

Отчет по лабораторной работе №1

Дисциплина: Информационная безопасность

Выполнила: Губина Ольга Вячеславовна

Содержание

1	Цель работы	6
1.0.1	Цель работы:	6
2	Задание	7
3	Теоретическое введение	8
4	Выполнение лабораторной работы	10
5	Выводы	36
	Список литературы	37

Список иллюстраций

4.1	Авторизация на github.com	10
4.2	Создание каталога work/2023-2024/Information_security	11
4.3	Создание репозитория по шаблону	11
4.4	Создание репозитория по шаблону	12
4.5	Клонирование репозитория	12
4.6	Проверка git	13
4.7	Создание виртуальной машины	14
4.8	Параметры VM	14
4.9	Параметры VM	15
4.10	Итог - параметры VM	15
4.11	VM ovgubina	16
4.12	Запуск VM	16
4.13	Выбор языка	17
4.14	Параметры настройки установки образа ОС	18
4.15	Выбор программ	19
4.16	Отключение KDUMP	20
4.17	Место установки	21
4.18	Сеть и имя узла	22
4.19	Выбор раскладки клавиатуры	23
4.20	Выбор времени и даты	24
4.21	Политика безопасности	25
4.22	Установка пароля для root	26
4.23	Завершение настроек, начало установки	27
4.24	Успешное завершение установки образа	28
4.25	Начало работы в Rocky Linux	29
4.26	Доступ к геолокации	29
4.27	Подключение гугл-аккаунта	30
4.28	Имя пользователя	30
4.29	Установка пароля пользователя	31
4.30	Окончание настройки ОС	31
4.31	Подключение образа диска дополнений гостевой ОС	32
4.32	Запуск образа диска дополнений гостевой ОС	33
4.33	Подтверждение запуска	33
4.34	Последовательность загрузки системы	34
4.35	Анализ памяти	35
4.36	Информация о файловых системах	35

4.37 Последовательность монтирования ФС	35
---	----

Список таблиц

3.1	Описание некоторых функций языка разметки markdown	9
-----	--	---

1 Цель работы

1.0.1 Цель работы:

Создание репозитория дисциплины на github.com в соответствии с требованиями по шаблону и соблюдением соглашения наименований, соглашения для путей к файлам; Создание виртуальной машины в соответствии с соглашением о наименовании типа RedHat для возможность осуществления дальнейшей работы на курсе.

2 Задание

1. Создать рабочую директорию и репозиторий дисциплины на github.com в соответствии с:
 - шаблоном;
 - соглашением наименований;
2. Создать виртуальную машину.

3 Теоретическое введение

В данной работе мы будем взаимодействовать с **github.com**, где будем выгружать лабораторные работы.

Git — распределённая система управления версиями. Проект был создан *Линусом Торвальдсом* для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года. На сегодняшний день его поддерживает Джунио Хамано.

Для того, чтобы git узнал наше имя и электронную почту, необходимо выполнить следующие действия в командной строке:

```
git config --global user.name "Your Name" git config --global  
user.email "your_email@whatever.com"
```

Однако, с прошлого года сохранились некоторые данные, поэтому в моем случае этого делать не пришлось.

Отчеты по лабораторным работам данной дисциплины будем писать с использованием **markdown**.

Markdown — облегчённый язык разметки, созданный с целью обозначения форматирования в простом тексте, с максимальным сохранением его читаемости человеком, и пригодный для машинного преобразования в языки для продвинутых публикаций.

Теперь в табл. 3.1 рассмотрим основные сведения и функции языка разметки.

Таблица 3.1: Описание некоторых функций языка разметки markdown

Символ	Описание
#	Создание заголовка первого порядка (всего порядков шесть, кол-во символов # обозначаем номер порядка)
** *	Задаёт полужирное начертание текста
* *	Задаёт курсивное начертание текста
***	Задаёт курсивное полужирное начертание текста
- или *	Задаёт элемент неупорядоченного маркированного списка
` ``	Задаёт выделенные блоки кода

Более подробно о git и markdown см. в [1–4].

Также будем использовать дистрибутив Rocky Linux. **Rocky Linux** — это дистрибутив Linux, разработанный *Rocky Enterprise Software Foundation*. Предполагается, что это будет полный бинарно-совместимый выпуск, использующий исходный код операционной системы **Red Hat** Enterprise Linux.

Некоторые образы ISO, выпущенные Rocky Enterprise Software Foundation, не имеют прямых эквивалентов вышестоящего уровня. Они создаются для определенных целей, например, для предоставления живого загрузочного образа или для предоставления установочного носителя уменьшенного размера [5].

4 Выполнение лабораторной работы

1. Авторизуемся на github.com (рис. [4.1]).

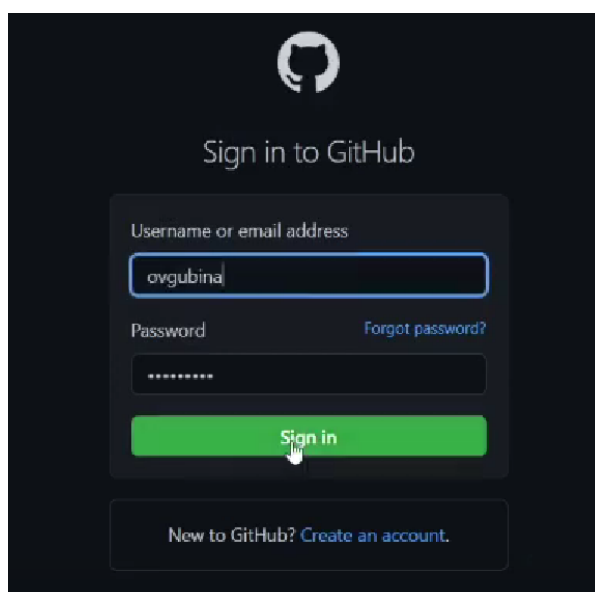


Рис. 4.1: Авторизация на github.com

2. Далее нам необходимо создать каталог, где будут располагаться все лабораторные работы, в моем случае: /d/work/2023-2024/Information_security(рис. 4.2). Полное название предмета написано на латиницей, поскольку при попытке сделать мейк возникала ошибка, то же и с пробелом - вместо него нижнее подчеркивание.

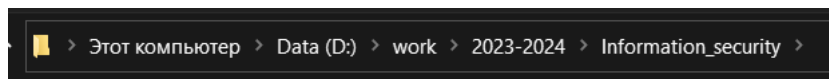


Рис. 4.2: Создание каталога work/2023-2024/Information_security

3. Создадим репозиторий по предложенному шаблону вручную через git.com с наименованием study_2023-2024_infosec (рис. [4.3]).

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

Repository template

yamadharm/course-directory-student-template

Start your repository with a template repository's contents.

☐ **Include all branches**
Copy all branches from yamadharm/course-directory-student-template and not just the default branch.

Owner * ovgubina / **Repository name *** study_2023-2024_infosec

study_2023-2024_infosec is available.

Great repository names are short and memorable. Need inspiration? How about [ubiquitous-tribble](#) ?

Description (optional)

Информационная безопасность 2023-2024

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

You are creating a public repository in your personal account.

[Create repository](#)

Рис. 4.3: Создание репозитория по шаблону

Видим, что репозиторий успешно создан в нашем профиле (рис. [4.4]).


```
echo "infosec" > COURSE
make
```

После этого Отправляем файлы на сервер (рис. [??]-[??]):

```
git add .
git commit -am '<commit_name>'
git push
```

```
slga@LAPTOP-CFFHTU0A MINGW64 /d/work/2023-2024/Information_security
$ cd infosec/
slga@LAPTOP-CFFHTU0A MINGW64 /d/work/2023-2024/Information_security/infosec (master)
$ echo "infosec" > COURSE
slga@LAPTOP-CFFHTU0A MINGW64 /d/work/2023-2024/Information_security/infosec (master)
$ make
slga@LAPTOP-CFFHTU0A MINGW64 /d/work/2023-2024/Information_security/infosec (master)
$ git add .
warning: in the working copy of 'COURSE', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'labs/README.md', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'presentation/README.md', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'presentation/README.ru.md', LF will be replaced by CRLF the next time Git touches it
slga@LAPTOP-CFFHTU0A MINGW64 /d/work/2023-2024/Information_security/infosec (master)
$ git commit -am 'make course structure'
[master 6ce0871] make course structure
151 files changed, 41045 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab1/presentation/Makefile
create mode 100644 labs/lab1/presentation/image/kulyabov.jpg
create mode 100644 labs/lab1/presentation/presentation.md
create mode 100644 labs/lab1/report/Makefile
create mode 100644 labs/lab1/report/bib/cite.bib
create mode 100644 labs/lab1/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab1/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab1/report/pandoc/filters/pandoc_egnos.py
create mode 100644 labs/lab1/report/pandoc/filters/pandoc_fignos.py
create mode 100644 labs/lab1/report/pandoc/filters/pandoc_secnos.py
create mode 100644 labs/lab1/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab1/report/pandoc/filters/pandocxnos/_init_.py
create mode 100644 labs/lab1/report/pandoc/filters/pandocxnos/core.py
```

```
slga@LAPTOP-CFFHTU0A MINGW64 /d/work/2023-2024/Information_security
$ git push
Enumerating objects: 37, done.
Counting objects: 100% (37/37), done.
Delta compression using up to 8 threads
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 342.11 KiB | 1.84 MiB/s, Total 35 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1
To github.com:ovgubina/study_2023-2024_infosec.git
cc1f074..6ce0871 master -> master
```

