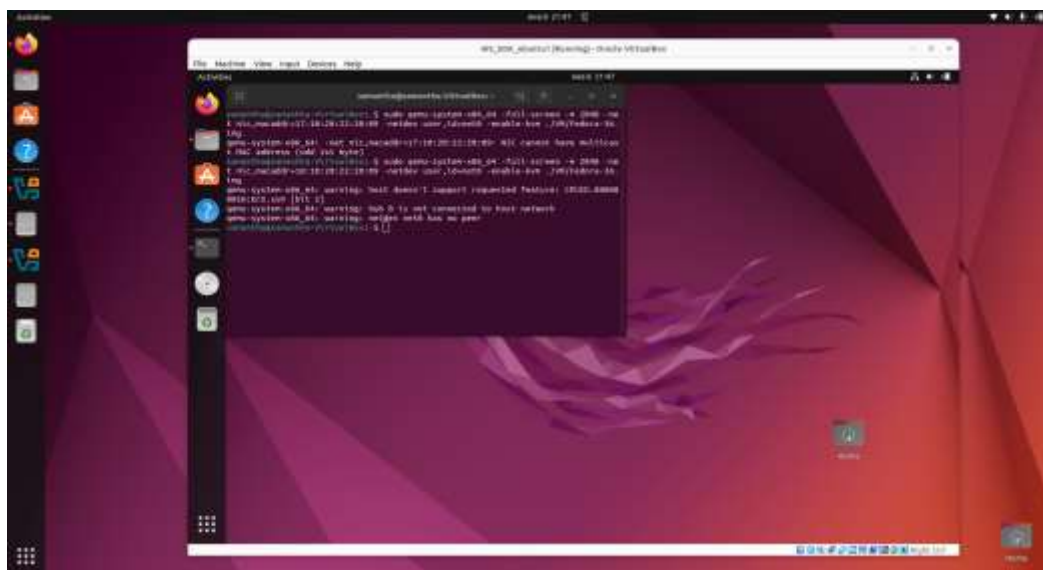


Рисунок 32 – Команда для запуска гостевой ОС

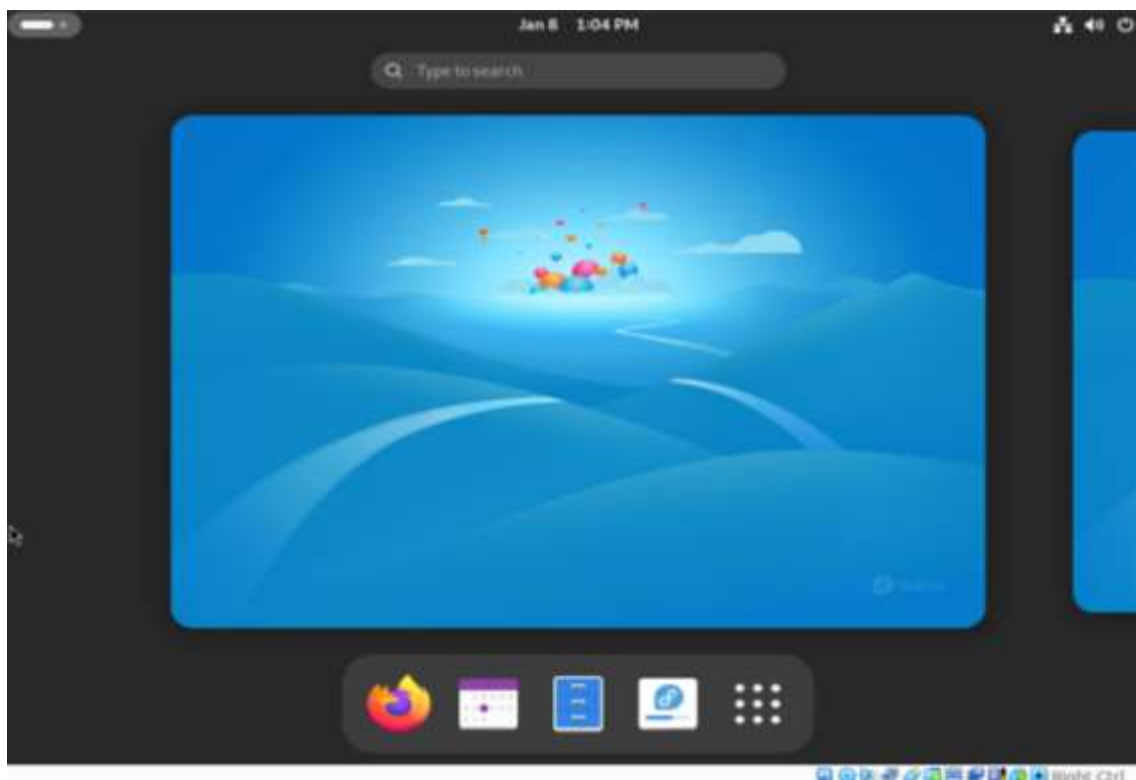


Рисунок 33 – Запуск Fedora после команды

7. Установка Virtual Man. Установите пакеты приложений для нормальной работы Virtual-Manger. Добавить своего пользователя в группу libvirt. Проверка что libvirt запущен и работает.

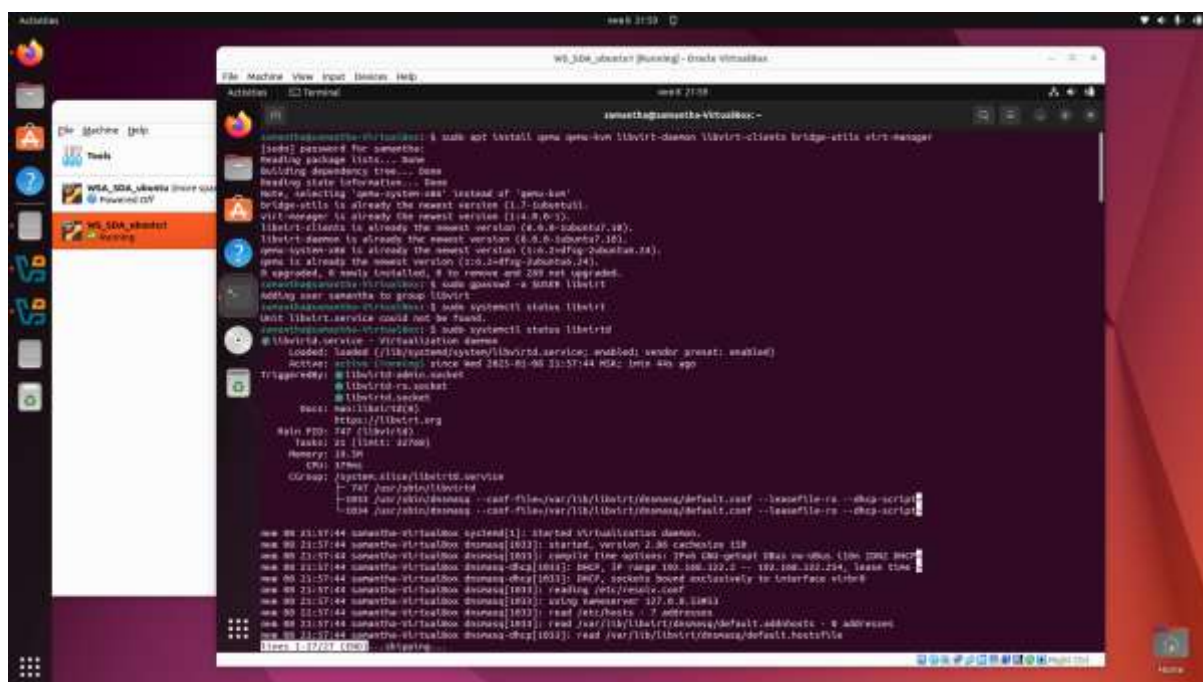


Рисунок 34 – Работа с Virtual Man

8. Проверка правильности установки системы.



Рисунок 35 – Проверка работоспособности KVM

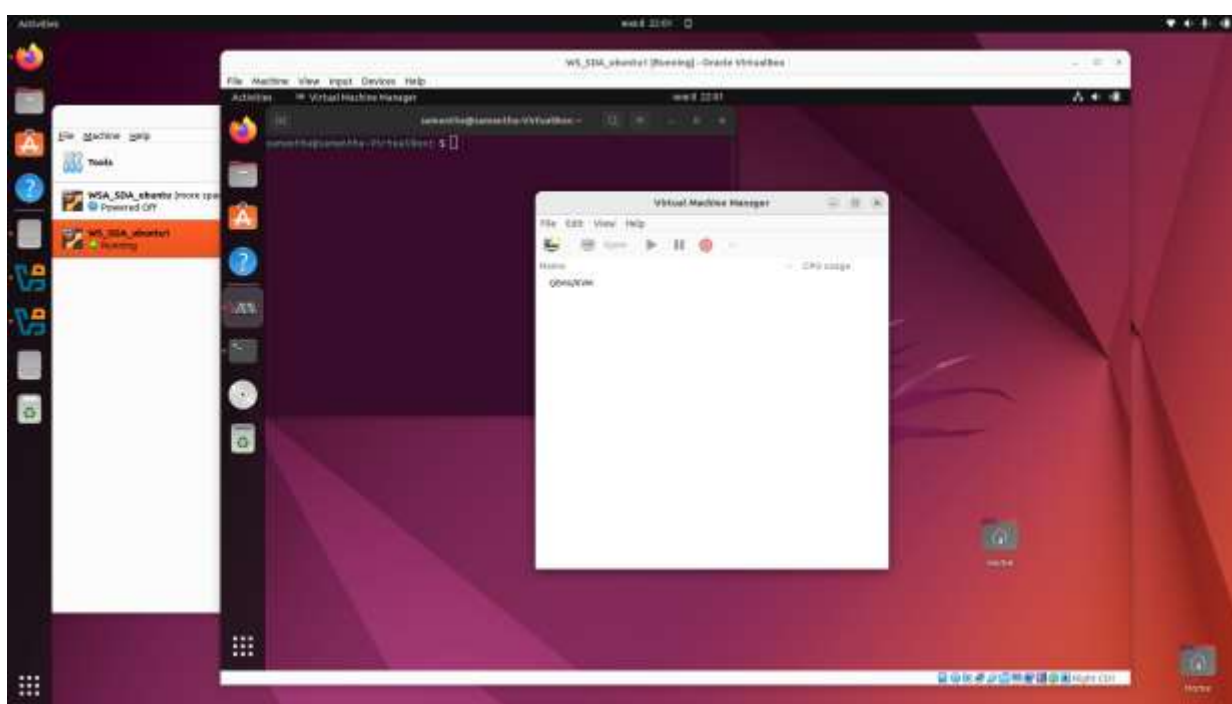


Рисунок 36 – Программа Virtual Machine Manager

9. После перезапуска найдите в приложениях программу Virtual Machine Manager. Создайте виртуальную машину с именем FIO-ваша_группа.

