

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

## ОТЧЕТ

### ПО ЛАБОРАТОРНОЙ РАБОТЕ №

дисциплина: Администрирование сетевых подсистем

Студент: Кармацкий Н.С.

Группа: НФИбд-01-21

МОСКВА

2023 г.

## Постановка задачи

1. Сформируйте box-файл с дистрибутивом Rocky Linux для VirtualBox.
2. Запустите виртуальные машины сервера и клиента и убедитесь в их работоспособности.
3. Внесите изменения в настройки загрузки образов виртуальных машин server и client, добавив пользователя с правами администратора и изменив названия хостов .
4. Скопируйте необходимые для работы с Vagrant файлы и box-файлы виртуальных машин на внешний носитель. Используя эти файлы, вы можете попробовать развернуть виртуальные машины на другом компьютере.

## Выполнение работы

### 1. Формирование box-файла с дистрибутивом Rocky Linux для VirtualBox

1. Загружаем нужные файлы для создание box-файла. Скачиваем Virtual Box, Packer, Vagrant, iso-образ Rocky и нужные файлы для создание образа.

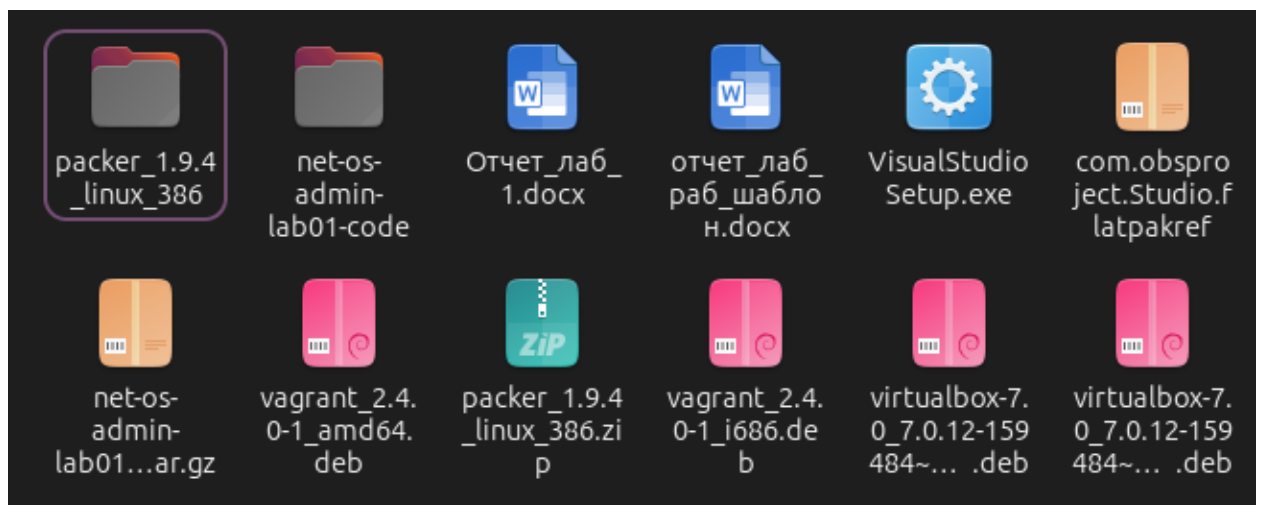


Рис.1.1: Скаченные файлы

2. Создаем каталоги для проекта. Так же в них поместим iso-образ и packer, так же необходимые файлы для дальнейшей работы сервера и клиента. В этих же каталогах создаём каталог provision с подкаталогами default, server и client, в которых будут размещаться скрипты, изменяющие настройки внутреннего окружения базового (общего) образа виртуальной машины, сервера или клиента соответственно.