Переходим на github и видим, что создалась папка labs - изменения внесены успешно (рис. [4.6]):

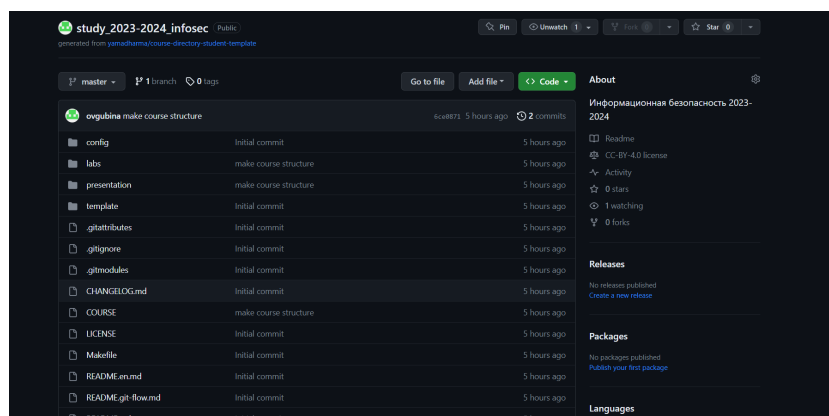


Рис. 4.6: Проверка git

5. Теперь перейдем к созданию виртуальной машины.

Для этого предварительно был скачан образ Rocky Linux 9.2 dvd.

Создаем виртуальную машину в соответствии с соглашением о наименовании, на диск D, используя установленный образ (рис. [4.7]):

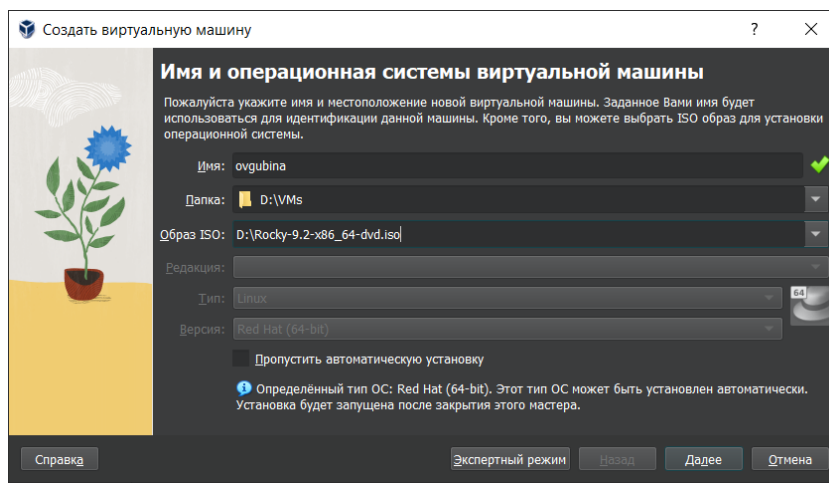


Рис. 4.7: Создание виртуальной машины

Выбираем параметры будущей машины - основная память 2Гб, 2 процессора, размер виртуального жесткого диска 40Гб (рис. [4.8]-[4.10])

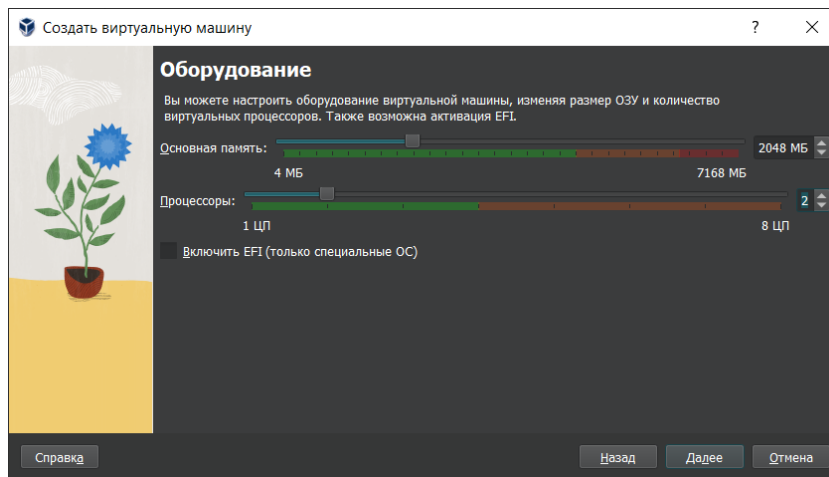


Рис. 4.8: Параметры ВМ

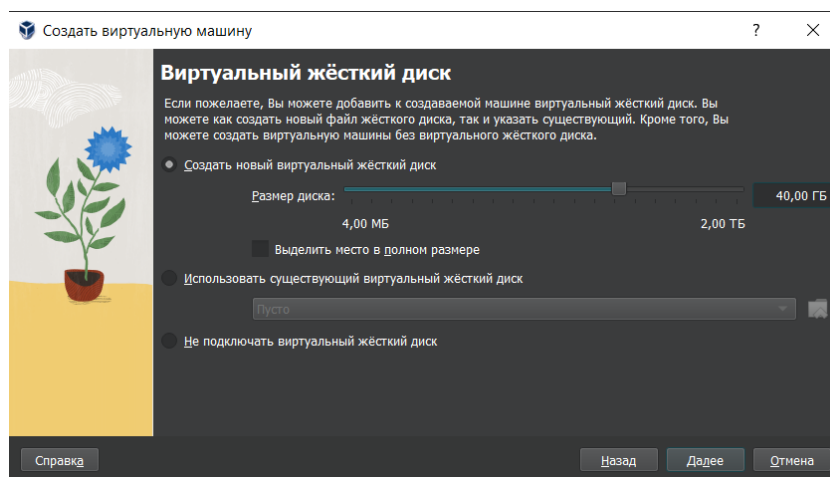


Рис. 4.9: Параметры VM

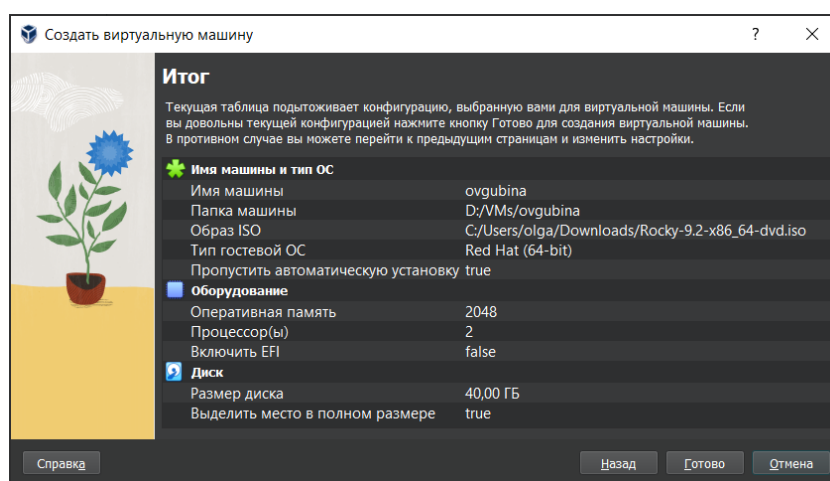


Рис. 4.10: Итог - параметры VM

Виртуальная машина успешно создана (рис. [4.11]):