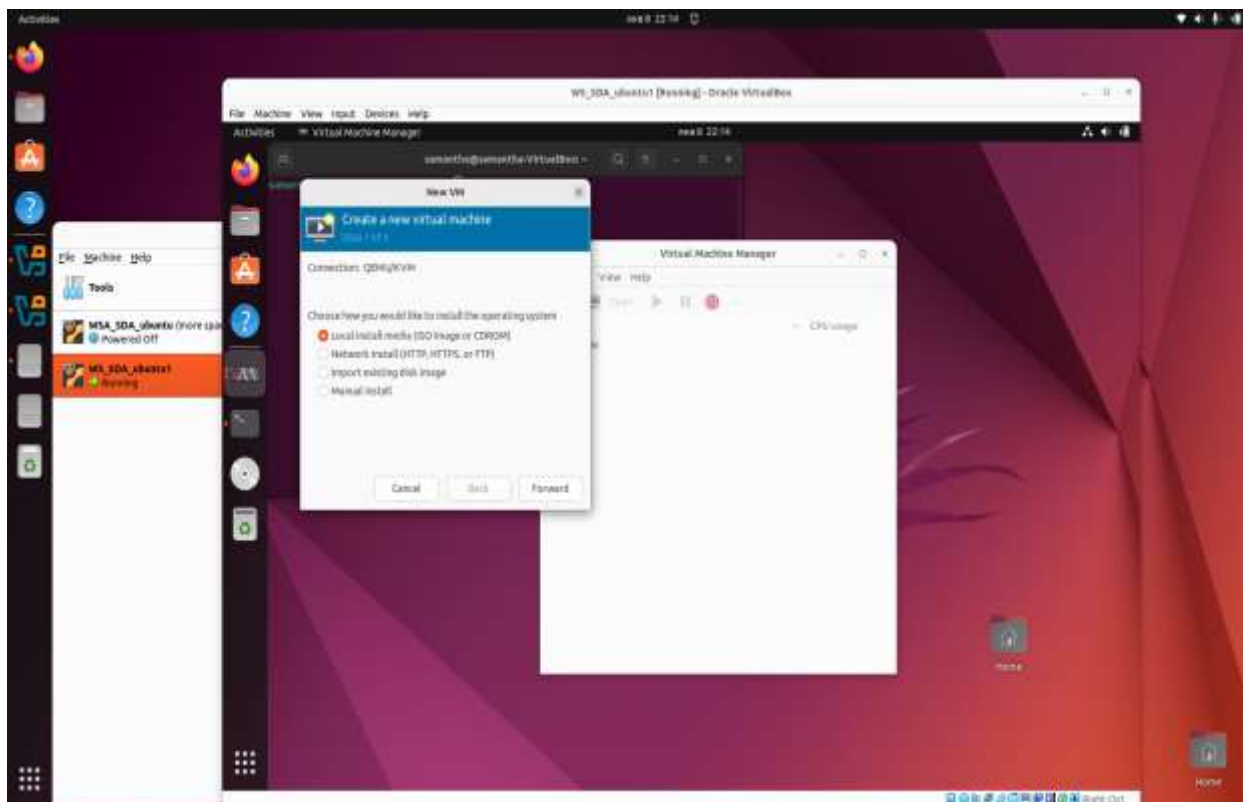


Рисунок 37 – Создание виртуальной машины

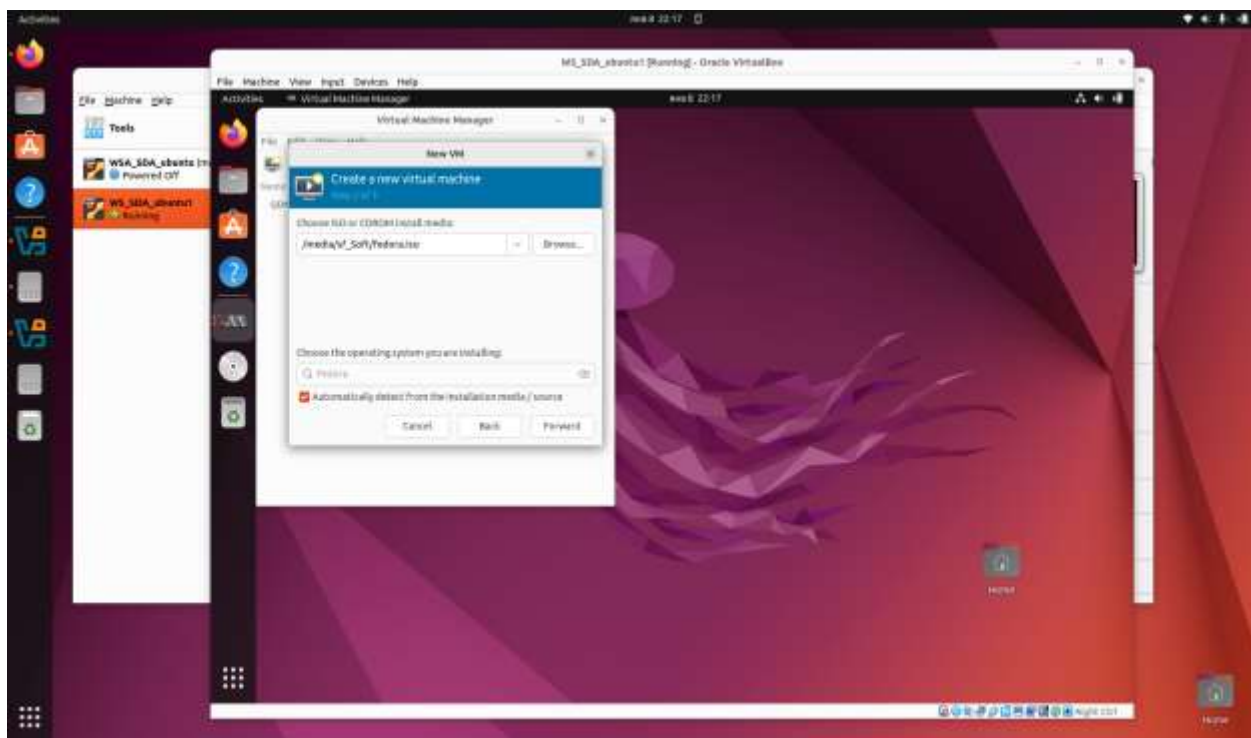


Рисунок 38 – Создание виртуальной машины

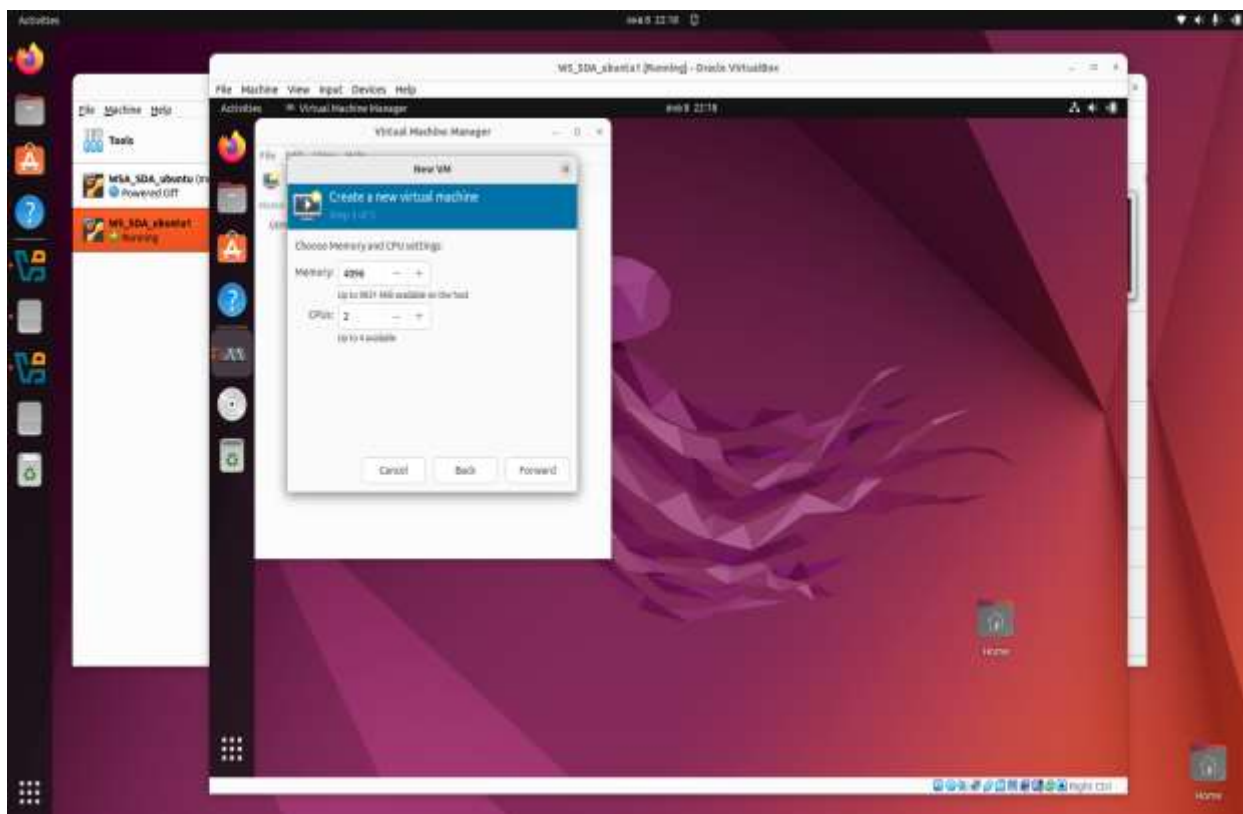


Рисунок 39 – Создание виртуальной машины

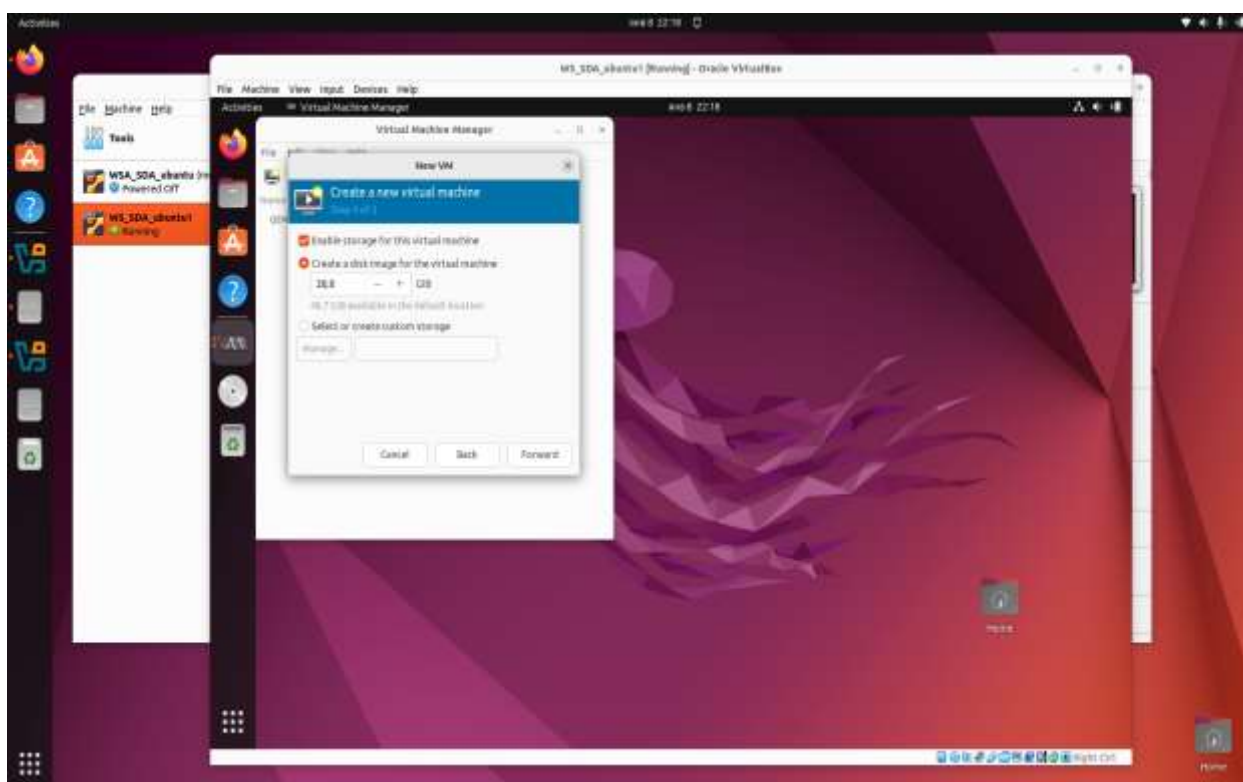


Рисунок 40 – Создание виртуальной машины

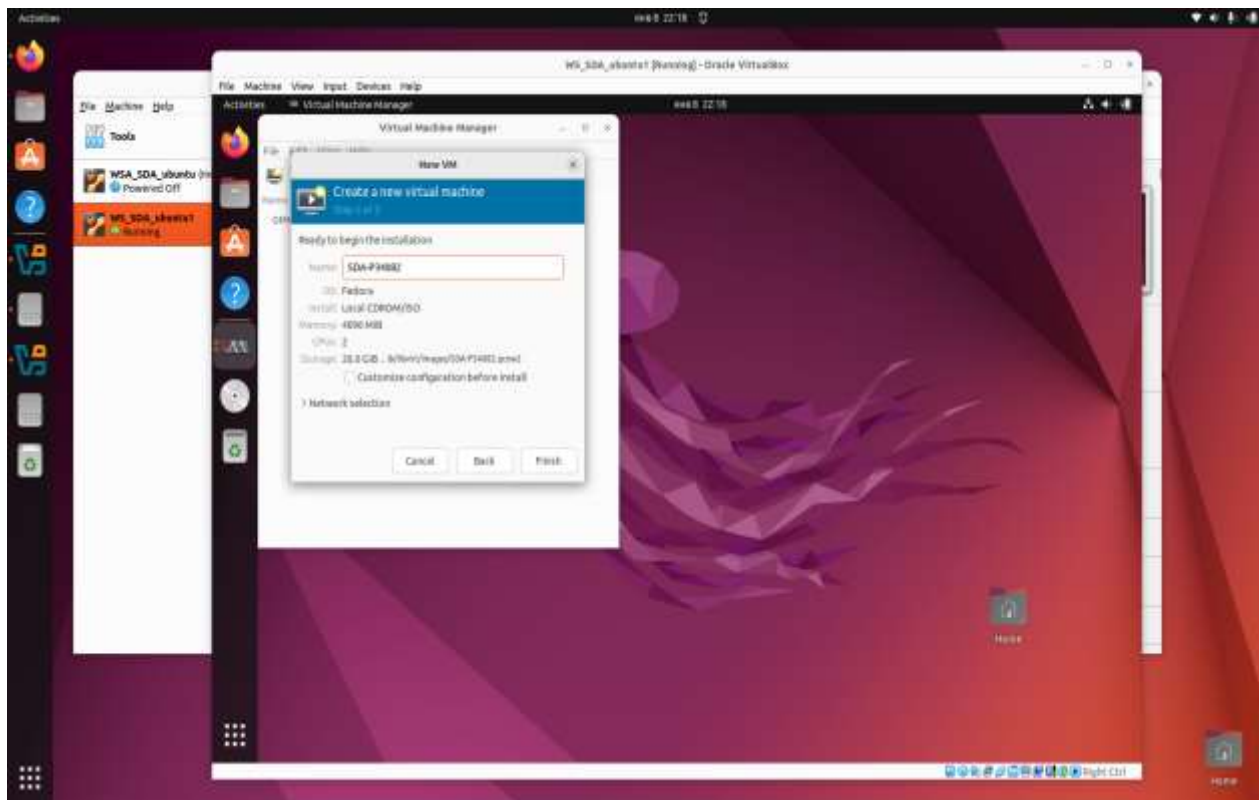


Рисунок 41 – Создание виртуальной машины

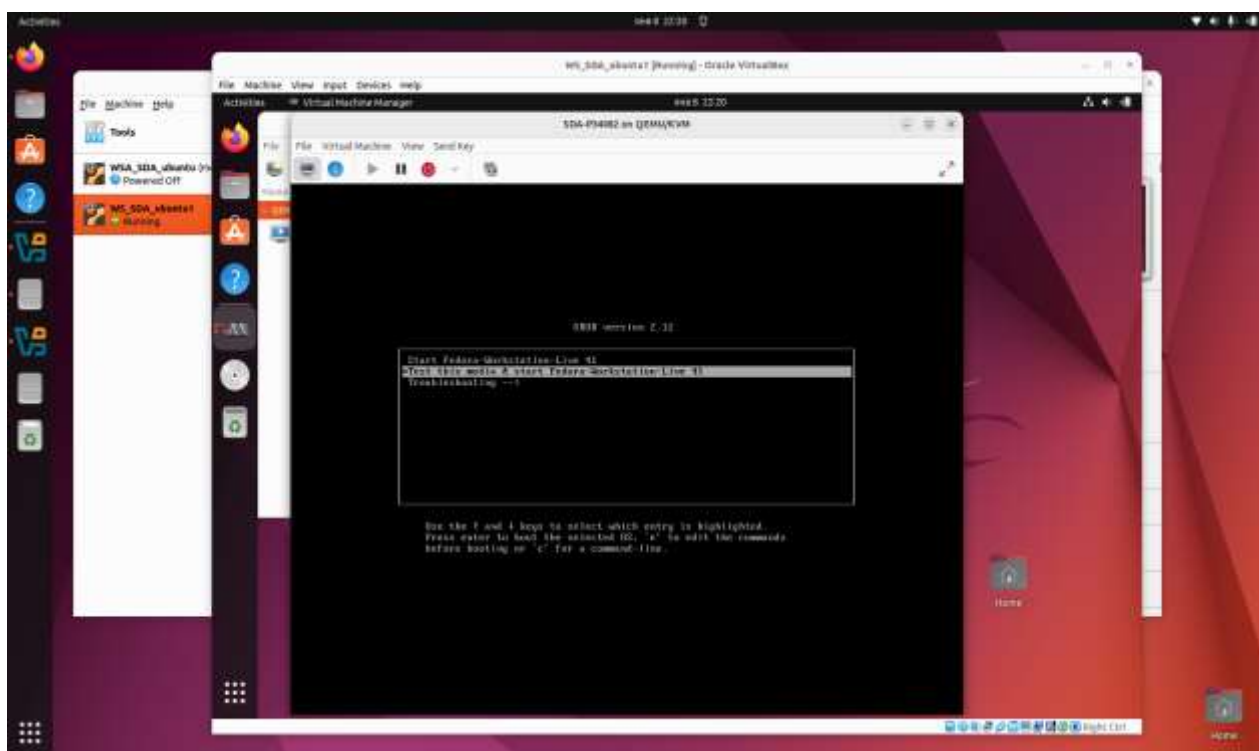


Рисунок 42 – Запуск виртуальной машины