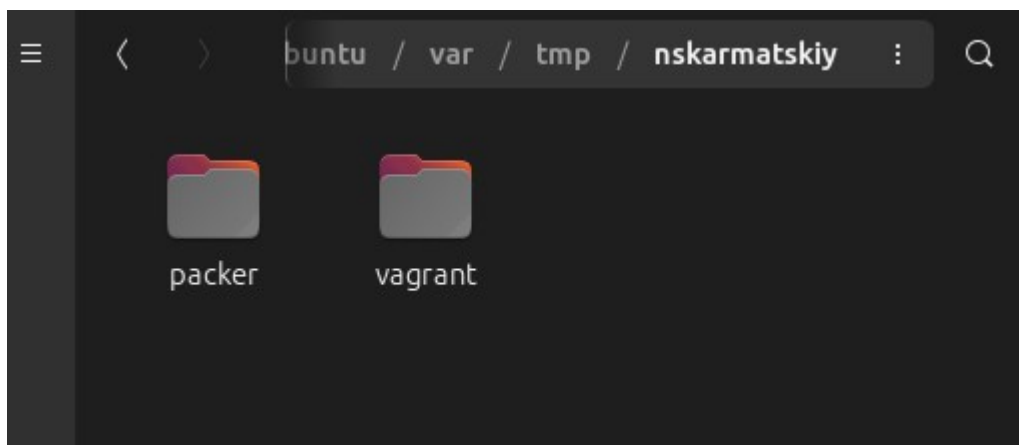


Рис.1.2: Созданный каталоги

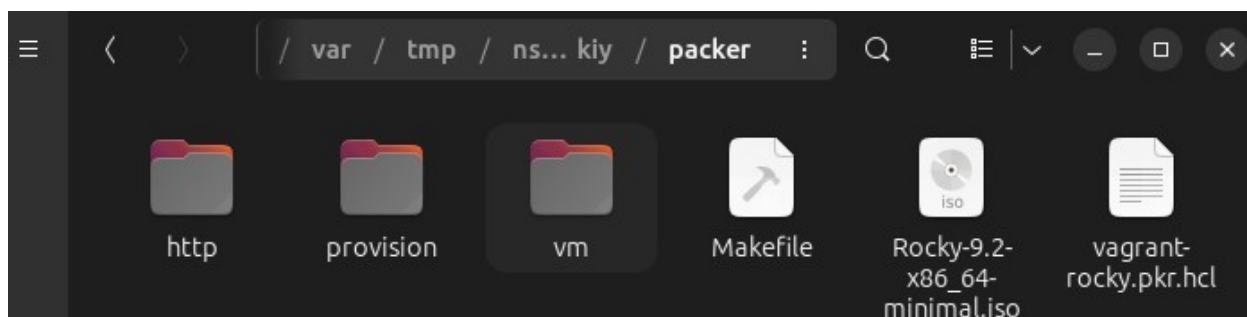


Рис.1.3: Файлы в каталоге Packer

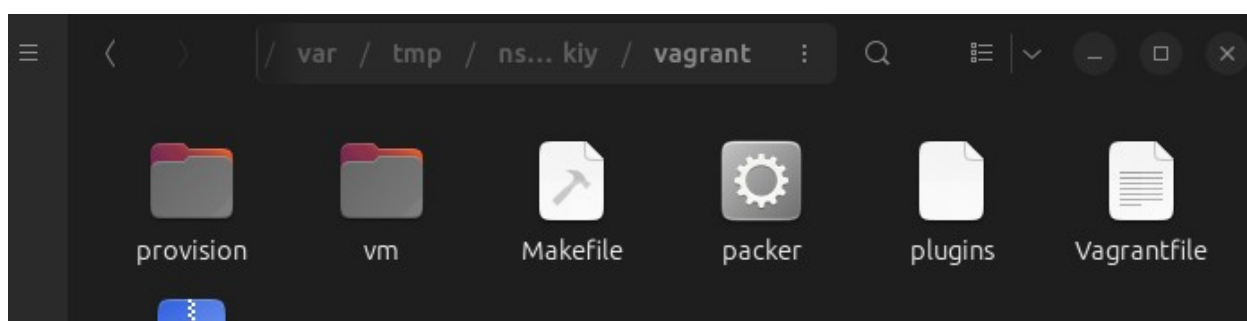
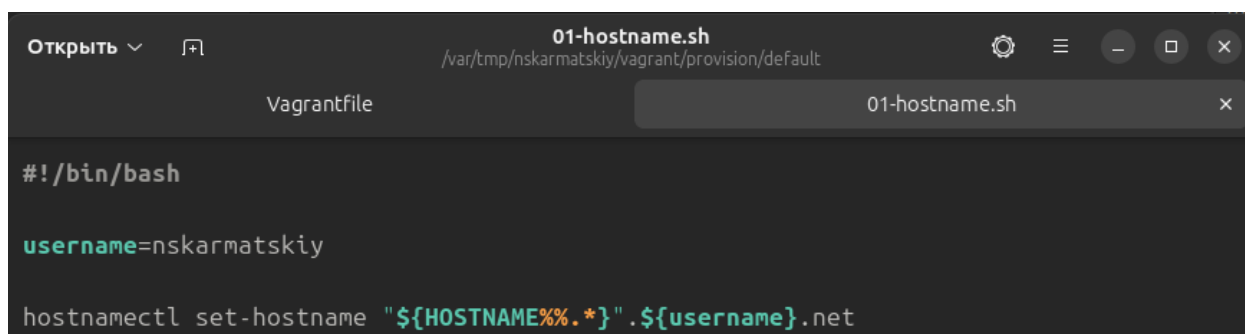