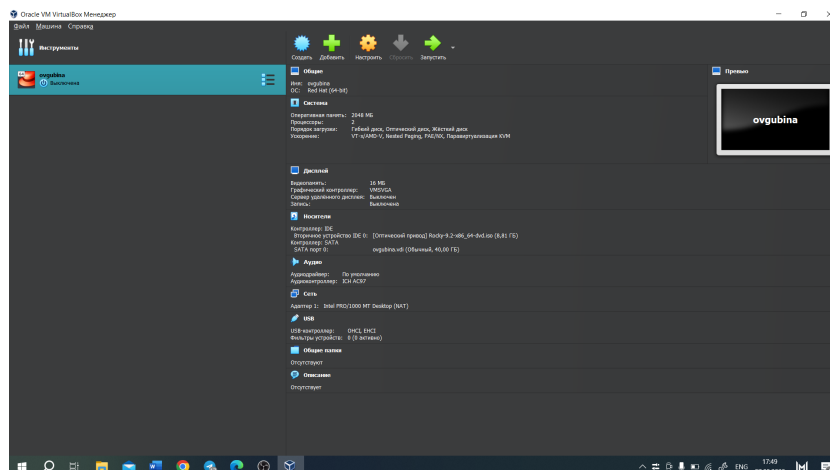


Рис. 4.11: BM ovgubina

6. Запускаем виртуальную машину (рис. [4.12]):

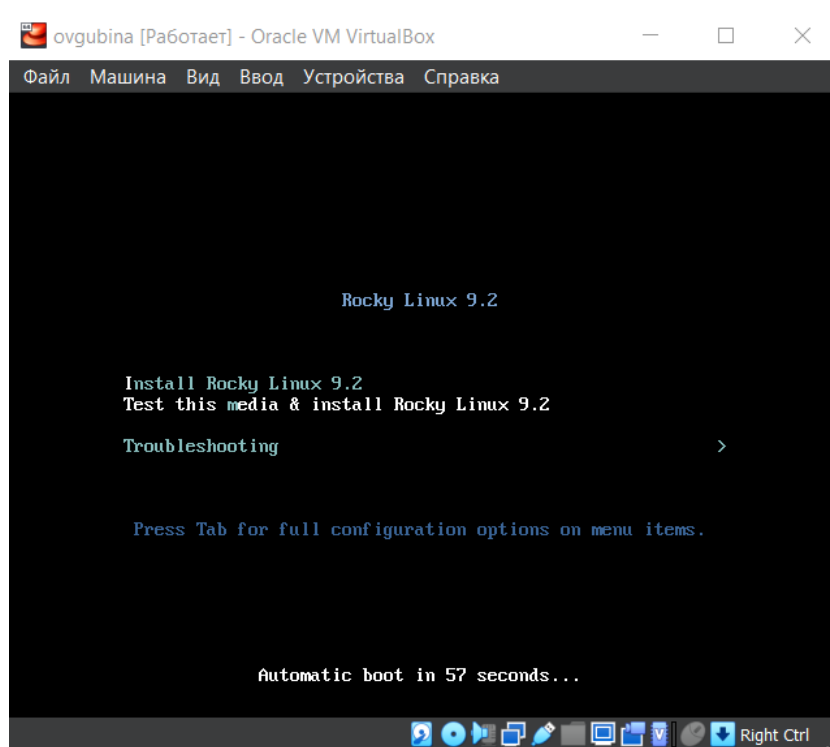


Рис. 4.12: Запуск BM

7. Далее необходимо произвести настройку Rocky Linux и выбрать язык (рис.

[4.13]-[4.14]).

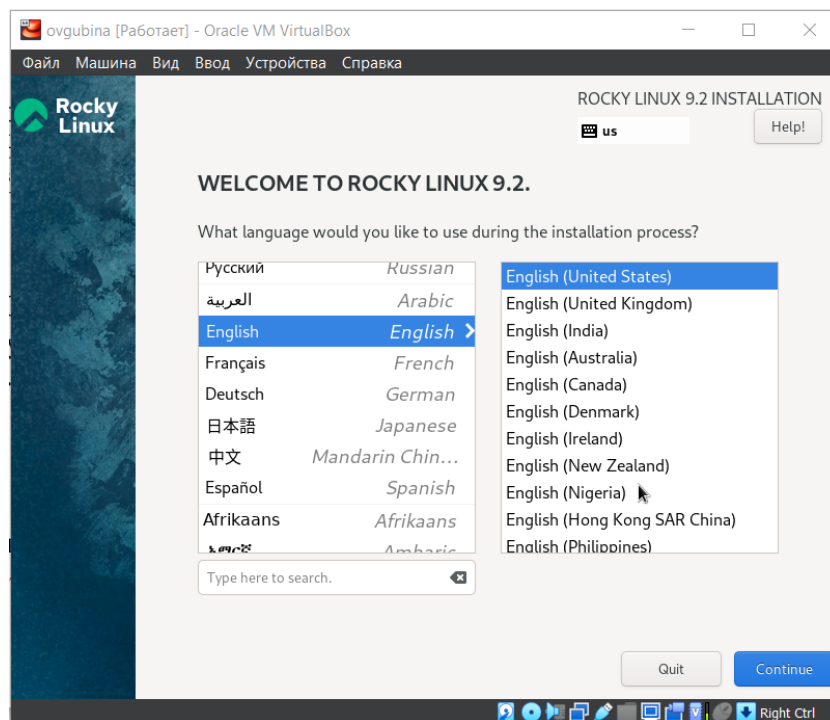


Рис. 4.13: Выбор языка

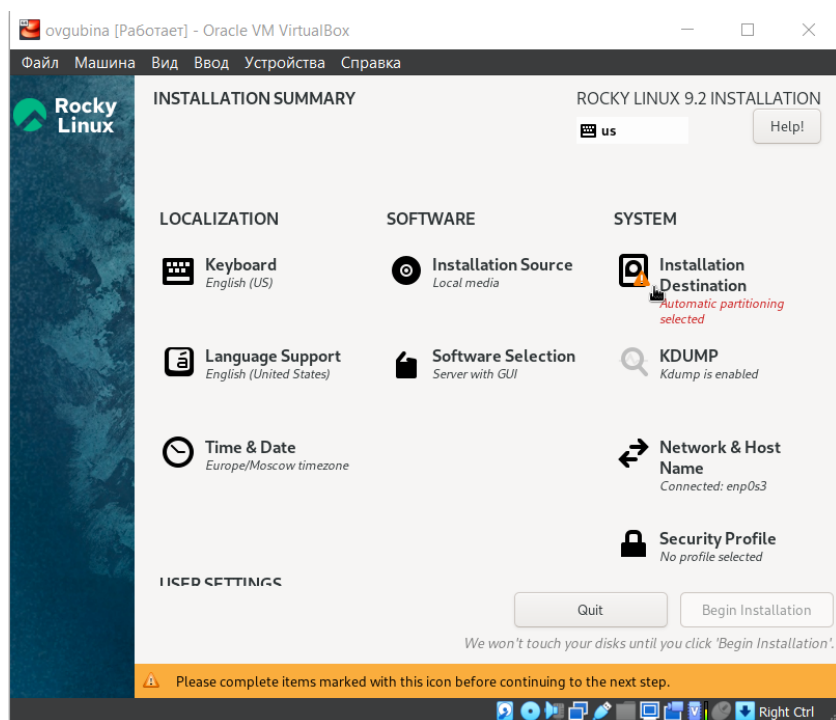


Рис. 4.14: Параметры настройки установки образа ОС

В разделе выбора программ указываем в качестве базового окружения Server with GUI, а в качестве дополнения — Development Tools (рис. [4.15])

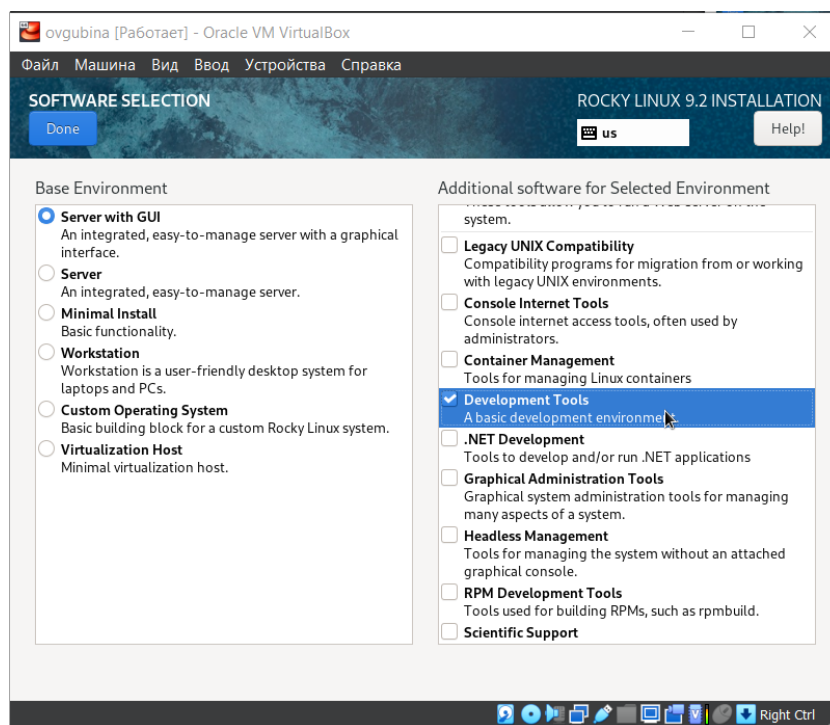


Рис. 4.15: Выбор программ

Отключаем KDUMP (рис. [4.16]):