При попытке запуска внутри гостевой системы происходит подобная Gnome Boxes ошибка(что логично, поскольку GnomeBoxes, так же как и VMM, это интерфейс для libvirt).

Поэтому дальнейшие действия будут выполняться на хост системе.

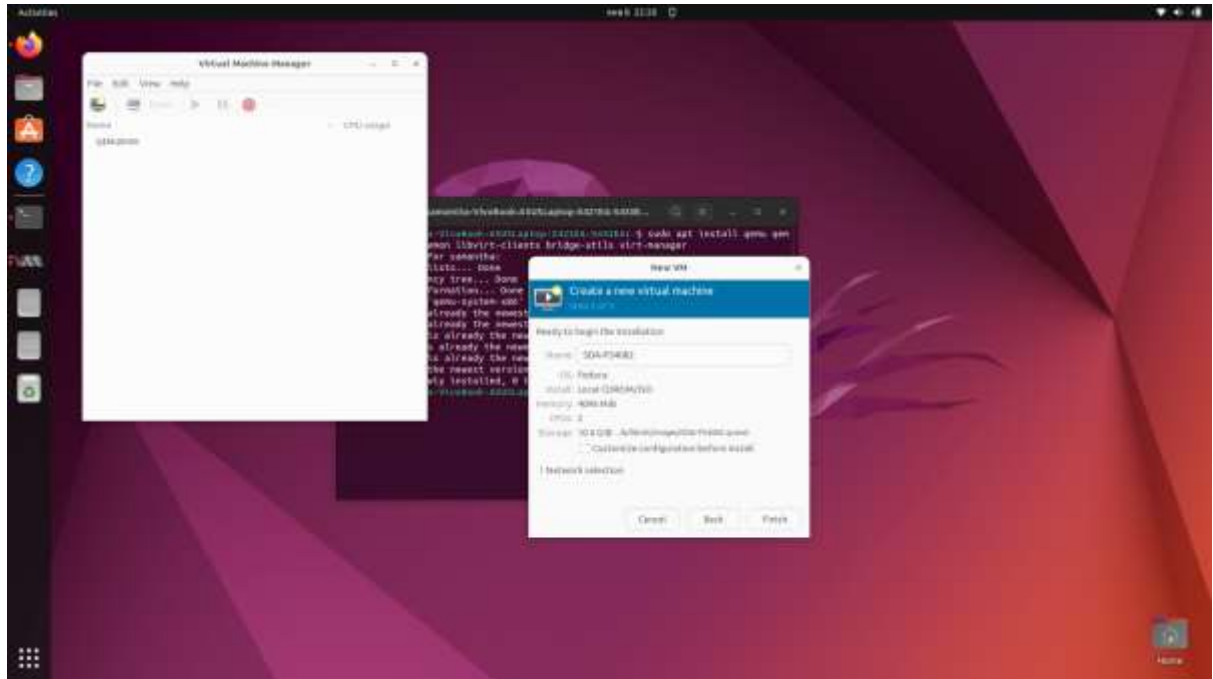


Рисунок 43 – Создание виртуальной машины

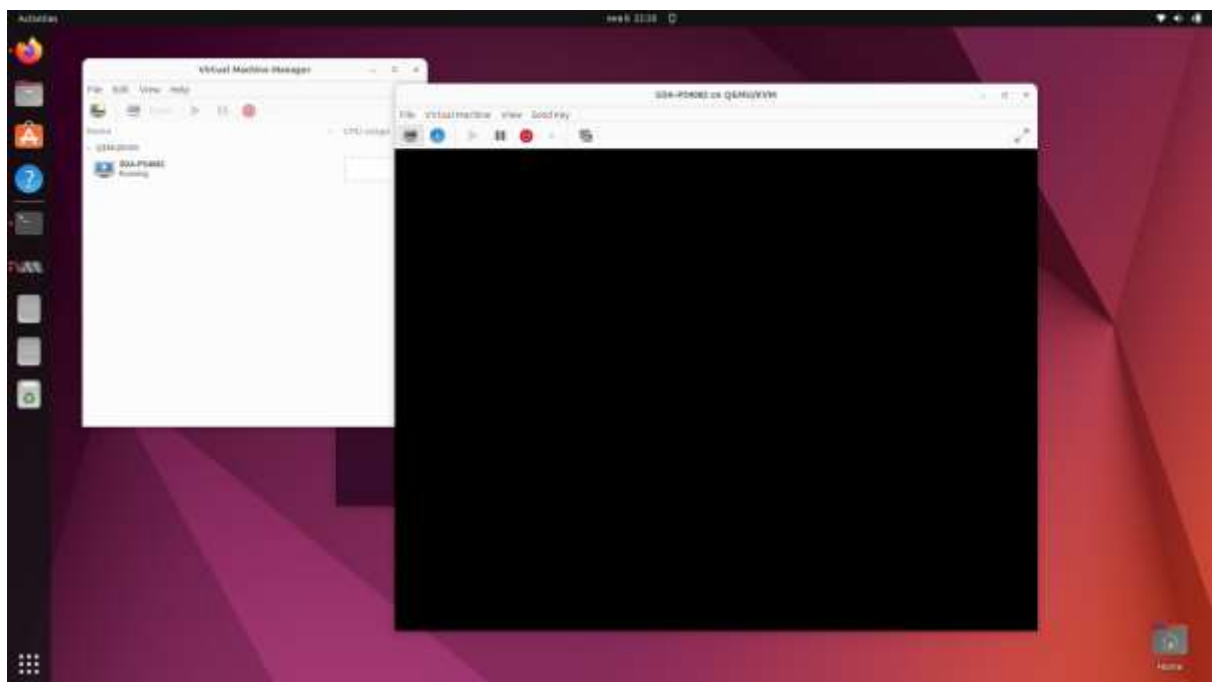


Рисунок 44 – Запуск виртуальной машины

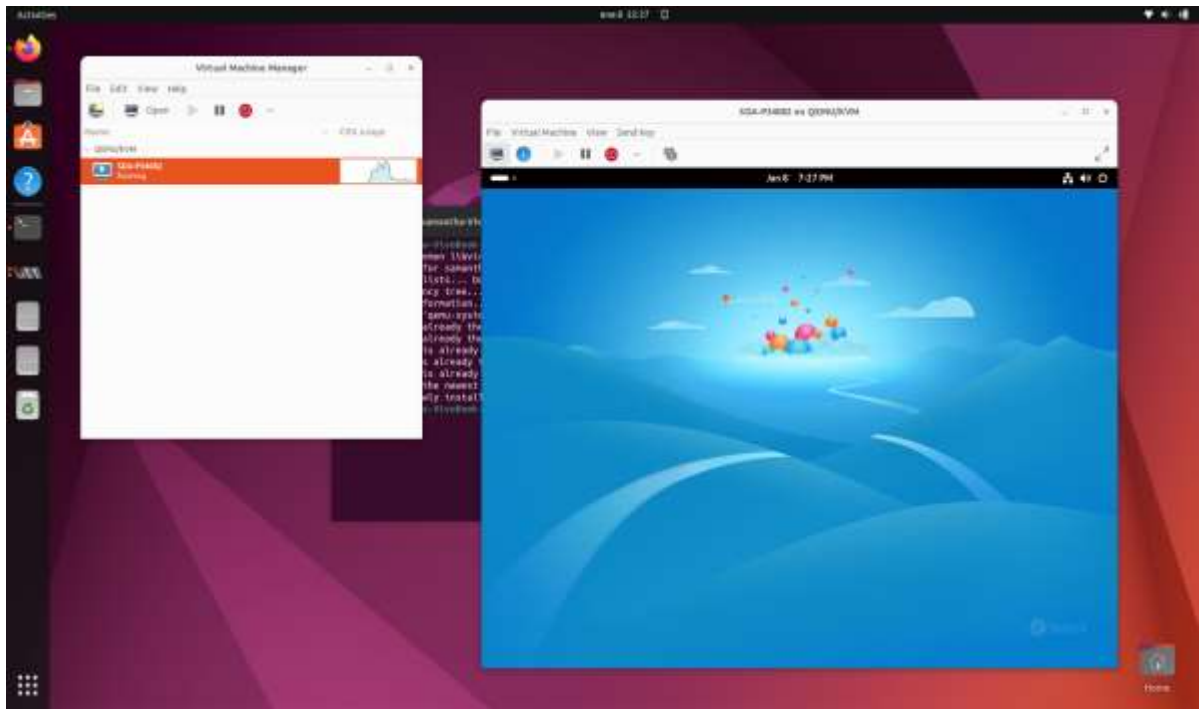


Рисунок 45 – Работоспособность виртуальной машины Fedora

10. Изучите свойства виртуальной машины созданной в VMM, какие настройки виртуальной машины можно выставлять? Какие параметры можно отслеживать в оперативном режиме?