Рис.1.4: Файлы в каталоге Vagrant

3. В каталоге default изменим скрипты 01-user.sh и 01-hostname.sh под себя, то есть вместо user впишем нашу фамилию и инициалы(nskarmatskiy). Меняем это значение в двух каталогах(vagrant и packer).



```
Открыть ▾ 01-hostname.sh
/var/tmp/nskarmatskiy/vagrant/provision/default

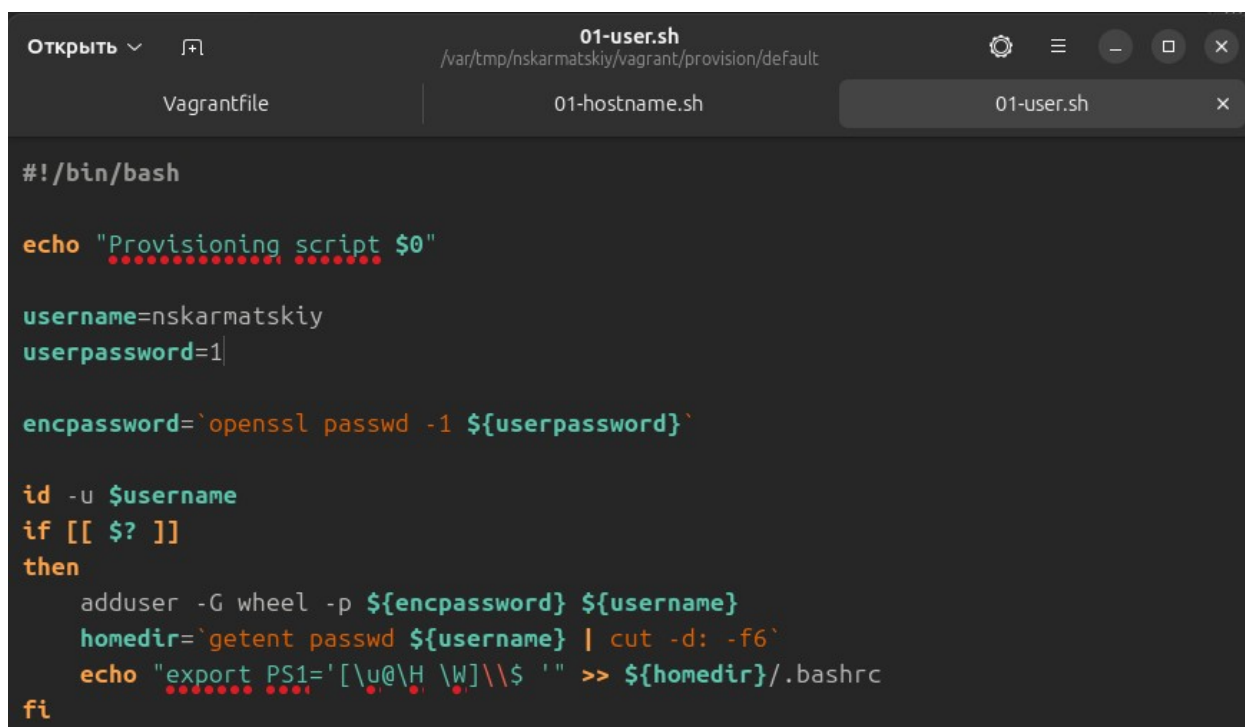
Vagrantfile 01-hostname.sh x

#!/bin/bash

username=nskarmatskiy

hostnamectl set-hostname "${HOSTNAME%%.*}.${username}.net
```