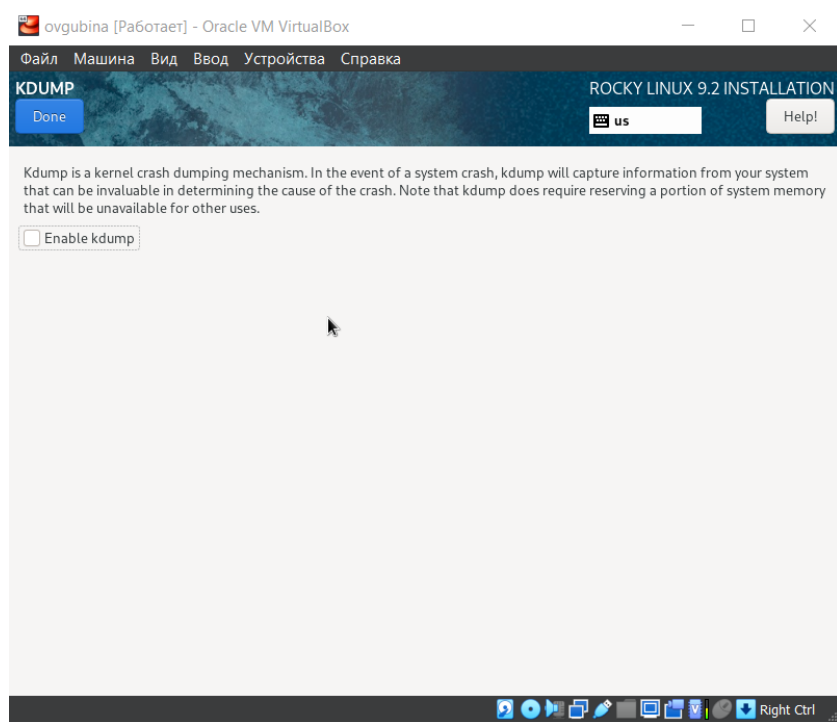


Рис. 4.16: Отключение KDUMP

Место установки ОС оставляем без изменений (рис. [4.17]):

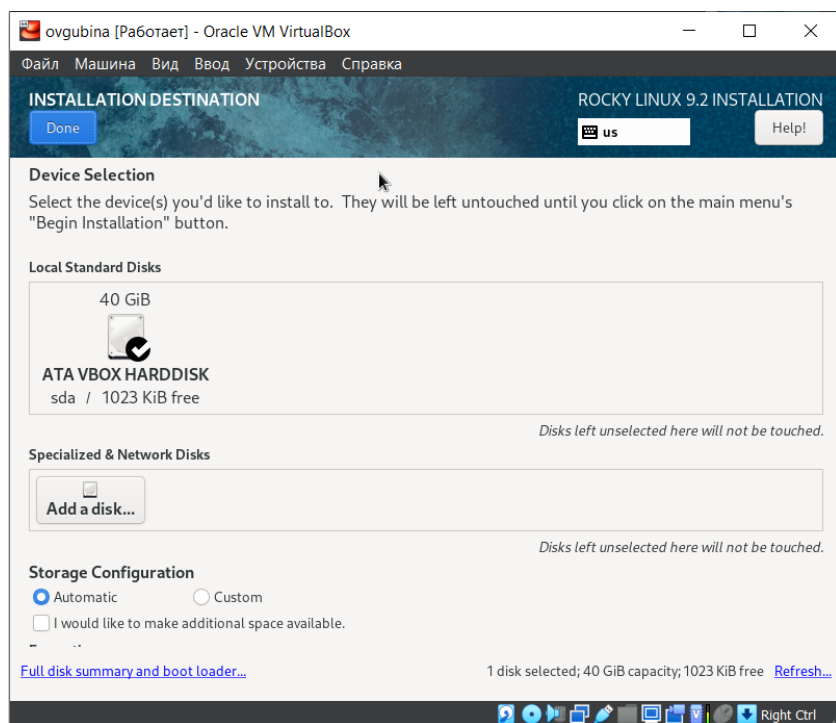


Рис. 4.17: Место установки

Включим сетевое соединение и в качестве имени узла указываем `ovgubina.localdomain` (рис. [4.18]) в соответствии с соглашением об именовании.

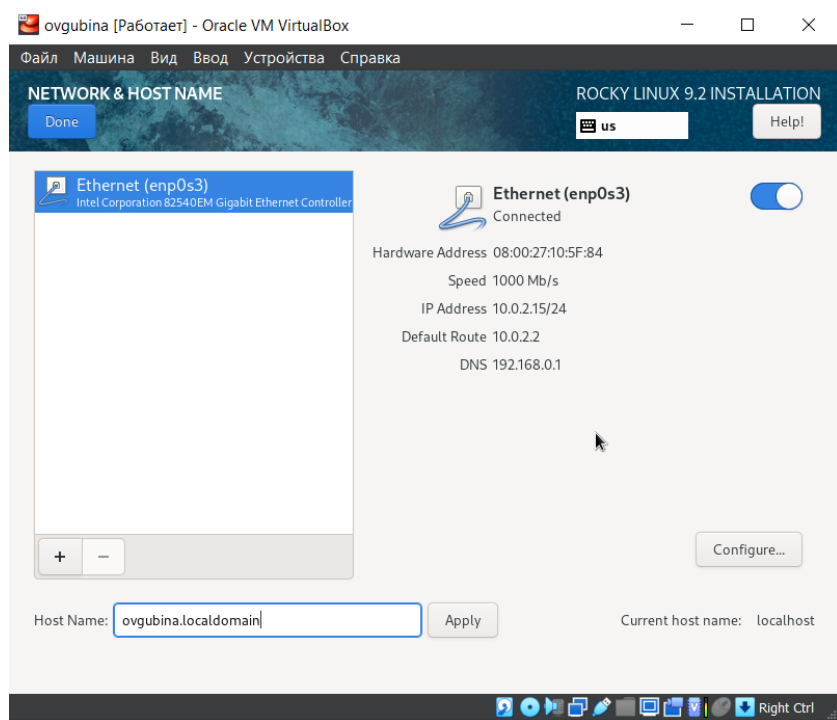


Рис. 4.18: Сеть и имя узла

Проверяем дополнительные настройки (рис. [4.19]-[4.20]):