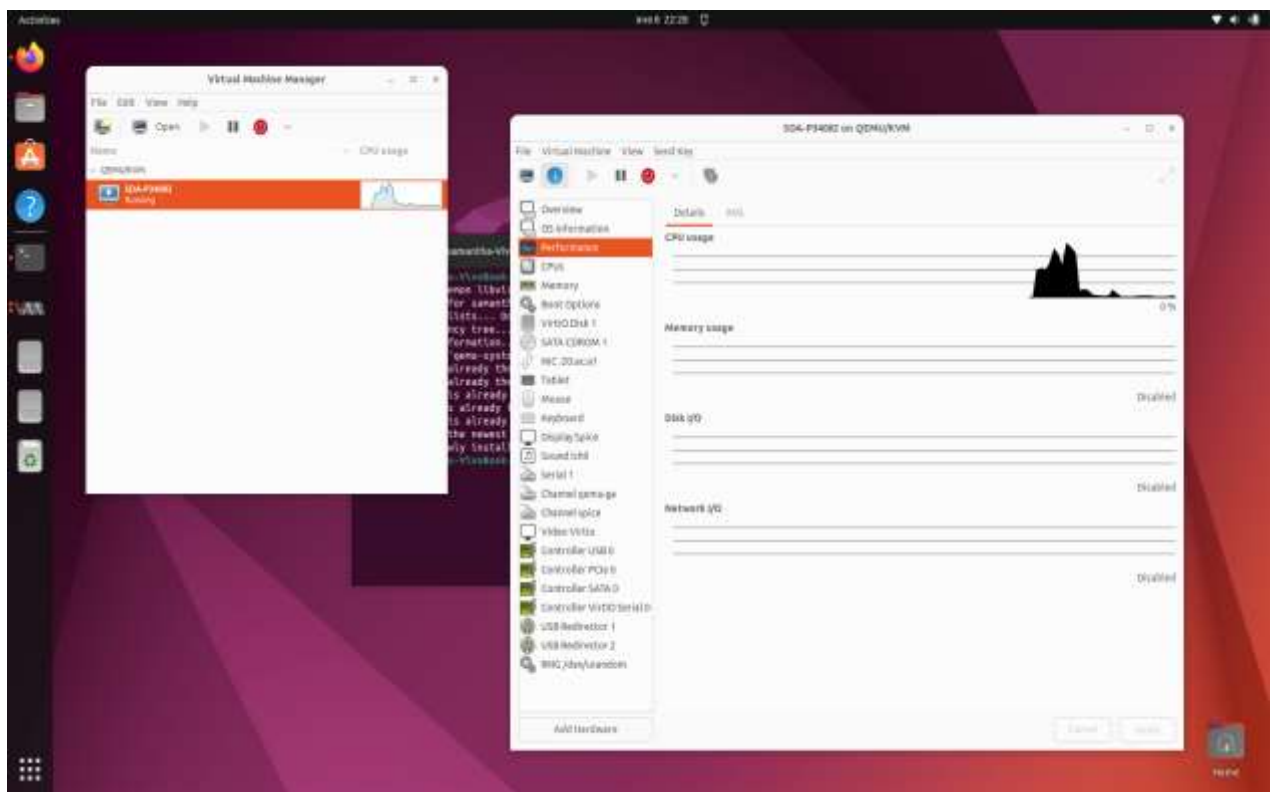


Рисунок 46 – Свойства виртуальной машины

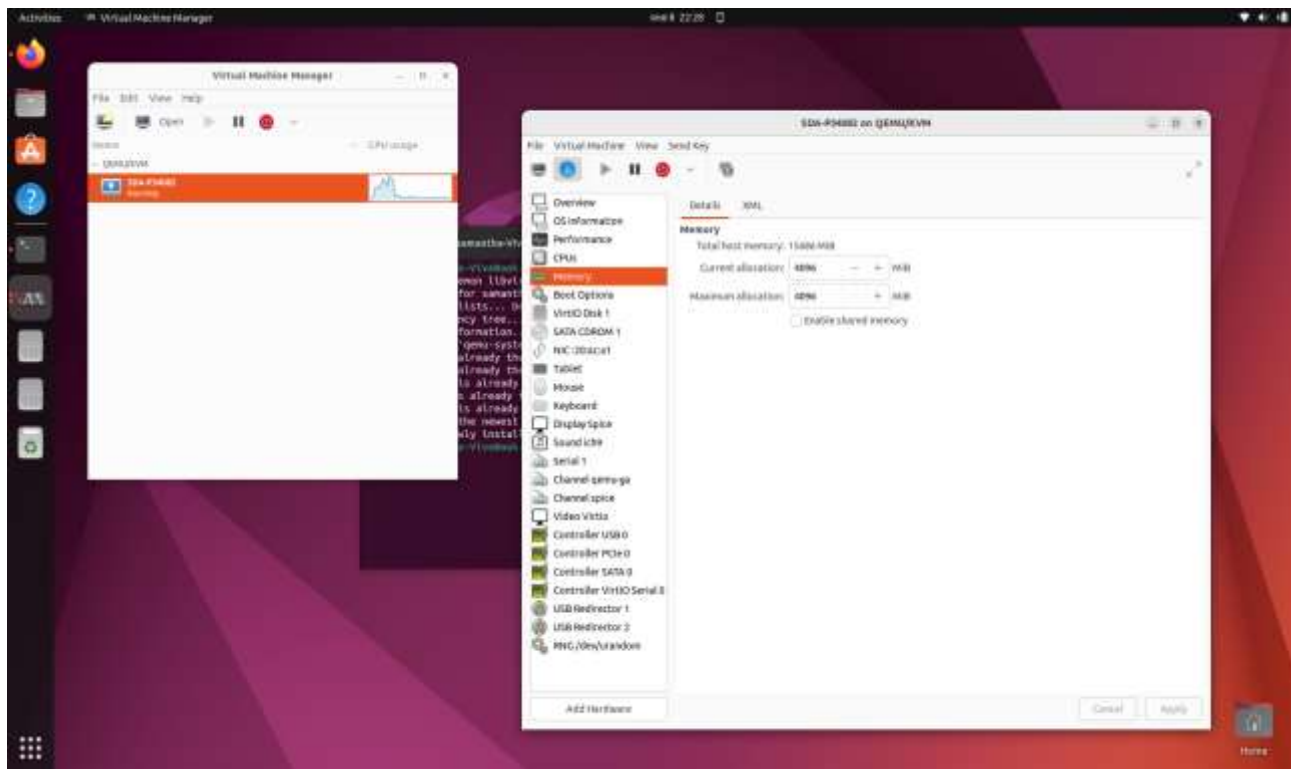


Рисунок 47 – Свойства виртуальной машины

11. Добавьте еще один жесткий диск объемом 10МиБ.

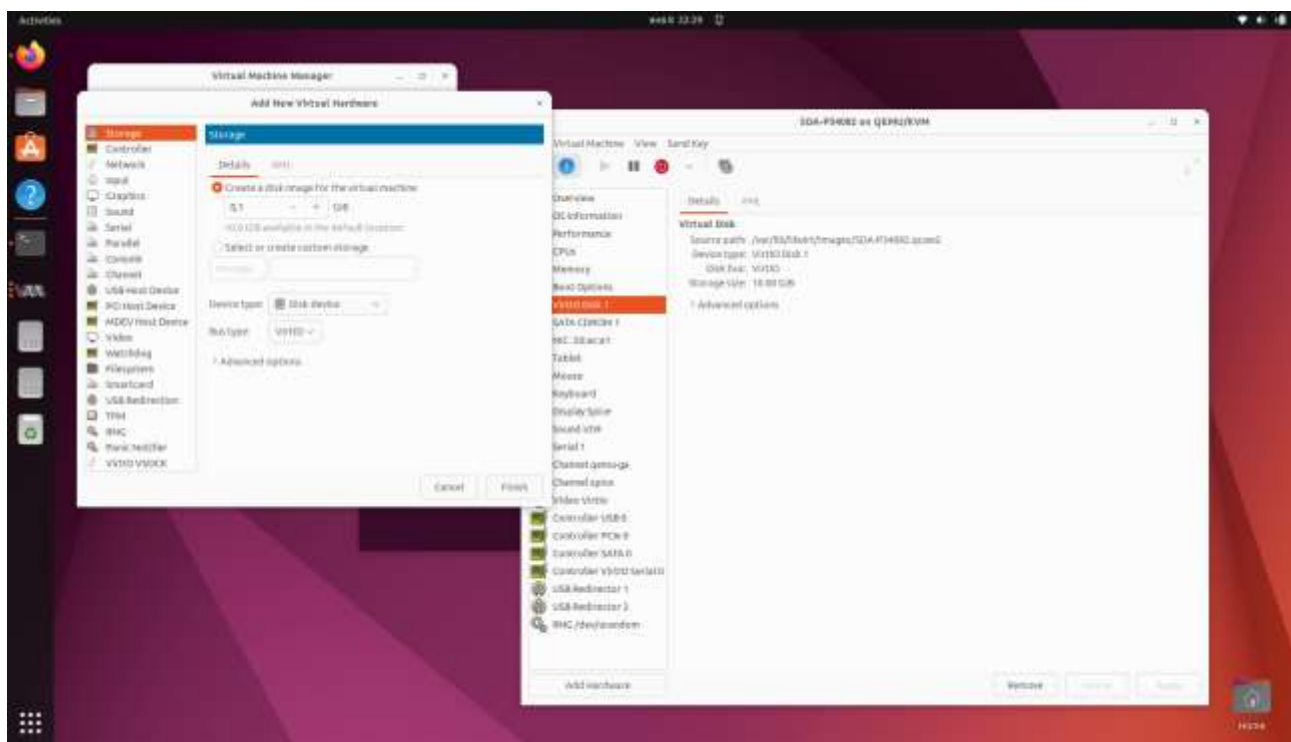


Рисунок 48 – Добавление жесткого диска

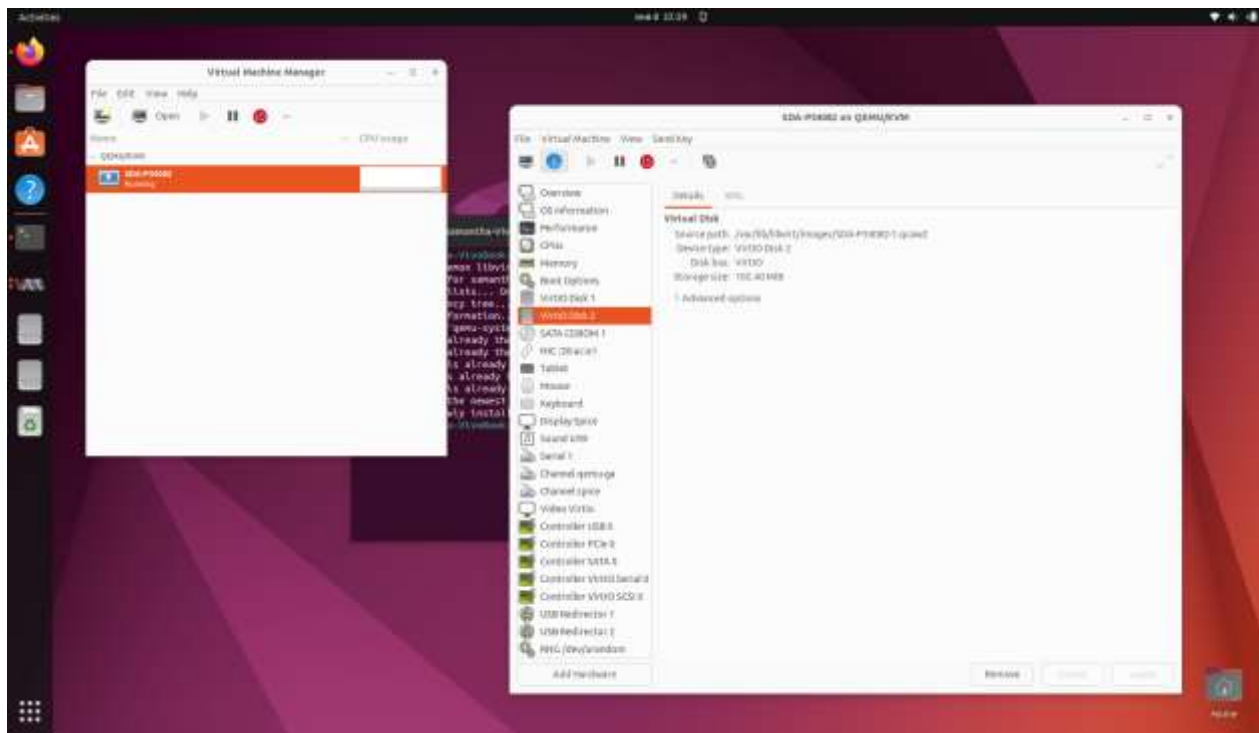


Рисунок 49 – Свойства виртуальной машины

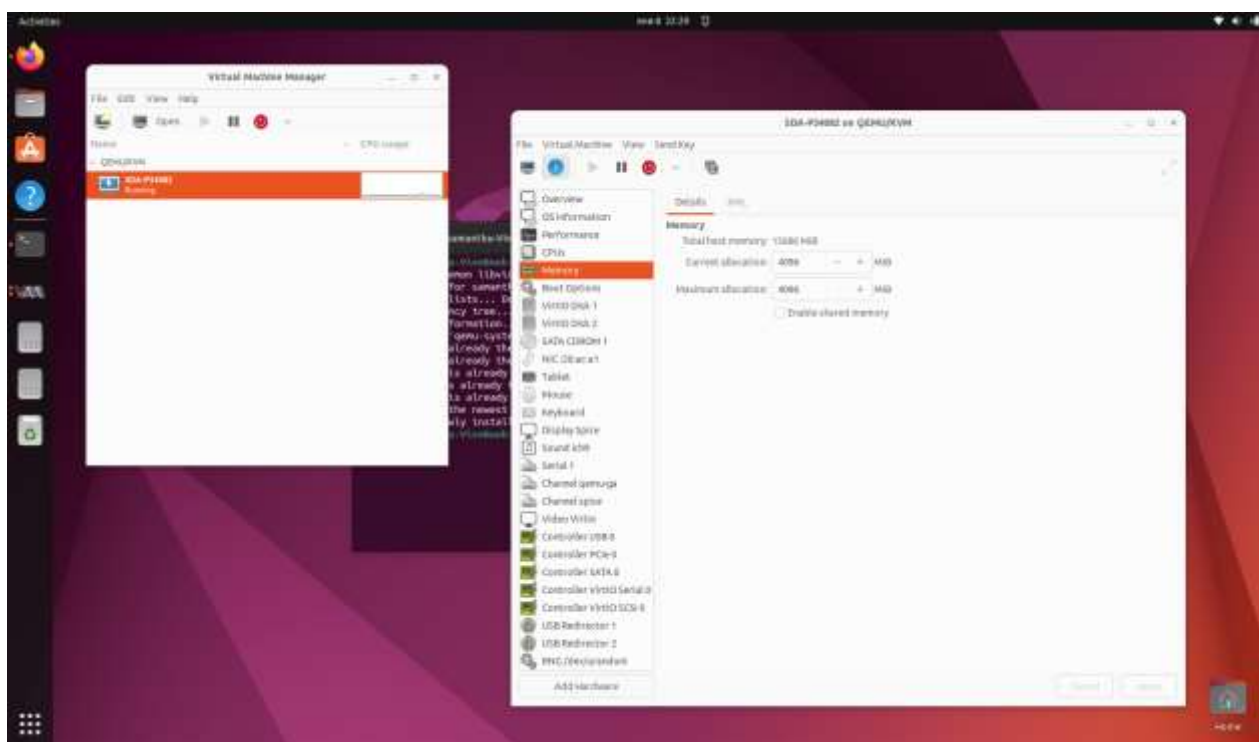


Рисунок 50 – Свойства виртуальной машины

12. Создать снимок состояния системы. Установить приложение Yandex Browser. Восстановить состояние системы в первоначальное состояние.