Рис.1.5: Измененный скрипт 01-hostname.sh



```
Открыть ▾ 01-user.sh
/var/tmp/nskarmatskiy/vagrant/provision/default

Vagrantfile 01-hostname.sh 01-user.sh x

#!/bin/bash

echo "Provisioning script $0"

username=nskarmatskiy
userpassword=1

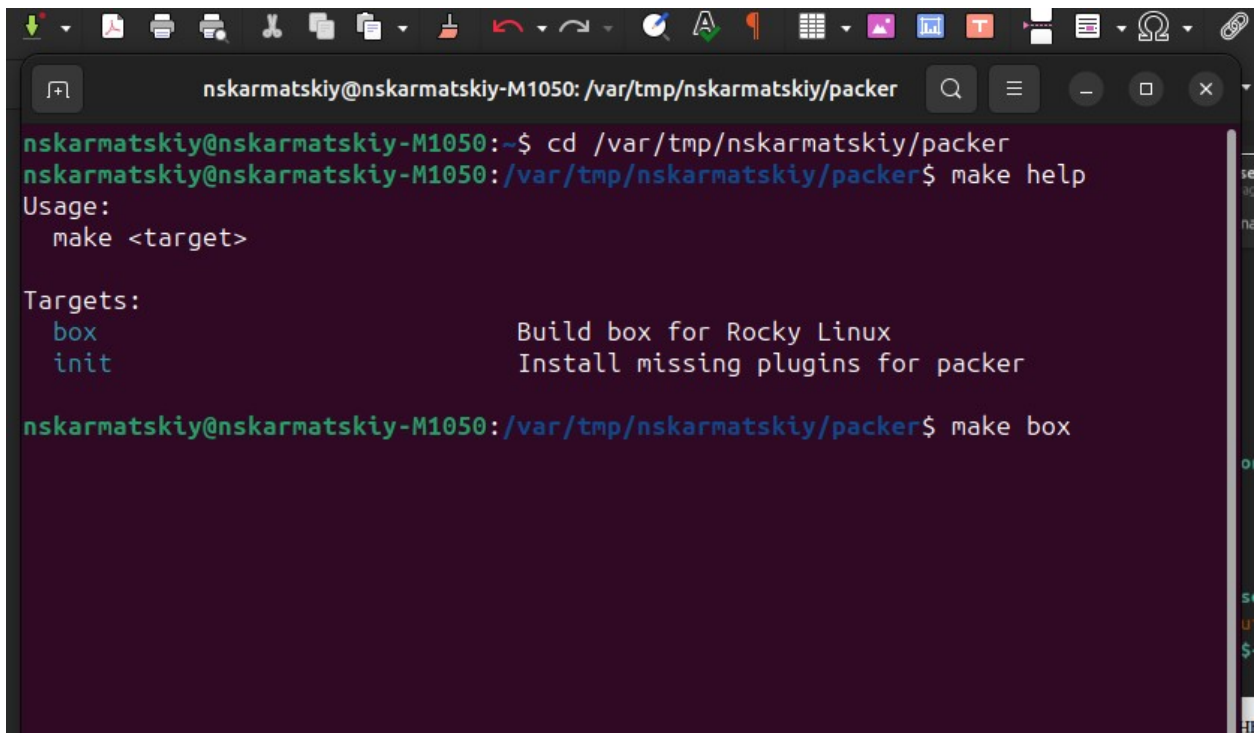
encpassword=`openssl passwd -1 ${userpassword}`

id -u $username
if [[ $? ]]
then
    adduser -G wheel -p ${encpassword} ${username}
    homedir=`getent passwd ${username} | cut -d: -f6`
    echo "export PS1='[\u@\H \W]\\$ ' >> ${homedir}/.bashrc
fi
```

Рис.1.6: Измененный скрипт 01-user.sh

2. Запустить виртуальные машины сервера и клиента и убедиться в их работоспособности, изменение в настройки внутреннего окружения виртуальной машины