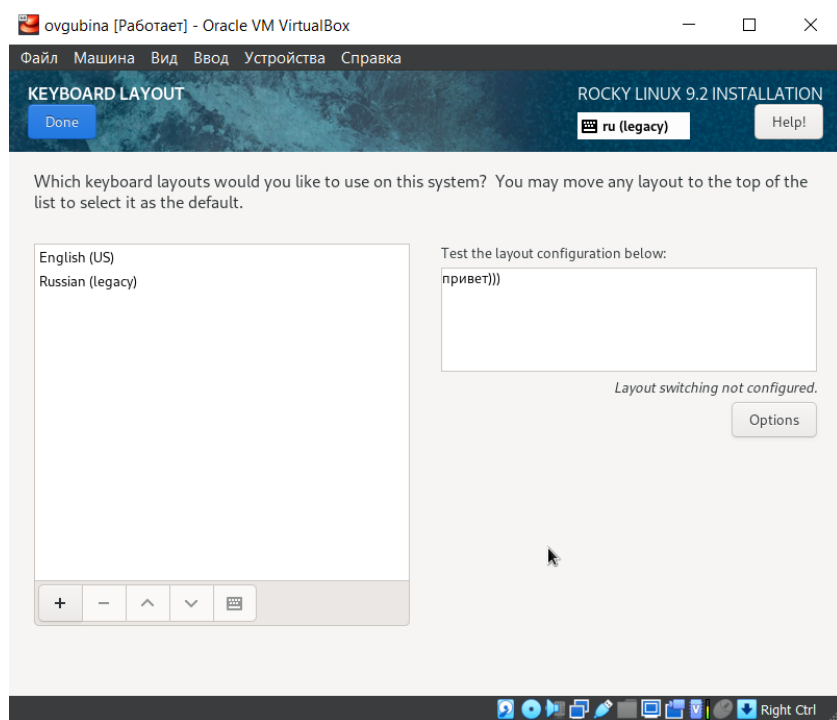


Рис. 4.19: Выбор раскладки клавиатуры

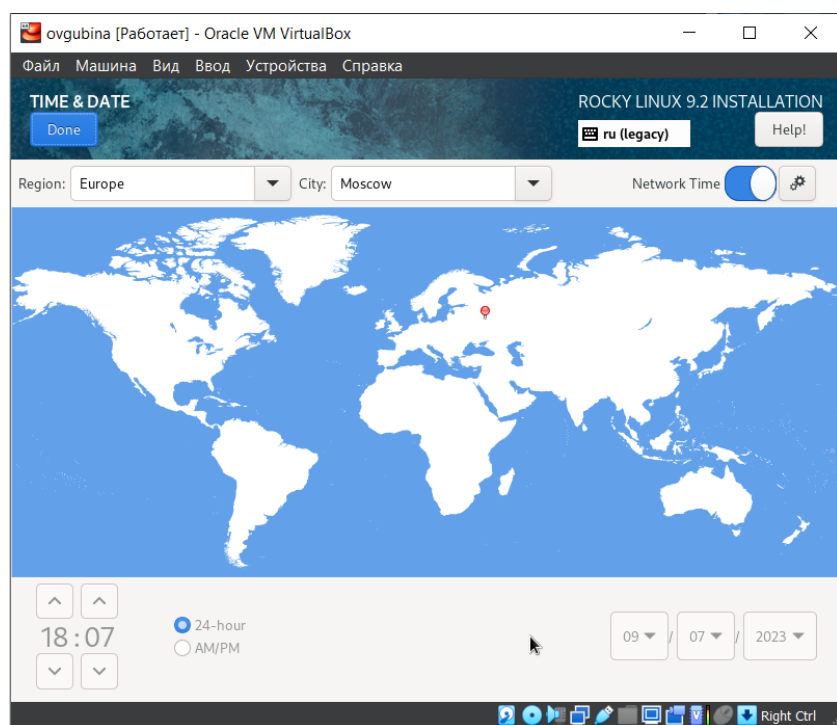


Рис. 4.20: Выбор времени и даты

Соглашаемся с политикой безопасности (рис. [4.21]):

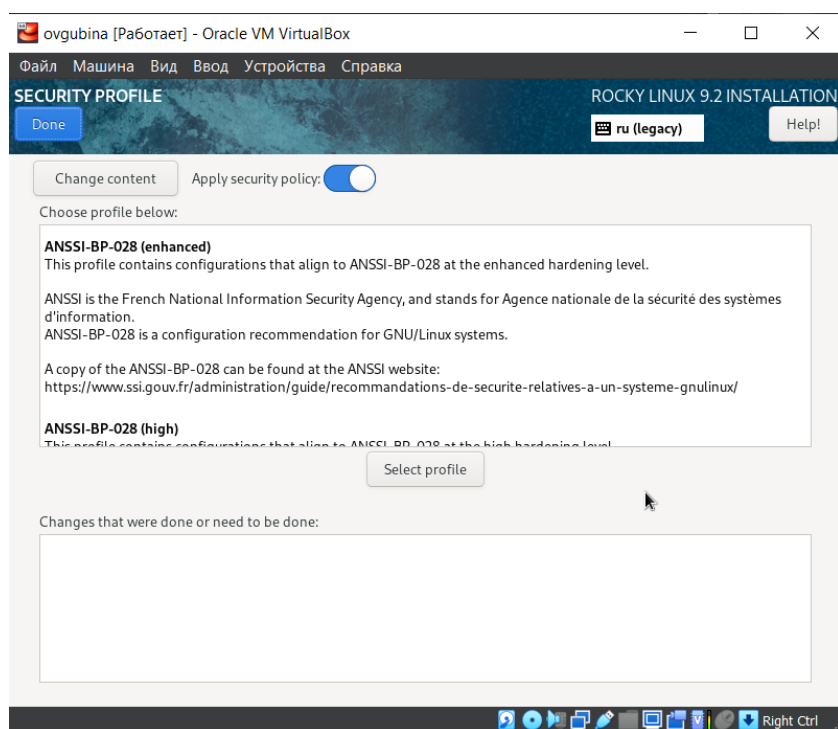


Рис. 4.21: Политика безопасности

Устанавливаем пароль для пользователя root (рис. [4.22]):

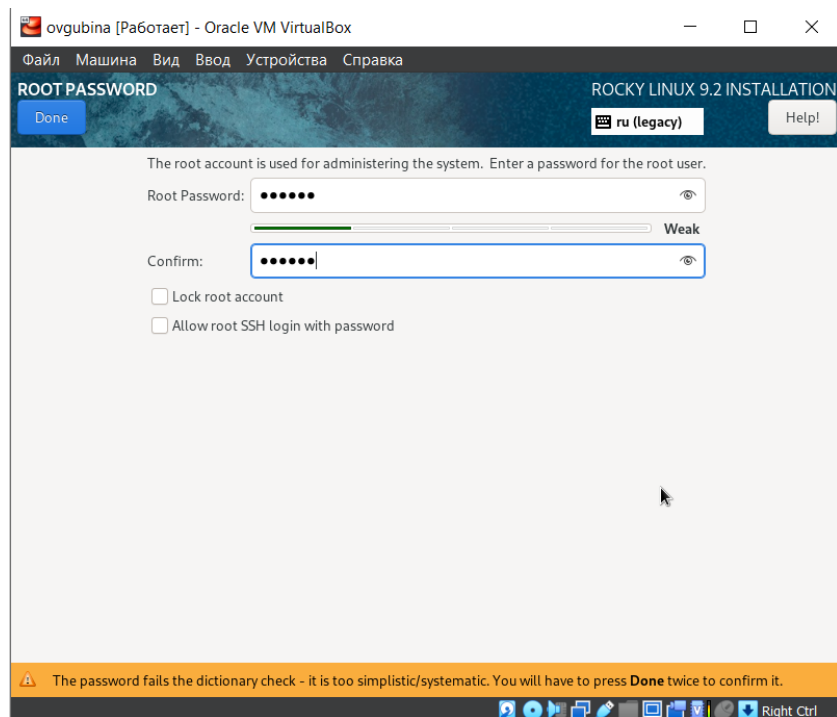


Рис. 4.22: Установка пароля для root

После этого мы можем начать установку (рис. [4.23]):

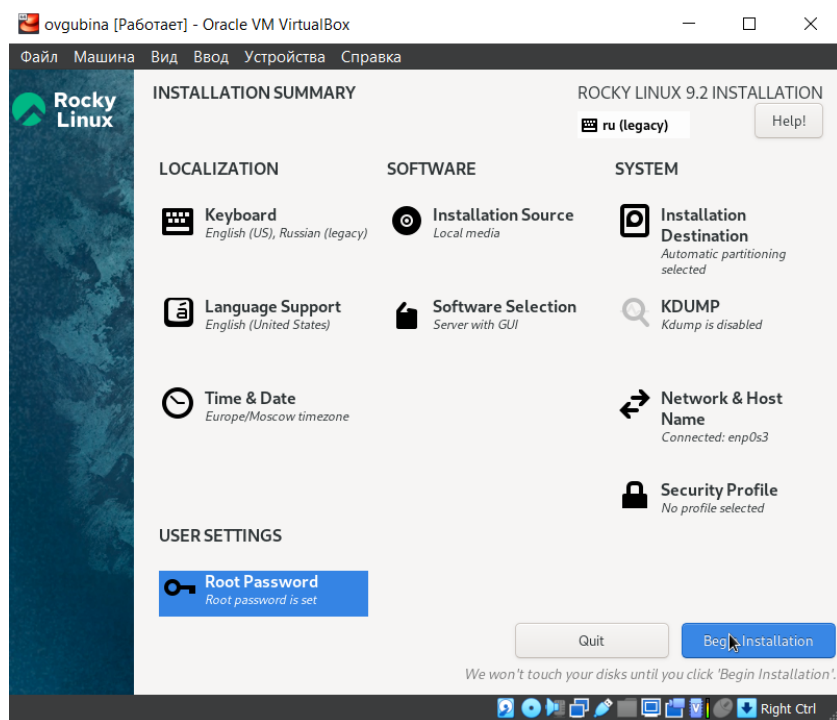


Рис. 4.23: Завершение настроек, начало установки

Видим, что установка завершена, перезапускаем систему (рис. [4.24]):