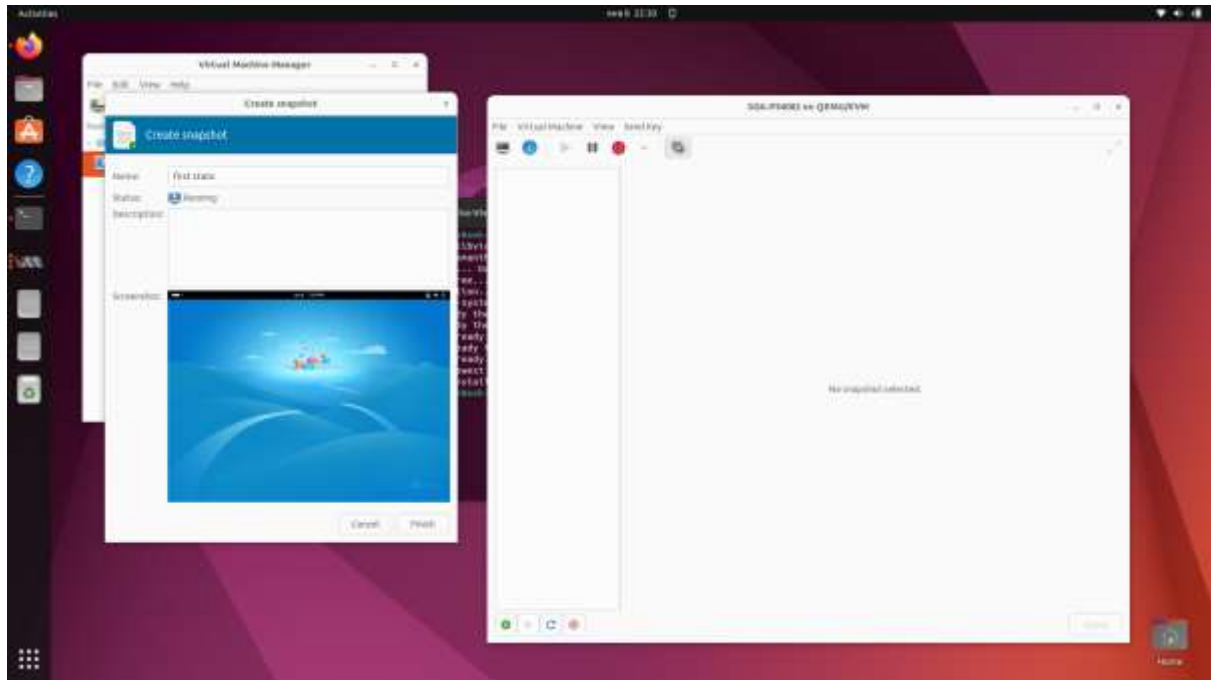


Рисунок 51 – Создание снимка состояния системы

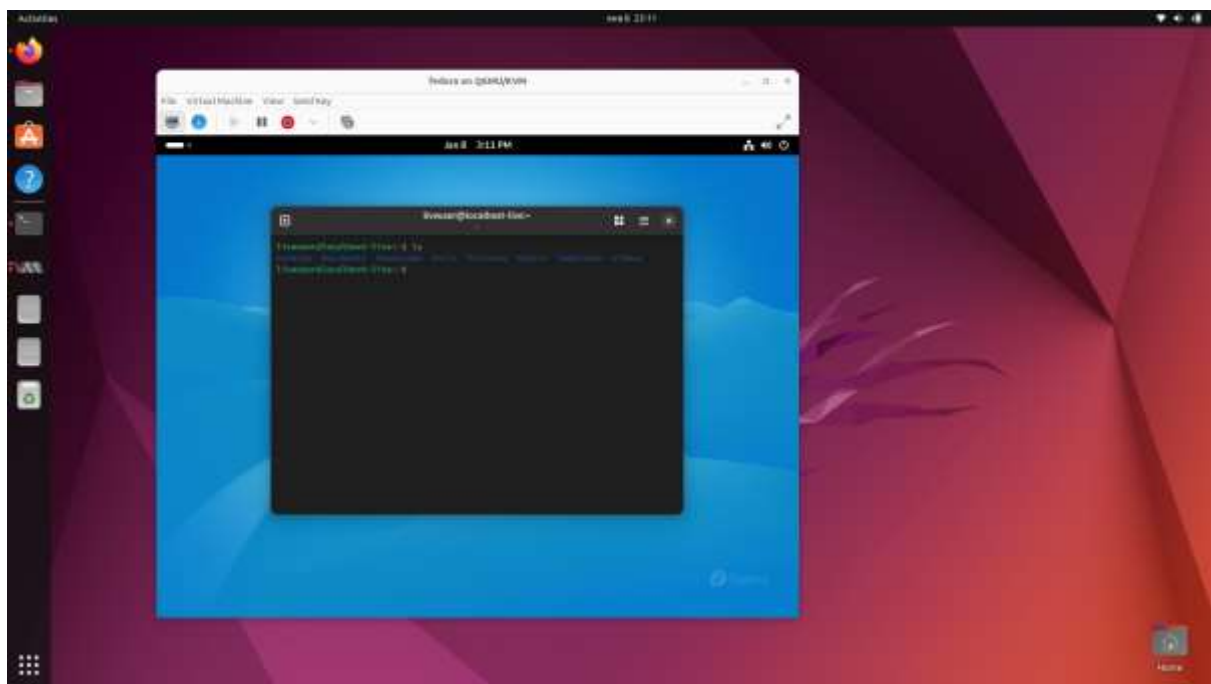


Рисунок 52 – Проверка отсутствия директории hello

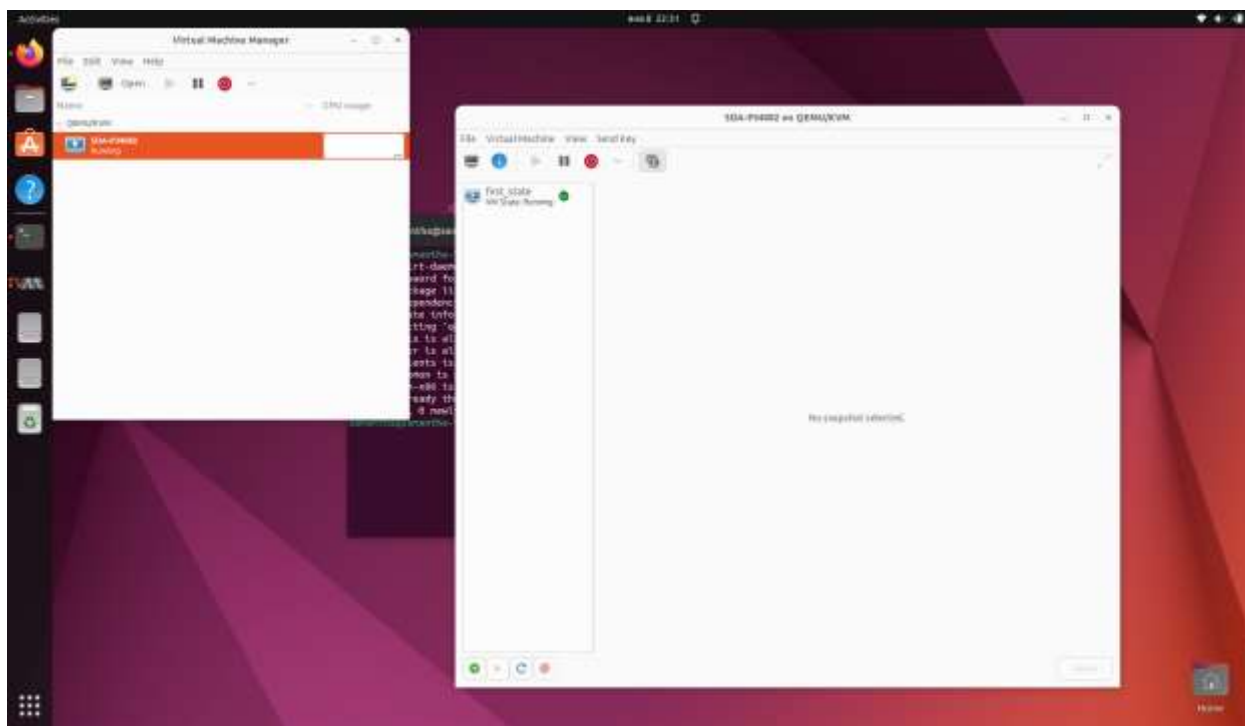


Рисунок 53 – Снимок системы

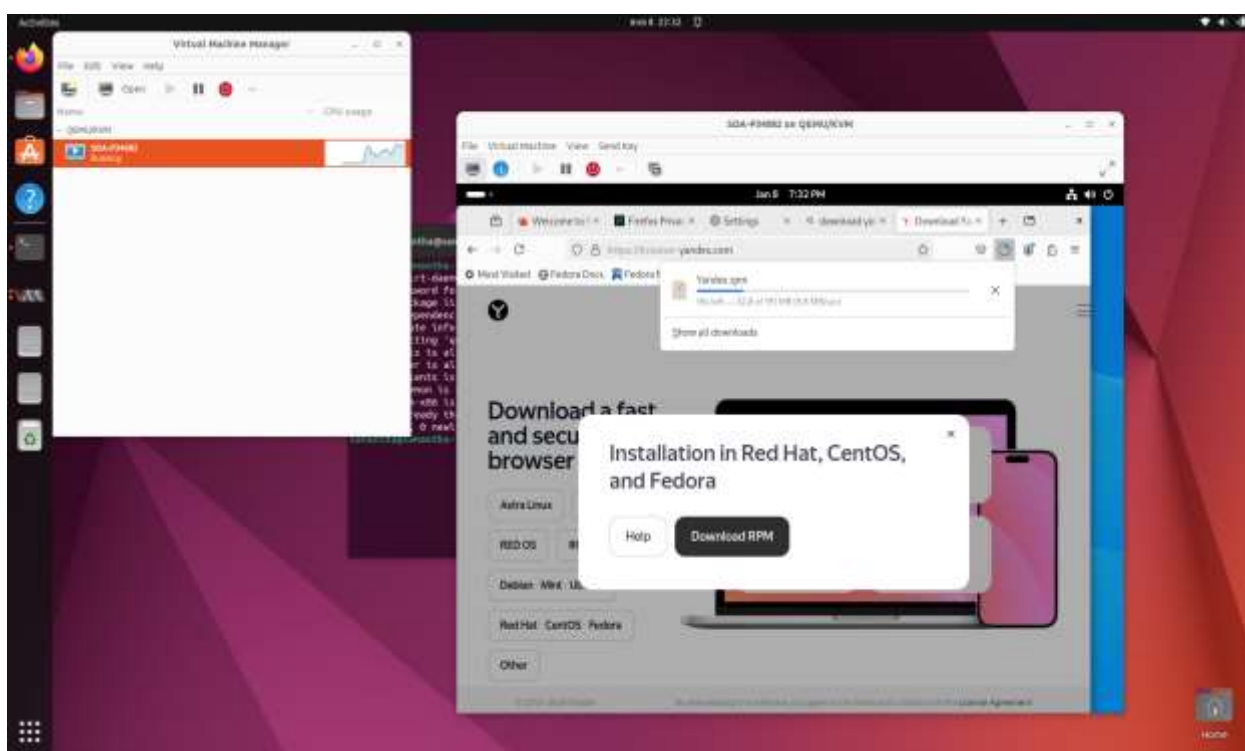


Рисунок 54 – Установка Yandex Browser

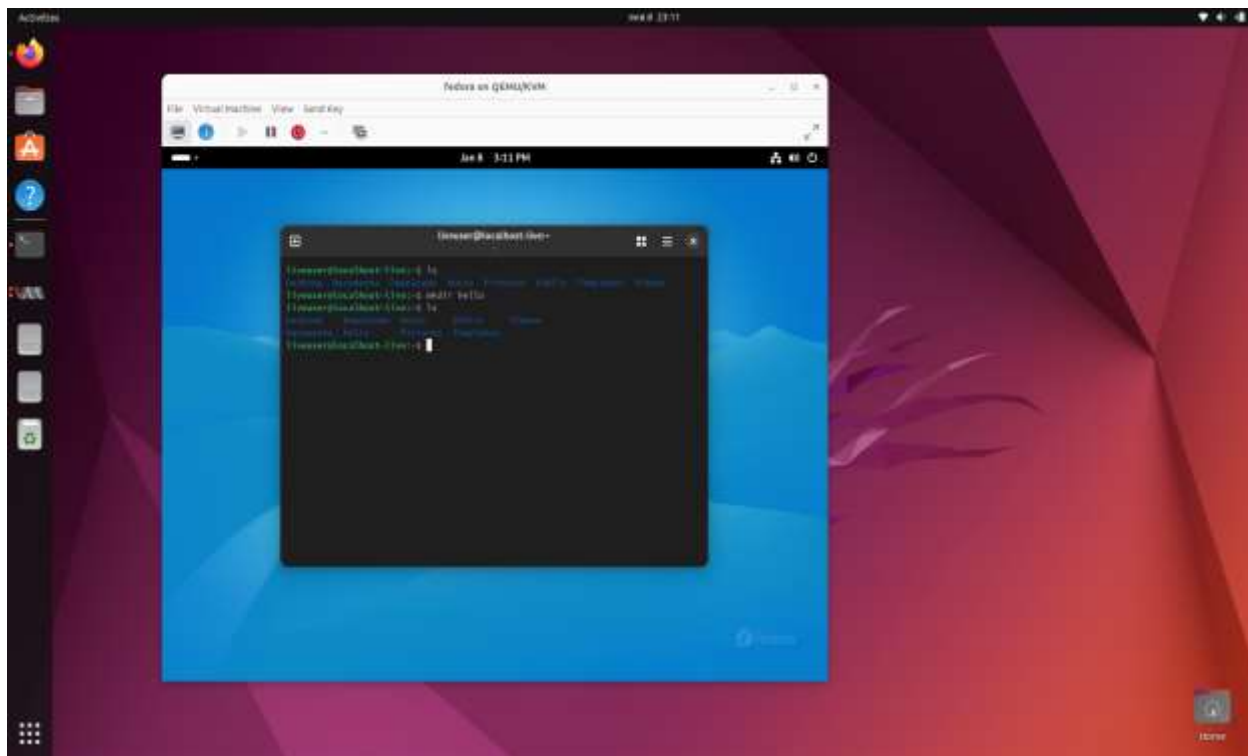


Рисунок 55 – Создание директории hello