1. Перейдем в каталог `cd /var/tmp/nskarmatskiy/packer`, запустим `makefile` и сформируем `box`-файл.



```
nskarmatskiy@nskarmatskiy-M1050: /var/tmp/nskarmatskiy/packer
nskarmatskiy@nskarmatskiy-M1050: /var/tmp/nskarmatskiy/packer$ make help
Usage:
  make <target>

Targets:
  box          Build box for Rocky Linux
  init         Install missing plugins for packer

nskarmatskiy@nskarmatskiy-M1050: /var/tmp/nskarmatskiy/packer$ make box
```

Рис.2.1: Создание box-файла

2. Сохраняем нужные файлы, чтобы сделать бэкап, если что-то пойдет не так, или если захотим запустить виртуальную машину на другом компьютере

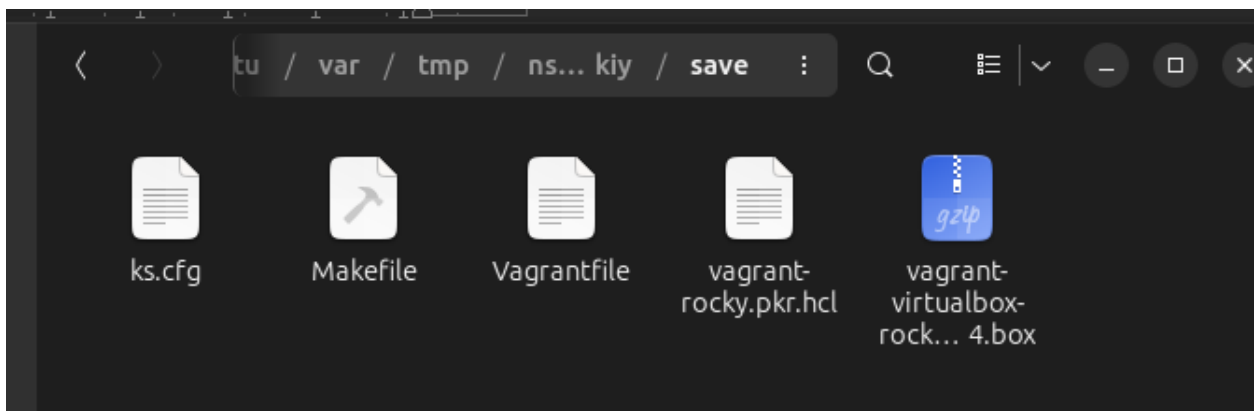


Рис.2.2: Сохраненные файлы

3. Далее переходим в каталог `vagrant` и с помощью `makefile` создаем две виртуальные машины и запускаем их с помощью команд `make server-up` и `make client-up`, после первого запуска вносим изменения, которые прописывали в файлах `01-user.sh` и `01-hostname.sh`, с помощью команд `make server-provision` и `make client-provision`.