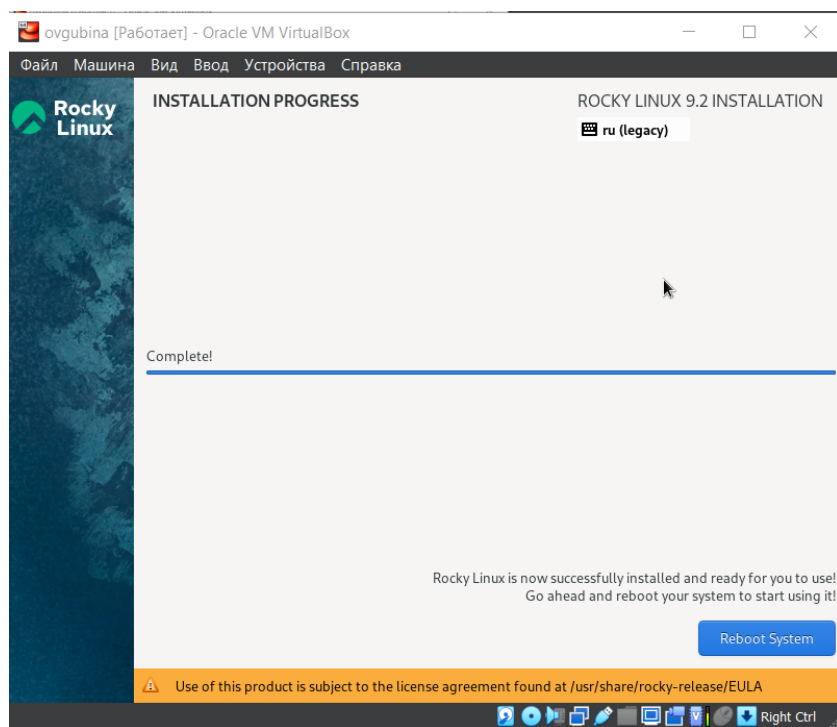


Рис. 4.24: Успешное завершение установки образа

После перезапуска настраиваем ОС под пользователя (рис. [4.25]-[4.30]).