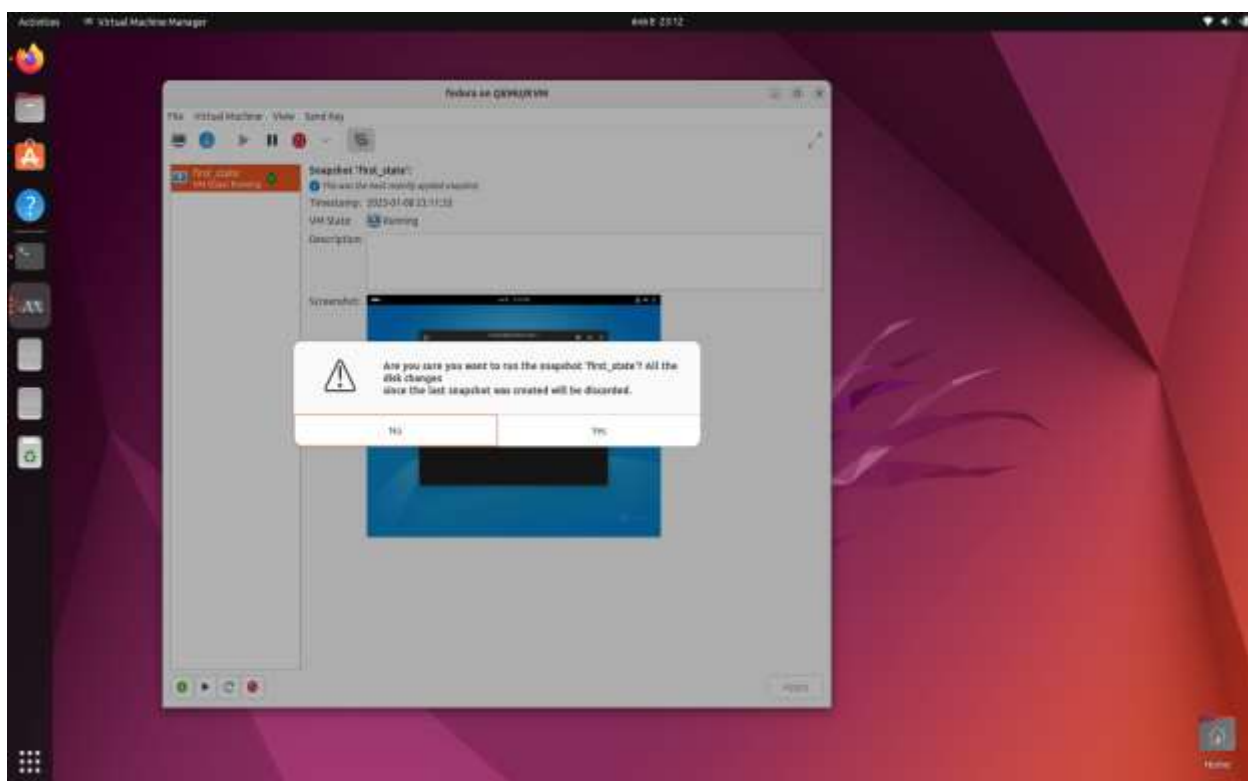


Рисунок 56 – Восстановление до первоначального состояния

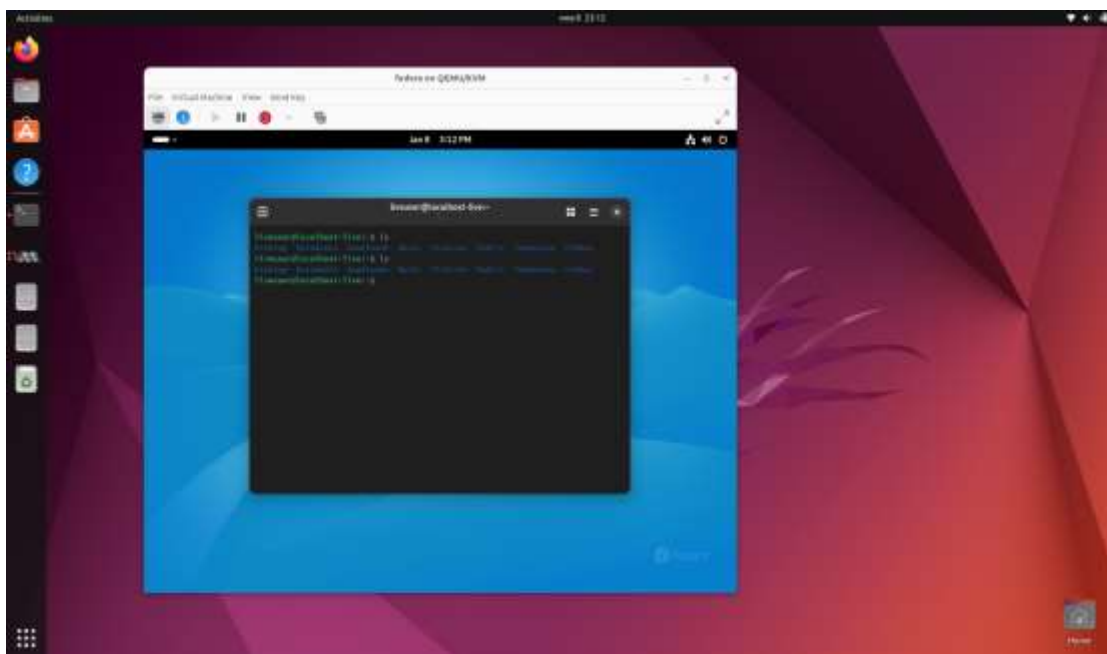


Рисунок 57 – Восстановление до первоначального состояния

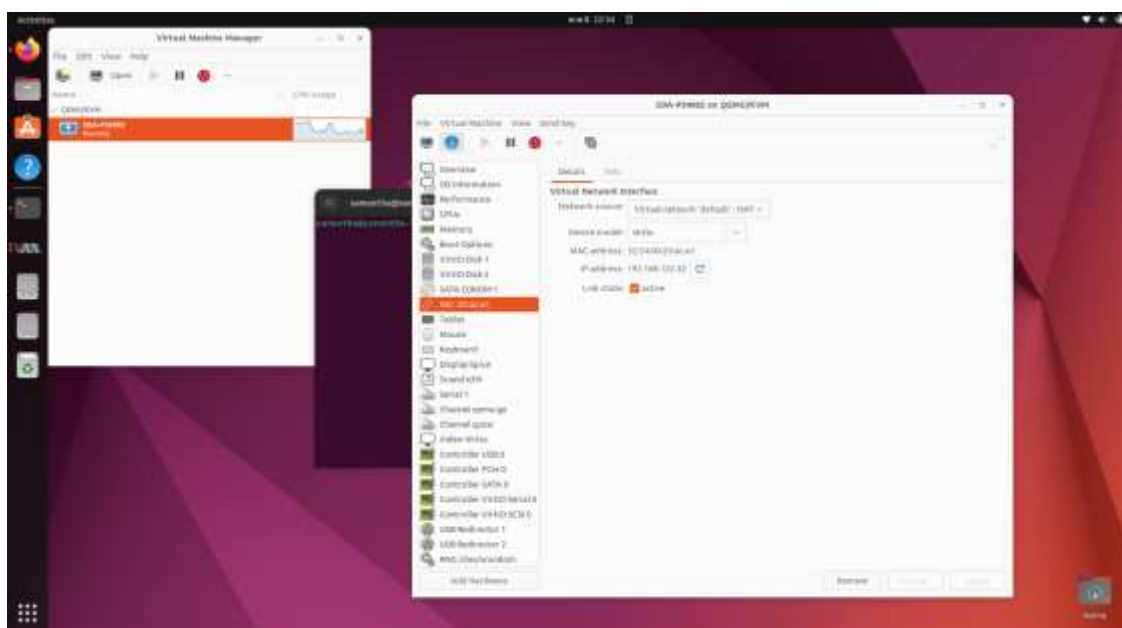
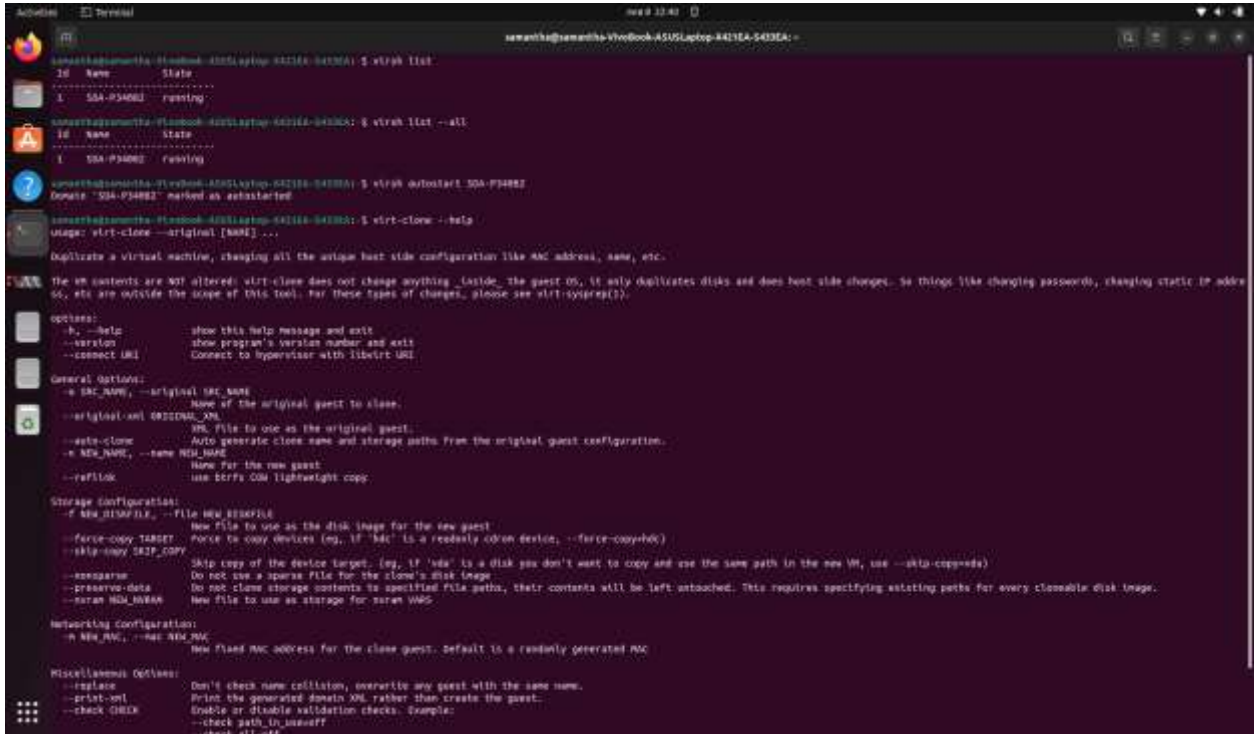


Рисунок 58 – Сетевые настройки виртуальной машины

13. Какие сетевые настройки можно устанавливать гостевой машине?