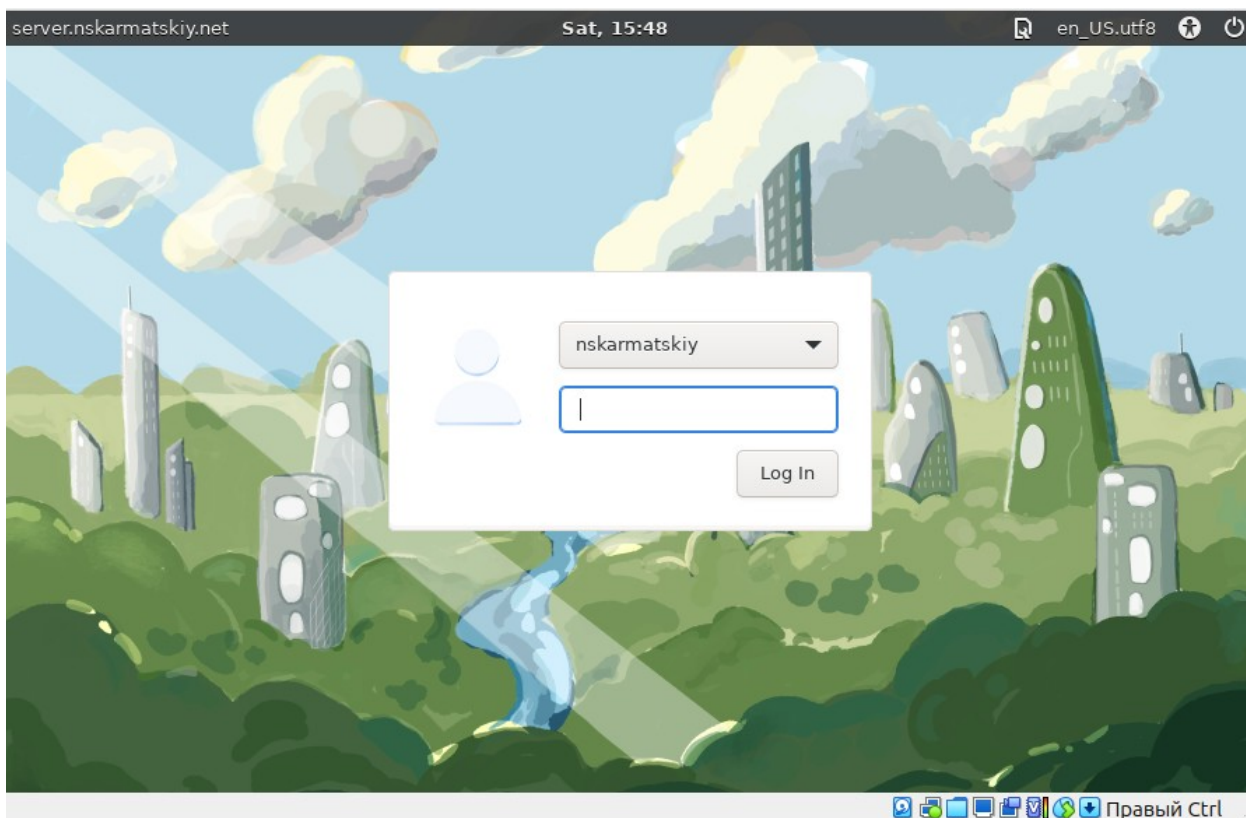


Рис.2.3: Запущенный server

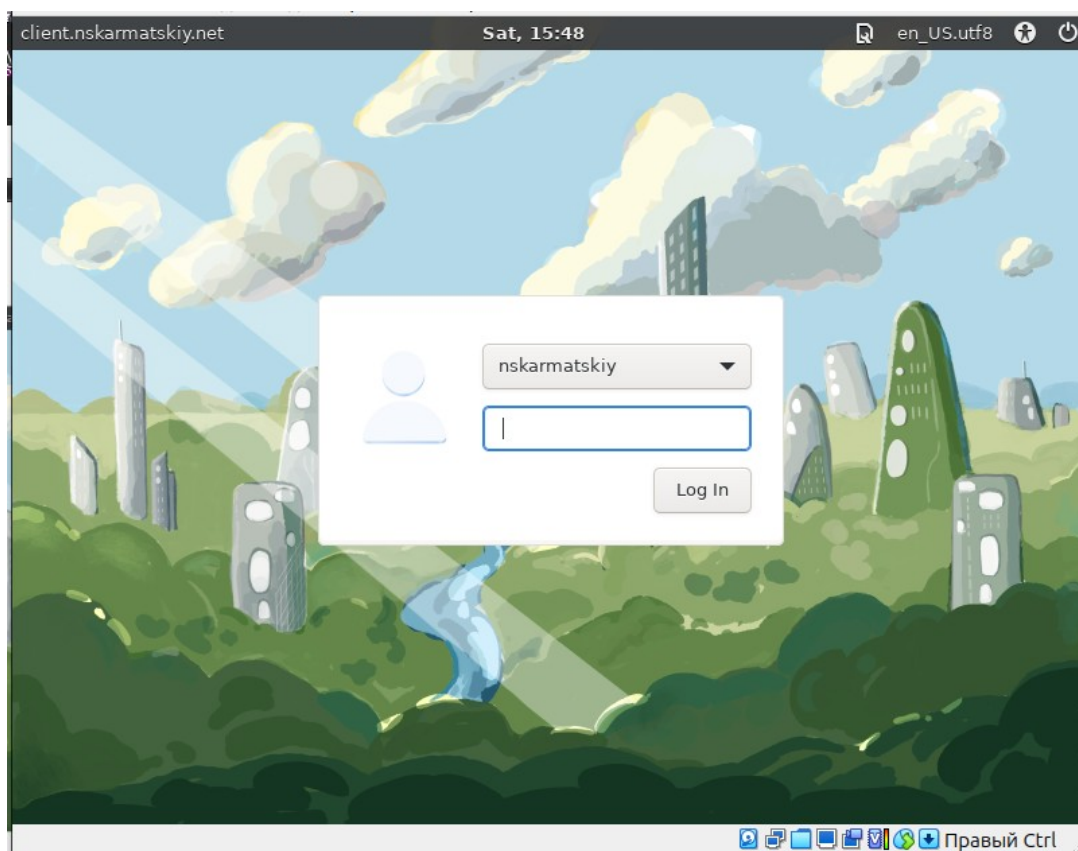


Рис.2.4: Запущенный client

3. Сохраним все файлы связанные с этими виртуальными машинами, чтобы можно

было их развернуть если что на другом устройстве.

**Вывод:** Мы приобрели практические навыки установки Rocky Linux на виртуальную машину с помощью инструмента Vagrant.

### Контрольные вопросы

1. Для чего предназначен Vagrant?

Vagrant – это **инструмент для создания и управления виртуальными средами разработки**. Он позволяет разработчикам легко создавать и настраивать виртуальные машины на своих компьютерах, что упрощает процесс настройки и согласования работы между членами команды.

2. Что такое box-файл? В чём назначение Vagrantfile?

box-файл (или Vagrant Box) — сохранённый образ виртуальной машины с развёрнутой в ней операционной системой; по сути, box-файл используется как основа для клонирования виртуальных машин с теми или иными настройками;

Vagrantfile — конфигурационный файл, написанный на языке Ruby, в котором указаны настройки запуска виртуальной машины.

3. Приведите описание и примеры вызова основных команд Vagrant.

- `vagrant help` — вызов справки по командам Vagrant;
- `vagrant box list` — список подключённых к Vagrant box-файлов;
- `vagrant box add` — подключение box-файла к Vagrant;
- `vagrant destroy` — отключение box-файла от Vagrant и удаление его из виртуального окружения;
- `vagrant init` — создание конфигурационного файла Vagrantfile для его последующего изменения;
- `vagrant up` — запуск виртуальной машины с использованием инструкций по запуску из конфигурационного файла Vagrantfile;