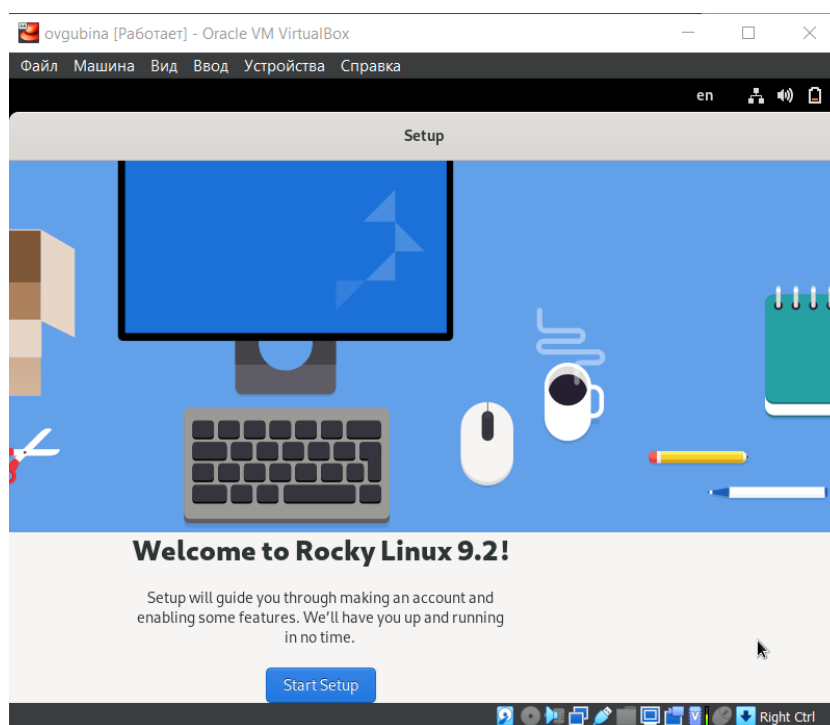


Рис. 4.25: Начало работы в Rocky Linux

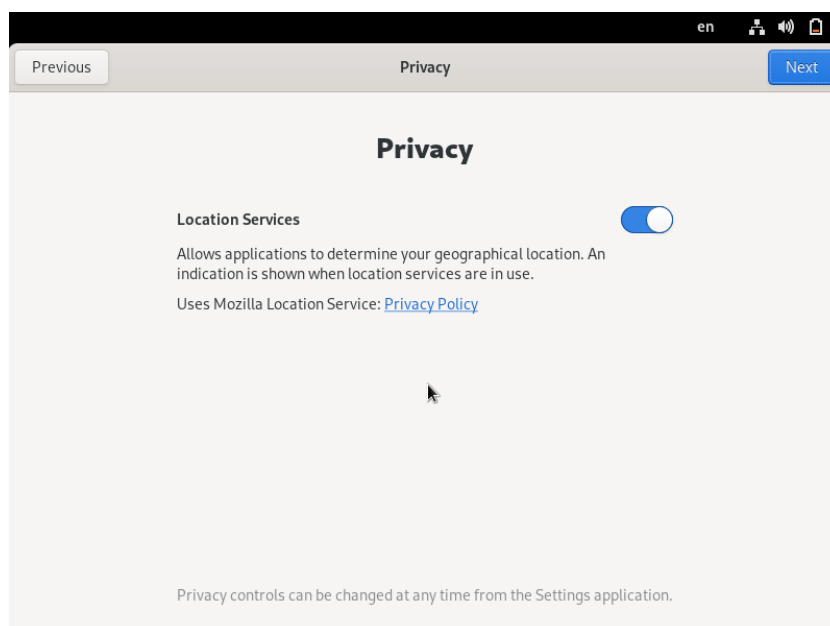


Рис. 4.26: Доступ к геолокации

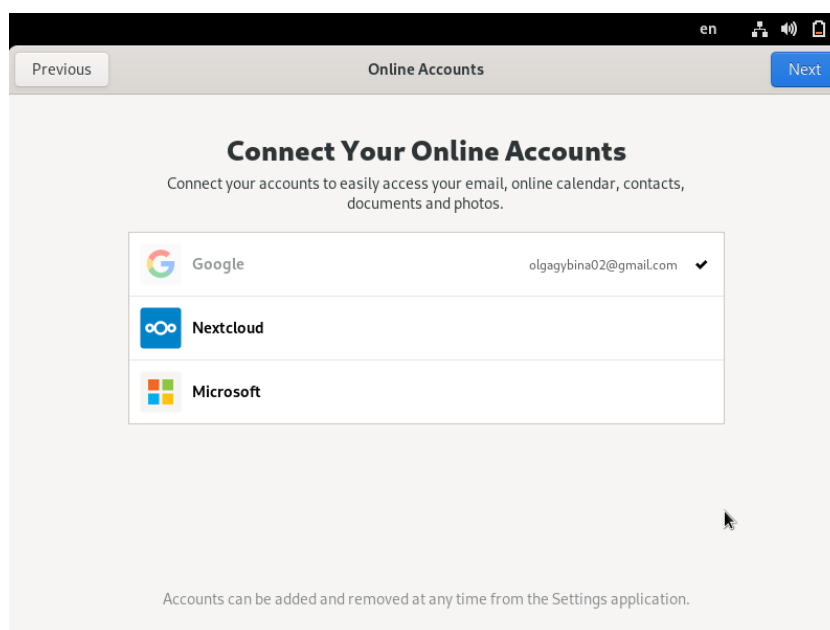


Рис. 4.27: Подключение гугл-аккаунта

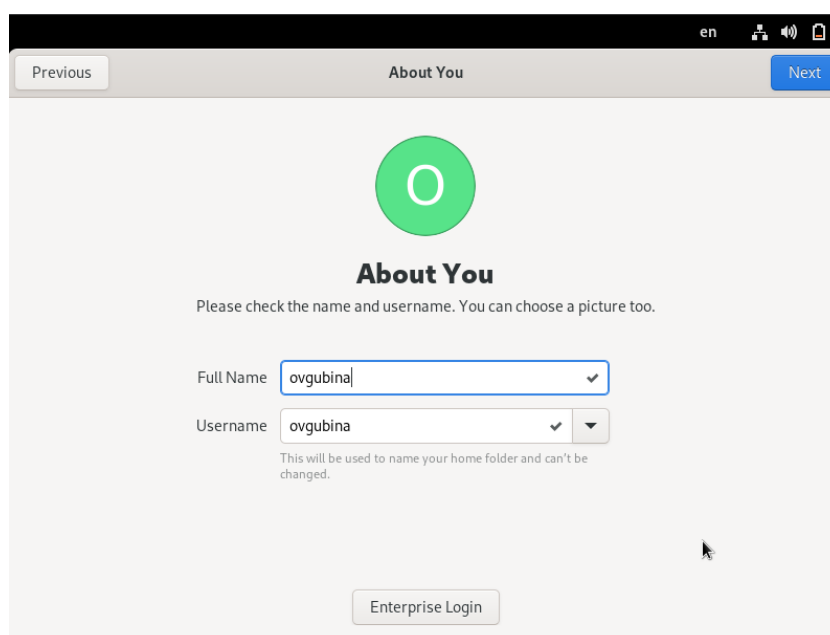


Рис. 4.28: Имя пользователя

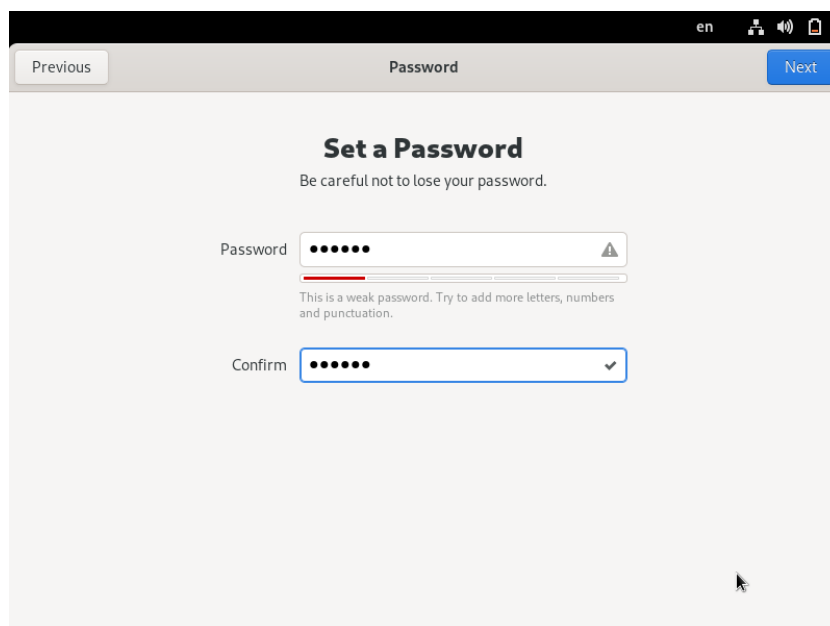


Рис. 4.29: Установка пароля пользователя

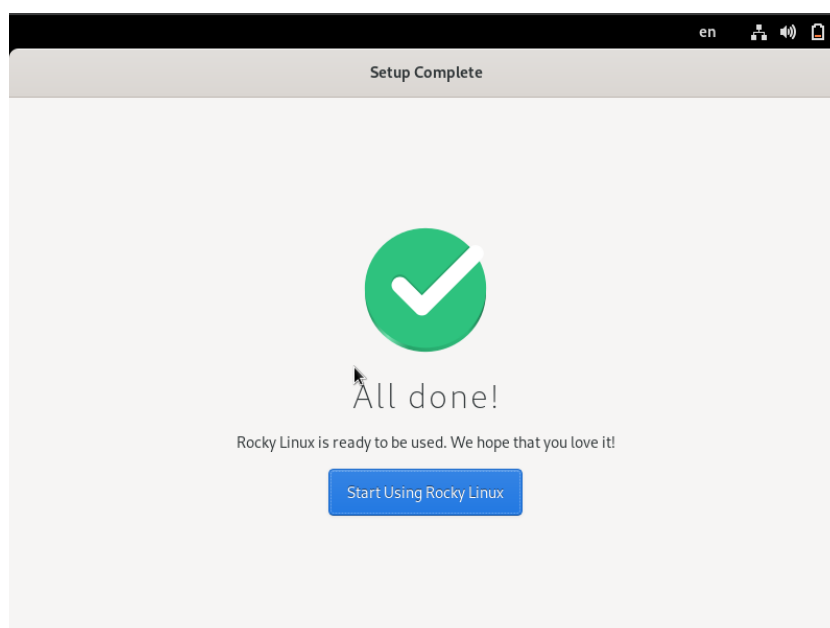


Рис. 4.30: Окончание настройки ОС

В меню “Устройства виртуальной машины” подключим образ диска дополнительный гостевой ОС (рис. [4.31]–[4.32]), вводим пароль пользователя root виртуальной

ОС (рис. [4.33]).