- Сетевой мост (Bridge)
- NAT (Network Address Translation)
- Сетевой адаптер (virtio, e1000 и др.)

14. Управление виртуальной машиной из командной строки virsh



```
samantha@samantha-VirtualBox: ~$ virsh list
# ID Name State
1 SDA-P34882 running

samantha@samantha-VirtualBox: ~$ virsh list --all
# ID Name State
1 SDA-P34882 running

samantha@samantha-VirtualBox: ~$ virsh autostart SDA-P34882
Domain 'SDA-P34882' marked as autostarted

samantha@samantha-VirtualBox: ~$ virsh clone --help
usage: virsh clone --original NAME [...] ...

Duplicate a virtual machine, changing all the unique host side configuration like MAC address, name, etc.

the vm contents are NOT altered: virsh clone does not change anything inside the guest OS, it only duplicates disks and does host side changes. So things like changing passwords, changing static IP address, etc are outside the scope of this tool. For these types of changes, please see virt-sysprep().

Options:
  -h, --help            show this help message and exit
  --version             show program's version number and exit
  --connect URI         Connect to Hypervisor with libvirt URI

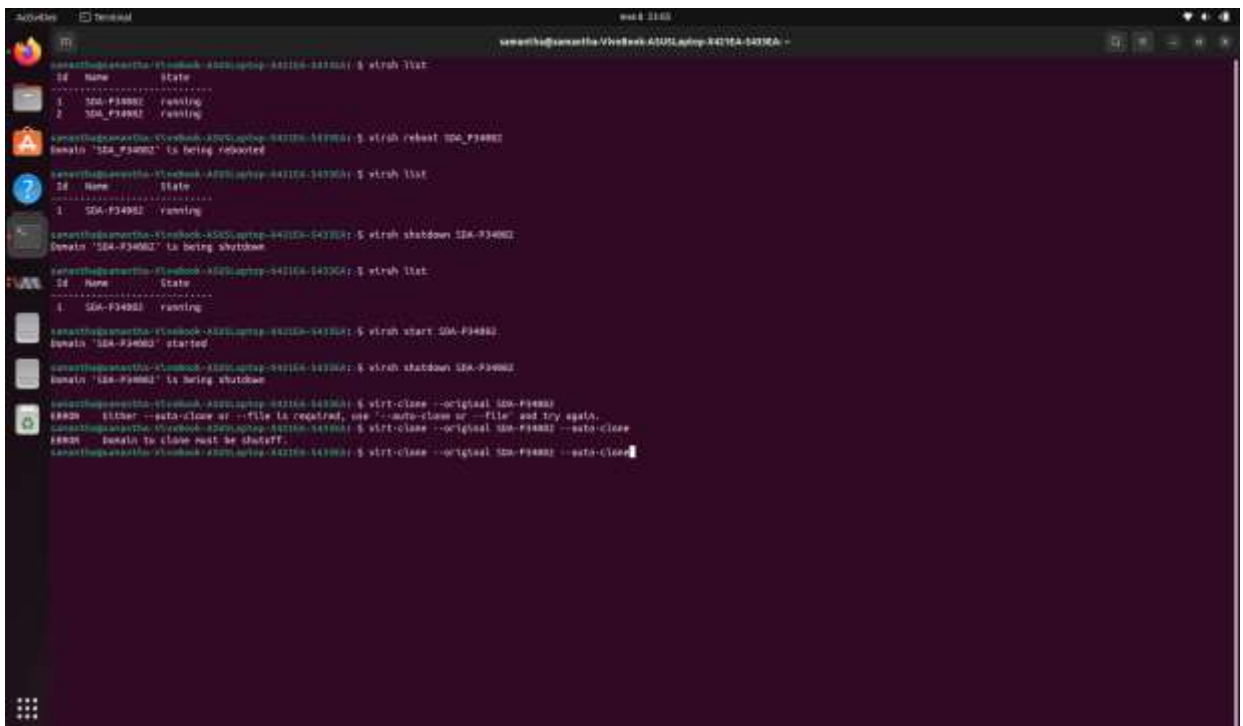
General Options:
  -o SRC_NAME, --original SRC_NAME
                        Name of the original guest to clone.
  --original-xml ORIGIN_XML
                        XML file to use as the original guest.
  --auto-clone          Auto generate clone name and storage paths from the original guest configuration.
  -n NEW_NAME, --name NEW_NAME
                        Name for the new guest.
  --reflink             use btrfs COW lightweight copy.

Storage Configuration:
  -f NEW_DISKFILE, --file NEW_DISKFILE
                        New file to use as the disk image for the new guest.
  --force-copy TARGET   Force to copy devices (eg. if 'hdc' is a readonly cdrom device, --force-copy-hdc)
  --skip-copy SKIP_COPY
                        Skip copy of the device target. (eg. if 'sda' is a disk you don't want to copy and use the same path in the new VM, use --skip-copy-sda)
  --nozero             Do not use a sparse file for the clone's disk image.
  --preserve-data       Do not clone storage contents to specified file path, their contents will be left untouched. This requires specifying existing paths for every cloneable disk image.
  --sran NEW_NBD_NAME
                        New file to use as storage for sran VMs.

Networking Configuration:
  -n NEW_MAC, --mac NEW_MAC
                        New fixed MAC address for the clone guest. Default is a randomly generated MAC.

Miscellaneous Options:
  --nocheck            Don't check name collision, overwrite any guest with the same name.
  --print-xml          Print the generated domain XML, rather than create the guest.
  --check CHECK        Enable or disable validation checks. Example:
                        --check path_in_use=off
                        --check all=off
```

Рисунок 59 – Управление виртуальной машиной из командной строки



```
samantha@samantha-VirtualBox: ~$ virsh list
# ID Name State
1 SDA-P34882 running
2 SDA-P34882 running

samantha@samantha-VirtualBox: ~$ virsh reboot SDA-P34882
Domain 'SDA-P34882' is being rebooted

samantha@samantha-VirtualBox: ~$ virsh list
# ID Name State
1 SDA-P34882 running

samantha@samantha-VirtualBox: ~$ virsh shutdown SDA-P34882
Domain 'SDA-P34882' is being shutdown

samantha@samantha-VirtualBox: ~$ virsh list
# ID Name State
1 SDA-P34882 running

samantha@samantha-VirtualBox: ~$ virsh start SDA-P34882
Domain 'SDA-P34882' started

samantha@samantha-VirtualBox: ~$ virsh shutdown SDA-P34882
Domain 'SDA-P34882' is being shutdown

samantha@samantha-VirtualBox: ~$ virsh clone --original SDA-P34882
ERROR: Either --auto-clone or --file is required, use --auto-clone or --file and try again.
samantha@samantha-VirtualBox: ~$ virsh clone --original SDA-P34882 --auto-clone
ERROR: Domain to clone must be shut off.
samantha@samantha-VirtualBox: ~$ virsh clone --original SDA-P34882 --auto-clone
```

Рисунок 60 – Управление виртуальной машиной из командной строки

3 ИСПОЛЬЗОВАНИЕ УТИЛИТЫ VIRT-INSTALL

Ниже представлено задание, которое необходимо выполнить в рамках данного раздела:

1. Установите в системы следующие пакеты:

\$sudo apt install virt-install libosinfo-bin

2. Проверьте какие ОС доступны из репозитория

\$osinfo-query os

3. Отсортируйте все ОС Windows и сохраните список

4. Отсортируйте все ОС Fedora и сохраните список

5. Добавить виртуальную машину с помощью virt-install

**virt-install **

**--name FIO_Group **

**--virt-type=kvm **

**--memory 2048 --vcpus=2 **

**--os-variant=fedora31 **

**--hvm **

**--cdrom=fedora.iso **

**--network default,model=virtio **

--disk

**path=~/.VM/fedora31.qcow2,size=8,bus=virtio,format=qcow2 **

**--graphics vnc **

--noautoconsole

6. Объясните значение параметров используем 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?

Далее поэтапно будем выполнять каждое из заданий.

\$sudo apt install virt-install libosinfo-bin



The screenshot shows a Kali Linux desktop environment. A terminal window is open, displaying the command `dpkg-query -f='\${Package} \${Version} \${Architecture} \${Homepage}\n' -W libc6:amd64`. The output of the command is a list of Linux distributions and their versions, including almalinux8, alpine3.10, altlinux, etc.

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	Alpline Linux 3.10	3.10	http://alpinelinux.org/alpinelinux/3.10
alpinelinux3.11	Alpline Linux 3.11	3.11	http://alpinelinux.org/alpinelinux/3.11
alpinelinux3.12	Alpline Linux 3.12	3.12	http://alpinelinux.org/alpinelinux/3.12
alpinelinux3.13	Alpline Linux 3.13	3.13	http://alpinelinux.org/alpinelinux/3.13
alpinelinux3.14	Alpline Linux 3.14	3.14	http://alpinelinux.org/alpinelinux/3.14
alpinelinux3.15	Alpline Linux 3.15	3.15	http://alpinelinux.org/alpinelinux/3.15
alpinelinux3.16	Alpline Linux 3.16	3.16	http://alpinelinux.org/alpinelinux/3.16
alpinelinux3.17	Alpline Linux 3.17	3.17	http://alpinelinux.org/alpinelinux/3.17
alpinelinux3.18	Alpline Linux 3.18	3.18	http://alpinelinux.org/alpinelinux/3.18
alpinelinux3.19	Alpline Linux 3.19	3.19	http://alpinelinux.org/alpinelinux/3.19
alpinelinux3.5	Alpline Linux 3.5	3.5	http://alpinelinux.org/alpinelinux/3.5
alpinelinux3.6	Alpline Linux 3.6	3.6	http://alpinelinux.org/alpinelinux/3.6
alpinelinux3.7	Alpline Linux 3.7	3.7	http://alpinelinux.org/alpinelinux/3.7
alpinelinux3.8	Alpline Linux 3.8	3.8	http://alpinelinux.org/alpinelinux/3.8
alpinelinux3.9	Alpline 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 62 – Операционные системы из репозитория

3. Отсортируйте все ОС Windows и сохраните список.

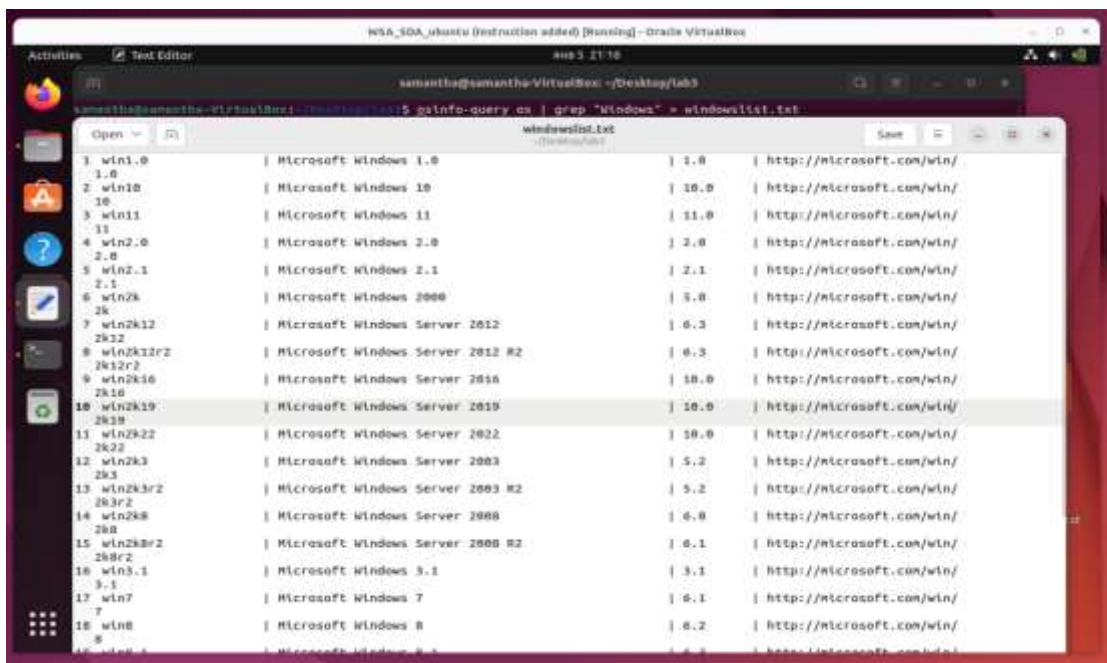


Рисунок 63 – Операционные системы Windows

4. Отсортируйте все ОС Fedora и сохраните список.