- `vagrant reload` — перезагрузка виртуальной машины;
- `vagrant halt` — остановка и выключение виртуальной машины;
- `vagrant provision` — настройка внутреннего окружения имеющейся виртуальной машины (например, добавление новых инструкций (скриптов) в ранее созданную виртуальную машину);
- `vagrant ssh` — подключение к виртуальной машине через ssh.

4. Дайте построчные пояснения содержания файлов `vagrant-rocky.pkr.hcl`, `ks.cfg`, `Vagrantfile`, `Makefile`.

`Vagrantfile`:

Первые две строки указывают на режим работы с `Vagrantfile` и использование языка Ruby.

- Затем идёт цикл `do`, заменяющий конструкцию `Vagrant.configure` далее по тексту на `config`.
- Строка `config.vm.box = "BOX_NAME"` задаёт название образа (бокс-файла) виртуальной машины (обычно выбирается из официального репозитория).
- Строка `config.vm.hostname = "HOST_NAME"` задаёт имя виртуальной машины.
- Конструкция `config.vm.network` задаёт тип сетевого соединения и может иметь следующие назначения:
  - `config.vm.network "private_network", ip: "xxx.xxx.xxx.xxx"` — адрес из внутренней сети;
  - `config.vm.network "public_network", ip: "xxx.xxx.xxx.xxx"` — публичный адрес, по которому виртуальная машина будет доступна;
  - `config.vm.network "private_network", type: "dhcp"` — адрес, назначаемый по протоколу DHCP.
- Строка `config.vm.define "VM_NAME"` задаёт название виртуальной машины, по которому можно обращаться к ней из `Vagrant` и `VirtualBox`.
- В конце идёт конструкция, определяющая параметры провайдера, а именно запуск виртуальной машины без графического интерфейса и с



выделением 1 ГБ памяти.