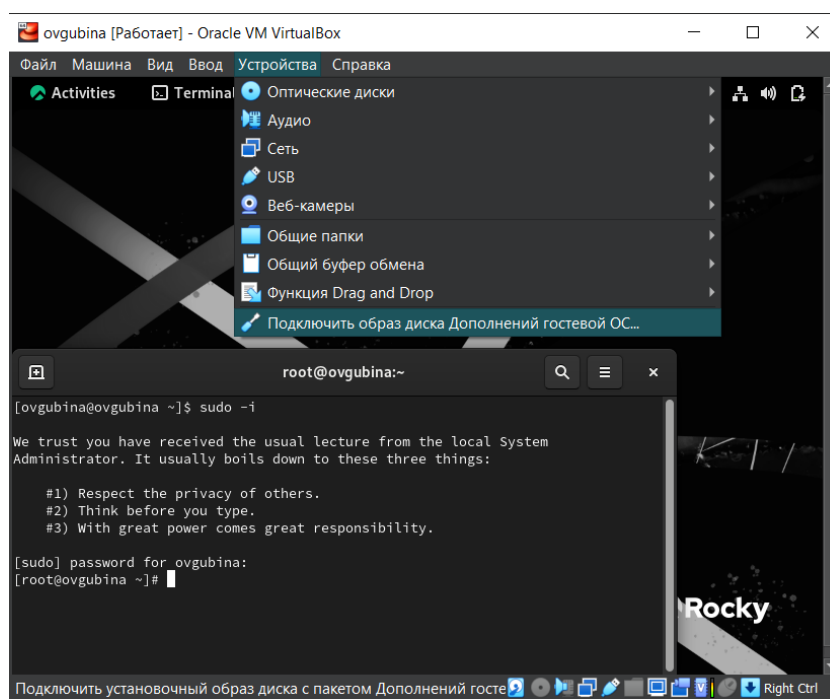


Рис. 4.31: Подключение образа диска дополнений гостевой ОС

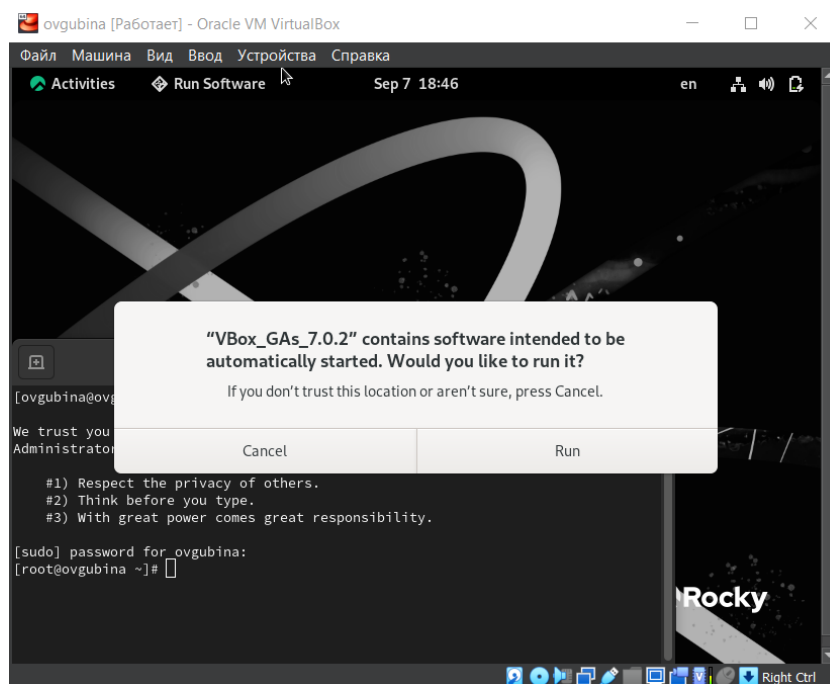


Рис. 4.32: Запуск образа диска дополнений гостевой ОС

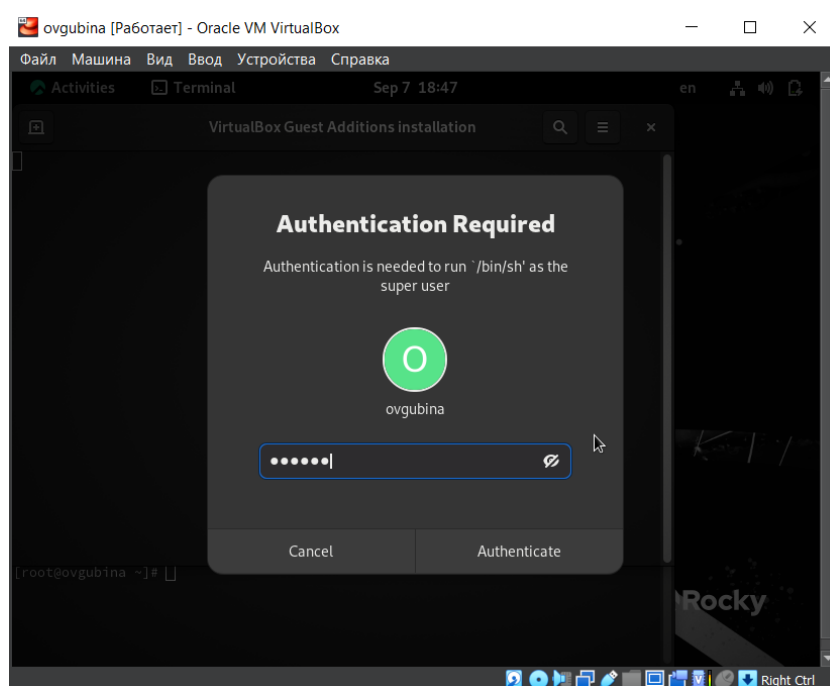


Рис. 4.33: Подтверждение запуска

8. После перезапуска начинаем работу с домашним заданием.

Дожидаемся загрузки графического окружения и открываем терминал. В окне терминала посмотрим последовательность загрузки системы, выполнив команду `dmesg | less` (рис. [4.34]).

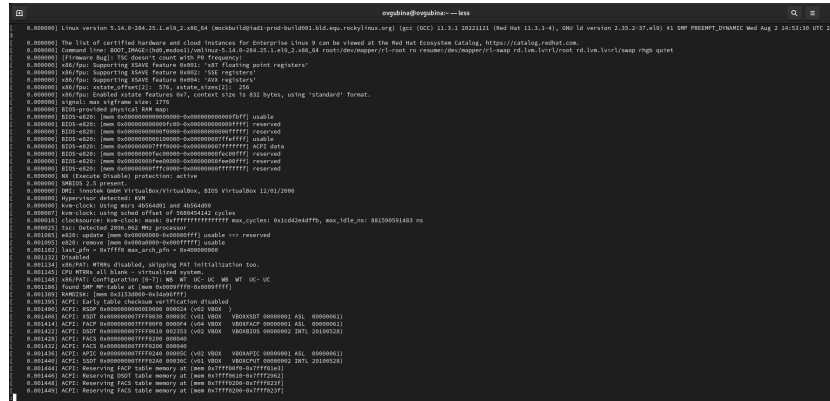


Рис. 4.34: Последовательность загрузки системы

Далее от нас требуется узнать: - Версия ядра Linux (Linux version) (рис. [??]).

```
```uname -r```
```

! [Информация о системе](image/34.png){#fig:38 width=70%}

- Частота процессора (Detected Mhz processor) (рис. [??]).

```
cat /proc/cpuinfo | grep "cpu MHz - выводит частоту обоих процессо-
ров - 2096.862MHz.
```

- Модель процессора (CPU0).

lscpu - раздел Vendor ID (рис. [??]).

- Объем доступной оперативной памяти (Memory available) (рис. [4.35]).

free

```

[ovgubina@ovgubina ~]$ free -g

```

	total	used	free	shared	buff/cache	available
Mem:	1	1	0	0	0	0
Swap:	2	0	2			

```

[ovgubina@ovgubina ~]$ free

```

	total	used	free	shared	buff/cache	available
Mem:	2017284	1342456	134044	23376	730308	674828
Swap:	2150396	1560	2148836			

```

[ovgubina@ovgubina ~]$

```

Рис. 4.35: Анализ памяти

- Тип обнаруженного гипервизора (Hypervisor detected).

`lscpu` - раздел Virtualization features (рис. [??]).

- Тип файловой системы корневого раздела.

`lsblk -f` - тип xfs (рис. [4.36])

```

[ovgubina@ovgubina ~]$ lsblk -f

```

NAME	FSTYPE	FSVER	LABEL	UUID	FSAVAIL	FSUSE%	MOUNTPOINTS
sda							
└─sda1	xfs			db90a0a1-9a15-4131-999e-e10d9cd4c638	686M	32%	/boot
└─sda2	LVM2_member	LVM2 001		14W020-3E80-prdE-AkHf-LV2q-WJ07-Qc41Ed			
└─┌─rl-root	xfs			563ed222-b7f0-43d6-b320-dfc12343c79e	31.9G	14%	/
└─┌─rl-swap	swap	1		2b4a483-b156-4d2a-aa33-1b95f7ca47a			[SWAP]
sr0	iso9660		Joliet Extension VBox_GAs_7.0.2	2022-10-19-20-19-46-75	0	100%	/run/media/ovgubina/VBox_GAs_7.0.2

```

[ovgubina@ovgubina ~]$

```

Рис. 4.36: Информация о файловых системах

- Последовательность монтирования файловых систем (рис. [4.37]).

`mount`

```

[ovgubina@ovgubina ~]$ mount

```

proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=4096K,nr_inodes=244605,mode=755,inode64)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=4096K,nr_inodes=819200,mode=755,inode64)
group on /sys/fs/cgroup type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,mode=755,recursivestore)
store on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime,seclabel)
bin on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
/dev/mapper/rl-root on / type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)
systemd on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=18860)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)
huqene on /dev/huqene type huqene (rw,nosuid,nodev,noexec,relatime,seclabel)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabel)
tracefs on /sys/kernel/trace type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
configs on /sys/kernel/config type configs (rw,nosuid,nodev,noexec,relatime)
none on /run/credentials/systemd-tmpfiles-setup-dev.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)
none on /run/credentials/systemd-sysctl.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)
/dev/sda1 on /boot type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
none on /run/credentials/systemd-tmpfiles-setup.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=201728K,nr_inodes=50432,mode=700,uid=1000,gid=1000,inode64)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
/dev/sr0 on /run/media/ovgubina/VBox_GAs_7.0.2 type iso9660 (ro,nosuid,nodev,relatime,nojoliet,check=s,map=n,blocksize=2048,uid=1000,gid=1000,dmode=500,fmode=400,uhelper=udisks2)

```

[ovgubina@ovgubina ~]$

```

Рис. 4.37: Последовательность монтирования ФС

## 5 Выводы

Создала репозиторий `study_2023-2024_infosec` дисциплины Информационная базопасность на `git.com` в соответствии с требованиями по шаблону и соблюдением соглашения наименований, соглашения для путей к файлам; Создала ВМ на основе образа Rocky Linux 9.2 dvd; Написала отчет при использовании языка разметки `markdown` по выполненной работе.

# Список литературы

1. Git [Электронный ресурс]. 2023. URL: <https://ru.wikipedia.org/wiki/Git>.
2. Работа с git [Электронный ресурс]. 2023. URL: [https://esystem.rudn.ru/pluginfile.php/1971716/mod\\_folder/content/0/git.pdf](https://esystem.rudn.ru/pluginfile.php/1971716/mod_folder/content/0/git.pdf).
3. Markdown [Электронный ресурс]. 2022. URL: <https://ru.wikipedia.org/wiki/Markdown>.
4. Язык Markdown [Электронный ресурс]. 2023. URL: [https://esystem.rudn.ru/pluginfile.php/1971716/mod\\_folder/content/0/markdown.pdf](https://esystem.rudn.ru/pluginfile.php/1971716/mod_folder/content/0/markdown.pdf).
5. Rocky Linux [Электронный ресурс]. 2023. URL: [https://ru.wikipedia.org/wiki/Rocky\\_Linux](https://ru.wikipedia.org/wiki/Rocky_Linux).