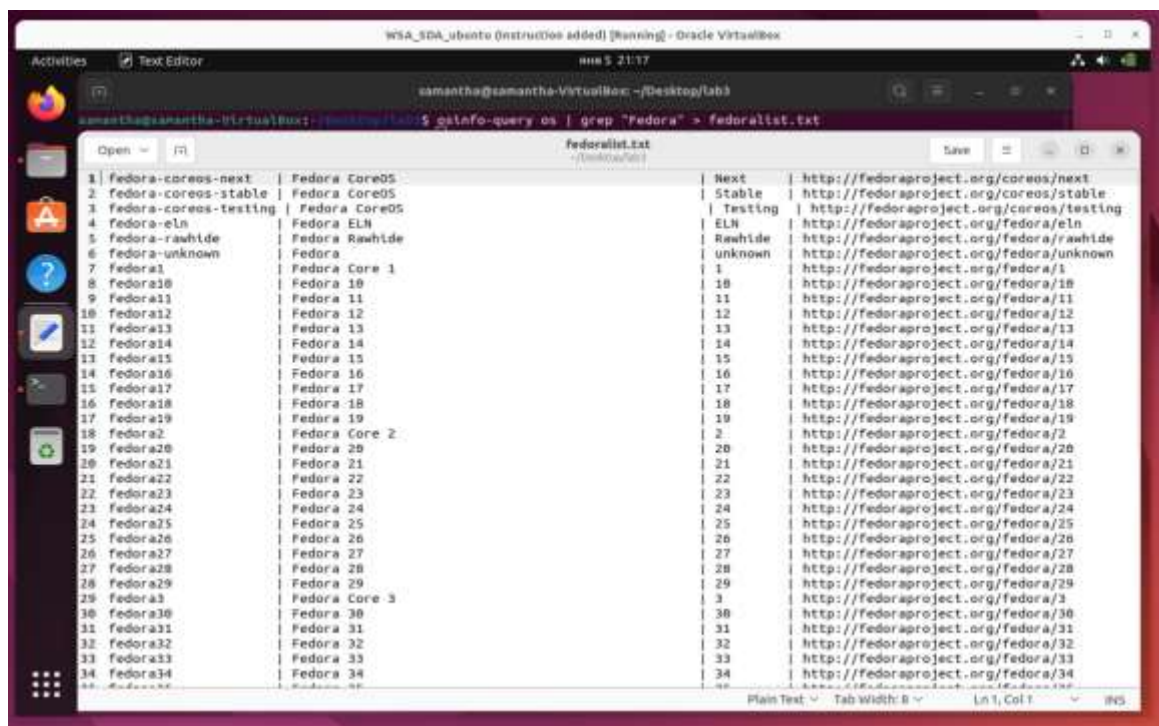


Рисунок 64 – Операционные системы Fedora

5. Добавить виртуальную машину с помощью virt-install.

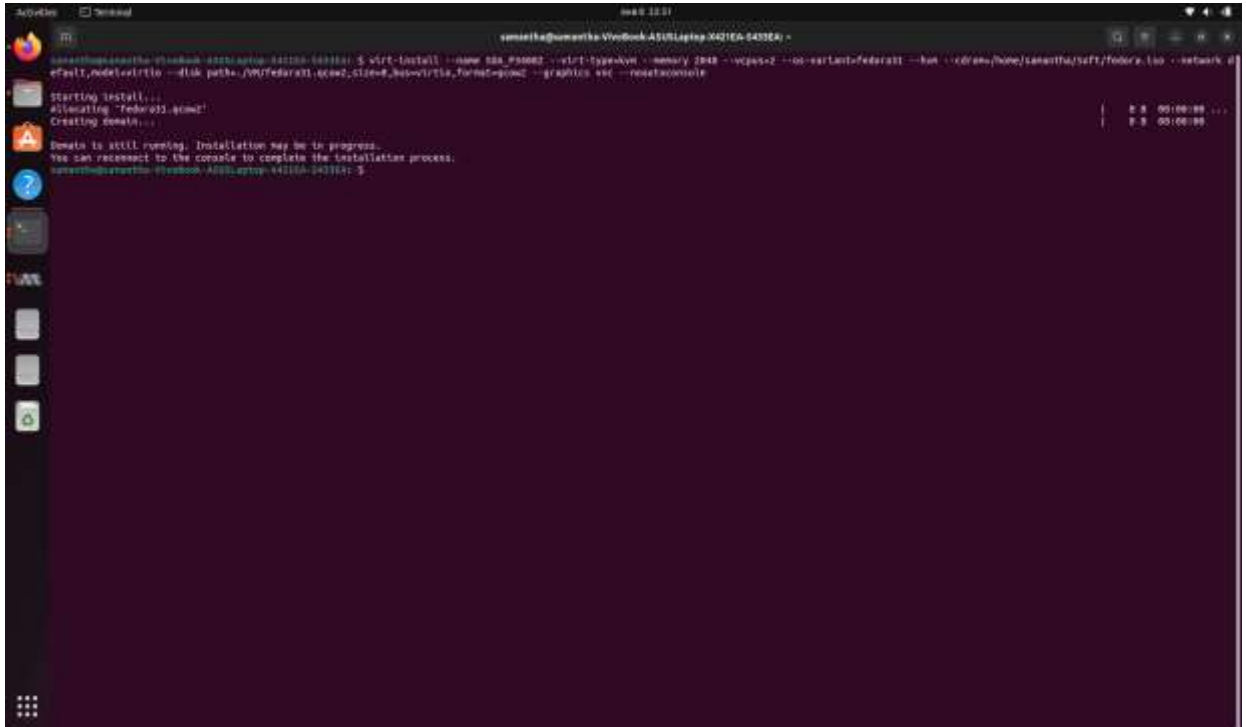


Рисунок 65 – Добавление виртуальной машины с помощью virt-install

6. Объясните значение параметров используем при вводе данной команды.

- name FIO_Group: Имя виртуальной машины.
- virt-type=kvm: Тип виртуализации.
- memory 2048: Размер памяти в МБ.
- vcpus=2: Количество процессоров.
- os-variant=fedora31: Дистрибутив ОС.
- cdrom=fedora.iso: Путь к ISO-образу.
- network default,model=virtio: Настройка сетевой карты.
- disk path=~ / VM / fedora31.qcow2: Путь к диску и его размер.
- graphics vnc: Использование VNC для графики.
- noautoconsole: Отключить консоль после установки.

7. Проверьте запуск виртуальной машины.

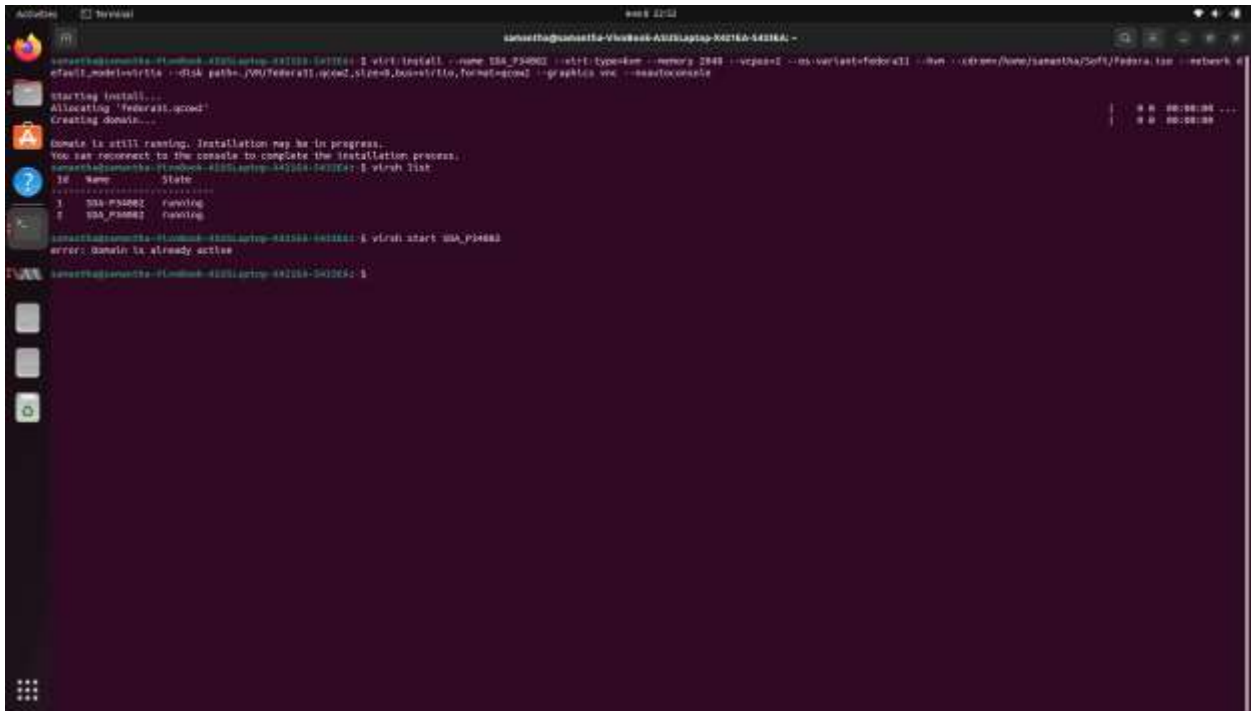


Рисунок 66 – Проверка запуска виртуальной машины

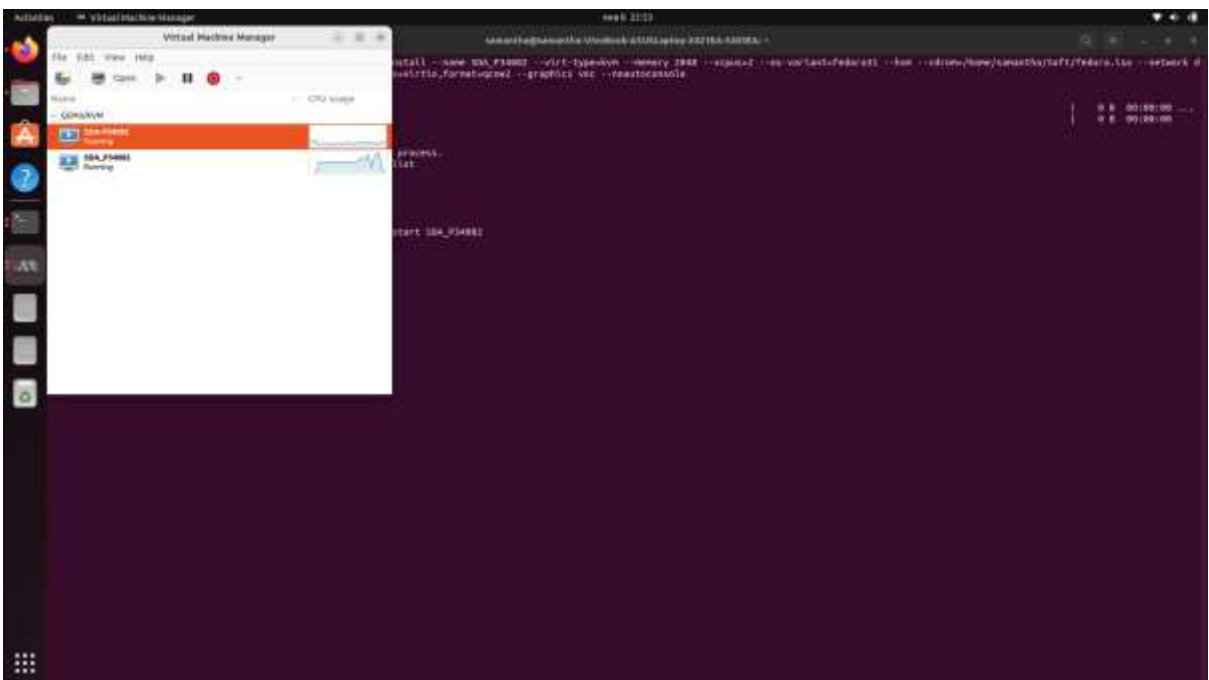


Рисунок 67 – Проверка запуска виртуальной машины

8. Как можно запустить машину в фоновом режиме?

Добавить параметр `--noautoconsole` для запуска машины без автоматического подключения.

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы были изучены основные принципы работы с виртуальными гипервизорами на базе Unix-систем. Рассмотрены и освоены такие инструменты, как Gnome Boxes, QEMU/KVM, Virtual Machine Manager (VMM) и утилита virt-install, которые используются для создания, настройки и управления виртуальными машинами.

В процессе работы успешно решены задачи по добавлению виртуальных дисков, настройке сетевых подключений, созданию и применению снимков состояния системы, а также удаленному подключению к виртуальным и физическим системам с использованием протоколов SSH и RDP. Особое внимание было уделено исследованию свойств виртуальных машин, настройке их параметров и мониторингу работы в реальном времени.

Лабораторная работа продемонстрировала важность навыков работы с гипервизорами и виртуализацией для решения различных задач в области системного администрирования и разработки программного обеспечения. Полученные знания и опыт могут быть применены для создания изолированных вычислительных сред, тестирования программного обеспечения и оптимизации инфраструктуры.

Таким образом, выполнение данной работы позволяет сделать вывод, что виртуализация является мощным и гибким инструментом, обеспечивающим эффективное использование ресурсов и упрощение управления вычислительными системами.